

UNIVERSITY OF THE PHILIPPINES MANILA
COLLEGE OF ARTS AND SCIENCES
DEPARTMENT OF PHYSICAL SCIENCES AND MATHEMATICS

**Health Insurance
Corporation
Customer Relationship Management Tool
(HICorpCRM)**

A special problem in partial fulfillment
of the requirements for the degree of
Bachelor of Science in Computer Science

Submitted by:

Paul John N. Macaraeg

June 2015

Permission is given for the following people to have access to this SP:

Available to the general public	Yes
Available only after consultation with author/ SP adviser	No
Available only to those bound by confidentiality agreement	No

ACCEPTANCE SHEET

The Special Problem entitled “Health Insurance Corporation Customer Relationship Management Tool (HICorpCRM)” prepared and submitted by Paul John N. Macaraeg in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended or acceptance.

Ma. Sheila A. Magboo, M.Sc.
Adviser

EXAMINERS:

	Approved	Disapproved
1. Gregorio B. Baes, Ph.D.	_____	_____
2. Avegail D. Carpio, M.Sc.	_____	_____
3. Perlita E. Gasmien, M.Sc.	_____	_____
4. Richar Bryann L. Chua, M.Sc.	_____	_____
5. Vincent Peter C. Magboo, M.D., M.Sc.	_____	_____
6. Bernie B. Terrado, M.Sc.	_____	_____

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

Ma. Sheila A. Magboo, M.Sc.

Unit Head

Mathematical and Computing Sciences Unit
Department of Physical Sciences
and Mathematics

Marcelina B. Lirazan, Ph.D.

Chair

Department of Physical Sciences
and Mathematics

Alex C. Gonzaga, Ph.D., Dr.Eng.

Dean

College of Arts and Sciences

Abstract

The health insurance industry is saturated by big companies competing for market share. This saturation has induced a change of focus towards this health insurance companies towards their customers which went from customer acquisition to customer retention. A need for a more holistic approach towards customer relationship management is needed which entails the need for SMS, Telephone and Email helplines. No SMS enabled customer relationship management tool is of existence in the health insurance market hence utilization of the pros brought about by the said technology hasn't been felt by its players yet. This need for an SMS enabled CRM tool could be substantiated through the utilization of a GSM modem and KANNEL for WAP based communication with a CodeIgniter based PHP web application that utilizes MySQL database so as cost and setup would be minimized for this software.

Kay Words: health insurance company, sms enabled crm tool

Contents

Acceptance Sheet	i
Abstract	ii
List of Figures	v
List of Tables	vi
I. Introduction	1
A. Background of the Study	1
B. Statement of the Problem	4
C. Objectives of the Study	7
D. Significance of the Project	15
E. Scope and Limitations	18
F. Assumptions	19
II. Review of Related Literature	20
III. Theoretical Framework	24
IV. Design and Implementation	36
A. Entity Relationship Diagram	37
B. Database Dictionary	38
C. Object Diagram	41
D. Format Messages	44
E. Technical Architecture	46
V. Results	47
VI. Discussion	62
VII. Conclusion	64
VIII. Recommendation	65
IX. References	66

X.	Appendix	70	
	A.	Source Code	71
XI.	Acknowledgement		

List of Figures

Table		Page
1	Short Messaging Service	25
2	Context Diagram	36
3	Entity Relationship Diagram	37
4	Object Diagrams	41
5	Login Screen	47
6	Dashboard	48
7	Customer Profile	48
8	Customer Registration	49
9	Customer Upload	49
10	Customer Update	50
11	Customer List	50
12	CustomerMessaging	51
13	Customer Referral – Company Selection	51
14	Customer Referral – Service Selection	52
15	Customer Advisory	52
16	SMS History	53
17	Companies	53
18	Company Registration	54
19	Company Partner Update	54
20	Company Profile	55
21	Company Details	55
22	Company Personnel	56
23	Personnel Registration	56
24	Personnel Update	57
25	Personnel List	57
26	Company Services	58

27	Service Registration	58
28	Service Update	59
29	Service List	59
30	System Accounts	60
31	User Registration	60
32	User Update	61
33	User Profiles	61

List of Tables

Table		Page
1	Transaction Number Specification	8
2	Data Dictionary	38

I. Introduction

A. Background of the Study

Article II, Section 15 of the Philippine Constitution (1987) upholds the right of Filipino citizens to avail quality and affordable health care. The health condition of an individual is regarded with so much importance because of its economic and social benefit. The World Health Organization (2015) even recognizes the role of health in development and productivity. For instance, economic values pertaining to how much an individual could be billed for in exchange for life saving surgeries could hit up to \$53,400 or Php 2,349,600 per hospital stay as the advisory board company (2014) dictates, on procedures such as heart valve procedures. These facts substantiate the claim that the health insurance industry is of great importance for they provide people with aid in terms of keeping themselves healthy and well.

The health industry in the Philippines shares about 1.4% of the country's total 2012 GDP, about \$3,808,000,000 or Php 167,552,000,000 of the total expenditures for health. It is such a profitable market that the private sector and government agencies progressively invest upon. This has led to the emergence of various companies offering health care insurance for a wide range of customers with varying age groups, economic stratas and religious factions. As per the Commission on Insurance (2014) there are 31 big life insurance companies dominating the health care market in which PhilAm Life & Gen. tops the hierarchy in terms of networth by having Php 28,775,973,266 in their sleeves. This number of health insurance companies posed a challenge in terms of the membership they incur through time. As the paper: "Factors Influencing Customer Loyalty of Mobile Phone Service: Empirical Evidence from Koreans" (Lee, 2009) points out, companies nowadays operate on fields of similar interests which saturates the various disciplines hence unsatisfactory perception towards one could lead to a

company's demise for consumers could easily shift to others (Harrington, 2014). This has lead marketing departments to focus more on customer retention rather than on customer acquisition efforts (Lee, 2009) for good words of mouth are better than fresh bundles cash as it serves as an indicator of a companies' stability, integrity and commitment, based on the report made by the U.S. Survey Report as well (2014). This has lead companies to invest more on efforts to make consumers feel a greater sense of value towards them by providing services as efficiently and as effectively as possible (Roddy, 2014).

The great promises of various information and communications infrastructures ranging from SMS, Emails to Telephone helplines has been of great value to health insurance companies (Roddy, 2014) for they provide cheap means of reaching out to customers ranging from a dollar to 500 bucks or more per customer which in turn returns to them 10x more of their initial investment, figures visualized by FrugalTech (2014). Of these technological pathways Emails and Telephone helplines dominates the customer relations management area of health insurance companies, as reflected by the means utilized by the top 10 life insurance companies in terms of net worth.

Emails and Telephone helplines provide good quality of service to members , as they have the perception that they are interacting with real persons on the other end, however these technologies has a proven drawback which involves customers waiting for responses in a long period of time especially on telephone helplines which requires them to be held on the line for an average duration of 56 seconds or more (Roddy, 2014). This length of time needed for a consumer to avail of the customer relations services offered by companies has been proven to be a primary factor for them to feel less of importance towards that company hence minimizing their retention towards it, as what the U.S. Survey Report (2014) has found out. Other technologies such as SMS has already been proven, by the University of California

concluded with their Group Case Study on Boomerang (2011) , to be of great value for it provides a method with which companies could easily respond to customer requests for an automated response system could be put in place of the human factor needed for the two previously cited (Riley, Schmidt, & Tubin, 2011).

The SMS mode of conversing with consumers has not been explored that much by health insurance companies in the Philippines although with it poses a great promise if utilized properly. The reason as to why this happens is inconclusive and no study has been put up in place to do so.

B. Statement of the Problem

The time-to-respond drawbacks posed about by the leading information and communications technology infrastructures being utilized by health insurance companies which are Email and Telephone Helplines poses a great risk towards the efforts of these companies to promote customer retention for the said drawback leeches consumers' perception of value of the company. No Health Insurance companies in the Philippines has been into SMS- based CRM tools yet. Most companies rely on their Email and Telephone Helplines up to the moment of writing of this paper. A need for a more efficient yet still effective mode of communicating with members of the health care industry is prevalent and SMS poses to be one of the best of them.

The implementation of a Customer Relationship Management (CRM) application which utilizes SMS as one of its core modules requires its integration to the existing systems of health insurance companies. This requires the CRM to communicate with customers whose information already exists at some other infrastructure that has already be in place in the company while taking in consideration the need for the CRM to be independent of those systems so as there will be little to no changes required to be done on them, being sensitive to the legacy data in existence, which leads to the CRM being a plug-and-play application.

For health insurance companies to get the most out of this CRM, it should be able to provide billing notifications on upcoming and missed premiums by itself in an automated fashion so as customers could continuously enjoy their health benefits and for the company's monetary funds to remain stable.

The CRM being a customer-centered tool should also be able to cater various customer groups which exists to cover the most number of possible clientèle from different economic

strata which can be from members belonging in charity-oriented groups whose premiums are being payed by a third party to freelance individuals who take care of their own bills.

CRM on its grounds capitalizes on customer retention which entails its social responsibility of providing utmost customer welfare improvement. This responsibility requires the CRM to have a capability to refer customers to services provided by the company's partner such as seminars, trainings and the like. These partners vary from private institutions to non-governmental organizations providing services besides health insurance with which the company could team up to provide its customers with utmost care hence making them feel much more important. Integration with these services should be done such that all parties, the customers and partner organizations, are well informed once referrals are done which details the CRM to provide notifications to them.

Besides functionalities which require no human intervention, the CRM tool should be able to disseminate information customized by authorized health personnel to its customer which can be filtered by different categories like location, payment type, membership, etc.

The CRM tool is idealized to be a proactive tool with which customers could also send requests besides receive information from. These request should include but not be limited to pin and billing inquiries. Pin in this scenario serves as the customers' tool to communicate with the system so as authentication could be made by the system and vital information will not be available to inappropriate persons.

Tracking of messages should also be made possible so as once fault arise the proper person responsible could be addressed. This tracking should include transaction numbers, timestamps, sender and receiver fields for each message.

The CRM tool being idealized should be a proactive, efficient, effective, secure system that customers and the company could trust.

C. Objectives

The primary goal of this paper is to provide a solution to the implementation of the SMS sending and receiving module, which is a component of the Health Insurance Corporation Customer Relationship Management Tool (HICorpCRM), that health insurance companies could utilize and integrate to their preexisting information and communications technology infrastructures. The secondary objective is to establish a way for health insurance companies to schedule performance of certain jobs pertaining to the accounts of its members. Beneficial optional goals is the integration of the said module in a mock health insurance company web application. A further specification of these objectives and the particulars of the HICorpCRM are as follows:

1. SMS and E-Mail Capabilities

This breakdown of functionality will be made available upon the information available per user. Certain actions may be restricted to SMS or E-Mail alone, if the other is not available for a user, but for the best case both SMS and E-Mail modes of communication will be utilized if possible.

1. Sends the following:

- Automated messages
 - billing to freelancers, 2 weeks prior to the due date of their premium
 - reminder to freelancers of its premium payment on its due date
 - alert to freelancers on missed premium payment
 - referral information to health insurance companies' partner organizations/
companies that provide other social welfare services
 - referral instruction to sponsored and freelancers on referral
- Human- intervention needed messages
 - advisory to sponsored and freelancers on relevant health information

- social intervention update to members
- Receives the following (SMS):
 - pin information request from members
 - referral status request from sponsored and freelancer members

2. Generates the following:

- a 17 character reference number for each message with the following format:

TYMMDD0000000000, specified as:

Index	Specification	Acknowledged Characters	
		Value	Description
T	type of message	B	Billing Prompt (2 weeks prior to due date)
		R	Billing Reminder (on due date of premium)
		A	Billing Advisory (on missed payment)
		N	News and Reminders
		I	Referrals and Interventions
		P	Pin
YY	last two digits of the year the message was processed (e.g. 2011 will have 11 as YY)	00 to 99	numeric combination (0 – 9) of the last two digits of the year
MM	Numerical equivalent of the	01	January
		02	February

	month the message was processed	03	March
		04	April
		05	May
		06	June
		07	July
		08	August
		09	September
		10	October
		11	November
		12	December
DD	day as to when the message was processed	01 - 31	For months January, March, May, July, August, October, December
		01 - 30	For months April, June, September, November
		01 - 28 01 - 29	For the month of February, max is 29 if its a leap year and 28 if not
0000000000	10 digit numeric identifier that is auto-incremental per type of message		

2. File Management

- Accepts the following:
 - Health insurance company member registry via .csv file, to incorporate large data files, with the following column format:
pin, first name, middle name, last name, birth day, birth month, birth year, home region, email address, cellphone number, notifiable, membership type, payment type, premium amount

where columns could take the following values:

pin:	12 character long alphanumeric text
first name:	1-15 characters long alphanumeric text
middle name:	1-15 characters long alphanumeric text
last name:	1-15 characters long alphanumeric text
birth day:	values ranging from 1 – 31 depending on the month
birth month:	numeric values from 1-12 representing months January – December
birth year:	4 character representation of birth year
home region:	13, National Capital Region philippine standard 14, Cordillera Administrative Region
geographic code	01, Ilocos Region 02, Cagayan Valley 03, Central Luzon 04, CALABARZON 17, MIMAROPA 05, Bicol Region 06, Western Visayas

- 07, Central Visayas
- 08, Eastern Visayas
- 09, Zamboanga Peninsula
- 10, Northern Mindanao
- 11, Davao Region
- 12, SOCCSKSARGEN
- 16, Caraga
- 15, Autonomous Region in Muslim
Mindanao

email address: maximum of 255 character long address

cellphone number: maximum of 12 character long number

notifiable: 1, yes

0, no

- membership type:
- 01, Corporate Employees
 - 02, Individually Paying Members
 - 03, Company Sponsored
 - 04, NGO Sponsored
 - 05, Lifetime and Senior Citizens

payment type: 01, Quarterly

02, BiAnnually

03, Annually

04, Monthly

3. Web Application Management

- Establish the following user accounts with respective capabilities:
- **Administrator**

- Logging In
 - Log into the system and gain Administrator privileges
- Customer Management
 - Profiles
 - Register a new customer
 - Update an existing customer in the system
 - Upload a CSV file containing an archive of customer's details
 - List all customers in the system
 - Messaging
 - Generate referrals for customers
 - Send referral information and instruction to member
 - Send referral notification to partner companies' authorized personnel
 - Advisory
 - Send advisories to customers utilizing filters:
 - Home Region
 - Membership
 - Payment Mode
 - History
 - View a list of all the messages processed by the system
 -
- Partner Company Management
 - Company
 - Register a new partner company into the system
 - Update a system existing company's details
 - List all companies registered into the system

- Services
 - Register a new service into a company existing into the system
 - Update a service of a company in the system
 - List all services belonging to a company in the system
- Personnel
 - Register a new personnel into a company existing into the system
 - Update a personnel of a company in the system
 - List all personnel belonging to a company in the system
- System
 - Accounts
 - Register new users of the web applications
 - May provide privileges: Administrator and Standard User
 - Update an existing user of the web application
 - List all the system's users
 - **Standard Account**
 - Logging In
 - Log into the system and gain Administrator privileges
 - Customer Management
 - Profiles
 - Update an existing customer in the system
 - List all customers in the system
 - Messaging
 - Generate referrals for customers
 - Send referral information and instruction to member

- Send referral notification to partner companies' authorized personnel
 - Advisory
 - Send advisories to customers utilizing filters:
 - Home Region
 - Membership
 - Payment Mode
 - History
 - View a list of all the messages processed by the system
 -
- Partner Company Management
 - Company
 - List all companies registered into the system
 - Services
 - List all services belonging to a company in the system
 - Personnel
 - Update a personnel's details
 - List all personnel belonging to a company in the system
- System
 - Accounts
 - Update an personal user account
 - List all the system's users

D. Significance of the Study

Health insurance companies in the Philippines utilize Email and Telephone helplines in order to cater the needs of their consumers. These ICT infrastructures poses drawbacks involving time-to-wait problems (Roddy, 2014) which has already been proven to be a factor by which customers feels less value of significance for the companies which they subscribe to hence less possibility of retention to them (Lee, 2009). SMS on the other hand offers timely and effective delivery of some of the services the two previously stated modes cater (Riley, Schmidt, & Tubin, 2011) . An alert and communications module that could empower health insurance companies with SMS capabilities poses a great possibility of making up to the drawbacks set forth by the technologies already being utilized by health insurance companies for customer relationship management (Roddy, 2014).

This Health Insurance Corporation Customer Relationship Management Tool (HICorpCRM) will be a easy to use, plug-and-play which health insurance companies would have less of a trouble integrating to their preexisting systems. This CRM application will be able to provide support to legacy data while staying independent from other infrastructures set upon by the company. Hence setup and utilization can be done within a day's time with little expertise of setting it up.

This tool will also be able to help the company reach out to their customers and provide them with up-to-date billing updates, so as customers could continue enjoying their health insurance benefits, promotes better customer-corporate interaction, increase of customer satisfaction and stabilization of the company's funds for their customers are constantly updated with regards to their billing.

Having the capability to categorize members on various economic stratas and locations,

messages could be provided to them efficiently for targeting the appropriate audience could be done easily by the filters set about by these categories.

Given that the tool has a way to integrate the health insurance company's services with those of its partners, customers will be able to view the company as a one-stop-shop where all their errands with regards to their welfare could be addressed like seminars, trainings, etc. Proper notifications being received by the members and these partner organizations builds up the an identity to the health insurance company as entity which they could trust will be able to take care of the bond they have formed in between them.

The functionality of the tool which enables an authorized personnel to disseminate news and updates to a targeted audience ensures their customers that they do not miss a thing or information that may pose as very vital to them. This also enables the company to have a way to establish close ties with their customers for these messages could be crafted the way they would want it to be. A personal touch of them could be added with these messages they send.

This CRM tool also empowers customers by giving them the ability to send inquiries to the health insurance company on matters which vary from pin requests to billing inquiries. This pro-activeness instills to the customers the thought that they are valued by the company for the company responds quickly and as effectively as possible to them.

The authentication being required by the system for customer sent inquiries induces a sense of security for vital information like pins, billing details could only be provided to clientèles who provide the proper set of credentials along with their requests.

Fault correction is easier to be done for tracking mechanisms like transaction numbers,

sending and receiving persons are being monitored by the system hence fallacious messages that could possibly be sent out through the tool will have the responsible personnel properly sanctioned for his/ her actions. Verification of transactions of customers by authorized health insurance company could also be done through this.

This tool is a proactive, efficient, effective, secure system that customers and the company could enjoy.

E. Scope and Limitations

This Health Insurance Corporation Customer Relationship Management Tool (HICorpCRM) shall be entirely in accordance with the specifications which has already been laid out in the Objectives module of this chapter. Further specifications as to the scope of this paper are as follows:

All functionalities stated in the objectives will be done

1. This paper must be able to produce an implementation of the HICorpCRM which conforms to the following conditions:
 1. The functional requirements laid upon in the objectives part of this chapter
 2. The database structure provided by customer
 3. Technical documentation of the said module which includes the following but are not limited to:
 1. Technical setup of resources, which considers:
 1. GSM modem
 2. MySQL database
 2. Module setup
 3. Module usage
 4. Module restrictions
 2. Legacy data support will be done via a csv file, containing the membership detail of the health insurance company, which will be crunched by HICorpCRM and integrate to its system.

With the provisions on the functionalities and tasks this HICorpCRM development will be accomplished with associated limitations, specifically the following:

1. Conforms to the limitations found on SMS which is inclusive of the following:
 1. Messages sent not to contain special characters not included in this list: : . , ? | ; - + # * () \ ' " _ @ / < > =

F. Assumptions

1. The data file containing the members of the company is to be provided via csv file by the health insurance company which conforms to the format set about by the objectives of this chapter.
2. Instructional materials will be provided by the company detailing format of messages to be utilized in communicating with HICorpCRM.
3. Necessary updates to the customers account should be properly addressed by the customer in the soonest possible time for them to properly avail of the services being offered by the health insurance company. The customer not doing so cannot hold the insurance company liable for any damages that could be incurred through such scenarios.
4. Delivery of messages to the customers and partner organizations cannot be guaranteed by the health insurance company for it has an external dependency with the SMS center which provides it the capability to send and receive messages. Archiving being done by the SMS center does not ensure the customers that they'll receive messages they didn't get while they are unreachable by it.
5. Server time and date are en-synced with Philippine Standard Time.
6. Server's integrity will not be covered by this Proposed Generic Health Insurance Company Customer Relations Management Tool.

II. Review of Related Literature

Customer Relationship Management, CRM, has proven itself to be a vital aspect of company operations as FrugalTech (2014) exemplifies in their discussion of it. With the market being saturated by companies playing on similar fields, service providers nowadays focus more on customer retention rather than on acquiring new subscribers to increase market shares (Roddy, 2014). As U.S. Survey Report (2014) puts it, innovations in the Information Technology field has enabled companies to have a better reach to their subscribers via the various Information and Communications Technology (ICT) infrastructures made available to them wherein some of which are utilizing e-mail, cellular calls, short messaging services (SMS) etc. Of all of these tools, short messaging service stands to be one of the most feasible in the Philippines for the country has posed as the undisputed SMS leader in the World back in 2011 as GMANews reports, wherein 600 text messages are being sent a month, an intensity which is 43% more than US counterparts. With the findings of BuddeCom (2012), SMS in the Philippines continuously proves to be a very great tool for the said endeavor as the Philippines had a 110% mobile penetration, ratio of mobile users to population, in 2014 as 110 million mobile subscribers came about when about only 100 million registered citizens exists in the Philippines back then. It is such a great tool for marketing that service providers are able to charge companies around Php 500 per customer of theirs who are to receive text blasts from them, a costly damage to some but not to these companies for returns are 10x as much of the cost they had in this strategy as what is reflected by statistics in the US, as indicated by FrugalTech (2014), Mobile Marketing proves to be a key growth area in customer relationship management for it offers immediacy and interactivity (Barnes, S., Scornavacca, E., 2004) hence SMS marketing when implemented righteously leads to consumers having greater perceived value which leads them to see the company to which they subscribe to as having better service quality which induces to them loyalty and retention to it (Lee, 2014).

However, the benefits of SMS based CRM tools has not been that well utilized by Philippine companies and most institutions also take a narrow view of CRM hence its benefits have been limited. Although with the emergence of second generation CRM which encompasses the totality of an organization (Enterprise CRM), success in the utilization of the such has still not been observed (Peppard, 2000).

SMS marketing providers vary from those providing monthly text blasts to companies offering application program interface (api) with which companies could use by adding them to web applications that they have commissioned so as to have a better customer relations model for they could attach modules pertaining to customer's finances, services, etc. with it, a feature that is rarely provided by those of the latter. For a customer relationship management tool to be successful it must be a cross-functional, process-oriented approach at strategic levels which includes strategy development, value creation, multichannel integration, information management and performance assessment processes that all gears towards the creation of a singular tool for a company and its customer's utilization (Payne, Frow, 2005). The Philippine government itself, especially the Civil Service Commission admits that it would make easier for people to contact their agencies through SMS based CRM tools, the reason behind its propulsion for a study for TXT CSC – an SMS based service of the said agency for reporting corrupt and unlawful government personnel (Lallana, C., Civil Service Commission).

The enumerable pros brought about by SMS marketing strategies has with it, cons which are not little to be ignored. Some of these cons involves security and reliability of the technology being utilized (Lamarre, Galarneau, &Boeck, 2012). Security in these terms comprises of the authenticity and authorization of the users of the service to which Banking companies give solution to by encrypting the messages they send and receive from customers (Lamarre, Galarneau, &Boeck, 2012). Reliability on the other hand requires the service to always be up

for the customers to avail and for the system to provide accurate and precise information being required by consumers (Lamarre, Galarneau, &Boeck, 2012). Integrity is also a factor in consideration for no assurance could be made as to the sanity of the data transferred to SMS. Suggestions as to keeping it includes the messages being encrypted symmetrically or assymmetrically using keys shared by parties although doing so requires a much more heavier burden to the SMS centers and mobile peripherals (Medani A., Gani A., Zakaria, O., Zaldan, A., Zaldan, B. ,2011). Other options as to how security will be improved is to mimic the handshake protocols being done in network communications where in a dedicated path is allotted between devices to communicate by sending SYN and ACK messages between them (Moshir, S., Moshir, K., Shafiezadeh, H., 2009). In addition to these two concerns is the dangers of SMS marketing posing as a span tool to send unsolicited messages to people hence mobile marketing permission and acceptance are core issues needed to be settled in the implementation of such (Barnes, S., Scornavacca, E., 2004)

A similar system which uses SMS as a primary component of its customer relationship management tool is the one being utilized by the Land Transportation Office (LTO) which it named as TextBilis. It is a system deployed by LTO so as people especially car buyers could check if a car they are eyeing has outstanding records or not. This is done by sending a formatted message to LTO indicating the plate number of the car and LTO will respond with the details of the car like its color, registration date and current status. The said system was featured at TopGearPH, Tsikot and MotorCyclePhilippines, organizations which has built credibility as leading online magazines for automobile concerns. These institutions boasts of the systems' capability to respond to requests after a minute or so. However besides the great promise of the system being a quick way to access some problems were reported as to the accuracy of the results the said system yields. AutoIndustriya's July 2014 report states cases of a Mitsubishi Strada and a Honda Civic which where both reported to the authorities as

carnapped yet the system prompted them that no outstanding record exists for the said two cars, along with some discrepancies as to the date of registration of the civic which the system pegs at 2013 though it was stolen in 2009. As of the writing of this document the said TextBilis service of LTO is undergoing maintenance and improvement.

This paper aims to provide an alert and communications module which health insurance companies that would cut costs for them by having a local implementations of CRM tools utilizing SMS and E-Mail capabilities while ensuring security for its members via authenticating them through their IDs which is expected that only them knows about.

III. Theoretical Framework

This paper utilizes various technologies available in the Computing Sciences field to accomplish its goal of crafting a web application adhering to standards set forth in the objectives part of this paper. These areas of knowledge comprises of, but not limited to, the following:

Health Insurance Industry

Health Insurance in the Philippines boasts as one of Asia's pioneers with Former President Ramos' signing of Republic Act No. 7875 which aims to provide every Filipino with health insurance coverage hence the creation of the Philippine Health Insurance Corporation (PhilHealth) (lapps, 2015). Although being lambasted by many inadequacies of it PhilHealth sets up as a strong and largely politically independent institution for the development of Social Health Insurance in the Philippines (Obermann, K. Jowett, MR., Alcantara MO., Banzon, EP., Bodart, C.2006). The kick start made by PhilHealth initiated the boom of the health insurance industry now with 31 big companies competing in its playing field, enhancing its share in the Philippines' GDP into 1.2% (Commission on Insurance).

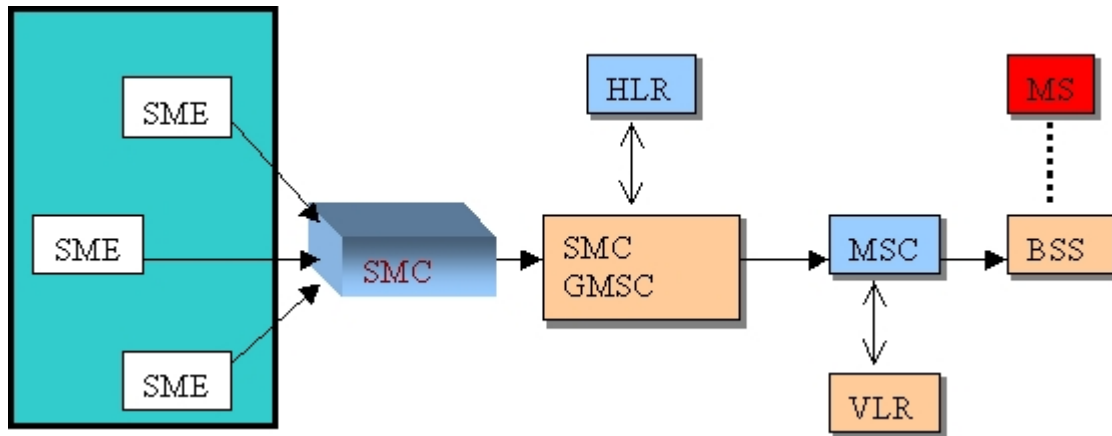
Short Messaging Service (SMS)

A mechanism utilized for the delivery of short messages, varying from 70 to 224 characters depending on the encoding, via mobile networks. Short Messaging Service (SMS) caches messages on a Central SMS that constantly queries receivers' availability for messages to be forwarded to them. Caching of messages by mobile networks guarantee receivers that they get the message once they get on-the-line even though those messages were sent while they are off. Caching lasts for some time depending on

the discretion of the networks to which they are hosted (Gupta, P.).

SMS proves to be a very feasible CRM pathway in the Philippines for the availability of network access is still of great scarcity in the country.

The abstracted process as to how SMS works is as follows:



The Short Message Center (SMC) serves as the router of the system, sending messages to and from Short Message Entities (SME) which are located at mobile stations. These SMEs receive and send short messages to mobile devices.

The SMS gateway MSC (SMS GWMS) serves as the pathway across various telecommunication companies. The SMS GWMS uses the Home Location Register (HLR) which houses the information on the mapping of where mobile devices are hosted in the various SMEs. The SMS GWMS point of contact with other telecommunication infrastructures is the Gateway MSC (GMSC) (Gupta, P.).

Kannel WAP and SMS Gateway

WAP Gateways enables devices to communicate through Wireless Markup Languages which is similar to the Unified Resource Locators (URLs) being used in HTMLs to send and receive messages (SendBulkSMS, 2014).

Kannel is the Free and Open Source Software (FOSS) Community's take on WAP and SMS gateways. It utilizes WAP to push and pull SMS while utilizing mobile internet connection (Kannel).

Kannel setup in a machine is done via:

Kannel Configuration and Setup

System Requirements

- Software Environment
 - C compiler and development libraries and related tools.
 - The gnome-xml (a.k.a. libxml) library, version 2.2.0 or newer. We recommend that you use libxml version 2.2.5. If you are installing it from your distribution's packages, you'll need libxml2-dev in addition to run-time libxml2 package libraries.

Note: there is a bug in libxml version 2.2.3 and 2.2.4 that causes problems with character encoding in attributes. This problem is fixed in 2.2.5. See <http://xmlsoft.org/xml.html>.

- GNU Make.
- POSIX threads (pthread.h).
- Ubuntu-based linux distros (\geq release 14.04)

- WVDialConf
- Minicom
- Wader-Core
- Hardware Specifications
 - 400 MHz Pentium II
 - 128 MB RAM
 - Network Connectivity (≥ 1.0 Mbps)
 - Huawei E160E GSM modem
 - GSM – Broadband Modem

Configuration and Setup

1. Establish needed software requirements for the Ubuntu-based linux distro
 1. * Execute terminal commands to setup gnome-xml:


```
sudo apt-get install libxml2 libxml2-dbg libxml2-dev libxml2-doc libxml2-
utils libxml2-utils-dbg
```
 2. * Install kannel via terminal commands:


```
sudo apt-get install kannel kannel-dev kannel-docs kannel-extras kannel-
sqlbox
```
 3. * Include minicom and wvdial for modem configuration via terminal
 commands:


```
sudo apt-get install minicom wvdial
```
 4. Restart device to ensure synchronization of new applications with system
2. Configure GSM modem device with system

1. Make the system recognize the GSM modem via terminal:

```
* sudo wvdialconf
```

Take note of the configurations yielded by the recognition similar to:

Found a modem on /dev/ttyUSB0.

Modem configuration written to /etc/wvdial.conf.

```
ttyUSB0<Info>: Speed 9600; init "ATQ0 V1 E1 S0=0 &C1 &D2
+FCLASS=0"
```

```
ttyUSB1<Info>: Speed 9600; init "ATQ0 V1 E1 S0=0 &C1 &D2
+FCLASS=0"
```

2. Configure system to utilize proper settings for the utilization of the GSM modem via minicom:

```
* sudo minicom -s
```

> Select "Serial Port Setup" > Change "A – Serial Device" > Set A – Serial Device with settings found through wvdial, however for this device let A be set to "/dev/ttyUSB1" for Huawei E160E modems utilize "/dev/ttyUSB1" for its GSM modem > Modify "E – Bps/Par/Bits" with the speed identified via wvdial which is "9600" > Select "Save setup as df" > Select "Exit" so system could configure the modem with configurations set

3. Let the system recognize the GSM modem more firmly via terminal commands:

```
lsusb
```

Take note of the configuration brought about by Huawei E160E like:

Bus 002 Device 002: ID 12d1:1003 Huawei Technologies Co., Ltd. E220
HSDPA Modem / E230/E270/E870 HSDPA/HSUPA Modem

* sudo nano /etc/modules

Append "usbserial vendor=0x12d1 product=0x1003" to the said file

4. Restart system so as it could properly recognize the modem.
5. Allow kannel to have priviledges to the GSM modem via terminal commands:

* sudo usermod -a -G dialout kannel

* sudo chown kannel /dev/ttyUSB0

* sudo chown kannel /dev/ttyUSB1

3. Setup Kannel Configuration via command:

sudo nano /etc/kannel/kannel.conf

Replace the contents of kannel.conf with:

```
-----  
group = core  
admin-port = 13000  
admin-password = bar  
status-password = foo  
smsbox-port = 13001  
box-deny-ip = "*. *.*.*"  
box-allow-ip = "127.0.0.1"  
log-file = "/var/log/kannel/bearerbox.log"  
log-level = 0
```

```

group = modems

id = "huawei"           // replace with gsm modem manufacturer
name = "huawei"        // replace with gsm modem manufacturer
detect-string = "huawei" // replace with gsm modem manufacturer
detect-string = "e160e" // replace with gsm modem make
                        // identify init-string and speed via wvdialconf
init-string = "ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0;"
init-string = "AT+CNMI=1,2,0,1,0;"
speed = 9600

group = smsc

smc = at

modemtype = "huawei"   // replace with gsm modem manufacturer
device = "/dev/ttyUSB1" // identify via wvdialconf
speed = 9600           // identify via wvdialconf
validityperiod = 167

group = smsbox

bearerbox-host = "127.0.0.1"

sendsms-port = 13013

global-sender = 13013

sendsms-chars = "0123456789 +-"

log-file = "/var/log/kannel/smsbox.log"

log-level = 0

access-log = "/var/log/kannel/access.log"

```

```
group = sendsms-user
username = kannel
password = kannel
max-messages = 3
concatenation = true
```

```
group = sms-service
keyword = default
```

```
// URL to sms receiver logic
```

```
get-url = "http://localhost/sagip/index.php/receive/receivesms/?sendingnumber
%p&receivingnumber=%P&message=%b&smcid=%i"
accept-x-kannel-headers = true
catch-all = true
max-messages = 3
concatenation = true
```

4. Run Kannel with terminal commands:

```
* sudo bearerbox -v 0 /etc/kannel/kannel.conf
* sudo smsbox -v 0 /etc/kannel/kannel.conf
* sudo tail -f /var/log/kannel/bearerbox.log
```

Note:

* - Terminal commands which requires administrative priviledges to execute. Need for authentication credentials is required.

Task Scheduling and CRON

Task Schedulers enable developers to set process to automatically perform tasks on predefined times, days, or frequencies. Application of scheduling could vary from events being triggered once certain stimulus happens up to events set to run on specified times. Task Schedulers commonly run as daemon processes in systems in which they are deployed (Microsoft).

CRON is a Linux system daemon used to execute desired tasks at designated times. Schedules are set on the crontab which can be edited via command-line utilities (Ubuntu).

Setting cron jobs can be done via:

1. Follow crontab's format string:

```
1 2 3 4 5 /path/to/command arg1 arg2
```

where:

1 – minute (0-59)

2 – hour (0-23)

3 – day (0-31)

4 – month (0-12)

5 – day of the week (0-7)

2. To edit your cronjobs key in:

```
crontab -e
```

3. Quick tip for running urls:

```
* * * * * wget http://example.com/check
```


PHP and CodeIgniter

PHP: Hypertext Preprocessor (PHP) is a server-side interpreted, non-compiled, scripting language. PHP can be integrated along with HTML to serve purposes ranging from user interface to logic. Although it has been 11 years since it was developed, 65.45% of the top 1 million websites still utilizes the said technology, some of which are Facebook and Wikipedia (PHP).

Codeigniter is an open-source PHP framework developed in 2006 (Ellislab). It posed as the industry leader in 2008 for it has risen from that time, when frameworks leaves large footprints and undocumented, as a deviant utility (Peck, J.). As for 2014, CodeIgniter now lies under the management of the British Columbia Institute of Technology .

CSS and Foundation

Cascading Style Sheets (CSS) defines how HTML elements are to be displayed via style definitions which may be inline or from external files. CSS allows webpages to adapt to various screen sizes or printers. It is independent of HTML and may be used with other XML-based marked up language. This separation of structure of CSS makes it easier to maintain sites and tailor pages to various environments (w3schools).

Foundation is a CSS framework which easily and efficiently scales websites from phones to tablets to desktops. Foundation utilizes the 12 grid column with which it is highly relying on for dynamic web page display. Foundation covers an extensive list of components from drop-downs to progress bars and many others (Zurb).

Information System

An integration of digital and human resources inclusive of hardware, software, trained personnel and the like organized so as to provide storage, computing, distribution and communication of information to organizations, which in turn they use to facilitate their planning, control, coordination and decision making (britannica).

Web Applications

Computer programs whose components, which may or may not entirely reside in the Web, is being downloaded into its clients' devices so as they are empowered with capabilities offered by the functionalities of the said programs which are not limited to database queries, logic implementations using a user interface that is ideally user-friendly in nature (SearchSoftwareQuality).

Client/ Server Architecture

Architectures wherein devices/ processes fall under one of the two categories: server, client. Servers are powerful computers solely dedicated to processing information freeing the clients the necessity to incur great technical specifications as they provide them with the storage and processing speed they may need. Clients on the other end requests these services or resources from the servers via the Web (BusinessDictionary).

MySQL Database Server

MySQL Database Server is a Free, Open Source Software available which implements SQL scripting capabilities (SiteGround). It is a database management system that is very fast, reliable, scalable, and easy to use and can be deployed in client/server or embedded systems (MySQL).

Apache Web Server

A public-domain open source HTTP web server which was developed in 1995. Apache HTTP Server Project is available for both UNIX and Windows NT. Apache was crafted to provide secure, efficient and extensible server that is en sync with HTTP standards (Apache).

IV. Design and Implementation

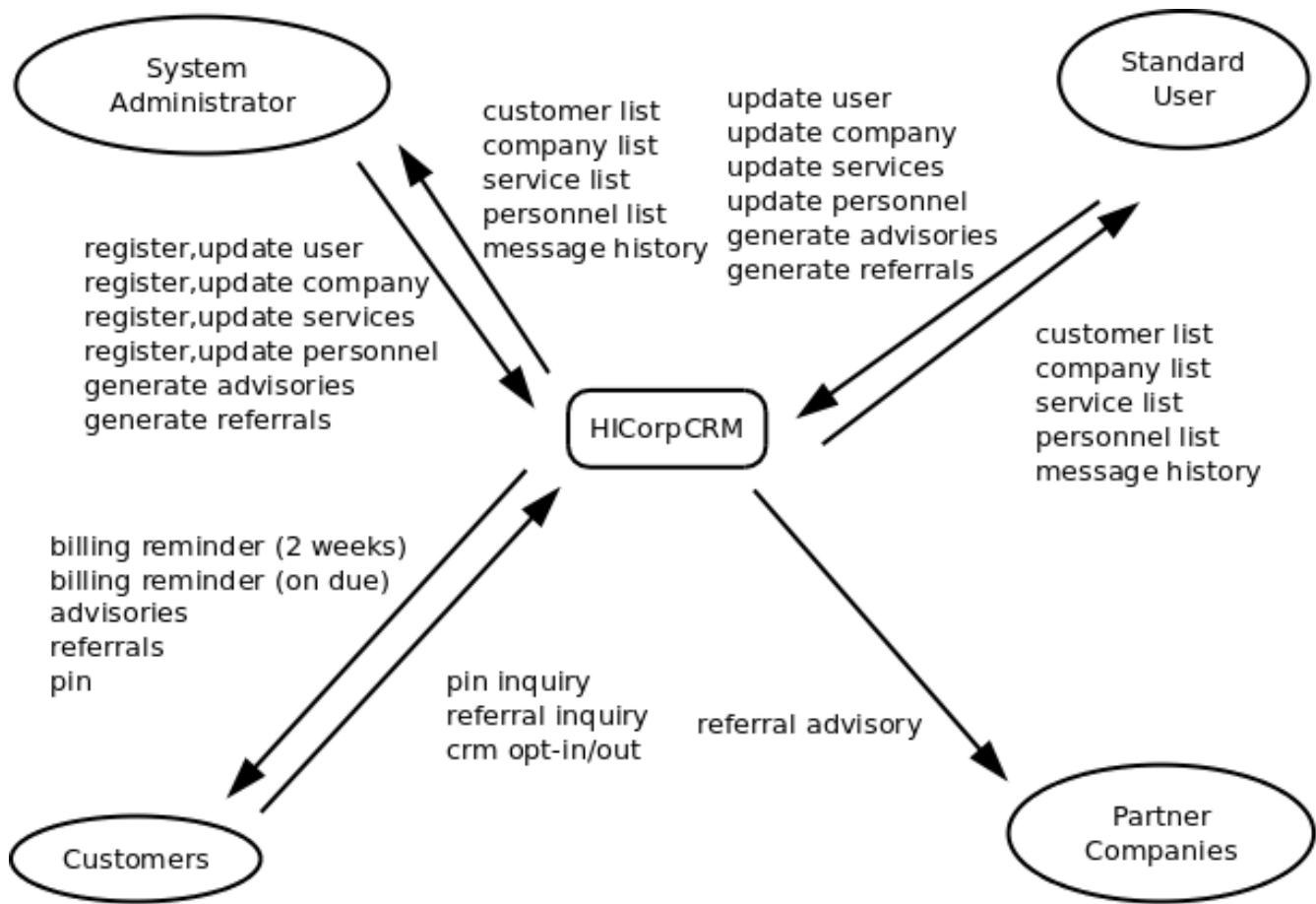


Figure 1.1

Context Diagram

Details how each stakeholder interacts with the system. A System Administrator could register and update users, companies – its services and personnel and also generate advisories and referral. Each action will yield information in list form of the results of these actions which varies from lists of customers, companies – its services and personnel and the messaging history. A Standard User could perform actions similar to a System Administrator with a few constraints specifically in registering users, companies, services, personnel. A customer will be able to receive billing reminders from the system together with advisories and referral in turn he/ she could perform inquiries. A Partner Company receives referral advisories.

Entity Relationship Diagram

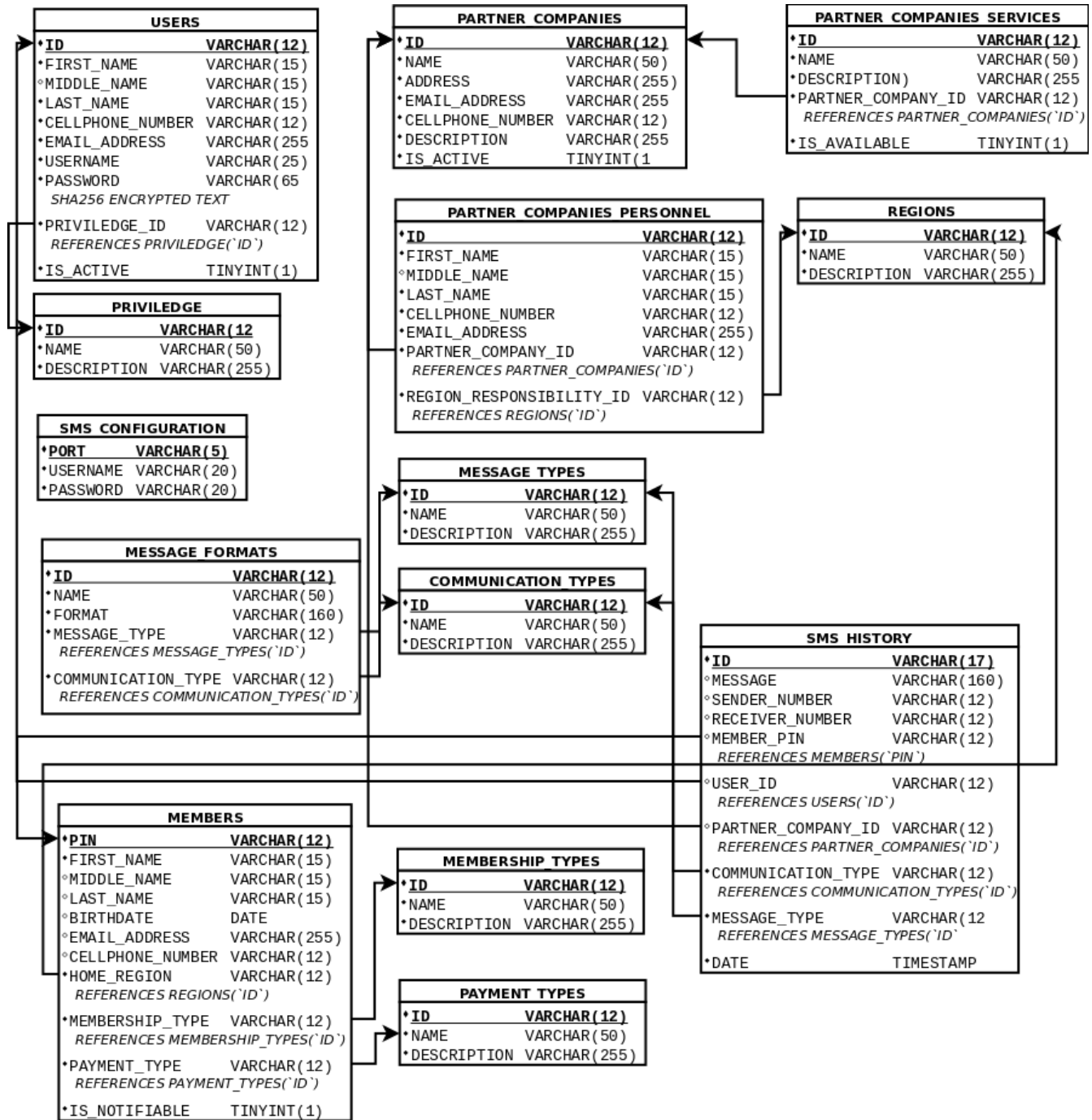


Figure 2.1

Entity Relationship Diagram

Describes how each entity depends to each other. Further discussion of these entities in the schema could be found in the data dictionary.

Data Dictionary

TABLE NAME	PURPOSE OF TABLE	FIELD	DATA TYPE	ATTRIBUTES	REFERENCED TABLE	DESCRIPTION
SMS_CONFIGURAT ION	Store configuration settings that the web application uses to establish wap services with KANNEL	PORT	VARCHAR(5)	PRIMARY NOT NULL		Network port for KANNEL adapter
		USERNAME	VARCHAR(20)	NOT NULL		KANNEL username for sms-center
		PASSWORD	VARCHAR(20)	NOT NULL		KANNEL password for sms-center
PRIVILEGE	Categorize users of the web application.	ID	VARCHAR(12)	PRIMARY NOT NULL		Privilege type identifier
		NAME	VARCHAR(50)	NOT NULL		Privilege
		DESCRIPTION	VARCHAR(255)	NOT NULL		Capabilities of privilege type
PARTNER_COMPANIES	Details of companies and/ or organizations with which health insurance companies partner up to provide better welfare service to its customers.	ID	VARCHAR(12)	PRIMARY NOT NULL		Partner company identifier
		NAME	VARCHAR(50)	NOT NULL		Partner company
		ADDRESS	VARCHAR(255)	NOT NULL		Address of partner company
		EMAIL_ADDRESS	VARCHAR(255)	NOT NULL		Company email of partner
		CELLPHONE_NUMBER	VARCHAR(12)	NOT NULL		Company cellphone number of partner
		DESCRIPTION	VARCHAR(255)	NOT NULL		Partner company description
		IS_ACTIVE	TINYINT(1)	NOT NULL		Identifier if the company is still in partnership
REGIONS	Regions of the Republic of the Philippines	ID	VARCHAR(12)	PRIMARY NOT NULL		Region identifier
		NAME	VARCHAR(50)	NOT NULL		Descriptor of the Region
		DESCRIPTION	VARCHAR(255)	NOT NULL		Areas included in the Region
MEMBERSHIP_TYPES	Member categorization as Corporate Employees, Freelancers, Charity, Sponsored, Lifetime and Senior Citizens	ID	VARCHAR(12)	PRIMARY NOT NULL		Member Category identifier
		NAME	VARCHAR(50)	NOT NULL		Member Category descriptor
		DESCRIPTION	VARCHAR(255)	NOT NULL		Details as to the category
PAYMENT_TYPES	Payment types which maybe Quarterly, BiAnnually, Annually	ID	VARCHAR(12)	PRIMARY NOT NULL		Payment Type identifier
		NAME	VARCHAR(50)	NOT NULL		Payment Type Descriptor
		DESCRIPTION	VARCHAR(255)	NOT NULL		Frequency as to payment
COMMUNICATION_TYPES	Communication which maybe Incoming or Outgoing	ID	VARCHAR(12)	PRIMARY NOT NULL		Communication category identifier
		NAME	VARCHAR(50)	NOT NULL		Communication category descriptor
		DESCRIPTION	VARCHAR(255)	NOT NULL		How a category is performed
MESSAGE_TYPES	Classification of messages whether for billing, reminder, advisory, missed payments, referrals and interventions, feedback or pin	ID	VARCHAR(12)	PRIMARY NOT NULL		Message type identifier
		NAME	VARCHAR(50)	NOT NULL		Message type descriptor
		DESCRIPTION	VARCHAR(255)	NOT NULL		Message type details

Figure 3.1

Data Dictionary

Details how the database is designed. Description of what each entity and row is attached

TABLE NAME	PURPOSE OF TABLE	FIELD	DATA TYPE	ATTRIBUTES	REFERENCED TABLE	DESCRIPTION
MEMBERS	Health Insurance Company's customer details	PIN	VARCHAR(12)	PRIMARY NOT NULL		Member's pin
		FIRST_NAME	VARCHAR(15)	NOT NULL		Member's first name
		MIDDLE_NAME	VARCHAR(15)			Member's middle name
		LAST_NAME	VARCHAR(15)			Member's last name
		BIRTHDATE	DATE			Members birth date yyyy-mm-dd
		EMAIL_ADDRESS	VARCHAR(255)			Member's email address
		CELLPHONE_NUMBER	VARCHAR(12)			Member's cellphone number
		HOME_REGION	VARCHAR(12)	NOT NULL	REGIONS('ID')	Member's region as to where he/she resides
		MEMBERSHIP_TYPE	VARCHAR(12)	NOT NULL	MEMBERSHIP_TYPES('ID')	Membership category he/she is at
		PAYMENT_TYPE	VARCHAR(12)	NOT NULL	PAYMENT_TYPES('ID')	Payment mode member is enrolled with
		IS_NOTIFIABLE	TINYINT(1)	NOT NULL		If messages could be sent to member or not
MESSAGE_FORMATS	Message templates used by the application to send and receive messages.	ID	VARCHAR(12)	PRIMARY NOT NULL		Message format identifier
		NAME	VARCHAR(50)	NOT NULL		Message format descriptor
		FORMAT	VARCHAR(160)	NOT NULL		Message format regular expression
		MESSAGE_TYPE	VARCHAR(12)	NOT NULL	MESSAGE_TYPES('ID')	Message type identifier
		COMMUNICATION_TYPE	VARCHAR(12)	NOT NULL	COMMUNICATION_TYPES('ID')	Communication category identifier
SMS_HISTORY	Archive of incoming and outgoing message with tracking capabilities to identify sender, receiver and transaction number	ID	VARCHAR(17)	PRIMARY NOT NULL		SMS tracking number: TYMMDD00000000
		MESSAGE	VARCHAR(160)			Message
		SENDER_NUMBER	VARCHAR(12)			Sender of message
		RECEIVER_NUMBER	VARCHAR(12)			Receiver of message
		MEMBER_PIN	VARCHAR(12)		MEMBERS('PIN')	Pin of member if applicable
		USER_ID	VARCHAR(12)		USERS('ID')	Identifier of personnel who sent message
		PARTNER_COMPANY_ID	VARCHAR(12)		PARTNER_COMPANIES('ID')	Partner company identifier
		COMMUNICATION_TYPE	VARCHAR(12)	NOT NULL	COMMUNICATION_TYPES('ID')	Communication category identifier
		MESSAGE_TYPE	VARCHAR(12)	NOT NULL	MESSAGE_TYPES('ID')	Message type identifier
		DATE	TIMESTAMP	NOT NULL		Timestamp of message

Figure 3.2

Data Dictionary

Details how the database is designed. Description of what each entity and row is attached.

TABLE NAME	PURPOSE OF TABLE	FIELD	DATA TYPE	ATTRIBUTES	REFERENCED TABLE	DESCRIPTION
SMS_CONFIGURAT ION	Store configuration settings that the web application uses to establish wap services with KANNEL	PORT	VARCHAR(5)	PRIMARY NOT NULL		Network port for KANNEL adapter
		USERNAME	VARCHAR(20)	NOT NULL		KANNEL username for sms-center
		PASSWORD	VARCHAR(20)	NOT NULL		KANNEL password for sms-center
PRIVILEGE	Categorize users of the web application.	ID	VARCHAR(12)	PRIMARY NOT NULL		Privilege type identifier
		NAME	VARCHAR(50)	NOT NULL		Privilege
		DESCRIPTION	VARCHAR(255)	NOT NULL		Capabilities of privilege type
PARTNER_COMPANIES	Details of companies and/ or organizations with which health insurance companies partner up to provide better welfare service to its customers.	ID	VARCHAR(12)	PRIMARY NOT NULL		Partner company identifier
		NAME	VARCHAR(50)	NOT NULL		Partner company
		ADDRESS	VARCHAR(255)	NOT NULL		Address of partner company
		EMAIL_ADDRESS	VARCHAR(255)	NOT NULL		Company email of partner
		CELLPHONE_NUMBER	VARCHAR(12)	NOT NULL		Company cellphone number of partner
		DESCRIPTION	VARCHAR(255)	NOT NULL		Partner company description
		IS_ACTIVE	TINYINT(1)	NOT NULL		Identifier if the company is still in partnership
REGIONS	Regions of the Republic of the Philippines	ID	VARCHAR(12)	PRIMARY NOT NULL		Region identifier
		NAME	VARCHAR(50)	NOT NULL		Descriptor of the Region
		DESCRIPTION	VARCHAR(255)	NOT NULL		Areas included in the Region
MEMBERSHIP_TY PES	Member categorization as Corporate Employees, Freelancers, Charity, Sponsored, Lifetime and Senior Citizens	ID	VARCHAR(12)	PRIMARY NOT NULL		Member Category identifier
		NAME	VARCHAR(50)	NOT NULL		Member Category descriptor
		DESCRIPTION	VARCHAR(255)	NOT NULL		Details as to the category
PAYMENT_TY PES	Payment types which maybe Quarterly, BiAnnually, Annually	ID	VARCHAR(12)	PRIMARY NOT NULL		Payment Type identifier
		NAME	VARCHAR(50)	NOT NULL		Payment Type Descriptor
		DESCRIPTION	VARCHAR(255)	NOT NULL		Frequency as to payment
COMMUNICATION_TY PES	Communication which maybe Incoming or Outgoing	ID	VARCHAR(12)	PRIMARY NOT NULL		Communication category identifier
		NAME	VARCHAR(50)	NOT NULL		Communication category descriptor
		DESCRIPTION	VARCHAR(255)	NOT NULL		How a category is performed
MESSAGE_TY PES	Classification of messages whether for billing, reminder, advisory, missed payments, referrals and interventions, feedback or pin	ID	VARCHAR(12)	PRIMARY NOT NULL		Message type identifier
		NAME	VARCHAR(50)	NOT NULL		Message type descriptor
		DESCRIPTION	VARCHAR(255)	NOT NULL		Message type details

Figure 3.3

Data Dictionary

Details how the database is designed. Description of what each entity and row is attached.

Object Diagrams

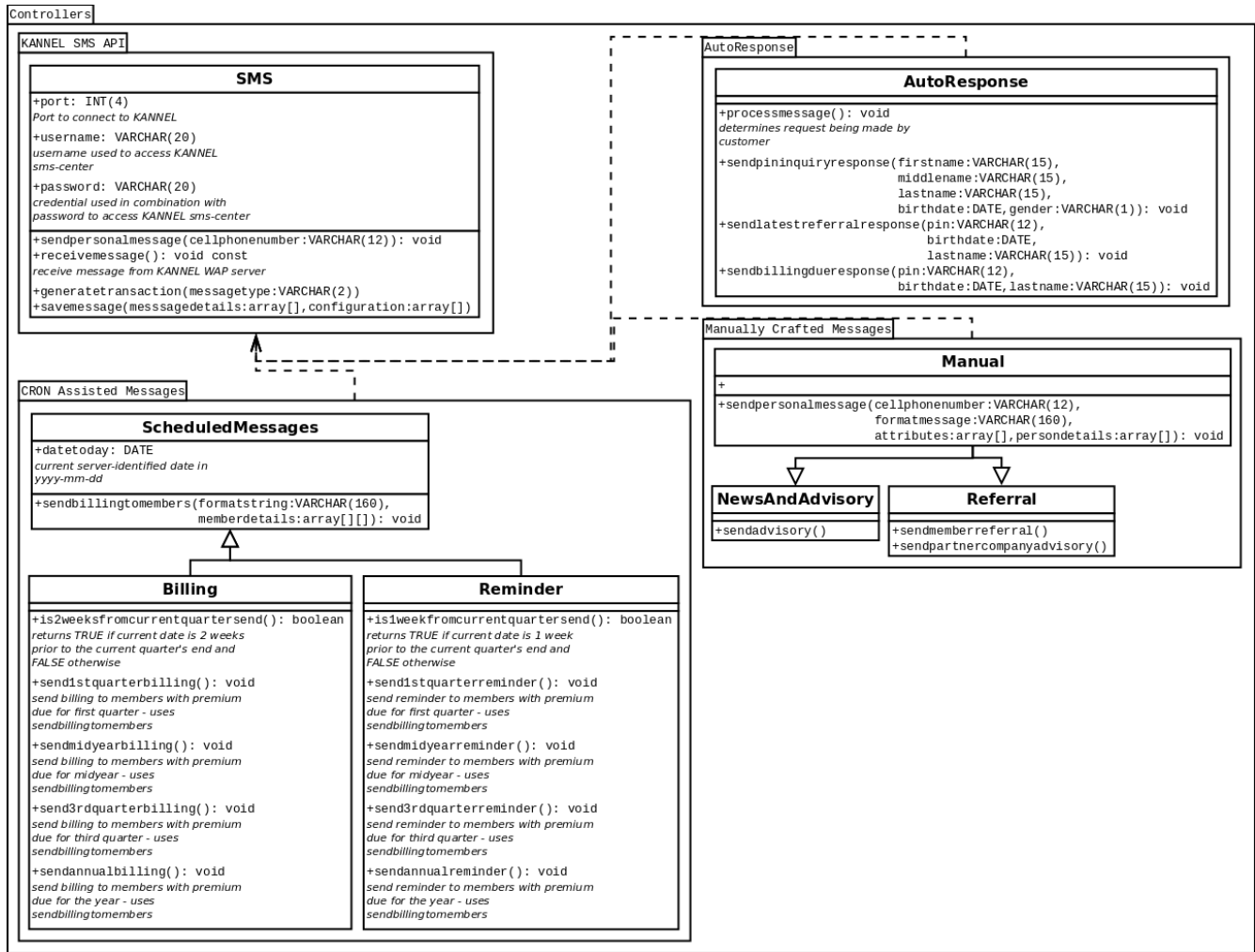


Figure 4.1

Controllers' Object Diagram

Details how core features of the HiCorpCRM functions in tandem with others. SMS serves as one of the core for it manages customer sent and received messages and also how the wap/sms gate way is configured. Scheduled Messages tackle the need for scheduled messages especially those concerning billing. It is being utilized in coordination with CRON. AutoResponse handles inquiries from the customer and serves information queried from the script generated by their inquiries. Billing serves reminders 2 weeks prior to a customer's premium due date. Reminder prompts customers on the day that their premium is due.

Manual caters to the human-induced messages to customers. Advisory manages messages crafted by employees and sent to customers regarding up-to-date news and reminders to the customers. Referral generates referral request and sends out information to the member and advisory to the partner companies regarding referrals made.

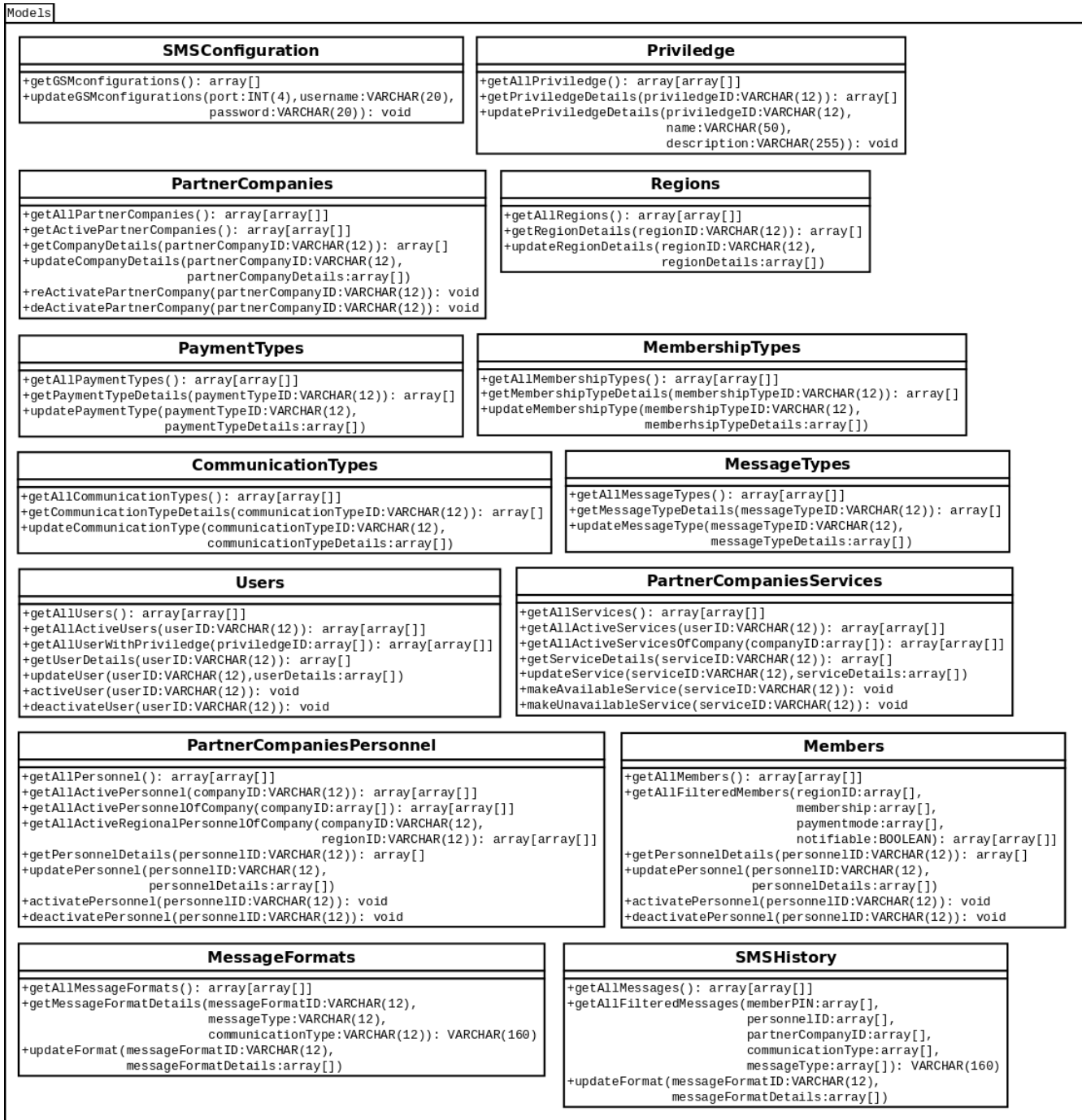


Figure 4.2

Models' Object Diagram

Specifies how models are to be crafted for the system. Details as to what each model processes can be traced back to the data dictionary for there is a one-to-one correspondence between a model and an entity,

SMS Formats

Outgoing

- Billing : <company_name>: <customer's first name>! Your
P<premium_amount> <payment_type> health insurance
premium is due on <date>. TN: <transaction_no>
- Billing Reminder: <company_name>: <customer's first name>! Today is the
due for your P<premium_amount> <payment_type>
health insurance premium. TN: <transaction_no>
- Billing Alert: <company_name>: <customer's first name>! You missed
your P<premium_amount> <payment_type>
health insurance premium. TN: <transaction_no>
- Member Referral: <company_name>: <customer's first name>! Your
<partner_company> <referred_service> was waived. See
<personnel_name> at their <address> office. TN:
<transaction_no>
- Partner Referral: <company_name>: <customer's first name>, with PIN
<member_pin> is referred to <service> of your organization.
TN: <transaction_no>
- Advisories: <company_name>: <advisory> TN: <transaction_no>
- Successful PIN inq.: <company_name>: <customer's first name>! Your PIN is <pin>.
Please keep this PIN safe for future transactions.
- Failed PIN inq.: <company_name>: No records found for your PIN inquiry
credentials.
- Successful Ref inq.: <referral_message>
- Failed Ref inq.: <company_name>: <customer's first name>! No Referral

Request has been made yet with your account.

Successful Opt: <company_name>: <customer's first name>! Your request to opt-<option> of the crm service has been approved.

Incoming

Pin Inquiry: PIN/<first name>/<middle name>/<last name>/<birthdate>

Advisory:

ADVISORY/<pin_number>/<last_name>/<birth_date>/<in/out>

Latest Referral Inquiry:

REFERRAL/<pin_number>/<last_name>/<birth_date>

Minimum Technical Architecture

Server

- Disk Space: 800MB free memory
- Processor: 2 CPU Cores, 64-bit (≥ 1.0 GHz)
- RAM: 2 GB RAM
- GSM: Huawei E160 Series GSM Modem
- Software: Linux, Ubuntu-based Operating System (≥ 14.04 LTS)

Apache Web Server

PHP 5.6 Compliant

MySQL Database

Kannel WAP Gateway

ANSI-C Compiler

GNU C Compiler (GCC)

Build System

Perl 5

GNOME-XML (libxml, libxml2-dev, libxml2) library ($\geq 2.2.5$)

GNU Make

POSIX threads (pthread.h)

- Other: Accurate Time Keeping

Client

- Software: Web Browser
 - Internet Explorer (≥ 9)
 - Google Chrome
 - Mozilla Firefox
 - Safari

V. Results

Series of images portraying how the system looks together with the description as to how each page works to serve the unity of the goals of the information system.

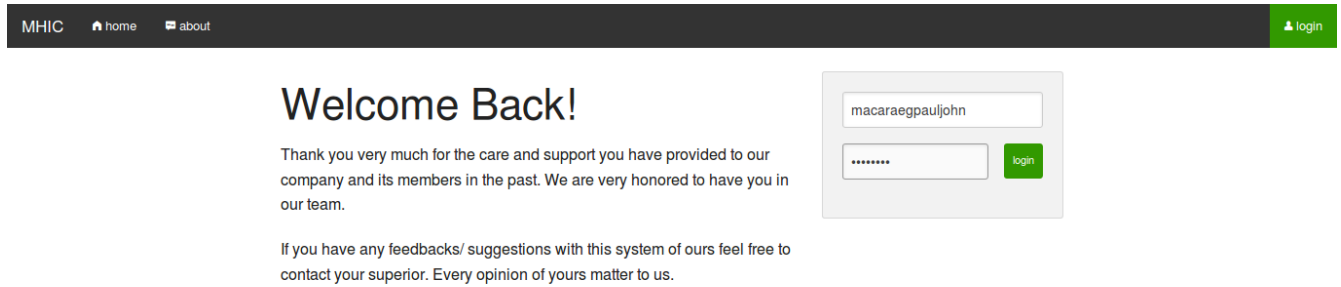


Figure 5.1

Login Screen

User will be able to input credentials in the username and password field. A valid credential will redirect the user into the dashboard and an incorrect one will stay put in the login page.

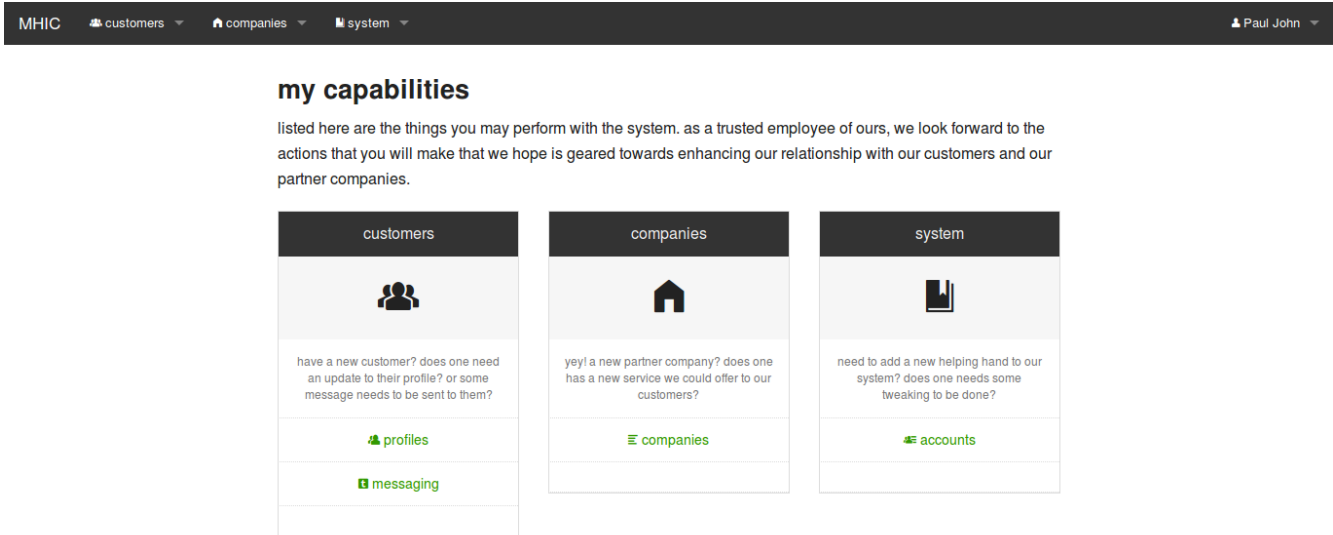


Figure 5.2

Dashboard

Prompts a user of the capabilities he could execute with the system namely manage customers, companies or users of the information system.

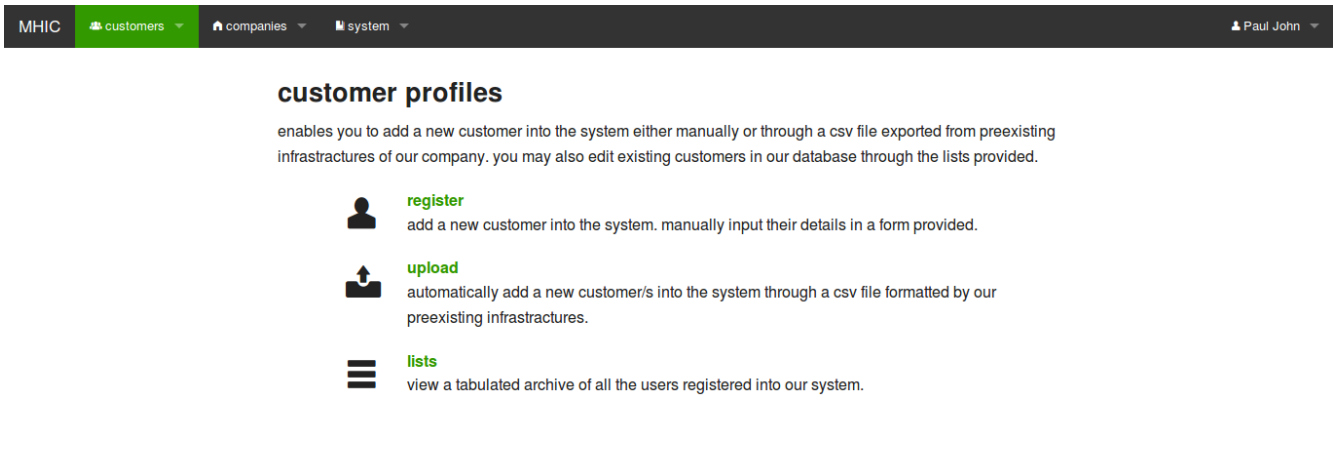


Figure 5.3

Customer Profiles

Things one can do to manage customers in the system: register, upload csv, list

MHIC customers companies system Paul John

customer registration
 please provide the latest details with regards to our customer. the membership pin, cellphone number and email address are the most important fields for this form.

Member PIN

First Name Middle Name Last Name

yyyy-mm-dd

09XX XXX XXX

username@email.com

home region Ilocos Region

membership Company Managed Member payment Monthly

notifiable non-notifiable

register

Figure 5.4

Customer Registration

Allows the registration of a customer. Employees registering a customer is advised not to leave the cellphone number field blank. Unique Member PIN is required.

MHIC customers companies system Paul John

customer upload
 register new customers via csv file generated by preexisting infrastructures.

Browse... No file selected.

upload

Figure 5.5

Customer Upload

Enables users to upload a .csv file containing the customer's details.

MHIC customers companies system Paul John

customer update
 please provide the latest details with regards to our customer. the membership pin, cellphone number and email address are the most important fields for this form.

pin 982716539262

Paul Hamilton Madeleine

1976-12-09

9322922728

iampauljohnmacaraeg@gmail.com

home region Cagayan Valley

membership Company Managed Member payment Monthly

notifiable non-notifiable

update

Figure 5.7

Customer Update

Enables the editing of the profile of the customer.

MHIC customers companies system Paul John

customer profiles
 a collation of a quick summary about our customers. you may do queries with the list through the search field.

Show 10 entries Search:

member's pin	name	membership	payment mode	notifiable
982716539262	Paul Madeleine	Company Managed Member	Monthly	- unnotify
982716540285	Austin Andrew	Individually Paying Member	Quarterly	- unnotify
982716541341	Jake Megan	Charity	BiAnnually	+ notify
982716542430	Maria Irene	NGO Sponsored Members	Annually	+ notify
982716543552	Julia Lucas	Lifetime Members and Senior Citizens	Monthly	+ notify
982716544707	Austin Ryan	Company Managed Member	Quarterly	- unnotify
982716545895	Alexander Cameron	Individually Paying Member	BiAnnually	- unnotify
982716547116	Audrey Dan	Charity	Annually	- unnotify
982716548370	Thomas Sally	NGO Sponsored Members	Monthly	- unnotify
982716549657	Stewart Nicholas	Lifetime Members and Senior Citizens	Quarterly	+ notify

Figure 5.6

Customer List

Lists all the customers existing in the system. You can edit/ change notification of users.

customer messaging

empowers you to generate referrals and broadcast advisories to our customers. you may see a full recap of all the messages sent and received by the system through the history tab.



referral

generate a new referral for a customer. referrals can be availed starting from the day of request generation.



advisory

broadcast up-to-date and relevant advisories to our customers with varying home regions, membership and payment types.



history

have a clear overview of what messages the system has sent and received through SMS in the past.

Figure 5.7

Customer Messaging

Provides functionalities to communicate with the customers via SMS and E-mail

customer referrals

provide welfare services offered by our partner companies to our customer.

9827165392

partner company My Red Cross

select services

Figure 5.8

Customer Referral – Company Selection

Generates Referral for a customer. Provision of Customer's PIN and Company is needed. Will be redirected to service selection afterwards.

MHIC customers companies system Paul John

customer referrals
provide welfare services offered by our partner companies to our customer.

9827165392

partner company My Red Cross

referred service Disaster Management Services

receiving personnel Gwendolyn Pang

refer

Figure 5.9

Customer Referral – Service Selection

Provides option on the services a member could avail from a company.

MHIC customers companies system Paul John

customer advisory
issue up-to-date news and reminders to our customers. filter the receiving audience through our various filters: home region, membership and payment mode

home region

membership

Company Managed Member NGO Sponsored Members

Individually Paying Member Lifetime Members and Senior Citizens

Charity

payment mode

Advisory

refer

Figure 5.9

Customer Advisory

Enables the sending of advisory to customers filtered on their home region, membership type and payment mode.

sms history

log of all the messages that came in and out of the system.

Show 10 entries Search:

message	sender	receiver	communication	message	timestamp
MHIC: Paul! Your MaKiBaKa Educational Assistance referral was waived. See Tita Crissy at their National Capital Region office. TN:11506090000000000		9322922728	Outgoing Message	Intervention and Referral	2015-06-09 12:37:25
MHIC: Paul! Your MaKiBaKa Educational Assistance referral was waived. See Tita Crissy at their National Capital Region office. TN:11506090000000000		09322922728	Outgoing Message	Intervention and Referral	2015-06-09 12:37:25
MHIC: Paul! Your MRC Disaster Management Services referral was waived. See Gwendolyn Pang at their Ilocos Region office. TN:11506090000000002		9322922728	Outgoing Message	Intervention and Referral	2015-06-09 12:52:56
MHIC: Paul, with PIN is referred to Disaster Management Services of		09322922728	Outgoing	Intervention	2015-06-09

Figure 5.10

SMS History

Lists all the messages the system has processed in the past.

companies

gives you the capability to register new companies then their services and personnel. you may edit companies preexisting in this system through the lists view.



-  **register**
add a new partner company into the system. manually input their details in a form provided. you are also required to register services and personnel associated with it.
-  **lists**
view a tabulated archive of all the partner companies registered into our system.

Figure 5.11

Companies

List capabilities one can do with companies. Register a new one or List them all.

MHIC customers companies system Paul John

company registration
 add a partner company into the system together with its most up-to-date details. you will be required to also add services and personnel associated with it.

Acronym Company Name

Company Address

username@email.com

09XX XXX XXXX

Description

active inactive

register

Figure 5.12

Company Registration

Registers a new partner company into the system. Provision of services and personnel is also expected.

MHIC customers companies system Paul John

company partner update
 edit a partner company's details with its most up-to-date information.

MRC My Red Cross

address Bonifacio Drive, Port Area, Manila 1018

email myredcross143@gmail.com

cellphone 09322922728

Born officially in 1947, but with roots that traces back to the revolutionary days, the Philippine Red Cross has truly become the premier humanitarian organization in the country, committed to provide quality life-saving services that protect the life.

active inactive

update

Figure 5.13

Company Partner Update

Updates the details of a pre-existing partner company

MHIC customers companies system Paul John

company profiles

a collation of a quick summary about our partner companies. you may do queries with the list through the search field.

Show 10 entries Search:

acronym	name	services	personnel	active
MRC	My Red Cross	services	personnel	deactivate
MaKiBaKa	MaKiBaKa Foundation	services	personnel	deactivate

Showing 1 to 2 of 2 entries

Previous 1 Next

Figure 5.14

Company Profiles

Lists all partner companies in the system. One may view its services and personnel dashboard here. You can also edit a company's profile via gear widget.

MHIC customers companies system Paul John

My Red Cross details

provides a brief overview of how you could manage My Red Cross's details.



-  **services**
provides a summary of the services My Red Cross offers.
-  **personnel**
provides a summary of the My Red Cross's personnel.

Figure 5.15

Company Details

Quick links to the services and personnel dashboard are here.

My Red Cross's personnel

manage personnel connected with My Red Cross. register new ones, list and update preexisting ones.



-  **register**
add a new personnel to My Red Cross
-  **lists**
list personnel belonging to My Red Cross

Figure 5.16

Company Personnel

Lists all capabilities in managing the personnel of a company. Register or List personnel.

personnel registration

please provide the latest details with regards to the company's personnel. please do not forget to authenticate the region of responsibility of theirs.

First Name	Middle Name	Last Name
09XX XXX XXXX		
username@email.com		
company	My Red Cross	
region covered	Ilocos Region	
<input checked="" type="radio"/> active	<input type="radio"/> inactive	
<input type="button" value="register"/>		

Figure 5.17

Personnel Registration

Registers a personnel to a partner company.

MHIC customers companies system Paul John

personnel update
 please provide the latest details with regards to the company's personnel. please do not forget to authenticate the region of responsibility of theirs.

Gwendolyn Middle Name Pang

09322922728

myredcross143@gmail.com

company My Red Cross

region covered Ilocos Region

active inactive

update

Figure 5.18

Personnel Update

Updates the details of a personnel of a partner company

MHIC customers companies system Paul John

My Red Cross's personnel list
 a collation of a quick summary of personnel of the My Red Cross, [register](#) a new one?

Show 10 entries Search:

name	region covered	active
Gwendolyn Pang	Ilocos Region	deactivate
Regina Nucup	Cagayan Valley	deactivate
Kabalen Katapang	Central Luzon	deactivate
Malakas Dimagiba	CALABARZON	deactivate
Inday Garutay	Bicol Region	deactivate
Ricardo Tiu	National Capital Region	deactivate

Showing 1 to 6 of 6 entries

Previous1Next

Figure 5.19

Personnel List

Lists all personnel of company. Update of a personnel can be made here.

My Red Cross's service

manage services offered by My Red Cross. register new ones, list and update preexisting ones.



-  **register**
add a new service to My Red Cross
-  **lists**
list services belonging My Red Cross

Figure 5.20

Company Services

Provides capabilities as to how an employee could manage the services offered by companies.

service registration
register a new service to our system.

available inavailable

Figure 5.21

Service Registration

Registers a service to a partner company

MHIC customers companies system Paul John

service update
update pre-existing service to our system.

Disaster Management Services

We provide aid to disaster-stricken areas which covers pre, post and onslaught disaster management.

My Red Cross

available inavailable

update

Figure 5.22

Service Update

Edits a service preexisting to a partner company in the system

MHIC customers companies system Paul John

My Red Cross's services list
a collation of a quick summary about services offered by My Red Cross, register a new one?

Show 10 entries Search:

name	availability
Disaster Management Services	make unavailable
Safety Services	make unavailable
National Blood Services	make unavailable
Health Services	make unavailable
Social Services	make unavailable
Red Cross Youth	make unavailable
International Humanitarian Law	make unavailable

Showing 1 to 7 of 7 entries

Previous1Next

Figure 5.23

Service List

Lists all the services available to a partner company.

system accounts

provide authorized personnel, priviledges to utilize this system of ours. view a list of all users authenticated to utilize this crm tool.



register

register a new user of the system. specify the priviledges he could utilize in accessing the crm tool.



lists

view a tabulated archive of all the users of our system.

Figure 5.24

System Accounts

Manage users of the HICorpCRM

user registration

please provide the latest details with regards to our customer. the membership pin, cellphone number and email address are the most important fields for this form.

First Name	Middle Name	Last Name
09XX XXX XXX		
username@email.com		
Username		
Password		
Retyped Password		
System Administrator		
<input checked="" type="radio"/> active <input type="radio"/> inactive		
register		

Figure 5.25

User Registration

Registers a user of the HICorpCRM to the system

MHIC customers companies system Paul John

user update
 please provide the latest details with regards to a user of this system.

System	Middle Name	Administrator
cellphone	09322922728	
email	macaraegpauljohn@gmail.com	
username	administrator	
privilege	System Administrator	
<input checked="" type="radio"/> active <input type="radio"/> inactive		
update		

Figure 5.26

User Update

Updates a preexisting user of the HICorpCRM

MHIC customers companies system Paul John

user profiles
 a collation of a quick summary about registered users of the system. you may do queries with the list through the search field.

Show 10 entries Search:

username	name	privilege	active
administrator	System Administrator	System Administrator	deactivate
macaraegpauljohn	Paul John Macaraeg	Standard User	deactivate

Showing 1 to 2 of 2 entries Previous1Next

Figure 5.27

User Profiles

Lists all the users of the HICorpCRM and their details.

VI. Discussion

The Philippine Health Insurance market shares a relatively profitable part of the country's GDP, although it only composed the 1.2% of it, the reflected amount tops to Php 167,552,000,000. This is the reason why companies sought after the industry of health insurance. As of the writing of this paper, there are 31 top insurance companies existing in the Philippines. This density in the competition grounds poses a new risk to companies which is customer retention. Customer retention involves making the customer feel that he is always remembered hence the need for Customer Relationship Management Tools (CRM). The current trend in the industry now is to have telephone helplines and email pathways to initiate the communication between the companies and the customers. These technologies though having great benefits along with them especially the sense that a real person is transacting with you has its cons with. Long queue lines and waiting time is regarded with telephone helplines while email pathways entails the need for the customers to have access to network infrastructures which the Philippines has not developed fully yet. This is where the SMS enabled CRM tools comes along. With the 110% mobile penetration of SMS in the Philippines, a great avenue can be looked upon by health insurance companies offering wider coverage and faster, dynamic communication between customers through the said technology.

This paper presents an implementation of a generic CRM tool which utilizes SMS modules in communicating with customers. Customers are empowered to receive billing reminders, advisories to the most up-to-date news and reminders from their health insurance company, be given referrals and for them to interact with the system by issuing formatted request messages regarding their pin, most recent referral and for them to opt-in or out of the SMS prompting module of the system.

The health insurance companies on the other end will be able to send out billing reminders 2 weeks before the due of its customer's premium and on the date it is and also reminder customers who missed to pay their premium. With regards to non-automated tasks a personnel will be able to generate referrals for its customers and messages regarding this will be sent out to the customer and the partner company, offering the service, as well. They will also be able to send advisories to a filtered set of members based on the home region where they belong to, their payment modes and membership categories. Add-ons for this web application is the capability for personnel to register other users into the system depending on their priviledges.

VII. Conclusion

The completion of the project Health Insurance Corporation Customer Relationship Management Tool entails a great deal of opportunity for health insurance companies and their efforts to interact and enhance customer relations through their CRM tools. This web application was crafted so as little to no change will be needed by the the company to suit its legacy data with the system. These pros, namely: easy setup of the web application and integration to existing infrastructures, compounds the opportunities offered by SMS itself which timely and dynamic communication between the company and its customers. This along with the other services offered by company like telephone helplines and email pathways will help bridge the gaps between the company and its customers hence increasing the satisfaction of the customers and promoting customer retention in them which in turn gives solution to the primary problem of this paper which is how health insurance companies will be able to enhance their Customer Relationship Management tools (CRM) to promote customer retention in a playing field saturated with 31 top competitors.

VIII. Recommendations

Further improvements to this paper primary focuses on the technical aspects of the development. Besides this the following points is ideally considered for future developments:

1. Revamped development time table. As problems with regards to insufficient development and testing phases have been met.
2. Mocked user acceptance testing as some discrepancies with the idealization of the developer and supposedly target market is foreseen.

Technical Recommendations

A primary concern of this paper is the technical capabilities of the devices utilized in its development, which calls for enhancement for future development cycles:

1. Better server, may be a local implementation or a production one, which ideally runs at clock speeds greater than 3.0 ghz so as processing of the logic will be done in a faster manner given that this application is idealized to process around 100 million records top in a given period of time.
2. Better GSM modem for sending and receiving for an average of 6 minutes I needed to send out 250 messages which entails around 1653 days worth of GSM modem time needed to process 100 million messages to be sent at a time.
3. Oracle database implementation for most of the industry players utilize this instead of MySQL.

IX. References

- Barnes, S., Scornavacca, E. (2004). Mobile marketing: the role of permission and acceptance
- Bhargava, A., Jamison, D., Lau, L., Murray, C. Modeling the Effects of Health on Economic Growth.
- Dimensional Research. (2013). Customer Service and Business Results: A Survey of Customer Service from Mid-Size Companies
- Krishna, V., Anurag, R., Prabhune, S.S. (2014). Short Messaging Service as an Alternative for pushing information to build efficient information passing systems in academic institutions
- Lamarre, A., Galarneau, S., Boeck, H. (2012). Mobile Marketing and Consumer Behaviour Current Research Trend.
- Lee, H. (2009). Factors Influencing Customer Loyalty of Mobile Phone Service: Empirical Evidence from Koreans.
- Lallana, E. (2004) TXT CSC: SMS Service for the Philippines Civil Service Commission
- Medani A., Gani A., Zakaria, O., Zaldan, A., Zaldan, B. (2011). Review of mobile short message service security issues and techniques towards the solution.
- Moshir, S., Moshir, K., Shafieezadeh, H. (2009). Systems and methods for secure short messaging service and multimedia messaging service
- Obermann, K. Jowett, MR., Alcantara MO., Banzon, EP., Bodart, C. (2006) Social Health insurance in a developing country: the case of the Philippines
- Payne A., Frow P. (2005) A Strategic Framework for Customer Relationship Management. Journal of Marketing: October 2005, Vol. 69, No. 4, pp. 167-176.
- Peppard J. (2000). Customer Relationship Management (CRM) in financial services.
- Riley, B., Schmidt A., Tubin, G. (2011). SMS in Financial Services: Accessing Your Customers on Their Terms.

Roddy, M. (2014). SMS for Customer Care.

U.S. Survey Report. (2014). The High Demand for Customer Service via Text Messages.

University of California. (2011). Boomerang: UCSF Medical Informatics Group Case Study.

The 1987 Constitution of the Republic of the Philippines

Apache. Apache HTTP Server Project retrieved from

http://httpd.apache.org/ABOUT_APACHE.html

Encyclopedia Britannica. Information System retrieved from

<http://www.britannica.com/EBchecked/topic/287895/information-system>

BulkSMS. (2014). What is Kannel – retrieved from

<http://support.sendbulksms.com/support/solutions/articles/20422-what-is-kannel>

BuddeComm. (2014). Philippines - Mobile Communications, Forecasts and Broadcasting

Market retrieved from [http://www.budde.com.au/Research/Philippines-Mobile-](http://www.budde.com.au/Research/Philippines-Mobile-Communications-Forecasts-and-Broadcasting-Market.html)

[Communications-Forecasts-and-Broadcasting-Market.html](http://www.budde.com.au/Research/Philippines-Mobile-Communications-Forecasts-and-Broadcasting-Market.html).

BusinessDictionary. Client-Server Architecture retrieved from

<http://www.businessdictionary.com/definition/client-server-architecture.html>

Commission of Insurance. (2014). Tentative Ranking of Life Insurance Companies According to Networth as of December 31, 2014 retrieved from

http://www.insurance.gov.ph/htm/..%5C_@dmin%5Cupload%5Cstatistics

[%5CRanking_Life_Networth2014.pdf](http://www.insurance.gov.ph/htm/..%5C_@dmin%5Cupload%5Cstatistics%5CRanking_Life_Networth2014.pdf).

Dimacali, T. (2010). Philippines still text messaging champ -US study retrieved from

<http://www.gmanetwork.com/news/story/198832/scitech/philippines-still-text-messaging-champ-us-study>.

Ellislab. CodeIgniter User Guide Version 2.2.0 retrieved from

<https://ellislab.com/codeigniter/user-guide/>

Eli The Computer Guy. (2013). CRM Customer Relationship Management) Software

Introduction retrieved from https://www.youtube.com/watch?v=OiYtQ_Tq0Dk.

FrugalTech. (2014). Introduction to CRM - Customer Relationship Management Systems
retrieved from <https://www.youtube.com/watch?v=SEIp-Gfgf1g>.

Harrington, C. (2014). 10 Need-to-Know Customer Dissatisfaction Stats retrieved from
<http://customerthink.com/10-need-to-know-customer-dissatisfaction-stats/>

Gupta, P. Short Message Service: What, How and Where? Retrieved from:
<http://www.wirelessdevnet.com/channels/sms/features/sms.html>

IctDATA. (2011). SMS Update 2011 retrieved from <http://www.ictdata.org/2012/06/sms-update-2011.html>.

Kannel. Kannel: Open Source WAP and SMS gateway retrieved from:
<http://www.kannel.org/>

Microsoft. Task Scheduler retrieved from <https://msdn.microsoft.com/en-us/library/windows/desktop/aa383614%28v=vs.85%29.aspx>

MySQL. What is MySQL retrieved from <https://dev.mysql.com/doc/refman/5.1/en/what-is-mysql.html>

Peck, J. What is CodeIgniter and why should I use it? Retrieved from
<http://www.lynda.com/CodeIgniter-tutorials/What-CodeIgniter-why-should-I-use/126122/141742-4.html>

PHP. What is PHP? Retrieved from <http://php.net/manual/en/intro-what-is.php>

Pronelos, V. (2014) Stolen Cars show up with no alarms on Text LTO retrieved from
<http://www.autoindustriya.com/auto-industry-news/stolen-cars-show-up-with-no-alarms-on-text-lto.html>

SearchSoftwareQuality. Web application (Web app) retrieved from
<http://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app>

SiteGround. What is MySQL retrieved from <https://www.siteground.com/tutorials/php-mysql/mysql.html>

The Advisory Board Company. (2014). Heart Valve Procedures Incur the Costliest Hospital

Stays retrieved from <http://www.advisory.com/daily-briefing/2014/02/20/ahrq-the-five-most-expensive-surgeries-in-america>.

The Atlantic. Why Texting is the Most Important Information Service in the Worlds retrieved from <http://www.theatlantic.com/technology/archive/2011/08/why-texting-is-the-most-important-information-service-in-the-world/242951/>

Trading Economics. (2009). Health Expenditure - Public (% of GDP) in Philippines retrieved from <http://www.tradingeconomics.com/philippines/health-expenditure-public-percent-of-gdp-wb-data.html>.

Tyson, J. How WAP Works retrieved from:

<http://computer.howstuffworks.com/wireless-internet3.htm>

Ubuntu. CronHowTo retrieved from <https://help.ubuntu.com/community/CronHowto>

w3schools. CSS Introduction retrieved from http://www.w3schools.com/css/css_intro.asp

World Health Organization. (2014). Financial Crisis and Global Health retrieved from http://www.who.int/topics/financial_crisis/en/.

Zurb. History of Foundation retrieved from <http://foundation.zurb.com/learn/about.html>

X. Appendix

Source Code

.htaccess

```
1: RewriteEngine on
2: RewriteCond $1 !^(index\.php|robots\.txt)
3: RewriteCond %{REQUEST_FILENAME} !-f
4: RewriteCond %{REQUEST_FILENAME} !-d
5: RewriteRule ^(.*)$ index.php/$1 [L,QSA]
```

index.php

```
1: <?php
2:
3: /*
4: * -----
5: * APPLICATION ENVIRONMENT
6: * -----
7: *
8: * You can load different configurations depending on your
9: * current environment. Setting the environment also influences
10: * things like logging and error reporting.
11: *
12: * This can be set to anything, but default usage is:
13: *
14: * development
15: * testing
16: * production
17: *
18: * NOTE: If you change these, also change the error_reporting() code below
19: *
20: */
21: define ( 'ENVIRONMENT', 'development' );
22: /*
23: * -----
24: * ERROR REPORTING
25: * -----
26: *
27: * Different environments will require different levels of error reporting.
28: * By default development will show errors but testing and live will hide them.
29: */
30:
31: if (defined ( 'ENVIRONMENT' )) {
32:     switch (ENVIRONMENT) {
33:         case 'development' :
34:             error_reporting ( E_ALL );
35:             break;
36:
37:         case 'testing' :
38:         case 'production' :
39:             error_reporting ( 0 );
40:             break;
41:
42:         default :
43:             exit ( 'The application environment is not set correctly.' );
44:     }
45: }
46:
47: /*
48: * -----
49: * SYSTEM FOLDER NAME
50: * -----
51: *
52: * This variable must contain the name of your "system" folder.
53: * Include the path if the folder is not in the same directory
54: * as this file.
55: *
56: */
57: $system_path = 'system';
58:
59: /*
60: * -----
61: * APPLICATION FOLDER NAME
62: * -----
63: *
```


index.php

```
64: * If you want this front controller to use a different "application"
65: * folder then the default one you can set its name here. The folder
66: * can also be renamed or relocated anywhere on your server. If
67: * you do, use a full server path. For more info please see the user guide:
68: * http://codeigniter.com/user\_guide/general/managing\_apps.html
69: *
70: * NO TRAILING SLASH!
71: *
72: */
73: $application_folder = 'application';
74:
75: /*
76: * -----
77: * DEFAULT CONTROLLER
78: * -----
79: *
80: * Normally you will set your default controller in the routes.php file.
81: * You can, however, force a custom routing by hard-coding a
82: * specific controller class/function here. For most applications, you
83: * WILL NOT set your routing here, but it's an option for those
84: * special instances where you might want to override the standard
85: * routing in a specific front controller that shares a common CI installation.
86: *
87: * IMPORTANT: If you set the routing here, NO OTHER controller will be
88: * callable. In essence, this preference limits your application to ONE
89: * specific controller. Leave the function name blank if you need
90: * to call functions dynamically via the URI.
91: *
92: * Un-comment the $routing array below to use this feature
93: *
94: */
95: // The directory name, relative to the "controllers" folder. Leave blank
96: // if your controller is not in a sub-folder within the "controllers" folder
97: // $routing['directory'] = '';
98:
99: // The controller class file name. Example: Mycontroller
100: // $routing['controller'] = '';
101:
102: // The controller function you wish to be called.
103: // $routing['function'] = '';
104:
105: /*
106: * -----
107: * CUSTOM CONFIG VALUES
108: * -----
109: *
110: * The $assign_to_config array below will be passed dynamically to the
111: * config class when initialized. This allows you to set custom config
112: * items or override any default config values found in the config.php file.
113: * This can be handy as it permits you to share one application between
114: * multiple front controller files, with each file containing different
115: * config values.
116: *
117: * Un-comment the $assign_to_config array below to use this feature
118: *
119: */
120: // $assign_to_config['name_of_config_item'] = 'value of config item';
121:
122: // -----
123: // END OF USER CONFIGURABLE SETTINGS. DO NOT EDIT BELOW THIS LINE
124: // -----
125:
126: /*
```

index.php

```
127: * -----
128: * Resolve the system path for increased reliability
129: * -----
130: */
131:
132: // Set the current directory correctly for CLI requests
133: if (defined ( 'STDIN' )) {
134:     chdir ( dirname ( __FILE__ ) );
135: }
136:
137: if (realpath ( $system_path ) !== FALSE) {
138:     $system_path = realpath ( $system_path ) . '/';
139: }
140:
141: // ensure there's a trailing slash
142: $system_path = rtrim ( $system_path, '/' ) . '/';
143:
144: // Is the system path correct?
145: if (! is_dir ( $system_path )) {
146:     exit ( "Your system folder path does not appear to be set correctly. Please open the following file and correct this: " . pathinfo ( __FILE__, PATHINFO_BASENAME ) );
147: }
148:
149: /*
150: * -----
151: * Now that we know the path, set the main path constants
152: * -----
153: */
154: // The name of THIS file
155: define ( 'SELF', pathinfo ( __FILE__, PATHINFO_BASENAME ) );
156:
157: // The PHP file extension
158: // this global constant is deprecated.
159: define ( 'EXT', '.php' );
160:
161: // Path to the system folder
162: define ( 'BASEPATH', str_replace ( "\\", "/", $system_path ) );
163:
164: // Path to the front controller (this file)
165: define ( 'FCPATH', str_replace ( SELF, '', __FILE__ ) );
166:
167: // Name of the "system folder"
168: define ( 'SYSDIR', trim ( strrchr ( trim ( BASEPATH, '/' ), '/' ), '/' ) );
169:
170: // The path to the "application" folder
171: if (is_dir ( $application_folder )) {
172:     define ( 'APPPATH', $application_folder . '/' );
173: } else {
174:     if (! is_dir ( BASEPATH . $application_folder . '/' )) {
175:         exit ( "Your application folder path does not appear to be set correctly. Please open the following file and correct this: " . SELF );
176:     }
177:
178:     define ( 'APPPATH', BASEPATH . $application_folder . '/' );
179: }
180:
181: /*
182: * -----
183: * LOAD THE BOOTSTRAP FILE
184: * -----
185: *
186: * And away we go...
187: *
```

index.php

```
188: */
189: require_once BASEPATH . 'core/CodeIgniter.php';
190:
191: /* End of file index.php */
192: /* Location: ./index.php */
```

.project

```
1: <?xml version="1.0" encoding="UTF-8"?>
2: <projectDescription>
3:     <name>mhic</name>
4:     <comment></comment>
5:     <projects>
6:     </projects>
7:     <buildSpec>
8:         <buildCommand>
9:             <name>org.eclipse.wst.common.project.facet.core.builder</name>
10:            <arguments>
11:            </arguments>
12:        </buildCommand>
13:        <buildCommand>
14:            <name>org.eclipse.wst.jsdt.core.javascriptValidator</name>
15:            <arguments>
16:            </arguments>
17:        </buildCommand>
18:        <buildCommand>
19:            <name>org.eclipse.wst.validation.validationbuilder</name>
20:            <arguments>
21:            </arguments>
22:        </buildCommand>
23:        <buildCommand>
24:            <name>org.eclipse.dltk.core.scriptbuilder</name>
25:            <arguments>
26:            </arguments>
27:        </buildCommand>
28:    </buildSpec>
29:    <natures>
30:        <nature>org.eclipse.php.core.PHPNature</nature>
31:        <nature>org.eclipse.wst.jsdt.core.jsNature</nature>
32:        <nature>org.eclipse.wst.common.project.facet.core.nature</nature>
33:    </natures>
34: </projectDescription>
```

.buildpath

```
1: <?xml version="1.0" encoding="UTF-8"?>
2: <buildpath>
3:     <buildpathentry kind="src" path=""/>
4:     <buildpathentry kind="con" path="org.eclipse.php.core.LANGUAGE"/>
5: </buildpath>
```

.htaccess

1: Deny from all

index.html

```
1: <html>
2: <head>
3: <title>403 Forbidden</title>
4: </head>
5: <body>
6:
7:     <p>Directory access is forbidden.</p>
8:
9: </body>
10: </html>
```

index.html

```
1: <html>
2: <head>
3: <title>403 Forbidden</title>
4: </head>
5: <body>
6:
7:     <p>Directory access is forbidden.</p>
8:
9: </body>
10: </html>
```


index.html

```
1: <html>
2: <head>
3: <title>403 Forbidden</title>
4: </head>
5: <body>
6:
7:     <p>Directory access is forbidden.</p>
8:
9: </body>
10: </html>
```

form_custom_validation_helper.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: if (! function_exists ( 'valid_cellphone_number' )) {
5:     function valid_cellphone_number($str = '') {
6:         $CI = & get_instance ();
7:         $cellphone_number_format = '/09\d{9}/';
8:         if ($str != '' && preg_match ( $cellphone_number_format, $str )) {
9:             return TRUE;
10:        }
11:        $CI->form_validation->set_message ( 'valid_cellphone_number', 'The %s f
ield should be a valid cellphone number.' );
12:        return FALSE;
13:    }
14: }
15: if (! function_exists ( 'valid_member' )) {
16:     function valid_member($str = '') {
17:         $CI = & get_instance ();
18:         $CI->load->database ();
19:         $CI->load->model ( 'Members' );
20:         $query = $CI->Members->get_member_details ( $str );
21:         if (issetandnotempty ( $query )) {
22:             return TRUE;
23:         }
24:         $CI->form_validation->set_message ( 'valid_member', 'The %s field shoul
d be a valid PIN number.' );
25:         return FALSE;
26:     }
27: }
28: if (! function_exists ( 'valid_member_with_cellphone_number' )) {
29:     function valid_member_with_cellphone_number($str = '') {
30:         $CI = & get_instance ();
31:         $CI->load->database ();
32:         $CI->load->model ( 'Members' );
33:         $query = $CI->Members->get_member_details ( $str );
34:         if (issetandnotempty ( $query ) && issetandnotempty ( $query ['cellphon
e_number'] )) {
35:             return TRUE;
36:         }
37:         $CI->form_validation->set_message ( 'valid_member_with_cellphone_number
', 'The %s field should be a valid PIN number with cellphone number.' );
38:         return FALSE;
39:     }
40: }
41: ?>
```

session_helper.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: if (! function_exists ( 'authenticate_log_in' )) {
5:     function authenticate_log_in() {
6:         $CI = & get_instance ();
7:         $CI->load->library ( 'session' );
8:         $CI->load->helper ( 'variable' );
9:         $CI->load->helper ( 'import' );
10:        $CI->load->helper ( 'url' );
11:        if (! issetandnotempty ( $CI->session->userdata ( 'logged_in' ) )) {
12:            return redirect ( 'login' );
13:        } else if (! $CI->session->userdata ( 'logged_in' )) {
14:            return redirect ( 'login' );
15:        }
16:    }
17: }
18: if (! function_exists ( 'is_system_administrator' )) {
19:     function is_system_administrator() {
20:         $CI = & get_instance ();
21:         $CI->load->library ( 'session' );
22:         $CI->load->helper ( 'variable' );
23:         if (issetandnotempty ( $CI->session->userdata ( 'priviledge_id' ) ) &&
$CI->session->userdata ( 'priviledge_id' ) == '0') {
24:             return TRUE;
25:         } else if (! $CI->session->userdata ( 'logged_in' )) {
26:             return FALSE;
27:         }
28:     }
29: }
30: ?>
```

variable_helper.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: if (! function_exists ( 'issetandnotempty' )) {
5:     function issetandnotempty($var = '', $arr = '') {
6:         if ($arr != '') {
7:             if (array_key_exists ( $var, $arr ) && ! is_null ( $arr [$var]
) && isset ( $arr [$var] ) && (is_numeric ( $arr [$var] ) || ! empty ( $arr [$var] ))) {
8:                 return true;
9:             }
10:            return false;
11:        } else {
12:            if (! is_null ( $var ) && isset ( $var ) && (is_numeric ( $var
) || ! empty ( $var ))) {
13:                return true;
14:            }
15:        }
16:        return false;
17:    }
18: }
19: ?>
```

import_helper.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: if (! function_exists ( 'js_tag' )) {
5:     function js_tag($src = '', $type = 'text/javascript', $charset = '', $defer = F
ELSE, $async = FALSE) {
6:         $CI = & get_instance ();
7:         $script = '<script ';
8:         if (strpos ( $src, '://' ) !== FALSE) {
9:             $script .= 'src="' . $src . '" ';
10:        } else {
11:            $script .= 'src="' . $CI->config->slash_item ( 'base_url' ) . $
src . '" ';
12:        }
13:        $script .= 'type="' . $type . '" ';
14:        if ($charset != '') {
15:            $script .= 'charset="' . $charset . '" ';
16:        }
17:        if ($defer) {
18:            $script .= 'defer ';
19:        }
20:        if ($async) {
21:            $script .= 'async ';
22:        }
23:        $script .= '></script>';
24:        return $script;
25:    }
26: }
27: if (! function_exists ( 'img_tag' )) {
28:     function img_tag($src = '', $alt = '', $align = '', $height = '', $width = '',
$crossorigin = '', $ismap = FALSE, $usemap = '') {
29:         $CI = & get_instance ();
30:         $script = 'config->slash_item ( 'base_url' ) . $
src . '" ';
36:        }
37:
38:        if ($alt != '') {
39:            $script .= 'alt="' . $alt . '" ';
40:        }
41:        if ($align != '') {
42:            $script .= 'align="' . $align . '" ';
43:        }
44:        if ($height != '') {
45:            $script .= 'height="' . $height . '" ';
46:        }
47:        if ($width != '') {
48:            $script .= 'width="' . $width . '" ';
49:        }
50:        if ($crossorigin != '') {
51:            $script .= 'crossorigin="' . $crossorigin . '" ';
52:        }
53:        if ($ismap != '') {
54:            $script .= 'ismap ';
55:        }
56:        if ($usemap != '') {
57:            $script .= 'usemap="' . $usemap . '" ';
58:        }
59:        $script .= '></img>';
```

import_helper.php

```
60:         return $script;
61:     }
62: }
63: if (! function_exists ( 'css_tag' )) {
64:     function css_tag($href = '', $type = 'text/css', $rel = 'stylesheet') {
65:         $CI = & get_instance ();
66:         $script = '<link ';
67:         if (strpos ( $href, '://' ) !== FALSE) {
68:             $script .= 'href="' . $href . '" ';
69:         } else {
70:             $script .= 'href="' . $CI->config->slash_item ( 'base_url' ) .
$href . '" ';
71:         }
72:         $script .= 'rel="' . $rel . '" ';
73:         $script .= 'type="' . $type . '" ';
74:         $script .= '>';
75:         return $script;
76:     }
77: }
78: ?>
```

index.html

```
1: <html>
2: <head>
3: <title>403 Forbidden</title>
4: </head>
5: <body>
6:
7:     <p>Directory access is forbidden.</p>
8:
9: </body>
10: </html>
```

login.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Login extends CI_Controller {
5:     private function company_details() {
6:         $this->load->model ( 'Company_details' );
7:         $queried_company_details = $this->Company_details->get_company_details
();
8:         $session_details = array (
9:             'company_acronym' => $queried_company_details ['acronym
'],
10:             'company_site' => $queried_company_details ['site']
11:         );
12:         $this->session->set_userdata ( $session_details );
13:     }
14:     public function index() {
15:         $this->company_details ();
16:         if (issetandnotempty ( $this->session->userdata ( 'logged_in' ) ) && $t
his->session->userdata ( 'logged_in' ) ) {
17:             redirect ( 'dashboard/', 'refresh' );
18:         } else {
19:             $this->load->view ( 'login/login' );
20:         }
21:     }
22:     public function process_login() {
23:         $config = array (
24:             array (
25:                 'field' => '',
26:                 'label' => '',
27:                 'rules' => 'required|trim|xss_clean'
28:             ),
29:             array (
30:                 'field' => '',
31:                 'label' => '',
32:                 'rules' => 'required|trim|xss_clean'
33:             )
34:         );
35:         $this->form_validation->set_rules ( $config );
36:         if ( $this->form_validation->run () == FALSE ) {
37:         } else {
38:         }
39:
40:         $username = $this->input->post ( 'username' );
41:         $password = $this->input->post ( 'password' );
42:         $this->load->model ( 'Users' );
43:         $verified_user_credentials = $this->Users->get_user_details_with_creden
tials ( $username, $password );
44:         if (issetandnotempty ( $verified_user_credentials ) && $verified_user_c
redentials ['username'] == $username) {
45:             $verified_user_credentials = $this->Users->get_user_details_wit
h_usernae ( $username );
46:             $this->load->model ( 'Priviledge' );
47:             $verified_user_priviledge = $this->Priviledge->get_priviledge_d
etails ( $verified_user_credentials ['priviledge_id'] );
48:             $session_details = array (
49:                 'user_id' => $verified_user_credentials ['id'],
50:                 'first_name' => $verified_user_credentials ['fi
rst_name'],
51:                 'priviledge_id' => $verified_user_priviledge ['
id'],
52:                 'priviledge_name' => $verified_user_priviledge
['name'],
53:                 'logged_in' => TRUE
```


login.php

```
54:                                     );
55:                                     $this->session->set_userdata ( $session_details );
56:                                     redirect ( 'dashboard/', 'refresh' );
57:     } else {
58:         $this->index ();
59:     }
60: }
61: public function logout() {
62:     $userdata_list = array_keys ( $this->session->all_userdata () );
63:     foreach ( $userdata_list as $userdata_element ) {
64:         $this->session->unset_userdata ( $userdata_element );
65:     }
66:     $this->index ();
67: }
68: }
69: ?>
```

personnel.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Personnel extends CI_Controller {
5:     public function register($selected_partner_company_id = NULL) {
6:         authenticate_log_in ();
7:         $this->load->model ( 'Partner_companies' );
8:         $queried_partner_companies_list = $this->Partner_companies->get_all_partner_companies ();
9:         $data ['partner_company_id'] = array ();
10:        foreach ( $queried_partner_companies_list as $partner_companies_list )
11:            {
12:                $data ['partner_company_id'] = $data ['partner_company_id'] + array (
13:                    => $partner_companies_list ['partner_company_id']
14:                    => $partner_companies_list ['name']
15:                );
16:            }
17:        $data ['selected_partner_company_id'] = $selected_partner_company_id;
18:        $this->load->model ( 'Regions' );
19:        $queried_region_list = $this->Regions->get_all_regions ();
20:        $data ['region_responsibility_id'] = array ();
21:        foreach ( $queried_region_list as $region_list ) {
22:            $data ['region_responsibility_id'] = $data ['region_responsibility_id'] + array (
23:                => $region_list ['id'] => $region_list ['name']
24:            );
25:        }
26:        $this->load->view ( 'personnel/registration', $data );
27:    }
28:    public function process_registration() {
29:        authenticate_log_in ();
30:        $partner_company_personnel_details = array ();
31:        $partner_company_personnel_details ['partner_company_id'] = $this->input->post ( 'partner_company_id' );
32:        $config = array (
33:            array (
34:                'field' => 'first_name',
35:                'label' => 'First Name',
36:                'rules' => 'required|trim|xss_clean'
37:            ),
38:            array (
39:                'field' => 'middle_name',
40:                'label' => 'Middle Name',
41:                'rules' => 'trim|xss_clean'
42:            ),
43:            array (
44:                'field' => 'last_name',
45:                'label' => 'Last Name',
46:                'rules' => 'required|trim|xss_clean'
47:            ),
48:            array (
49:                'field' => 'cellphone_number',
50:                'label' => 'Cellphone Number',
51:                'rules' => 'callback_valid_cellphone_number|required|trim|xss_clean'
52:            ),
53:            array (
54:                'field' => 'email_address',
55:                'label' => 'Email Address',
56:                'rules' => 'valid_email|required|trim|x
```

personnel.php

```
ss_clean'
57:         )
58:     );
59:     $this->form_validation->set_rules ( $config );
60:     if ( $this->form_validation->run () == FALSE ) {
61:         $this->register ( $partner_company_personnel_details ['partner_
company_id'] );
62:     } else {
63:         $partner_company_personnel_details ['first_name'] = $this->inpu
t->post ( 'first_name' );
64:         $partner_company_personnel_details ['middle_name'] = $this->inp
ut->post ( 'middle_name' );
65:         $partner_company_personnel_details ['last_name'] = $this->input
->post ( 'last_name' );
66:         $partner_company_personnel_details ['cellphone_number'] = $this
->input->post ( 'cellphone_number' );
67:         $partner_company_personnel_details ['email_address'] = $this->i
nput->post ( 'email_address' );
68:         $partner_company_personnel_details ['region_responsibility_id']
= $this->input->post ( 'region_responsibility_id' );
69:         $partner_company_personnel_details ['is_active'] = ( $this->inpu
t->post ( 'is_active' ) == 'active' ? '1' : '0' );
70:         $this->load->model ( 'Partner_companies_personnel' );
71:         $this->Partner_companies_personnel->save_personnel ( $partner_c
ompany_personnel_details );
72:         $this->lists ( $partner_company_personnel_details ['partner_com
pany_id'] );
73:     }
74: }
75: public function update($personnel_id = NULL) {
76:     authenticate_log_in ();
77:     $data = array ();
78:     $this->load->model ( 'Partner_companies_personnel' );
79:     $queried_partner_company_personnel_details = $this->Partner_companies_p
ersonnel->get_personnel_details ( $personnel_id );
80:     $data ['personnel_id'] = $queried_partner_company_personnel_details ['i
d'];
81:     $data ['first_name'] = $queried_partner_company_personnel_details ['fir
st_name'];
82:     $data ['middle_name'] = $queried_partner_company_personnel_details ['mi
ddle_name'];
83:     $data ['last_name'] = $queried_partner_company_personnel_details ['last
_name'];
84:     $data ['cellphone_number'] = $queried_partner_company_personnel_details
['cellphone_number'];
85:     $data ['email_address'] = $queried_partner_company_personnel_details ['
email_address'];
86:     $data ['selected_partner_company_id'] = $queried_partner_company_person
nel_details ['partner_company_id'];
87:     $data ['selected_region_responsibility_id'] = $queried_partner_company_
personnel_details ['region_responsibility_id'];
88:     $data ['is_active'] = ( $queried_partner_company_personnel_details ['is_
active'] == '1' ? 'active' : 'inactive' );
89:     $this->load->model ( 'Partner_companies' );
90:     $queried_partner_companies_list = $this->Partner_companies->get_all_par
tner_companies ();
91:     $data ['partner_company_id'] = array ();
92:     foreach ( $queried_partner_companies_list as $partner_companies_list )
{
93:         $data ['partner_company_id'] = $data ['partner_company_id'] + a
rray (
94:             $partner_companies_list ['partner_company_id']
=> $partner_companies_list ['name']
```

personnel.php

```
95:         );
96:     }
97:     $this->load->model ( 'Regions' );
98:     $queried_region_list = $this->Regions->get_all_regions ();
99:     $data ['region_responsibility_id'] = array ();
100:     foreach ( $queried_region_list as $region_list ) {
101:         $data ['region_responsibility_id'] = $data ['region_responsibil
ity_id'] + array (
102:             $region_list ['id'] => $region_list ['name']
103:         );
104:     }
105:     $this->load->view ( 'personnel/update', $data );
106: }
107: public function process_update() {
108:     authenticate_log_in ();
109:
110:     $partner_company_personnel_id = $this->input->post ( 'personnel_id' );
111:
112:     $config = array (
113:         array (
114:             'field' => 'first_name',
115:             'label' => 'First Name',
116:             'rules' => 'required|trim|xss_clean'
117:         ),
118:         array (
119:             'field' => 'middle_name',
120:             'label' => 'Middle Name',
121:             'rules' => 'trim|xss_clean'
122:         ),
123:         array (
124:             'field' => 'last_name',
125:             'label' => 'Last Name',
126:             'rules' => 'required|trim|xss_clean'
127:         ),
128:         array (
129:             'field' => 'cellphone_number',
130:             'label' => 'Cellphone Number',
131:             'rules' => 'callback_valid_cellphone_nu
mber|required|trim|xss_clean'
132:         ),
133:         array (
134:             'field' => 'email_address',
135:             'label' => 'Email Address',
136:             'rules' => 'valid_email|required|trim|x
ss_clean'
137:         )
138:     );
139:     $this->form_validation->set_rules ( $config );
140:     if ( $this->form_validation->run () == FALSE ) {
141:         $this->update ( $partner_company_personnel_id );
142:     } else {
143:         $partner_company_personnel_details = array ();
144:         $partner_company_personnel_details ['first_name'] = $this->inpu
t->post ( 'first_name' );
145:         $partner_company_personnel_details ['middle_name'] = $this->inp
ut->post ( 'middle_name' );
146:         $partner_company_personnel_details ['last_name'] = $this->input
->post ( 'last_name' );
147:         $partner_company_personnel_details ['cellphone_number'] = $this
->input->post ( 'cellphone_number' );
148:         $partner_company_personnel_details ['email_address'] = $this->i
nput->post ( 'email_address' );
149:         $partner_company_personnel_details ['partner_company_id'] = $th
```

personnel.php

```
is->input->post ( 'partner_company_id' );
150:         $partner_company_personnel_details ['region_responsibility_id']
= $this->input->post ( 'region_responsibility_id' );
151:         $partner_company_personnel_details ['is_active'] = ($this->input->post ( 'is_active' ) == 'active' ? '1' : '0');
152:         $this->load->model ( 'Partner_companies_personnel' );
153:         $this->Partner_companies_personnel->update_personnel ( $partner_company_personnel_id, $partner_company_personnel_details );
154:         $this->lists ( $partner_company_personnel_details ['partner_company_id'] );
155:     }
156: }
157: public function lists($partner_company_id = NULL) {
158:     authenticate_log_in ();
159:     $this->load->model ( 'Partner_companies_personnel' );
160:     $this->load->model ( 'Partner_companies' );
161:     $queried_partner_company_personnel_list = $this->Partner_companies_personnel->get_all_personnel_of_company_id ( $partner_company_id );
162:     $data = array ();
163:     $queried_partner_company_details = $this->Partner_companies->get_company_details ( $partner_company_id );
164:     $data ['partner_company_name'] = $queried_partner_company_details ['name'];
165:     $data ['partner_company_id'] = $partner_company_id;
166:     $data ['partner_company_personnel_list'] = array ();
167:     if (issetandnotempty ( $queried_partner_company_personnel_list )) {
168:         foreach ( $queried_partner_company_personnel_list as $partner_company_personnel_list ) {
169:             $queryresult = array ();
170:             $queryresult ['personnel_id'] = $partner_company_personnel_list ['id'];
171:             $queryresult ['name'] = $partner_company_personnel_list ['first_name'] . ' ' . $partner_company_personnel_list ['last_name'];
172:             $queryresult ['cellphone_number'] = $partner_company_personnel_list ['cellphone_number'];
173:             $queryresult ['email_address'] = $partner_company_personnel_list ['email_address'];
174:             $queried_partner_company_details = $this->Partner_companies->get_company_details ( $partner_company_personnel_list ['partner_company_id'] );
175:             $queryresult ['partner_company_id'] = $queried_partner_company_details ['name'];
176:             $this->load->model ( 'Regions' );
177:             $queried_region_details = $this->Regions->get_region_details ( $partner_company_personnel_list ['region_responsibility_id'] );
178:             $queryresult ['region_responsibility_id'] = $queried_region_details ['name'];
179:             $queryresult ['is_active'] = ($partner_company_personnel_list ['is_active'] == '1' ? 'active' : 'inactive');
180:             array_push ( $data ['partner_company_personnel_list'], $queryresult );
181:         }
182:     }
183:     $this->load->view ( 'personnel/list', $data );
184: }
185: public function activate($user_id = NULL) {
186:     authenticate_log_in ();
187:     $this->load->model ( 'Partner_companies_personnel' );
188:     $this->Partner_companies_personnel->activate_personnel ( $user_id );
189:     $queried_partner_company_personnel_details = $this->Partner_companies_personnel->get_personnel_details ( $user_id );
190:     $this->lists ( $queried_partner_company_personnel_details ['partner_company_id'] );
191: }
```

personnel.php

```
192:         public function deactivate($user_id = NULL) {
193:             authenticate_log_in ();
194:             $this->load->model ( 'Partner_companies_personnel' );
195:             $this->Partner_companies_personnel->deactivate_personnel ( $user_id );
196:             $queried_partner_company_personnel_details = $this->Partner_companies_p
ersonnel->get_personnel_details ( $user_id );
197:             $this->lists ( $queried_partner_company_personnel_details ['partner_com
pany_id'] );
198:         }
199:         function valid_cellphone_number($str) {
200:             return valid_cellphone_number ( $str );
201:         }
202:     }
203: ?>
```

partner.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Partner extends CI_Controller {
5:     public function register() {
6:         authenticate_log_in ();
7:         $this->load->view ( 'partner/registration' );
8:     }
9:     public function process_registration() {
10:        authenticate_log_in ();
11:
12:        $config = array (
13:            array (
14:                'field' => 'acronym',
15:                'label' => 'Acronym',
16:                'rules' => 'required|trim|xss_clean'
17:            ),
18:            array (
19:                'field' => 'name',
20:                'label' => 'Name',
21:                'rules' => 'required|trim|xss_clean'
22:            ),
23:            array (
24:                'field' => 'address',
25:                'label' => 'Address',
26:                'rules' => 'required|trim|xss_clean'
27:            ),
28:            array (
29:                'field' => 'description',
30:                'label' => 'Description',
31:                'rules' => 'required|trim|xss_clean'
32:            )
33:        );
34:        $this->form_validation->set_rules ( $config );
35:        if ( $this->form_validation->run () == FALSE ) {
36:            $this->register ();
37:        } else {
38:            $partner_company_details = array ();
39:            $partner_company_details ['acronym'] = $this->input->post ( 'ac
ronym' );
40:            $partner_company_details ['name'] = $this->input->post ( 'name'
);
41:            $partner_company_details ['address'] = $this->input->post ( 'ad
dress' );
42:            $partner_company_details ['description'] = $this->input->post (
'description' );
43:            $partner_company_details ['is_active'] = ( $this->input->post (
'is_active' ) == 'active' ? '1' : '0' );
44:            $this->load->model ( 'Partner_companies' );
45:            $this->Partner_companies->save_company_details ( $partner_compa
ny_details );
46:            $this->lists ();
47:        }
48:    }
49:    public function update($company_id = NULL) {
50:        authenticate_log_in ();
51:        $data = array ();
52:        $this->load->model ( 'Partner_companies' );
53:        $queried_partner_company_details = $this->Partner_companies->get_compan
y_details ( $company_id );
54:        $data ['partner_company_id'] = $queried_partner_company_details ['partn
er_company_id'];
55:        $data ['acronym'] = $queried_partner_company_details ['acronym'];
```

partner.php

```
56:         $data ['name'] = $queried_partner_company_details ['name'];
57:         $data ['address'] = $queried_partner_company_details ['address'];
58:         $data ['description'] = $queried_partner_company_details ['description'
];
59:         $data ['is_active'] = $queried_partner_company_details ['is_active'];
60:         $this->load->view ( 'partner/update', $data );
61:     }
62:     public function process_update() {
63:         authenticate_log_in ();
64:
65:         $partner_company_id = $this->input->post ( 'partner_company_id' );
66:
67:         $config = array (
68:             array (
69:                 'field' => 'acronym',
70:                 'label' => 'Acronym',
71:                 'rules' => 'required|trim|xss_clean'
72:             ),
73:             array (
74:                 'field' => 'name',
75:                 'label' => 'Name',
76:                 'rules' => 'required|trim|xss_clean'
77:             ),
78:             array (
79:                 'field' => 'address',
80:                 'label' => 'Address',
81:                 'rules' => 'required|trim|xss_clean'
82:             ),
83:             array (
84:                 'field' => 'description',
85:                 'label' => 'Description',
86:                 'rules' => 'required|trim|xss_clean'
87:             )
88:         );
89:         $this->form_validation->set_rules ( $config );
90:         if ( $this->form_validation->run () == FALSE ) {
91:             $this->update ( $partner_company_id );
92:         } else {
93:             $partner_company_details = array ();
94:             $partner_company_details ['acronym'] = $this->input->post ( 'ac
ronym' );
95:             $partner_company_details ['name'] = $this->input->post ( 'name'
);
96:             $partner_company_details ['address'] = $this->input->post ( 'ad
dress' );
97:             $partner_company_details ['description'] = $this->input->post (
'description' );
98:             $partner_company_details ['is_active'] = ( $this->input->post (
'is_active' ) == 'active' ? '1' : '0' );
99:             $this->load->model ( 'Partner_companies' );
100:            $this->Partner_companies->update_company_details ( $partner_com
pany_id, $partner_company_details );
101:            $this->lists ();
102:        }
103:    }
104:    public function lists() {
105:        authenticate_log_in ();
106:        $this->load->model ( 'Partner_companies' );
107:        $this->load->model ( 'Partner_companies_personnel' );
108:        $this->load->model ( 'Partner_companies_services' );
109:        $queried_partner_company_list = $this->Partner_companies->get_all_partn
er_companies ();
110:        $data = array ();
```


partner.php

```
111:         $data ['partner_company_list'] = array ();
112:         if (issetandnotempty ( $queried_partner_company_list )) {
113:             foreach ( $queried_partner_company_list as $partner_company_lis
t ) {
114:                 $queryresult = array ();
115:                 $queryresult ['partner_company_id'] = $partner_company_
list ['partner_company_id'];
116:                 $queryresult ['acronym'] = $partner_company_list ['acro
nym'];
117:                 $queryresult ['name'] = $partner_company_list ['name'];
118:                 $queryresult ['address'] = $partner_company_list ['addr
ess'];
119:                 $queryresult ['description'] = $partner_company_list ['
description'];
120:                 $queryresult ['is_active'] = ($partner_company_list ['i
s_active'] == '1' ? 'active' : 'inactive');
121:
122:                 $queryresult ['active_personnel_count'] = $this->Partne
r_companies_personnel->count_active_personnel ( $queryresult ['partner_company_id'] );
123:                 $queryresult ['active_service_count'] = $this->Partner_
companies_services->count_active_service ( $queryresult ['partner_company_id'] );
124:
125:                 array_push ( $data ['partner_company_list'], $queryresu
lt );
126:             }
127:         }
128:         $this->load->view ( 'partner/list', $data );
129:     }
130:     public function activate($user_id = NULL) {
131:         authenticate_log_in ();
132:         $this->load->model ( 'Partner_companies' );
133:         $this->Partner_companies->activate_partner_company ( $user_id );
134:         $this->lists ();
135:     }
136:     public function deactivate($user_id = NULL) {
137:         authenticate_log_in ();
138:         $this->load->model ( 'Partner_companies' );
139:         $this->Partner_companies->deactivate_partner_company ( $user_id );
140:         $this->lists ();
141:     }
142: }
143: ?>
```

user.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class User extends CI_Controller {
5:     public function register() {
6:         authenticate_log_in ();
7:         $this->load->model ( 'Priviledge' );
8:         $queried_priviledge_list = $this->Priviledge->get_all_priviledge ();
9:         $data ['priviledge_id'] = array ();
10:        foreach ( $queried_priviledge_list as $priviledge_list ) {
11:            $data ['priviledge_id'] = array_merge ( $data ['priviledge_id']
, array (
12:                $priviledge_list ['id'] => $priviledge_list ['n
ame']
13:            ) );
14:        }
15:        $this->load->view ( 'user/registration', $data );
16:    }
17:    public function process_registration() {
18:        authenticate_log_in ();
19:
20:        $config = array (
21:            array (
22:                'field' => 'first_name',
23:                'label' => 'First Name',
24:                'rules' => 'required|trim|xss_clean'
25:            ),
26:            array (
27:                'field' => 'middle_name',
28:                'label' => 'Middle Name',
29:                'rules' => 'trim|xss_clean'
30:            ),
31:            array (
32:                'field' => 'last_name',
33:                'label' => 'Last Name',
34:                'rules' => 'required|trim|xss_clean'
35:            ),
36:            array (
37:                'field' => 'cellphone_number',
38:                'label' => 'Cellphone Number',
39:                'rules' => 'callback_valid_cellphone_nu
mber|required|trim|xss_clean'
40:            ),
41:            array (
42:                'field' => 'email_address',
43:                'label' => 'Email Address',
44:                'rules' => 'valid_email|required|trim|x
ss_clean'
45:            ),
46:            array (
47:                'field' => 'username',
48:                'label' => 'Username',
49:                'rules' => 'is_unique[USERS.USERNAME]|r
equired|trim|xss_clean'
50:            ),
51:            array (
52:                'field' => 'password',
53:                'label' => 'Password',
54:                'rules' => 'required|trim|xss_clean|mat
ches[retyped_password]'
55:            ),
56:            array (
57:                'field' => 'retyped_password',
```

user.php

```
58:                                     'label' => 'Retyped Password',
59:                                     'rules' => 'required|trim|xss_clean|mat
ches[password]\'
60:                                     )
61:                                 );
62: $this->form_validation->set_rules ( $config );
63: if ($this->form_validation->run () == FALSE) {
64:     $this->register ();
65: } else {
66:     $user_details = array ();
67:     $user_details ['first_name'] = $this->input->post ( 'first_name
\' );
68:     $user_details ['middle_name'] = $this->input->post ( 'middle_na
me' );
69:     $user_details ['last_name'] = $this->input->post ( 'last_name'
);
70:     $user_details ['cellphone_number'] = $this->input->post ( 'cell
phone_number' );
71:     $user_details ['email_address'] = $this->input->post ( 'email_a
ddress' );
72:     $user_details ['username'] = $this->input->post ( 'username' );
73:     $user_details ['password'] = hash ( "sha256", $this->input->pos
t ( 'password' ) );
74:     $user_details ['priviledge_id'] = $this->input->post ( 'privile
dge_id' );
75:     $isactive = TRUE;
76:     if ($this->input->post ( 'is_active' ) === 'inactive') {
77:         $isactive = FALSE;
78:     }
79:     $user_details ['is_active'] = $isactive;
80:     $this->load->model ( 'Users' );
81:     $this->Users->save_user ( $user_details );
82:     $this->lists ();
83: }
84: }
85: public function update($user_id = NULL) {
86:     authenticate_log_in ();
87:     $data = array ();
88:     $this->load->model ( 'Users' );
89:     $queried_user_details = $this->Users->get_user_details ( $user_id );
90:     $data ['user_id'] = $queried_user_details ['id'];
91:     $data ['first_name'] = $queried_user_details ['first_name'];
92:     $data ['middle_name'] = $queried_user_details ['middle_name'];
93:     $data ['last_name'] = $queried_user_details ['last_name'];
94:     $data ['cellphone_number'] = $queried_user_details ['cellphone_number']
;
95:     $data ['email_address'] = $queried_user_details ['email_address'];
96:     $data ['selected_priviledge_id'] = $queried_user_details ['priviledge_i
d'];
97:     $data ['is_active'] = $queried_user_details ['is_active'];
98:     $this->load->model ( 'Priviledge' );
99:     $queried_priviledge_id = $this->Priviledge->get_all_priviledge ();
100:    $data ['priviledge_id'] = array ();
101:    foreach ( $queried_priviledge_id as $priviledge_id ) {
102:        $data ['priviledge_id'] = array_merge ( $data ['priviledge_id']
, array (
103:            $priviledge_id ['id'] => $priviledge_id ['name'
]
104:        ) );
105:    }
106:    $this->load->view ( 'user/update', $data );
107: }
108: public function process_update() {
```

user.php

```
109:         authenticate_log_in ();
110:
111:         $config = array (
112:             array (
113:                 'field' => 'first_name',
114:                 'label' => 'First Name',
115:                 'rules' => 'required|trim|xss_clean'
116:             ),
117:             array (
118:                 'field' => 'middle_name',
119:                 'label' => 'Middle Name',
120:                 'rules' => 'trim|xss_clean'
121:             ),
122:             array (
123:                 'field' => 'last_name',
124:                 'label' => 'Last Name',
125:                 'rules' => 'required|trim|xss_clean'
126:             ),
127:             array (
128:                 'field' => 'cellphone_number',
129:                 'label' => 'Cellphone Number',
130:                 'rules' => 'callback_valid_cellphone_number|required|trim|xss_clean'
131:             ),
132:             array (
133:                 'field' => 'email_address',
134:                 'label' => 'Email Address',
135:                 'rules' => 'valid_email|required|trim|xss_clean'
136:             )
137:         );
138:         $this->form_validation->set_rules ( $config );
139:         if ( $this->form_validation->run () == FALSE ) {
140:             $this->update ();
141:         } else {
142:             $user_id = $this->input->post ( 'user_id' );
143:             $user_details = array ();
144:             $user_details ['first_name'] = $this->input->post ( 'first_name' );
145:             $user_details ['middle_name'] = $this->input->post ( 'middle_name' );
146:             $user_details ['last_name'] = $this->input->post ( 'last_name' );
147:             $user_details ['cellphone_number'] = $this->input->post ( 'cellphone_number' );
148:             $user_details ['email_address'] = $this->input->post ( 'email_address' );
149:             $user_details ['priviledge_id'] = $this->input->post ( 'priviledge_id' );
150:             $user_details ['is_active'] = ( $this->input->post ( 'is_active' ) == 'active' ? '1' : '0' );
151:             $this->load->model ( 'Users' );
152:             $this->Users->update_user ( $user_id, $user_details );
153:             $this->lists ();
154:         }
155:     }
156:     public function lists() {
157:         authenticate_log_in ();
158:         $this->load->model ( 'Users' );
159:         $queried_user_list = $this->Users->get_all_users ();
160:         $data = array ();
161:         $data ['user_list'] = array ();
162:         if ( issetandnotempty ( $queried_user_list ) ) {
```

user.php

```
163:         foreach ( $queried_user_list as $user_list ) {
164:             $queryresult = array ();
165:             $queryresult ['id'] = $user_list ['id'];
166:             $queryresult ['name'] = $user_list ['first_name'] . ' '
. $user_list ['last_name'];
167:             $queryresult ['username'] = $user_list ['username'];
168:             $queryresult ['is_active'] = ($user_list ['is_active']
== '1' ? 'active' : 'inactive');
169:             $this->load->model ( 'Priviledge' );
170:             $queried_priviledge_details = $this->Priviledge->get_pr
iviledge_details ( $user_list ['priviledge_id'] );
171:             $queryresult ['priviledge'] = $queried_priviledge_detai
ls ['name'];
172:             array_push ( $data ['user_list'], $queryresult );
173:         }
174:     }
175:     $this->load->view ( 'user/list', $data );
176: }
177: public function activate($user_id = NULL) {
178:     authenticate_log_in ();
179:     $this->load->model ( 'Users' );
180:     $this->Users->activate_user ( $user_id );
181:     $this->lists ();
182: }
183: public function deactivate($user_id = NULL) {
184:     authenticate_log_in ();
185:     $this->load->model ( 'Users' );
186:     $this->Users->deactivate_user ( $user_id );
187:     $this->lists ();
188: }
189: function valid_cellphone_number($str) {
190:     return valid_cellphone_number ( $str );
191: }
192: }
193: ?>
```

dashboard.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Dashboard extends CI_Controller {
5:     private function company_details() {
6:         $this->load->model ( 'Company_details' );
7:         $queried_company_details = $this->Company_details->get_company_details
( );
8:         $session_details = array (
9:             'company_acronym' => $queried_company_details ['acronym
'],
10:             'company_site' => $queried_company_details ['site']
11:         );
12:         $this->session->set_userdata ( $session_details );
13:     }
14:     public function index() {
15:         $this->company_details ( );
16:         authenticate_log_in ( );
17:
18:         $this->load->model ( 'Partner_companies' );
19:         $data ['company_count'] = $this->Partner_companies->count_companies ( );
20:
21:         $this->load->model ( 'Users' );
22:         $data ['user_count'] = $this->Users->count_users ( );
23:
24:         $this->load->model ( 'Members' );
25:         $data ['member_count'] = $this->Members->count_members ( );
26:
27:         $this->load->view ( 'dashboard/dashboard', $data );
28:     }
29:     public function accounts() {
30:         $this->load->model ( 'Users' );
31:         $data ['user_count'] = $this->Users->count_users ( );
32:
33:         authenticate_log_in ( );
34:         $this->load->view ( 'dashboard/accounts', $data );
35:     }
36:     public function companies() {
37:         $this->load->model ( 'Partner_companies' );
38:         $data ['company_count'] = $this->Partner_companies->count_companies ( );
39:
40:         authenticate_log_in ( );
41:         $this->load->view ( 'dashboard/companies', $data );
42:     }
43:     public function messaging() {
44:         authenticate_log_in ( );
45:         $this->load->view ( 'dashboard/messaging' );
46:     }
47:     public function partner($partner_company_id = NULL) {
48:         authenticate_log_in ( );
49:         $this->load->model ( 'Partner_companies' );
50:         $queried_company_details = $this->Partner_companies->get_company_detail
s ( $partner_company_id );
51:         $data ['partner_company_id'] = $partner_company_id;
52:         $data ['partner_company_name'] = $queried_company_details ['name'];
53:
54:         $this->load->model ( 'Partner_companies_personnel' );
55:         $this->load->model ( 'Partner_companies_services' );
56:         $data ['active_personnel_count'] = $this->Partner_companies_personnel->
count_active_personnel ( $partner_company_id );
57:         $data ['active_service_count'] = $this->Partner_companies_services->cou
nt_active_service ( $partner_company_id );
58:
```

dashboard.php

```
59:             $this->load->view ( 'dashboard/partner', $data );
60:         }
61:     public function personnel($partner_company_id = NULL) {
62:         authenticate_log_in ();
63:         $this->load->model ( 'Partner_companies' );
64:         $queried_company_details = $this->Partner_companies->get_company_detail
s ( $partner_company_id );
65:         $data ['partner_company_id'] = $partner_company_id;
66:         $data ['partner_company_name'] = $queried_company_details ['name'];
67:         $this->load->view ( 'dashboard/personnel', $data );
68:     }
69:     public function profiles() {
70:         $this->load->model ( 'Members' );
71:         $data ['member_count'] = $this->Members->count_members ();
72:
73:         authenticate_log_in ();
74:         $this->load->view ( 'dashboard/profiles', $data );
75:     }
76:     public function services($partner_company_id = NULL) {
77:         authenticate_log_in ();
78:         $this->load->model ( 'Partner_companies' );
79:         $queried_company_details = $this->Partner_companies->get_company_detail
s ( $partner_company_id );
80:         $data ['partner_company_id'] = $partner_company_id;
81:         $data ['partner_company_name'] = $queried_company_details ['name'];
82:         $this->load->view ( 'dashboard/services', $data );
83:     }
84: }
85: ?>
```

service.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Service extends CI_Controller {
5:     public function register($selected_partner_company_id = NULL) {
6:         authenticate_log_in ();
7:         $this->load->model ( 'Partner_companies' );
8:         $queried_partner_companies_list = $this->Partner_companies->get_all_partner_companies ();
9:         $data ['partner_company_id'] = array ();
10:        foreach ( $queried_partner_companies_list as $partner_companies_list )
11:        {
12:            $data ['partner_company_id'] = $data ['partner_company_id'] + array (
13:                $partner_companies_list ['partner_company_id'] => $partner_companies_list ['name']
14:            );
15:        }
16:        $data ['selected_partner_company_id'] = $selected_partner_company_id;
17:        $this->load->view ( 'service/registration', $data );
18:    }
19:    public function process_registration() {
20:        authenticate_log_in ();
21:        $partner_company_service_details = array ();
22:        $partner_company_service_details ['partner_company_id'] = $this->input->post ( 'partner_company_id' );
23:        $config = array (
24:            array (
25:                'field' => 'name',
26:                'label' => 'Service',
27:                'rules' => 'required|trim|xss_clean'
28:            ),
29:            array (
30:                'field' => 'description',
31:                'label' => 'Description',
32:                'rules' => 'required|trim|xss_clean'
33:            )
34:        );
35:        $this->form_validation->set_rules ( $config );
36:        if ($this->form_validation->run () == FALSE) {
37:            $this->register ( $partner_company_service_details ['partner_company_id'] );
38:        } else {
39:            $partner_company_service_details ['name'] = $this->input->post ( 'name' );
40:            $partner_company_service_details ['description'] = $this->input->post ( 'description' );
41:            $partner_company_service_details ['is_available'] = ($this->input->post ( 'is_available' ) == 'available' ? '1' : '0');
42:            $this->load->model ( 'Partner_companies_services' );
43:            $this->Partner_companies_services->save_service ( $partner_company_service_details );
44:            $this->lists ( $partner_company_service_details ['partner_company_id'] );
45:        }
46:    }
47: }
48: public function update($service_id = NULL) {
49:     authenticate_log_in ();
50:     $data = array ();
51:     $this->load->model ( 'Partner_companies_services' );
52:     $queried_partner_company_service_details = $this->Partner_companies_ser
```


service.php

```
vices->get_service_details ( $service_id );
53:         $data ['service_id'] = $queried_partner_company_service_details ['id'];
54:         $data ['name'] = $queried_partner_company_service_details ['name'];
55:         $data ['description'] = $queried_partner_company_service_details ['description'];
56:         $data ['selected_partner_company_id'] = $queried_partner_company_service_details ['partner_company_id'];
57:         $data ['is_available'] = $queried_partner_company_service_details ['is_available'];
58:         $this->load->model ( 'Partner_companies' );
59:         $queried_partner_companies_list = $this->Partner_companies->get_all_partner_companies ();
60:         $data ['partner_company_id'] = array ();
61:         foreach ( $queried_partner_companies_list as $partner_companies_list )
        {
62:             $data ['partner_company_id'] = $data ['partner_company_id'] + array (
63:                 $partner_companies_list ['partner_company_id']
=> $partner_companies_list ['name']
64:             );
65:         }
66:         $this->load->view ( 'service/update', $data );
67:     }
68:     public function process_update() {
69:         authenticate_log_in ();
70:
71:         $partner_company_service_id = $this->input->post ( 'service_id' );
72:         $config = array (
73:             array (
74:                 'field' => 'name',
75:                 'label' => 'Service',
76:                 'rules' => 'required|trim|xss_clean'
77:             ),
78:             array (
79:                 'field' => 'description',
80:                 'label' => 'Description',
81:                 'rules' => 'required|trim|xss_clean'
82:             )
83:         );
84:         $this->form_validation->set_rules ( $config );
85:         if ( $this->form_validation->run () == FALSE ) {
86:             $this->update ( $partner_company_service_id );
87:         } else {
88:             $partner_company_service_details = array ();
89:             $partner_company_service_details ['name'] = $this->input->post ( 'name' );
90:             $partner_company_service_details ['description'] = $this->input->post ( 'description' );
91:             $partner_company_service_details ['partner_company_id'] = $this->input->post ( 'partner_company_id' );
92:             $partner_company_service_details ['is_available'] = ( $this->input->post ( 'is_available' ) == 'available' ? '1' : '0' );
93:             $this->load->model ( 'Partner_companies_services' );
94:             $this->Partner_companies_services->update_service ( $partner_company_service_id, $partner_company_service_details );
95:             $this->lists ( $partner_company_service_details ['partner_company_id'] );
96:         }
97:     }
98:     public function lists($partner_company_id = NULL) {
99:         authenticate_log_in ();
100:        $this->load->model ( 'Partner_companies_services' );
101:        $queried_partner_company_service_list = $this->Partner_companies_servic
```

service.php

```
es->get_all_services_of_company ( $partner_company_id );
102:         $data = array ();
103:         $data ['partner_company_id'] = $partner_company_id;
104:         $data ['partner_company_service_list'] = array ();
105:         if (issetandnotempty ( $queried_partner_company_service_list )) {
106:             foreach ( $queried_partner_company_service_list as $partner_com
pany_service_list ) {
107:                 $queryresult = array ();
108:                 $queryresult ['service_id'] = $partner_company_service_
list ['id'];
109:                 $queryresult ['name'] = $partner_company_service_list [
'name'];
110:                 $queryresult ['description'] = $partner_company_service
_list ['description'];
111:                 $this->load->model ( 'Partner_companies' );
112:                 $queried_partner_company_details = $this->Partner_compa
nies->get_company_details ( $partner_company_id );
113:                 $queryresult ['partner_company_id'] = $queried_partner_
company_details ['name'];
114:                 $data ['partner_company_name'] = $queryresult ['partner
_company_id'];
115:                 $queryresult ['is_available'] = ($partner_company_servi
ce_list ['is_available'] == '1' ? 'available' : 'inavailable');
116:                 array_push ( $data ['partner_company_service_list'], $q
ueryresult );
117:             }
118:         }
119:         $this->load->view ( 'service/list', $data );
120:     }
121:     public function make_available($service_id = NULL) {
122:         authenticate_log_in ();
123:         $this->load->model ( 'Partner_companies_services' );
124:         $this->Partner_companies_services->make_available_service ( $service_id
);
125:         $queried_service_details = $this->Partner_companies_services->get_servi
ce_details ( $service_id );
126:         $this->lists ( $queried_service_details ['partner_company_id'] );
127:     }
128:     public function make_unavailable($service_id = NULL) {
129:         authenticate_log_in ();
130:         $this->load->model ( 'Partner_companies_services' );
131:         $this->Partner_companies_services->make_unavailable_service ( $service_
id );
132:         $queried_service_details = $this->Partner_companies_services->get_servi
ce_details ( $service_id );
133:         $this->lists ( $queried_service_details ['partner_company_id'] );
134:     }
135: }
136: ?>
```

cron.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Cron extends CI_Controller {
5:     function index() {
6:         $this->load->helper ( 'url' );
7:         redirect ( 'messaging/billing_reminder' );
8:     }
9: }
```

messaging.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Messaging extends CI_Controller {
5:     public function referral() {
6:         authenticate_log_in ();
7:         $this->load->model ( 'Partner_companies' );
8:         if ($this->Partner_companies->companies_exist ()) {
9:             $queried_partner_companies_list = $this->Partner_companies->get
_all_partner_companies ();
10:             $data ['partner_company_id'] = array ();
11:             foreach ( $queried_partner_companies_list as $partner_companies
_list ) {
12:                 $data ['partner_company_id'] = $data ['partner_company_
id'] + array (
13:                     $partner_companies_list ['partner_compa
ny_id'] => $partner_companies_list ['name']
14:                 );
15:             }
16:             $this->load->view ( 'messaging/referral', $data );
17:         } else {
18:             redirect ( 'partner/register' );
19:         }
20:     }
21:     public function process_referral_get_service() {
22:         authenticate_log_in ();
23:         $config = array (
24:             array (
25:                 'field' => 'pin',
26:                 'label' => 'PIN',
27:                 'rules' => 'callback_valid_member_with_
cellphone_number|required|trim|xss_clean'
28:             ),
29:             array (
30:                 'field' => 'partner_company_id',
31:                 'label' => 'Partner Company',
32:                 'rules' => 'trim|xss_clean'
33:             )
34:         );
35:         $this->form_validation->set_rules ( $config );
36:         if ($this->form_validation->run () == FALSE) {
37:             $this->referral ();
38:         } else {
39:             $pin = $this->input->post ( 'pin' );
40:             $company_id = $this->input->post ( 'partner_company_id' );
41:             $this->load->model ( 'Members' );
42:             $queried_member_details = $this->Members->get_member_details (
$pin );
43:             $data ['pin'] = $pin;
44:             $data ['name'] = $queried_member_details ['first_name'] . ' ' .
$queried_member_details ['last_name'];
45:             $data ['company_id'] = $company_id;
46:             $this->load->model ( 'Partner_companies' );
47:             $queried_partner_companies_list = $this->Partner_companies->get
_all_partner_companies ();
48:             $data ['partner_company_id'] = array ();
49:             foreach ( $queried_partner_companies_list as $partner_companies
_list ) {
50:                 $data ['partner_company_id'] = $data ['partner_company_
id'] + array (
51:                     $partner_companies_list ['partner_compa
ny_id'] => $partner_companies_list ['name']
52:                 );
```

messaging.php

```
53:     }
54:     $this->load->model ( 'Partner_companies_services' );
55:     if ( $this->Partner_companies_services->services_exists ( $company_id )) {
56:         $queried_partner_company_services_list = $this->Partner_companies_services->get_all_active_services_of_company ( $company_id );
57:         $data ['partner_company_services_id'] = array ();
58:         foreach ( $queried_partner_company_services_list as $partner_company_services_list ) {
59:             $data ['partner_company_services_id'] = $data ['partner_company_services_id'] + array (
60:                 $partner_company_services_list ['id'] => $partner_company_services_list ['name']
61:             );
62:         }
63:     } else {
64:         return redirect ( 'service/register' );
65:     }
66:     $this->load->model ( 'Partner_companies_personnel' );
67:     if ( $this->Partner_companies_personnel->personnel_exists ( $company_id )) {
68:         $queried_partner_company_personnel_list = $this->Partner_companies_personnel->get_all_active_personnel_of_company_id ( $company_id );
69:         $data ['partner_company_personnel_id'] = array ();
70:         foreach ( $queried_partner_company_personnel_list as $partner_company_personnel_list ) {
71:             $data ['partner_company_personnel_id'] = $data ['partner_company_personnel_id'] + array (
72:                 $partner_company_personnel_list ['id'] => $partner_company_personnel_list ['first_name'] . ' ' . $partner_company_personnel_list ['last_name']
73:             );
74:         }
75:     }
76:     $this->load->view ( 'messaging/service', $data );
77: } else {
78:     return redirect ( 'personnel/register' );
79: }
80: }
81: }
82: public function process_referral() {
83:     authenticate_log_in ();
84:
85:     $pin = $this->input->post ( 'pin' );
86:     $name = $this->input->post ( 'name' );
87:     $this->load->model ( 'Members' );
88:     $queried_member_details = $this->Members->get_member_details ( $pin );
89:     $member_first_name = $queried_member_details ['first_name'];
90:     $member_cellphone_number = $queried_member_details ['cellphone_number'];
91:
92:     $partner_company_id = $this->input->post ( 'partner_company_id' );
93:     $this->load->model ( 'Partner_companies' );
94:     $queried_company_details = $this->Partner_companies->get_company_details ( $partner_company_id );
95:     $partner_company_acronym = $queried_company_details ['acronym'];
96:     $partner_company_services_id = $this->input->post ( 'partner_company_services_id' );
97:     $this->load->model ( 'Partner_companies_services' );
98:     $queried_partner_company_service_details = $this->Partner_companies_services->get_service_details ( $partner_company_services_id );
99:     $partner_company_service = $queried_partner_company_service_details ['name'];
100:     $partner_company_personnel_id = $this->input->post ( 'partner_company_p
```

messaging.php

```
ersonnel_id' );
100:         $this->load->model ( 'Partner_companies_personnel' );
101:         $queried_partner_company_personnel_details = $this->Partner_companies_p
ersonnel->get_personnel_details ( $partner_company_personnel_id );
102:         $partner_company_personnel_name = $queried_partner_company_personnel_de
tails ['first_name'] . ' ' . $queried_partner_company_personnel_details ['last_name'];
103:         $partner_region_responsibility_id = $queried_partner_company_personnel_
details ['region_responsibility_id'];
104:         $partner_company_personnel_cellphone_number = $queried_partner_company_
personnel_details ['cellphone_number'];
105:         $this->load->Model ( 'Regions' );
106:         $queried_region_details = $this->Regions->get_region_details ( $partner
_region_responsibility_id );
107:         $partner_region_responsibility = $queried_region_details ['name'];
108:         $this->load->model ( 'Company_details' );
109:         $queried_company_details = $this->Company_details->get_company_details
( );
110:         $company_acronym = $queried_company_details ['acronym'];
111:         $this->load->model ( 'Message_formats' );
112:         $this->load->model ( 'Sms_history' );
113:         $this->load->library ( 'session' );
114:         $format_id_member_referral = 'MHICCRMTXT04';
115:         $message_type_member_referral = 'I';
116:         $communication_type_member_referral = '0';
117:         $transaction_no_member_referral = $this->generate_transaction_number (
$message_type_member_referral );
118:         $queried_member_referral_format_parameters = array (
119:             $company_acronym,
120:             $member_first_name,
121:             $partner_company_acronym,
122:             $partner_company_service,
123:             $partner_company_personnel_name,
124:             $partner_region_responsibility,
125:             $transaction_no_member_referral
126:         );
127:         $queried_member_referral_format_details = $this->Message_formats->get_m
essage_format_details ( $format_id_member_referral, $message_type_member_referral, $communicat
ion_type_member_referral );
128:         $message_member_referral = vsprintf ( $queried_member_referral_format_d
etails, $queried_member_referral_format_parameters );
129:         $message_member_referral_details = array (
130:             'message' => $message_member_referral,
131:             'sender_number' => NULL,
132:             'receiver_number' => $member_cellphone_number,
133:             'member_pin' => $pin,
134:             'user_id' => $this->session->userdata ( 'user_id' ),
135:             'partner_company_id' => NULL,
136:             'communication_type' => $communication_type_member_refe
rral,
137:             'message_type' => $message_type_member_referral
138:         );
139:         $this->Sms_history->save_message ( $transaction_no_member_referral, $me
ssage_member_referral_details );
140:         $this->send ( $member_cellphone_number, $message_member_referral );
141:         $format_id_partner_referral = 'MHICCRMTXT05';
142:         $message_type_partner_referral = 'I';
143:         $communication_type_partner_referral = '0';
144:         $transaction_no_partner_referral = $this->generate_transaction_number (
$message_type_partner_referral );
145:         $queried_partner_referral_format_parameters = array (
146:             $company_acronym,
147:             $member_first_name,
148:             $pin,
```

messaging.php

```
149:             $partner_company_service,
150:             $transaction_no_partner_referral
151:         );
152:         $queried_partner_referral_format_details = $this->Message_formats->get_
message_format_details ( $format_id_partner_referral, $message_type_partner_referral, $communi
cation_type_partner_referral );
153:         $message_partner_referral = vsprintf ( $queried_partner_referral_format
_details, $queried_partner_referral_format_parameters );
154:         $message_partner_referral_details = array (
155:             'message' => $message_partner_referral,
156:             'sender_number' => NULL,
157:             'receiver_number' => $partner_company_personnel_cellpho
ne_number,
158:             'member_pin' => NULL,
159:             'user_id' => $this->session->userdata ( 'user_id' ),
160:             'partner_company_id' => $partner_company_id,
161:             'communication_type' => $communication_type_member_refe
rral,
162:             'message_type' => $message_type_member_referral
163:         );
164:         $this->Sms_history->save_message ( $transaction_no_partner_referral, $m
essage_partner_referral_details );
165:         $this->send ( $partner_company_personnel_cellphone_number, $message_par
tner_referral );
166:         $this->history ();
167:     }
168:     public function history() {
169:         authenticate_log_in ();
170:         $this->load->helper ( 'variable' );
171:         $this->load->model ( 'Sms_history' );
172:         $queried_message_list = $this->Sms_history->get_all_messages ();
173:         $data ['message_list'] = array ();
174:         if (issetandnotempty ( $queried_message_list )) {
175:             foreach ( $queried_message_list as $message_list ) {
176:                 $message_element = array (
177:                     'id' => $message_list ['id'],
178:                     'message' => $message_list ['message'],
179:                     'sender_number' => $message_list ['send
er_number'],
180:                     'receiver_number' => $message_list ['re
ceiver_number'],
181:                     'member_pin' => $message_list ['member_
pin'],
182:                     'user_id' => '',
183:                     'partner_company_id' => '',
184:                     'date' => $message_list ['date']
185:                 );
186:                 if (issetandnotempty ( 'user_id', $message_list )) {
187:                     $this->load->model ( 'Partner_companies_personn
el' );
188:                     $queried_personnel_details = $this->Partner_com
panies_personnel->get_personnel_details ( $message_list ['user_id'] );
189:                     $message_element ['user_id'] = $queried_personn
el_details ['first_name'] + $queried_personnel_details ['last_name'];
190:                 }
191:                 if (issetandnotempty ( 'partner_company_id', $message_l
ist )) {
192:                     $this->load->model ( 'Partner_companies' );
193:                     $queried_partner_company_details = $this->Partn
er_companies->get_company_details ( $message_list ['partner_company_id'] );
194:                     $message_element ['partner_company_id'] = $quer
ied_partner_company_details ['name'];
195:                 }
```

messaging.php

```
196:             $this->load->model ( 'Communication_types' );
197:             $queried_communication_type_details = $this->Communication_types->get_communication_type_details ( $message_list ['communication_type'] );
198:             $message_element ['communication_type'] = $queried_communication_type_details ['name'];
199:             $this->load->model ( 'Message_types' );
200:             $queried_message_type_details = $this->Message_types->get_message_type_details ( $message_list ['message_type'] );
201:             $message_element ['message_type'] = $queried_message_type_details ['name'];
202:             array_push ( $data ['message_list'], $message_element );
;
203:         }
204:     }
205:     $this->load->view ( 'messaging/history', $data );
206: }
207: public function advisory() {
208:     authenticate_log_in ();
209:     $data = array ();
210:     $this->load->model ( 'Regions' );
211:     $queried_region_list = $this->Regions->get_all_regions ();
212:     $data ['home_region'] = array ();
213:     foreach ( $queried_region_list as $region_list ) {
214:         $region_element = array (
215:             'id' => $region_list ['id'],
216:             'name' => $region_list ['name']
217:         );
218:         array_push ( $data ['home_region'], $region_element );
219:     }
220:     $this->load->model ( 'Membership_types' );
221:     $queried_membership_list = $this->Membership_types->get_all_membership_types ();
222:     $data ['membership'] = array ();
223:     foreach ( $queried_membership_list as $membership_list ) {
224:         $membership_element = array (
225:             'id' => $membership_list ['id'],
226:             'name' => $membership_list ['name']
227:         );
228:         array_push ( $data ['membership'], $membership_element );
229:     }
230:     $this->load->model ( 'Payment_types' );
231:     $queried_payment_type_list = $this->Payment_types->get_all_payment_types ();
232:     $data ['payment_type'] = array ();
233:     foreach ( $queried_payment_type_list as $payment_type_list ) {
234:         $payment_element = array (
235:             'id' => $payment_type_list ['id'],
236:             'name' => $payment_type_list ['name']
237:         );
238:         array_push ( $data ['payment_type'], $payment_element );
239:     }
240:     $this->load->view ( 'messaging/advisory', $data );
241: }
242: public function process_advisory() {
243:     authenticate_log_in ();
244:
245:     $home_region = $this->input->post ( 'home_region' );
246:     $membership = $this->input->post ( 'membership' );
247:     $payment_type = $this->input->post ( 'payment_type' );
248:     $notifiable = '1';
249:     $advisory = $this->input->post ( 'advisory' );
250:     $this->load->model ( 'Members' );
251:     $queried_member_list = $this->Members->get_all_filtered_members ( $home
```


messaging.php

```
_region, $membership, $payment_type, $notifiable, NULL );
252:     $this->load->model ( 'Message_formats' );
253:     $this->load->model ( 'Sms_history' );
254:     $this->load->library ( 'session' );
255:     $this->load->model ( 'Company_details' );
256:     $queried_company_details = $this->Company_details->get_company_details
();
257:     $company_acronym = $queried_company_details ['acronym'];
258:     $format_id_advisory = 'MHICCRMTXT06';
259:     $message_type_advisory = 'N';
260:     $communication_type_advisory = 'O';
261:     $queried_advisory_format_details = $this->Message_formats->get_message_
format_details ( $format_id_advisory, $message_type_advisory, $communication_type_advisory );
262:     foreach ( $queried_member_list as $member_list ) {
263:         $member_cellphone_number = $member_list ['cellphone_number'];
264:         if (issetandnotempty ( $member_cellphone_number )) {
265:             $pin = $member_list ['pin'];
266:             $transaction_no_advisory = $this->generate_transaction_
number ( $message_type_advisory );
267:             $queried_advisory_format_parameters = array (
268:                 $company_acronym,
269:                 $advisory,
270:                 $transaction_no_advisory
271:             );
272:             $message_advisory = vsprintf ( $queried_advisory_format
_details, $queried_advisory_format_parameters );
273:             $message_advisory_details = array (
274:                 'message' => $message_advisory,
275:                 'sender_number' => NULL,
276:                 'receiver_number' => $member_cellphone_
number,
277:                 'member_pin' => $pin,
278:                 'user_id' => $this->session->userdata (
'user_id' ),
279:                 'partner_company_id' => NULL,
280:                 'communication_type' => $communication_
type_advisory,
281:                 'message_type' => $message_type_advisor
y
282:             );
283:             $this->Sms_history->save_message ( $transaction_no_advi
sory, $message_advisory_details );
284:             $this->send ( $member_cellphone_number, $message_adviso
ry );
285:         }
286:     }
287:     $this->history ();
288: }
289: public function billing_reminder() {
290:     echo date ( 'Y-m-d' );
291:     $this->load->helper ( 'variable' );
292:     $month = date ( 'n' );
293:     $day = date ( 'j' );
294:     $lastday = date ( 't' );
295:     echo $lastday . ' - ' . $day . ' - ' . $month;
296:     $payment_type_array = array ();
297:     if (($month / 3 === 4)) {
298:         $payment_type_array = array (
299:             '1',
300:             '2',
301:             '3',
302:             '4'
303:         );
};
```

messaging.php

```
304:         } else if (($month / 3 === 3)) {
305:             $payment_type_array = array (
306:                 '1',
307:                 '2'
308:             );
309:         } else if (($month / 3 === 2)) {
310:             $payment_type_array = array (
311:                 '1',
312:                 '2',
313:                 '3'
314:             );
315:         } else if (($month / 3 === 1)) {
316:             $payment_type_array = array (
317:                 '1',
318:                 '2'
319:             );
320:         } else {
321:             $payment_type_array = array (
322:                 '1'
323:             );
324:         }
325:
326:         if ($day == $lastday) {
327:             $this->missed_payment ();
328:             $this->due_billing_reminder ( $payment_type_array );
329:         } elseif ($day == ($lastday - 14)) {
330:             $this->missed_payment ();
331:             $this->overhead_billing_reminder ( $payment_type_array );
332:         }
333:     }
334:     public function overhead_billing_reminder($payment_type) {
335:         $notifiable = '1';
336:         $this->load->model ( 'Members' );
337:         $queried_member_list = $this->Members->get_all_filtered_members ( NULL,
NULL, $payment_type, $notifiable, NULL );
338:         $this->load->model ( 'Message_formats' );
339:         $this->load->model ( 'Sms_history' );
340:         $this->load->library ( 'session' );
341:         $this->load->model ( 'Payment_types' );
342:         $this->load->model ( 'Company_details' );
343:         $queried_company_details = $this->Company_details->get_company_details
();
344:         $company_acronym = $queried_company_details ['acronym'];
345:         $format_id_billing = 'MHICCRMTXT01';
346:         $message_type_billing = 'B';
347:         $communication_type_billing = 'O';
348:         $queried_billing_format_details = $this->Message_formats->get_message_f
ormat_details ( $format_id_billing, $message_type_billing, $communication_type_billing );
349:         $due_date = date ( 't-M-Y' );
350:         foreach ( $queried_member_list as $member_list ) {
351:             $member_cellphone_number = $member_list ['cellphone_number'];
352:             if (issetandnotempty ( $member_cellphone_number )) {
353:                 $pin = $member_list ['pin'];
354:                 $first_name = $member_list ['first_name'];
355:                 $payment_amount = $member_list ['payment_amount'];
356:                 $queried_payment_type = $this->Payment_types->get_payme
nt_type_details ( $member_list ['payment_type'] );
357:                 $payment_type = $queried_payment_type ['name'];
358:                 $transaction_no_billing = $this->generate_transaction_n
umber ( $message_type_billing );
359:                 $queried_billing_format_parameters = array (
360:                     $company_acronym,
361:                     $first_name,
```

messaging.php

```
362:                                     $payment_amount,
363:                                     $payment_type,
364:                                     $due_date,
365:                                     $transaction_no_billing
366:                                 );
367:                                 $message_billing = vsprintf ( $queried_billing_format_d
etails, $queried_billing_format_parameters );
368:                                 $message_billing_details = array (
369:                                     'message' => $message_billing,
370:                                     'sender_number' => NULL,
371:                                     'receiver_number' => $member_cellphone_
number,
372:                                     'member_pin' => $pin,
373:                                     'user_id' => $this->session->userdata (
'user_id' ),
374:                                     'partner_company_id' => NULL,
375:                                     'communication_type' => $communication_
type_billing,
376:                                     'message_type' => $message_type_billing
377:                                 );
378:                                 $this->Sms_history->save_message ( $transaction_no_bill
ing, $message_billing_details );
379:                                 $this->send ( $member_cellphone_number, $message_billin
g );
380:                                 }
381:                             }
382:                         }
383:         public function due_billing_reminder($payment_type) {
384:             $notifiable = '1';
385:             $this->load->model ( 'Members' );
386:             $queried_member_list = $this->Members->get_all_filtered_members ( NULL,
NULL, $payment_type, $notifiable, NULL );
387:             $this->load->model ( 'Message_formats' );
388:             $this->load->model ( 'Sms_history' );
389:             $this->load->library ( 'session' );
390:             $this->load->model ( 'Payment_types' );
391:             $this->load->model ( 'Company_details' );
392:             $queried_company_details = $this->Company_details->get_company_details
( );
393:             $company_acronym = $queried_company_details ['acronym'];
394:             $format_id_billing = 'MHICCRMTXT02';
395:             $message_type_billing = 'R';
396:             $communication_type_billing = '0';
397:             $queried_billing_format_details = $this->Message_formats->get_message_f
ormat_details ( $format_id_billing, $message_type_billing, $communication_type_billing );
398:             foreach ( $queried_member_list as $member_list ) {
399:                 $member_cellphone_number = $member_list ['cellphone_number'];
400:                 if (issetandnotempty ( $member_cellphone_number )) {
401:                     $pin = $member_list ['pin'];
402:                     $first_name = $member_list ['first_name'];
403:                     $payment_amount = $member_list ['payment_amount'];
404:                     $queried_payment_type = $this->Payment_types->get_payme
nt_type_details ( $member_list ['payment_type'] );
405:                     $payment_type = $queried_payment_type ['name'];
406:                     $transaction_no_billing = $this->generate_transaction_n
umber ( $message_type_billing );
407:                     $queried_billing_format_parameters = array (
408:                         $company_acronym,
409:                         $first_name,
410:                         $payment_amount,
411:                         $payment_type,
412:                         $transaction_no_billing
413:                     );
```

messaging.php

```
414:             $message_billing = vsprintf ( $queried_billing_format_d
etails, $queried_billing_format_parameters );
415:             $message_billing_details = array (
416:                 'message' => $message_billing,
417:                 'sender_number' => NULL,
418:                 'receiver_number' => $member_cellphone_
number,
419:                 'member_pin' => $pin,
420:                 'user_id' => $this->session->userdata (
'user_id' ),
421:                 'partner_company_id' => NULL,
422:                 'communication_type' => $communication_
type_billing,
423:                 'message_type' => $message_type_billing
);
424:             );
425:             $this->Sms_history->save_message ( $transaction_no_bill
ing, $message_billing_details );
426:             $this->send ( $member_cellphone_number, $message_billin
g );
427:         }
428:     }
429: }
430:     public function missed_payment() {
431:         $notifiable = '1';
432:         $this->load->model ( 'Members' );
433:         $queried_member_list = $this->Members->get_all_filtered_members ( NULL,
NULL, NULL, $notifiable, '0' );
434:         $this->load->model ( 'Message_formats' );
435:         $this->load->model ( 'Sms_history' );
436:         $this->load->library ( 'session' );
437:         $this->load->model ( 'Payment_types' );
438:         $this->load->model ( 'Company_details' );
439:         $queried_company_details = $this->Company_details->get_company_details
();
440:         $company_acronym = $queried_company_details ['acronym'];
441:         $format_id_billing = 'MHICCRMTXT03';
442:         $message_type_billing = 'A';
443:         $communication_type_billing = '0';
444:         $queried_billing_format_details = $this->Message_formats->get_message_f
ormat_details ( $format_id_billing, $message_type_billing, $communication_type_billing );
445:         foreach ( $queried_member_list as $member_list ) {
446:             $member_cellphone_number = $member_list ['cellphone_number'];
447:             if (issetandnotempty ( $member_cellphone_number )) {
448:                 $pin = $member_list ['pin'];
449:                 $first_name = $member_list ['first_name'];
450:                 $payment_amount = $member_list ['payment_amount'];
451:                 $queried_payment_type = $this->Payment_types->get_payme
nt_type_details ( $member_list ['payment_type'] );
452:                 $payment_type = $queried_payment_type ['name'];
453:                 $transaction_no_billing = $this->generate_transaction_n
umber ( $message_type_billing );
454:                 $queried_billing_format_parameters = array (
455:                     $company_acronym,
456:                     $first_name,
457:                     $payment_amount,
458:                     $payment_type,
459:                     $transaction_no_billing
);
460:             );
461:             $message_billing = vsprintf ( $queried_billing_format_d
etails, $queried_billing_format_parameters );
462:             $message_billing_details = array (
463:                 'message' => $message_billing,
464:                 'sender_number' => NULL,
```

messaging.php

```
465:                                     'receiver_number' => $member_cellphone_
number,
466:                                     'member_pin' => $pin,
467:                                     'user_id' => $this->session->userdata (
'user_id' ),
468:                                     'partner_company_id' => NULL,
469:                                     'communication_type' => $communication_
type_billing,
470:                                     'message_type' => $message_type_billing
471:                                     );
472:                                     $this->Sms_history->save_message ( $transaction_no_bill
ing, $message_billing_details );
473:                                     $this->send ( $member_cellphone_number, $message_billin
g );
474:                                     }
475:                                 }
476:         }
477:     public function generate_transaction_number($message_type) {
478:         $this->load->helper ( 'date' );
479:         $time = time ();
480:         $yearstring = '%y';
481:         $year = mdate ( $yearstring, $time );
482:         $monthstring = '%m';
483:         $month = mdate ( $monthstring, $time );
484:         $daystring = '%d';
485:         $day = mdate ( $daystring, $time );
486:         $transaction_prefix = $message_type . $year . $month . $day;
487:         $this->db->select ( 'ID, DATE' );
488:         $this->db->from ( 'SMS_HISTORY' );
489:         $this->db->like ( 'ID', $transaction_prefix, 'after' );
490:         $this->db->order_by ( 'ID', 'DESC' );
491:         $this->db->limit ( 1 );
492:         $query = $this->db->get ();
493:         $counter = '0000000000';
494:         if ($query->num_rows () > 0) {
495:             $row = $query->row_array ();
496:             $transaction_no = $row ['ID'];
497:             $counter = substr ( $transaction_no, - 10 ) + 1;
498:         }
499:         $transaction_no = $transaction_prefix . str_pad ( $counter, 10, '0', ST
R_PAD_LEFT );
500:         return $transaction_no;
501:     }
502:     function send($cellphone_number, $message) {
503:         $message = urldecode ( $message );
504:         $message = urlencode ( $message );
505:         $wap_format_string = "http://localhost:%d/cgi-bin/sendsms?username=%s&p
assword=%s&to=%s&text=%s";
506:         $this->load->model ( 'Sms_configuration' );
507:         $queried_gsm_configurations = $this->Sms_configuration->get_gsm_configu
rations ();
508:         $port = $queried_gsm_configurations ['port'];
509:         $username = $queried_gsm_configurations ['username'];
510:         $password = $queried_gsm_configurations ['password'];
511:         $wap_send_command = sprintf ( $wap_format_string, $port, $username, $pa
ssword, $cellphone_number, $message );
512:         $ch = curl_init ();
513:         curl_setopt ( $ch, CURLOPT_URL, $wap_send_command );
514:         curl_setopt ( $ch, CURLOPT_HEADER, 0 );
515:         curl_setopt ( $ch, CURLOPT_RETURNTRANSFER, 1 );
516:         $my_var = curl_exec ( $ch );
517:         curl_close ( $ch );
518:         return $my_var;
```

messaging.php

```
519:     }
520:     public function receive() {
521:         $sending_number = $this->input->get ( 'sending_number' );
522:         $receiving_number = $this->input->get ( 'receiving_number' );
523:         $message = urldecode ( $this->input->get ( 'message' ) );
524:         $communication_type = 'I';
525:         $message_details = array (
526:             'message' => $message,
527:             'sender_number' => $sending_number,
528:             'receiver_number' => $receiving_number,
529:             'communication_type' => $communication_type
530:         );
531:         $this->load->model ( 'Message_formats' );
532:         $this->load->model ( 'Sms_history' );
533:         $pin_inquiry_format_id = 'MHICCRMTXT07';
534:         $pin_inquiry_message_type = 'P';
535:         $pin_inquiry_message_format = $this->Message_formats->get_message_format_details ( $pin_inquiry_format_id, $pin_inquiry_message_type, $communication_type );
536:         $referral_inquiry_format_id = 'MHICCRMTXT08';
537:         $referral_inquiry_message_type = 'I';
538:         $referral_inquiry_message_format = $this->Message_formats->get_message_format_details ( $referral_inquiry_format_id, $referral_inquiry_message_type, $communication_type );
539:         $advisory_inquiry_format_id = 'MHICCRMTXT09';
540:         $advisory_inquiry_message_type = 'C';
541:         $advisory_inquiry_message_format = $this->Message_formats->get_message_format_details ( $advisory_inquiry_format_id, $advisory_inquiry_message_type, $communication_type );
542:         $message_components = array ();
543:         $transaction_no = '';
544:         if (preg_match ( '/' . $pin_inquiry_message_format . '/', $message, $message_components )) {
545:             $message_details ['message_type'] = $pin_inquiry_message_type;
546:             $transaction_no = $this->generate_transaction_number ( $pin_inquiry_message_type );
547:             $this->Sms_history->save_message ( $transaction_no, $message_details );
548:             $this->process_pin_inquiry ( $sending_number, $message_components );
549:         } else if (preg_match ( '/' . $advisory_inquiry_message_format . '/', $message, $message_components )) {
550:             $message_details ['message_type'] = $advisory_inquiry_message_type;
551:             $transaction_no = $this->generate_transaction_number ( $advisory_inquiry_message_type );
552:             $this->Sms_history->save_message ( $transaction_no, $message_details );
553:             $this->process_advisory_inquiry ( $sending_number, $message_components );
554:         } else if (preg_match ( '/' . $referral_inquiry_message_format . '/', $message, $message_components )) {
555:             $message_details ['message_type'] = $referral_inquiry_message_type;
556:             $transaction_no = $this->generate_transaction_number ( $referral_inquiry_message_type );
557:             $this->Sms_history->save_message ( $transaction_no, $message_details );
558:             $this->process_referral_inquiry ( $sending_number, $message_components );
559:         } else {
560:             $this->load->model ( 'Company_details' );
561:             $queried_company_details = $this->Company_details->get_company_details ();
```

messaging.php

```
562:             $company_acronym = $queried_company_details ['acronym'];
563:             echo $company_acronym . ': invalid inquiry format string. please see flyers for more details.';
564:         }
565:     }
566:     public function process_pin_inquiry($cellphone_number, $message_components) {
567:         $communication_type = 'O';
568:         $this->load->helper ( 'variable' );
569:         $this->load->model ( 'Members' );
570:         $this->load->model ( 'Company_details' );
571:         $queried_company_details = $this->Company_details->get_company_details
( );
572:         $company_acronym = $queried_company_details ['acronym'];
573:         $queried_member_details = $this->Members->get_member_details_with ( $message_components [1], $message_components [2], $message_components [3] );
574:         $message = '';
575:         if (issetandnotempty ( $queried_member_details )) {
576:             $pin_inquiry_format_id = 'MHICCRMTXT10';
577:             $pin_inquiry_message_type = 'P';
578:             $pin_inquiry_message_format = $this->Message_formats->get_message_format_details ( $pin_inquiry_format_id, $pin_inquiry_message_type, $communication_type );
579:             $message = sprintf ( $pin_inquiry_message_format, $company_acronym, $queried_member_details ['first_name'], $queried_member_details ['pin'] );
580:         } else {
581:             $pin_inquiry_format_id = 'MHICCRMTXT11';
582:             $pin_inquiry_message_type = 'P';
583:             $pin_inquiry_message_format = $this->Message_formats->get_message_format_details ( $pin_inquiry_format_id, $pin_inquiry_message_type, $communication_type );
584:             $message = sprintf ( $pin_inquiry_message_format, $company_acronym );
585:         }
586:         echo $message;
587:     }
588:     public function process_referral_inquiry($cellphone_number, $message_components) {
589:         $this->load->model ( 'Sms_history' );
590:         $this->load->model ( 'Members' );
591:         $this->load->helper ( 'variable' );
592:         $queried_member_details = $this->Members->get_member_details ( $message_components [1] );
593:         if (issetandnotempty ( $queried_member_details )) {
594:             $member_pin = $message_components [1];
595:             $message_type = 'I';
596:             $communication_type = 'O';
597:             $queried_latest_message = $this->Sms_history->get_latest_message ( $member_pin, $communication_type, $message_type );
598:             $message = '';
599:             if (issetandnotempty ( $queried_latest_message )) {
600:                 $referral_inquiry_format_id = 'MHICCRMTXT12';
601:                 $referral_inquiry_message_format = $this->Message_formats->get_message_format_details ( $referral_inquiry_format_id, $message_type, $communication_type );
602:                 $message = sprintf ( $referral_inquiry_message_format, $queried_latest_message );
603:             } else {
604:                 $this->load->model ( 'Company_details' );
605:                 $queried_company_details = $this->Company_details->get_company_details ( );
606:                 $company_acronym = $queried_company_details ['acronym'];
607:                 $referral_inquiry_format_id = 'MHICCRMTXT13';
608:                 $referral_inquiry_message_format = $this->Message_formats->get_message_format_details ( $referral_inquiry_format_id, $message_type, $communication_type );
```

messaging.php

```
pe );
609:             $message = sprintf ( $referral_inquiry_message_format,
$company_acronym, $queried_member_details ['first_name'] );
610:                 }
611:                 echo $message;
612:             } else {
613:                 $this->load->model ( 'Company_details' );
614:                 $queried_company_details = $this->Company_details->get_company_
details ();
615:                 $company_acronym = $queried_company_details ['acronym'];
616:                 echo $company_acronym . ': No valid Referral credentials receiv
ed.';
617:             }
618:         }
619:     public function process_advisory_inquiry($cellphone_number, $message_components
) {
620:         $this->load->model ( 'Sms_history' );
621:         $this->load->model ( 'Members' );
622:         $this->load->helper ( 'variable' );
623:         $this->load->model ( 'Company_details' );
624:         $queried_company_details = $this->Company_details->get_company_details
();
625:         $company_acronym = $queried_company_details ['acronym'];
626:         $queried_member_details = $this->Members->get_member_details ( $message
_components [1]);
627:         if (issetandnotempty ( $queried_member_details )) {
628:             $member_pin = $message_components [1];
629:             $message_type = 'C';
630:             $communication_type = 'O';
631:             $advisory_inquiry_format_id = 'MHICCRMTXT14';
632:             $advisory_inquiry_message_format = $this->Message_formats->get_
message_format_details ( $advisory_inquiry_format_id, $message_type, $communication_type );
633:             $choice = $message_components [2];
634:             $this->load->model ( 'Members' );
635:             $message = $company_acronym . ': invalid advisory opt option re
ceived.';
636:             if ($choice == 'ON') {
637:                 $this->Members->make_notifiable_member ( $member_pin );
638:                 $message = sprintf ( $advisory_inquiry_message_format,
$company_acronym, $queried_member_details ['first_name'], $choice );
639:             } else if ($choice == 'OFF') {
640:                 $this->Members->make_unnotifiable_member ( $member_pin
);
641:                 $message = sprintf ( $advisory_inquiry_message_format,
$company_acronym, $queried_member_details ['first_name'], $choice );
642:             }
643:             echo $message;
644:         } else {
645:             echo $company_acronym . ': No valid Referral credentials receiv
ed.';
646:         }
647:     }
648:     function valid_member_with_cellphone_number($str) {
649:         return valid_member_with_cellphone_number ( $str );
650:     }
651: }
652: ?>
```


member.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Member extends CI_Controller {
5:     public function register() {
6:         authenticate_log_in ();
7:         $this->load->model ( 'Regions' );
8:         $queried_region_list = $this->Regions->get_all_regions ();
9:         $data [ 'home_region' ] = array ();
10:        foreach ( $queried_region_list as $region_list ) {
11:            $data [ 'home_region' ] = $data [ 'home_region' ] + array (
12:                $region_list [ 'id' ] => $region_list [ 'name' ]
13:            );
14:        }
15:        $this->load->model ( 'Membership_types' );
16:        $queried_membership_list = $this->Membership_types->get_all_membership_
types ();
17:        $data [ 'membership' ] = array ();
18:        foreach ( $queried_membership_list as $membership_list ) {
19:            $data [ 'membership' ] = $data [ 'membership' ] + array (
20:                $membership_list [ 'id' ] => $membership_list [ 'n
ame' ]
21:            );
22:        }
23:        $this->load->model ( 'Payment_types' );
24:        $queried_payment_list = $this->Payment_types->get_all_payment_types ();
25:        $data [ 'payment_mode' ] = array ();
26:        foreach ( $queried_payment_list as $payment_list ) {
27:            $data [ 'payment_mode' ] = $data [ 'payment_mode' ] + array (
28:                $payment_list [ 'id' ] => $payment_list [ 'name' ]
29:            );
30:        }
31:        $this->load->view ( 'member/registration', $data );
32:    }
33:    public function process_registration() {
34:        authenticate_log_in ();
35:
36:        $config = array (
37:            array (
38:                'field' => 'pin',
39:                'label' => 'PIN',
40:                'rules' => 'is_unique[MEMBERS.PIN]|requ
ired|trim|xss_clean'
41:            ),
42:            array (
43:                'field' => 'first_name',
44:                'label' => 'First Name',
45:                'rules' => 'required|trim|xss_clean'
46:            ),
47:            array (
48:                'field' => 'middle_name',
49:                'label' => 'Middle Name',
50:                'rules' => 'required|trim|xss_clean'
51:            ),
52:            array (
53:                'field' => 'last_name',
54:                'label' => 'Last Name',
55:                'rules' => 'required|trim|xss_clean'
56:            ),
57:            array (
58:                'field' => 'cellphone_number',
59:                'label' => 'Cellphone Number',
60:                'rules' => 'callback_valid_cellphone_nu
```

member.php

```
mber|required|trim|xss_clean'
61:         ),
62:         array (
63:             'field' => 'email_address',
64:             'label' => 'Email Address',
65:             'rules' => 'valid_email|required|trim|x
ss_clean'
66:         ),
67:         array (
68:             'field' => 'home_region',
69:             'label' => 'Home Region',
70:             'rules' => 'trim|xss_clean'
71:         ),
72:         array (
73:             'field' => 'membership_type',
74:             'label' => 'Membership',
75:             'rules' => 'trim|xss_clean'
76:         ),
77:         array (
78:             'field' => 'payment_type',
79:             'label' => 'Payment mode',
80:             'rules' => 'trim|xss_clean'
81:         ),
82:         array (
83:             'field' => 'is_notifiable',
84:             'label' => 'Notifiable',
85:             'rules' => 'trim|xss_clean'
86:         )
87:     );
88:     $this->form_validation->set_rules ( $config );
89:     if ( $this->form_validation->run () == FALSE ) {
90:         $this->register ();
91:     } else {
92:         $member_details = array ();
93:         $pin = $this->input->post ( 'pin' );
94:         $member_details ['first_name'] = $this->input->post ( 'first_na
me' );
95:         $member_details ['middle_name'] = $this->input->post ( 'middle_
name' );
96:         $member_details ['last_name'] = $this->input->post ( 'last_name
' );
97:         $member_details ['cellphone_number'] = $this->input->post ( 'ce
llphone_number' );
98:         $member_details ['email_address'] = $this->input->post ( 'email
_address' );
99:         $member_details ['home_region'] = $this->input->post ( 'home_re
gion' );
100:        $member_details ['membership_type'] = $this->input->post ( 'mem
bership' );
101:        $member_details ['payment_type'] = $this->input->post ( 'paymen
t_mode' );
102:        $member_details ['is_notifiable'] = ( $this->input->post ( 'is_n
otifiable' ) === 'notifiable' ? '1' : '0' );
103:        $this->load->model ( 'Members' );
104:        $this->Members->save_member ( $pin, $member_details );
105:        $this->lists ();
106:    }
107: }
108: public function update($pin = NULL) {
109:     authenticate_log_in ();
110:     $data = array ();
111:     $this->load->model ( 'Members' );
112:     $queried_member_details = $this->Members->get_member_details ( $pin );
```

member.php

```
113:         $data ['pin'] = $queried_member_details ['pin'];
114:         $data ['first_name'] = $queried_member_details ['first_name'];
115:         $data ['middle_name'] = $queried_member_details ['middle_name'];
116:         $data ['last_name'] = $queried_member_details ['last_name'];
117:         $data ['cellphone_number'] = $queried_member_details ['cellphone_number'];
118:
119:         $data ['email_address'] = $queried_member_details ['email_address'];
120:         $data ['selected_home_region'] = $queried_member_details ['home_region'];
121:
122:         $data ['selected_membership'] = $queried_member_details ['membership_type'];
123:         $data ['selected_payment_mode'] = $queried_member_details ['payment_type'];
124:
125:         $data ['is_notifiable'] = $queried_member_details ['is_notifiable'];
126:         $this->load->model ( 'Regions' );
127:         $queried_region_list = $this->Regions->get_all_regions ();
128:         $data ['home_region'] = array ();
129:         foreach ( $queried_region_list as $region_list ) {
130:             $data ['home_region'] = $data ['home_region'] + array (
131:                 $region_list ['id'] => $region_list ['name']
132:             );
133:         }
134:         $this->load->model ( 'Membership_types' );
135:         $queried_membership_list = $this->Membership_types->get_all_membership_types ();
136:         $data ['membership'] = array ();
137:         foreach ( $queried_membership_list as $membership_list ) {
138:             $data ['membership'] = $data ['membership'] + array (
139:                 $membership_list ['id'] => $membership_list ['name']
140:             );
141:         }
142:         $this->load->model ( 'Payment_types' );
143:         $queried_payment_list = $this->Payment_types->get_all_payment_types ();
144:         $data ['payment_mode'] = array ();
145:         foreach ( $queried_payment_list as $payment_list ) {
146:             $data ['payment_mode'] = $data ['payment_mode'] + array (
147:                 $payment_list ['id'] => $payment_list ['name']
148:             );
149:         }
150:         $this->load->view ( 'member/update', $data );
151:     }
152:     public function process_update() {
153:         authenticate_log_in ();
154:
155:         $pin = $this->input->post ( 'pin' );
156:
157:         $config = array (
158:             array (
159:                 'field' => 'first_name',
160:                 'label' => 'First Name',
161:                 'rules' => 'is_unique[MEMBERS.PIN]|required|trim|xss_clean'
162:             ),
163:             array (
164:                 'field' => 'middle_name',
165:                 'label' => 'Middle Name',
166:                 'rules' => 'required|trim|xss_clean'
167:             ),
168:             array (
169:                 'field' => 'last_name',
170:                 'label' => 'Last Name',
171:                 'rules' => 'required|trim|xss_clean'
```

member.php

```
169:         ),
170:         array (
171:             'field' => 'cellphone_number',
172:             'label' => 'Cellphone Number',
173:             'rules' => 'callback_valid_cellphone_nu
mber|required|trim|xss_clean'
174:         ),
175:         array (
176:             'field' => 'email_address',
177:             'label' => 'Email Address',
178:             'rules' => 'valid_email|required|trim|x
ss_clean'
179:         ),
180:         array (
181:             'field' => 'home_region',
182:             'label' => 'Home Region',
183:             'rules' => 'trim|xss_clean'
184:         ),
185:         array (
186:             'field' => 'membership_type',
187:             'label' => 'Membership',
188:             'rules' => 'trim|xss_clean'
189:         ),
190:         array (
191:             'field' => 'payment_type',
192:             'label' => 'Payment mode',
193:             'rules' => 'trim|xss_clean'
194:         ),
195:         array (
196:             'field' => 'is_notifiable',
197:             'label' => 'Notifiable',
198:             'rules' => 'trim|xss_clean'
199:         )
200:     );
201:     $this->form_validation->set_rules ( $config );
202:     if ( $this->form_validation->run ( ) == FALSE ) {
203:         $this->update ( $pin );
204:     } else {
205:         $member_details = array ( );
206:         $member_details [ 'first_name' ] = $this->input->post ( 'first_na
me' );
207:         $member_details [ 'middle_name' ] = $this->input->post ( 'middle_
name' );
208:         $member_details [ 'last_name' ] = $this->input->post ( 'last_name
' );
209:         $member_details [ 'cellphone_number' ] = $this->input->post ( 'ce
llphone_number' );
210:         $member_details [ 'email_address' ] = $this->input->post ( 'email
_address' );
211:         $member_details [ 'home_region' ] = $this->input->post ( 'home_re
gion' );
212:         $member_details [ 'membership_type' ] = $this->input->post ( 'mem
bership' );
213:         $member_details [ 'payment_type' ] = $this->input->post ( 'paymen
t_mode' );
214:         $member_details [ 'is_notifiable' ] = ( $this->input->post ( 'is_n
otifiable' ) === 'notifiable' ? '1' : '0' );
215:         $this->load->model ( 'Members' );
216:         $this->Members->update_member ( $pin, $member_details );
217:         $this->lists ( );
218:     }
219: }
220: public function lists() {
```

member.php

```
221:         authenticate_log_in ();
222:         $this->load->model ( 'Members' );
223:         $queried_member_list = $this->Members->get_all_members ();
224:         $data = array ();
225:         $data ['member_list'] = array ();
226:         if (issetandnotempty ( $queried_member_list )) {
227:             foreach ( $queried_member_list as $member_list ) {
228:                 $queryresult = array ();
229:                 $queryresult ['pin'] = $member_list ['pin'];
230:                 $queryresult ['name'] = $member_list ['first_name'] . '
' . $member_list ['last_name'];
231:                 $this->load->model ( 'Regions' );
232:                 $queried_region_details = $this->Regions->get_region_de
tails ( $member_list ['home_region'] );
233:                 $queryresult ['home_region'] = $queried_region_details
['name'];
234:                 $this->load->model ( 'Membership_types' );
235:                 $queried_membership_details = $this->Membership_types->
get_membership_type_details ( $member_list ['membership_type'] );
236:                 $queryresult ['membership_type'] = $queried_membership_
details ['name'];
237:                 $this->load->model ( 'Payment_types' );
238:                 $queried_payment_mode_details = $this->Payment_types->g
et_payment_type_details ( $member_list ['payment_type'] );
239:                 $queryresult ['payment_type'] = $queried_payment_mode_d
etails ['name'];
240:                 $queryresult ['is_notifiable'] = ($member_list ['is_not
ifiable'] == '1' ? 'notifiable' : 'nonnotifiable');
241:                 array_push ( $data ['member_list'], $queryresult );
242:             }
243:         }
244:         $this->load->view ( 'member/list', $data );
245:     }
246:     public function notify($pin = NULL) {
247:         authenticate_log_in ();
248:         $this->load->model ( 'Members' );
249:         $this->Members->make_notifiable_member ( $pin );
250:         $this->lists ();
251:     }
252:     public function unnotify($pin = NULL) {
253:         authenticate_log_in ();
254:         $this->load->model ( 'Members' );
255:         $this->Members->make_unnotifiable_member ( $pin );
256:         $this->lists ();
257:     }
258:     public function upload() {
259:         authenticate_log_in ();
260:         $this->load->view ( 'member/upload' );
261:     }
262:     public function process_upload() {
263:         authenticate_log_in ();
264:
265:         $this->form_validation->set_rules ( '', '', 'trim|xss_clean' );
266:         if ( $this->form_validation->run () == FALSE ) {
267:             } else {
268:             }
269:
270:         $config = array ();
271:         $config ['upload_path'] = './uploads/';
272:         $config ['allowed_types'] = 'csv';
273:         $config ['file_name'] = 'mhc_members_archive_' . date ( 'Y-m-d_H-i-s.'
) . $config ['allowed_types'];
274:         $this->load->library ( 'upload', $config );
```

member.php

```
275:         if (! $this->upload->do_upload ()) {
276:             redirect ( 'member/upload' );
277:         } else {
278:             $data = array (
279:                 'upload_data' => $this->upload->data ()
280:             );
281:         }
282:         $handle = @fopen ( "./uploads/" . $config ['file_name'], "r" );
283:         if ($handle) {
284:             while ( ($buffer = fgets ( $handle, 4096 )) !== false ) {
285:                 $row = explode ( ",", $buffer );
286:                 $member_details = array ();
287:
288:                 $pin = $row [0];
289:                 $member_details ['first_name'] = $row [1];
290:                 $member_details ['middle_name'] = $row [2];
291:                 $member_details ['last_name'] = $row [3];
292:                 $member_details ['home_region'] = $row [4];
293:                 $member_details ['email_address'] = $row [5];
294:                 $member_details ['cellphone_number'] = $row [6];
295:                 $member_details ['is_notifiable'] = $row [7];
296:                 $member_details ['membership_type'] = $row [8];
297:                 $member_details ['payment_type'] = $row [9];
298:                 $member_details ['premium'] = $row [10];
299:                 $member_details ['has_missed_payment'] = $row [11];
300:
301:                 $this->load->model ( 'Members' );
302:                 if ($this->Members->does_member_exists ( $pin )) {
303:                     $this->Members->update_member ( $pin, $member_d
etails );
304:                 } else {
305:                     $this->Members->save_member ( $pin, $member_det
ails );
306:                 }
307:             }
308:             if (! feof ( $handle )) {
309:                 echo "Error: unexpected fgets() fail.\n";
310:             }
311:             fclose ( $handle );
312:         }
313:         $this->lists ();
314:     }
315:     function valid_cellphone_number($str) {
316:         return valid_cellphone_number ( $str );
317:     }
318: }
```

index.html

```
1: <html>
2: <head>
3: <title>403 Forbidden</title>
4: </head>
5: <body>
6:
7:     <p>Directory access is forbidden.</p>
8:
9: </body>
10: </html>
```

members.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Members extends CI_Model {
5:     function __construct() {
6:         parent::__construct ();
7:         $this->load->database ();
8:     }
9:     function count_members() {
10:         $this->db->select ( 'PIN' );
11:         $this->db->from ( 'MEMBERS' );
12:         $query = $this->db->get ();
13:         return $query->num_rows ();
14:     }
15:     function members_exist() {
16:         $this->db->select ( 'PIN' );
17:         $this->db->from ( 'MEMBERS' );
18:         $query = $this->db->get ();
19:         if ($query->num_rows () > 0) {
20:             return TRUE;
21:         }
22:         return FALSE;
23:     }
24:     function get_all_members() {
25:         $this->db->select ( 'PIN, FIRST_NAME, MIDDLE_NAME, LAST_NAME, EMAIL_ADD
RESS, CELLPHONE_NUMBER, PAYMENT_AMOUNT, HOME_REGION, MEMBERSHIP_TYPE, PAYMENT_TYPE, IS_NOTIFIA
BLE' );
26:         $this->db->from ( 'MEMBERS' );
27:         $query = $this->db->get ();
28:         if ($query->num_rows () > 0) {
29:             $queryresults = array ();
30:             foreach ( $query->result_array () as $row ) {
31:                 $queryresult = array ();
32:                 $queryresult ['pin'] = $row ['PIN'];
33:                 $queryresult ['first_name'] = $row ['FIRST_NAME'];
34:                 $queryresult ['middle_name'] = $row ['MIDDLE_NAME'];
35:                 $queryresult ['last_name'] = $row ['LAST_NAME'];
36:                 $queryresult ['email_address'] = $row ['EMAIL_ADDRESS']
;
37:                 $queryresult ['cellphone_number'] = $row ['CELLPHONE_NU
MBER'];
38:                 $queryresult ['payment_amount'] = $row ['PAYMENT_AMOUNT
'];
39:                 $queryresult ['home_region'] = $row ['HOME_REGION'];
40:                 $queryresult ['membership_type'] = $row ['MEMBERSHIP_TY
PE'];
41:                 $queryresult ['payment_type'] = $row ['PAYMENT_TYPE'];
42:                 $queryresult ['is_notifiable'] = $row ['IS_NOTIFIABLE']
;
43:                 array_push ( $queryresults, $queryresult );
44:             }
45:             return $queryresults;
46:         }
47:         return NULL;
48:     }
49:     function get_all_filtered_members($region_id, $membership, $payment_mode, $noti
fiable, $missed_payment) {
50:         $this->load->helper ( 'variable' );
51:
52:         $this->db->select ( 'PIN, FIRST_NAME, MIDDLE_NAME, LAST_NAME, EMAIL_ADD
RESS, CELLPHONE_NUMBER, PAYMENT_AMOUNT, HOME_REGION, MEMBERSHIP_TYPE, PAYMENT_TYPE, IS_NOTIFIA
BLE' );
53:         $this->db->from ( 'MEMBERS' );
```


members.php

```
54:         if (issetandnotempty ( $region_id )) {
55:             $this->db->where_in ( 'HOME_REGION', $region_id );
56:         }
57:         if (issetandnotempty ( $membership )) {
58:             $this->db->where_in ( 'MEMBERSHIP_TYPE', $membership );
59:         }
60:         if (issetandnotempty ( $payment_mode )) {
61:             $this->db->where_in ( 'PAYMENT_TYPE', $payment_mode );
62:         }
63:         if (issetandnotempty ( $notifiable )) {
64:             $this->db->where_in ( 'IS_NOTIFIABLE', $notifiable );
65:         }
66:         if (issetandnotempty ( $missed_payment )) {
67:             $this->db->where_in ( 'MISSED_PAYMENT', $missed_payment );
68:         }
69:         $query = $this->db->get ();
70:         if ($query->num_rows () > 0) {
71:             $queryresults = array ();
72:             foreach ( $query->result_array () as $row ) {
73:                 $queryresult = array ();
74:                 $queryresult ['pin'] = $row ['PIN'];
75:                 $queryresult ['first_name'] = $row ['FIRST_NAME'];
76:                 $queryresult ['middle_name'] = $row ['MIDDLE_NAME'];
77:                 $queryresult ['last_name'] = $row ['LAST_NAME'];
78:                 $queryresult ['email_address'] = $row ['EMAIL_ADDRESS'];
;
79:                 $queryresult ['cellphone_number'] = $row ['CELLPHONE_NUM
MEMBER'];
80:                 $queryresult ['payment_amount'] = $row ['PAYMENT_AMOUNT
'];
81:                 $queryresult ['home_region'] = $row ['HOME_REGION'];
82:                 $queryresult ['membership_type'] = $row ['MEMBERSHIP_TY
PE'];
83:                 $queryresult ['payment_type'] = $row ['PAYMENT_TYPE'];
84:                 $queryresult ['is_notifiable'] = $row ['IS_NOTIFIABLE'];
;
85:                 array_push ( $queryresults, $queryresult );
86:             }
87:             return $queryresults;
88:         }
89:         return NULL;
90:     }
91:     function get_member_details($pin) {
92:         $this->load->helper ( 'variable' );
93:
94:         $this->db->select ( 'PIN, FIRST_NAME, MIDDLE_NAME, LAST_NAME, EMAIL_ADD
RESS, CELLPHONE_NUMBER, PAYMENT_AMOUNT, HOME_REGION, MEMBERSHIP_TYPE, PAYMENT_TYPE, IS_NOTIFIA
BLE' );
95:         $this->db->from ( 'MEMBERS' );
96:         if (issetandnotempty ( $pin )) {
97:             $this->db->where ( 'PIN', $pin );
98:         }
99:         $query = $this->db->get ();
100:        if ($query->num_rows () > 0) {
101:            $queryresult = array ();
102:            $row = $query->row_array ();
103:            $queryresult ['pin'] = $row ['PIN'];
104:            $queryresult ['first_name'] = $row ['FIRST_NAME'];
105:            $queryresult ['middle_name'] = $row ['MIDDLE_NAME'];
106:            $queryresult ['last_name'] = $row ['LAST_NAME'];
107:            $queryresult ['email_address'] = $row ['EMAIL_ADDRESS'];
108:            $queryresult ['cellphone_number'] = $row ['CELLPHONE_NUMBER'];
109:            $queryresult ['payment_amount'] = $row ['PAYMENT_AMOUNT'];
```

members.php

```
110:         $queryresult ['home_region'] = $row ['HOME_REGION'];
111:         $queryresult ['membership_type'] = $row ['MEMBERSHIP_TYPE'];
112:         $queryresult ['payment_type'] = $row ['PAYMENT_TYPE'];
113:         $queryresult ['is_notifiable'] = $row ['IS_NOTIFIABLE'];
114:         return $queryresult;
115:     }
116:     return NULL;
117: }
118: function get_member_details_with($first_name, $middle_name, $last_name) {
119:     $this->load->helper ( 'variable' );
120:
121:     $this->db->select ( 'PIN, FIRST_NAME' );
122:     $this->db->from ( 'MEMBERS' );
123:     if (! issetandnotempty ( $first_name )) {
124:         return NULL;
125:     }
126:     if (! issetandnotempty ( $middle_name )) {
127:         return NULL;
128:     }
129:     if (! issetandnotempty ( $last_name )) {
130:         return NULL;
131:     }
132:     $this->db->where ( 'FIRST_NAME', $first_name );
133:     $this->db->where ( 'MIDDLE_NAME', $middle_name );
134:     $this->db->where ( 'LAST_NAME', $last_name );
135:     $this->db->limit ( 1 );
136:     $query = $this->db->get ();
137:     if ($query->num_rows () > 0) {
138:         $queryresult = array ();
139:         $row = $query->row_array ();
140:         $queryresult ['pin'] = $row ['PIN'];
141:         $queryresult ['first_name'] = $row ['FIRST_NAME'];
142:         return $queryresult;
143:     }
144:     return NULL;
145: }
146: function get_user_with($pin, $last_name) {
147:     $this->load->helper ( 'variable' );
148:
149:     $this->db->select ( 'PIN, FIRST_NAME' );
150:     $this->db->from ( 'MEMBERS' );
151:     if (! issetandnotempty ( $pin )) {
152:         return NULL;
153:     }
154:     if (! issetandnotempty ( $last_name )) {
155:         return NULL;
156:     }
157:     $this->db->where ( 'PIN', $pin );
158:     $this->db->where ( 'LAST_NAME', $last_name );
159:     $this->db->limit ( 1 );
160:     $query = $this->db->get ();
161:     if ($query->num_rows () > 0) {
162:         $queryresult = array ();
163:         $row = $query->row_array ();
164:         $queryresult ['pin'] = $row ['PIN'];
165:         $queryresult ['first_name'] = $row ['FIRST_NAME'];
166:         return $queryresult;
167:     }
168:     return NULL;
169: }
170: function does_member_exists($pin) {
171:     $this->load->helper ( 'variable' );
172:
```

members.php

```
173:         $this->db->select ( 'PIN, FIRST_NAME' );
174:         $this->db->from ( 'MEMBERS' );
175:         if (issetandnotempty ( $pin )) {
176:             $this->db->where ( 'PIN', $pin );
177:             $this->db->limit ( 1 );
178:             $query = $this->db->get ();
179:             if ($query->num_rows () > 0) {
180:                 return TRUE;
181:             }
182:         }
183:         return FALSE;
184:     }
185:     function update_member($pin, $member_details) {
186:         $this->load->helper ( 'variable' );
187:
188:         if (issetandnotempty ( $pin )) {
189:             $this->db->set ( 'PIN', $pin );
190:         }
191:         if (issetandnotempty ( 'first_name', $member_details )) {
192:             $this->db->set ( 'FIRST_NAME', $member_details ['first_name'] )
;
193:         }
194:         if (issetandnotempty ( 'middle_name', $member_details )) {
195:             $this->db->set ( 'MIDDLE_NAME', $member_details ['middle_name']
);
196:         }
197:         if (issetandnotempty ( 'last_name', $member_details )) {
198:             $this->db->set ( 'LAST_NAME', $member_details ['last_name'] );
199:         }
200:         if (issetandnotempty ( 'email_address', $member_details )) {
201:             $this->db->set ( 'EMAIL_ADDRESS', $member_details ['email_adre
ss'] );
202:         }
203:         if (issetandnotempty ( 'cellphone_number', $member_details )) {
204:             $this->db->set ( 'CELLPHONE_NUMBER', $member_details ['cellphon
e_number'] );
205:         }
206:         if (issetandnotempty ( 'payment_amount', $member_details )) {
207:             $this->db->set ( 'PAYMENT_AMOUNT', $member_details ['payment_am
ount'] );
208:         }
209:         if (issetandnotempty ( 'home_region', $member_details )) {
210:             $this->db->set ( 'HOME_REGION', $member_details ['home_region']
);
211:         }
212:         if (issetandnotempty ( 'membership_type', $member_details )) {
213:             $this->db->set ( 'MEMBERSHIP_TYPE', $member_details ['membershi
p_type'] );
214:         }
215:         if (issetandnotempty ( 'payment_type', $member_details )) {
216:             $this->db->set ( 'PAYMENT_TYPE', $member_details ['payment_type
'] );
217:         }
218:         if (issetandnotempty ( 'is_notifiable', $member_details )) {
219:             $this->db->set ( 'IS_NOTIFIABLE', $member_details ['is_notifiab
le'] );
220:         }
221:         if (issetandnotempty ( 'premium', $member_details )) {
222:             $this->db->set ( 'PAYMENT_AMOUNT', $member_details ['premium']
);
223:         }
224:         if (issetandnotempty ( 'has_missed_payment', $member_details )) {
225:             $this->db->set ( 'MISSED_PAYMENT', $member_details ['has_missed
```

members.php

```
    _payment'] );
    226:         }
    227:         $this->db->where ( 'PIN', $pin );
    228:         $this->db->update ( 'MEMBERS' );
    229:     }
    230:     function save_member($pin, $member_details) {
    231:         $this->load->helper ( 'variable' );
    232:
    233:         if (issetandnotempty ( $pin )) {
    234:             $this->db->set ( 'PIN', $pin );
    235:         }
    236:         if (issetandnotempty ( $member_details ['first_name'] )) {
    237:             $this->db->set ( 'FIRST_NAME', $member_details ['first_name'] )
;
    238:         }
    239:         if (issetandnotempty ( $member_details ['middle_name'] )) {
    240:             $this->db->set ( 'MIDDLE_NAME', $member_details ['middle_name']
);
    241:         }
    242:         if (issetandnotempty ( $member_details ['last_name'] )) {
    243:             $this->db->set ( 'LAST_NAME', $member_details ['last_name'] );
    244:         }
    245:         if (issetandnotempty ( $member_details ['email_address'] )) {
    246:             $this->db->set ( 'EMAIL_ADDRESS', $member_details ['email_addre
ss'] );
    247:         }
    248:         if (issetandnotempty ( $member_details ['cellphone_number'] )) {
    249:             $this->db->set ( 'CELLPHONE_NUMBER', $member_details ['cellphon
e_number'] );
    250:         }
    251:         if (issetandnotempty ( 'payment_amount', $member_details )) {
    252:             $this->db->set ( 'PAYMENT_AMOUNT', $member_details ['payment_am
ount'] );
    253:         }
    254:         if (issetandnotempty ( $member_details ['home_region'] )) {
    255:             $this->db->set ( 'HOME_REGION', $member_details ['home_region']
);
    256:         }
    257:         if (issetandnotempty ( $member_details ['membership_type'] )) {
    258:             $this->db->set ( 'MEMBERSHIP_TYPE', $member_details ['membershi
p_type'] );
    259:         }
    260:         if (issetandnotempty ( $member_details ['payment_type'] )) {
    261:             $this->db->set ( 'PAYMENT_TYPE', $member_details ['payment_type
'] );
    262:         }
    263:         if (issetandnotempty ( $member_details ['is_notifiable'] )) {
    264:             $this->db->set ( 'IS_NOTIFIABLE', $member_details ['is_notifiab
le'] );
    265:         }
    266:         if (issetandnotempty ( 'premium', $member_details )) {
    267:             $this->db->set ( 'PAYMENT_AMOUNT', $member_details ['premium']
);
    268:         }
    269:         if (issetandnotempty ( 'has_missed_payment', $member_details )) {
    270:             $this->db->set ( 'MISSED_PAYMENT', $member_details ['has_missed
_payment'] );
    271:         }
    272:         $this->db->insert ( 'MEMBERS' );
    273:     }
    274:     function make_notifiable_member($pin) {
    275:         $data = array (
    276:             'IS_NOTIFIABLE' => '1'
```

members.php

```
277:         );
278:         $this->db->where ( 'PIN', $pin );
279:         $this->db->update ( 'MEMBERS', $data );
280:     }
281:     function make_unnotifiable_member($pin) {
282:         $data = array (
283:             'IS_NOTIFIABLE' => '0'
284:         );
285:         $this->db->where ( 'PIN', $pin );
286:         $this->db->update ( 'MEMBERS', $data );
287:     }
288: }
289: ?>
```

index.html

```
1: <html>
2: <head>
3: <title>403 Forbidden</title>
4: </head>
5: <body>
6:
7:     <p>Directory access is forbidden.</p>
8:
9: </body>
10: </html>
```

partner_companies_services.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Partner_companies_services extends CI_Model {
5:     function __construct() {
6:         parent::__construct ();
7:         $this->load->database ();
8:     }
9:     function count_active_service($company_id) {
10:         $this->db->select ( 'ID' );
11:         $this->db->from ( 'PARTNER_COMPANIES_SERVICES' );
12:         $this->db->where ( 'IS_AVAILABLE', '1' );
13:         $this->db->where ( 'PARTNER_COMPANY_ID', $company_id );
14:         $query = $this->db->get ();
15:         return $query->num_rows ();
16:     }
17:     function services_exists($company_id) {
18:         $this->db->select ( 'ID' );
19:         $this->db->from ( 'PARTNER_COMPANIES_SERVICES' );
20:         $this->db->where ( 'IS_AVAILABLE', '1' );
21:         if (issetandnotempty ( $company_id )) {
22:             $this->db->where_in ( 'PARTNER_COMPANY_ID', $company_id );
23:             $query = $this->db->get ();
24:             if ( $query->num_rows () > 0 ) {
25:                 return TRUE;
26:             }
27:         }
28:         return FALSE;
29:     }
30:     function get_all_services() {
31:         $this->db->select ( 'ID, NAME, DESCRIPTION, PARTNER_COMPANY_ID, IS_AVAI
32: LABEL' );
33:         $this->db->from ( 'PARTNER_COMPANIES_SERVICES' );
34:         $query = $this->db->get ();
35:         if ( $query->num_rows () > 0 ) {
36:             $queryresults = array ();
37:             foreach ( $query->result_array () as $row ) {
38:                 $queryresult = array ();
39:                 $queryresult ['id'] = $row ['ID'];
40:                 $queryresult ['name'] = $row ['NAME'];
41:                 $queryresult ['description'] = $row ['DESCRIPTION'];
42:                 $queryresult ['partner_company_id'] = $row ['PARTNER_CO
43: MPANY_ID'];
44:                 $queryresult ['is_available'] = $row ['IS_AVAILABLE'];
45:                 array_push ( $queryresults, $queryresult );
46:             }
47:             return $queryresults;
48:         }
49:         return NULL;
50:     }
51:     function get_all_active_services() {
52: LABEL' );
53:         $this->db->select ( 'ID, NAME, DESCRIPTION, PARTNER_COMPANY_ID, IS_AVAI
54:         $this->db->from ( 'PARTNER_COMPANIES_SERVICES' );
55:         $this->db->where ( 'IS_AVAILABLE', '1' );
56:         $query = $this->db->get ();
57:         if ( $query->num_rows () > 0 ) {
58:             $queryresults = array ();
59:             foreach ( $query->result_array () as $row ) {
60:                 $queryresult = array ();
61:                 $queryresult ['id'] = $row ['ID'];
62:                 $queryresult ['name'] = $row ['NAME'];
63:                 $queryresult ['description'] = $row ['DESCRIPTION'];
```

partner_companies_services.php

```
61:                                     $queryresult ['partner_company_id'] = $row ['PARTNER_CO
MPANY_ID'];
62:                                     $queryresult ['is_available'] = $row ['IS_AVAILABLE'];
63:                                     array_push ( $queryresults, $queryresult );
64:                                     }
65:                                     return $queryresults;
66:                                     }
67:                                     return NULL;
68:                                     }
69:                                     function get_all_active_services_of_company($company_id) {
70:                                     $this->db->select ( 'ID, NAME, DESCRIPTION, PARTNER_COMPANY_ID, IS_AVAI
71: LABEL' );
72:                                     $this->db->from ( 'PARTNER_COMPANIES_SERVICES' );
73:                                     $this->db->where ( 'IS_AVAILABLE', '1' );
74:                                     if (issetandnotempty ( $company_id )) {
75:                                     $this->db->where_in ( 'PARTNER_COMPANY_ID', $company_id );
76:                                     }
77:                                     $query = $this->db->get ();
78:                                     if ($query->num_rows () > 0) {
79:                                     $queryresults = array ();
80:                                     foreach ( $query->result_array () as $row ) {
81:                                     $queryresult = array ();
82:                                     $queryresult ['id'] = $row ['ID'];
83:                                     $queryresult ['name'] = $row ['NAME'];
84:                                     $queryresult ['description'] = $row ['DESCRIPTION'];
85:                                     $queryresult ['partner_company_id'] = $row ['PARTNER_CO
MPANY_ID'];
86:                                     $queryresult ['is_available'] = $row ['IS_AVAILABLE'];
87:                                     array_push ( $queryresults, $queryresult );
88:                                     }
89:                                     return $queryresults;
90:                                     }
91:                                     return NULL;
92:                                     }
93:                                     function get_all_services_of_company($company_id) {
94: LABEL' );
95:                                     $this->db->select ( 'ID, NAME, DESCRIPTION, PARTNER_COMPANY_ID, IS_AVAI
96:                                     $this->db->from ( 'PARTNER_COMPANIES_SERVICES' );
97:                                     if (issetandnotempty ( $company_id )) {
98:                                     $this->db->where_in ( 'PARTNER_COMPANY_ID', $company_id );
99:                                     }
100:                                    $query = $this->db->get ();
101:                                    if ($query->num_rows () > 0) {
102:                                    $queryresults = array ();
103:                                    foreach ( $query->result_array () as $row ) {
104:                                    $queryresult = array ();
105:                                    $queryresult ['id'] = $row ['ID'];
106:                                    $queryresult ['name'] = $row ['NAME'];
107:                                    $queryresult ['description'] = $row ['DESCRIPTION'];
108:                                    $queryresult ['partner_company_id'] = $row ['PARTNER_CO
MPANY_ID'];
109:                                    $queryresult ['is_available'] = $row ['IS_AVAILABLE'];
110:                                    array_push ( $queryresults, $queryresult );
111:                                    }
112:                                    return $queryresults;
113:                                    }
114:                                    return NULL;
115:                                    }
116:                                    function get_service_details($service_id) {
117: LABEL' );
118:                                    $this->db->select ( 'ID, NAME, DESCRIPTION, PARTNER_COMPANY_ID, IS_AVAI
119:                                    $this->db->from ( 'PARTNER_COMPANIES_SERVICES' );
120:                                    if (issetandnotempty ( $service_id )) {
```


partner_companies_services.php

```
118:             $this->db->where ( 'ID', $service_id );
119:         }
120:         $this->db->limit ( 1 );
121:         $query = $this->db->get ();
122:         if ($query->num_rows () > 0) {
123:             $queryresult = array ();
124:             $row = $query->row_array ();
125:             $queryresult ['id'] = $row ['ID'];
126:             $queryresult ['name'] = $row ['NAME'];
127:             $queryresult ['description'] = $row ['DESCRIPTION'];
128:             $queryresult ['partner_company_id'] = $row ['PARTNER_COMPANY_ID
'];
129:             $queryresult ['is_available'] = $row ['IS_AVAILABLE'];
130:             return $queryresult;
131:         }
132:         return NULL;
133:     }
134:     function update_service($service_id, $service_details) {
135:         if (issetandnotempty ( 'name', $service_details )) {
136:             $this->db->set ( 'NAME', $service_details ['name'] );
137:         }
138:         if (issetandnotempty ( 'description', $service_details )) {
139:             $this->db->set ( 'DESCRIPTION', $service_details ['description'
]);
140:         }
141:         if (issetandnotempty ( 'partner_company_id', $service_details )) {
142:             $this->db->set ( 'PARTNER_COMPANY_ID', $service_details ['partn
er_company_id'] );
143:         }
144:         if (issetandnotempty ( 'is_available', $service_details )) {
145:             $this->db->set ( 'IS_AVAILABLE', $service_details ['is_availabl
e'] );
146:         }
147:
148:         $this->db->where ( 'ID', $service_id );
149:         $this->db->update ( 'PARTNER_COMPANIES_SERVICES' );
150:     }
151:     function save_service($service_details) {
152:         if (issetandnotempty ( 'name', $service_details )) {
153:             $this->db->set ( 'NAME', $service_details ['name'] );
154:         }
155:         if (issetandnotempty ( 'description', $service_details )) {
156:             $this->db->set ( 'DESCRIPTION', $service_details ['description'
]);
157:         }
158:         if (issetandnotempty ( 'partner_company_id', $service_details )) {
159:             $this->db->set ( 'PARTNER_COMPANY_ID', $service_details ['partn
er_company_id'] );
160:         }
161:         if (issetandnotempty ( 'is_available', $service_details )) {
162:             $this->db->set ( 'IS_AVAILABLE', $service_details ['is_availabl
e'] );
163:         }
164:         $this->db->insert ( 'PARTNER_COMPANIES_SERVICES' );
165:     }
166:     function make_available_service($service_id) {
167:         $data = array (
168:             'IS_AVAILABLE' => '1'
169:         );
170:         $this->db->where ( 'ID', $service_id );
171:         $this->db->update ( 'PARTNER_COMPANIES_SERVICES', $data );
172:     }
173:     function make_unavailable_service($service_id) {
```

partner_companies_services.php

```
174:             $data = array (
175:                 'IS_AVAILABLE' => '0'
176:             );
177:             $this->db->where ( 'ID', $service_id );
178:             $this->db->update ( 'PARTNER_COMPANIES_SERVICES', $data );
179:         }
180:     }
181: ?>
```

message_formats.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Message_formats extends CI_Model {
5:     function __construct() {
6:         parent::__construct ();
7:         $this->load->database ();
8:     }
9:     function get_all_message_formats() {
10:         $this->db->select ( 'ID, NAME, FORMAT, MESSAGE_TYPE, COMMUNICATION_TYPE
' );
11:         $this->db->from ( 'MESSAGE_FORMATS' );
12:         $query = $this->db->get ();
13:         if ($query->num_rows () > 0) {
14:             $queryresults = array ();
15:             foreach ( $query->result_array () as $row ) {
16:                 $queryresult = array ();
17:                 $queryresult ['id'] = $row ['ID'];
18:                 $queryresult ['name'] = $row ['NAME'];
19:                 $queryresult ['format'] = $row ['FORMAT'];
20:                 $queryresult ['message_type'] = $row ['MESSAGE_TYPE'];
21:                 $queryresult ['communication_type'] = $row ['COMMUNICAT
ION_TYPE'];
22:                 array_push ( $queryresults, $queryresult );
23:             }
24:             return $queryresults;
25:         }
26:         return NULL;
27:     }
28:     function get_message_format_details($message_format_id, $message_type, $communi
cation_type) {
29:         $this->load->helper ( 'variable' );
30:
31:         $this->db->select ( 'FORMAT' );
32:         $this->db->from ( 'MESSAGE_FORMATS' );
33:         if (issetandnotempty ( $message_format_id )) {
34:             $this->db->where ( 'ID', $message_format_id );
35:         }
36:         if (issetandnotempty ( $message_type )) {
37:             $this->db->where ( 'MESSAGE_TYPE', $message_type );
38:         }
39:         if (issetandnotempty ( $communication_type )) {
40:             $this->db->where ( 'COMMUNICATION_TYPE', $communication_type );
41:         }
42:         $query = $this->db->get ();
43:
44:         if ($query->num_rows () > 0) {
45:             $row = $query->row_array ();
46:             return $row ['FORMAT'];
47:         }
48:         return NULL;
49:     }
50:     function update_format($message_format_id, $message_format_details) {
51:         $data = array (
52:             'NAME' => $message_format_details ['name'],
53:             'FORMAT' => $message_format_details ['format'],
54:             'MESSAGE_TYPE' => $message_format_details ['message_typ
e'],
55:             'COMMUNICATION_TYPE' => $message_format_details ['commu
nication_type']
56:         );
57:         $this->db->where ( 'ID', $message_format_id );
58:         $this->db->update ( 'MESSAGE_FORMATS', $data );
```

message_formats.php

```
59:         }  
60:     }  
61: ?>
```

sms_history.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Sms_history extends CI_Model {
5:     function __construct() {
6:         parent::__construct ();
7:         $this->load->database ();
8:     }
9:     function get_all_messages() {
10:        $this->load->helper ( 'variable' );
11:
12:        $this->db->select ( 'ID, MESSAGE, SENDER_NUMBER, RECEIVER_NUMBER, MEMBE
R_PIN, USER_ID, PARTNER_COMPANY_ID, COMMUNICATION_TYPE, MESSAGE_TYPE, DATE' );
13:        $this->db->from ( 'SMS_HISTORY' );
14:        $this->db->order_by ( 'DATE', 'desc' );
15:        $query = $this->db->get ();
16:        if ($query->num_rows () > 0) {
17:            $queryresults = array ();
18:            foreach ( $query->result_array () as $row ) {
19:                $queryresult = array ();
20:                $queryresult ['id'] = $row ['ID'];
21:                $queryresult ['message'] = $row ['MESSAGE'];
22:                $queryresult ['sender_number'] = $row ['SENDER_NUMBER']
;
23:                $queryresult ['receiver_number'] = $row ['RECEIVER_NUMB
ER'];
24:                $queryresult ['member_pin'] = $row ['MEMBER_PIN'];
25:                $queryresult ['user_id'] = $row ['USER_ID'];
26:                $queryresult ['partner_company_id'] = $row ['PARTNER_CO
MPANY_ID'];
27:                $queryresult ['communication_type'] = $row ['COMMUNICAT
ION_TYPE'];
28:                $queryresult ['message_type'] = $row ['MESSAGE_TYPE'];
29:                $queryresult ['date'] = $row ['DATE'];
30:                array_push ( $queryresults, $queryresult );
31:            }
32:            return $queryresults;
33:        }
34:        return NULL;
35:    }
36:    function get_all_filtered_messages($member_pin, $personnel_id, $partner_company
_id, $communication_type, $message_type) {
37:        $this->load->helper ( 'variable' );
38:
39:        $this->db->select ( 'ID, MESSAGE, SENDER_NUMBER, RECEIVER_NUMBER, MEMBE
R_PIN, USER_ID, PARTNER_COMPANY_ID, COMMUNICATION_TYPE, MESSAGE_TYPE, DATE' );
40:        $this->db->from ( 'SMS_HISTORY' );
41:        if (issetandnotempty ( $member_pin )) {
42:            $this->db->where_in ( 'MEMBER_PIN', $member_pin );
43:        }
44:        if (issetandnotempty ( $personnel_id )) {
45:            $this->db->where_in ( 'USER_ID', $personnel_id );
46:        }
47:        if (issetandnotempty ( $partner_company_id )) {
48:            $this->db->where_in ( 'PARTNER_COMPANY_ID', $partner_company_id
);
49:        }
50:        if (issetandnotempty ( $communication_type )) {
51:            $this->db->where_in ( 'COMMUNICATION_TYPE', $communication_type
);
52:        }
53:        if (issetandnotempty ( $message_type )) {
54:            $this->db->where_in ( 'MESSAGE_TYPE', $message_type );
```

sms_history.php

```
55:     }
56:     $this->db->order_by ( 'DATE', 'desc' );
57:     $query = $this->db->get ();
58:     if ($query->num_rows () > 0) {
59:         $queryresults = array ();
60:         foreach ( $query->result_array () as $row ) {
61:             $queryresult = array ();
62:             $queryresult ['id'] = $row ['ID'];
63:             $queryresult ['message'] = $row ['MESSAGE'];
64:             $queryresult ['sender_number'] = $row ['SENDER_NUMBER']
;
65:             $queryresult ['receiver_number'] = $row ['RECEIVER_NUMB
ER'];
66:             $queryresult ['member_pin'] = $row ['MEMBER_PIN'];
67:             $queryresult ['user_id'] = $row ['USER_ID'];
68:             $queryresult ['partner_company_id'] = $row ['PARTNER_CO
MPANY_ID'];
69:             $queryresult ['communication_type'] = $row ['COMMUNICAT
ION_TYPE'];
70:             $queryresult ['message_type'] = $row ['MESSAGE_TYPE'];
71:             $queryresult ['date'] = $row ['DATE'];
72:             array_push ( $queryresults, $queryresult );
73:         }
74:         return $queryresults;
75:     }
76:     return NULL;
77: }
78: function get_latest_message($member_pin, $communication_type, $message_type) {
79:     $this->load->helper ( 'variable' );
80:     $this->db->select ( 'MESSAGE, DATE' );
81:     $this->db->from ( 'SMS_HISTORY' );
82:     if (! issetandnotempty ( $member_pin )) {
83:         return NULL;
84:     }
85:     if (! issetandnotempty ( $communication_type )) {
86:         return NULL;
87:     }
88:     if (! issetandnotempty ( $message_type )) {
89:         return NULL;
90:     }
91:     $this->db->where ( 'MEMBER_PIN', $member_pin );
92:     $this->db->where ( 'MESSAGE_TYPE', $message_type );
93:     $this->db->where ( 'COMMUNICATION_TYPE', $communication_type );
94:     $this->db->order_by ( 'DATE', 'desc' );
95:     $this->db->limit ( 1 );
96:     $query = $this->db->get ();
97:     if ($query->num_rows () > 0) {
98:         $row = $query->row_array ();
99:         return $row ['MESSAGE'];
100:     }
101:     return NULL;
102: }
103: function save_message($message_id, $message_details) {
104:     $this->load->helper ( 'variable' );
105:     if (issetandnotempty ( $message_id )) {
106:         $this->db->set ( 'ID', $message_id );
107:     }
108:     if (issetandnotempty ( 'message', $message_details )) {
109:         $this->db->set ( 'MESSAGE', $message_details ['message'] );
110:     }
111:     if (issetandnotempty ( 'sender_number', $message_details )) {
112:         $this->db->set ( 'SENDER_NUMBER', $message_details ['sender_num
ber'] );
```

sms_history.php

```
113:         }
114:         if (issetandnotempty ( 'receiver_number', $message_details )) {
115:             $this->db->set ( 'RECEIVER_NUMBER', $message_details ['receiver
_number'] );
116:         }
117:         if (issetandnotempty ( 'member_pin', $message_details )) {
118:             $this->db->set ( 'MEMBER_PIN', $message_details ['member_pin']
);
119:         }
120:         if (issetandnotempty ( 'user_id', $message_details )) {
121:             $this->db->set ( 'USER_ID', $message_details ['user_id'] );
122:         }
123:         if (issetandnotempty ( 'partner_company_id', $message_details )) {
124:             $this->db->set ( 'PARTNER_COMPANY_ID', $message_details ['partn
er_company_id'] );
125:         }
126:         if (issetandnotempty ( 'communication_type', $message_details )) {
127:             $this->db->set ( 'COMMUNICATION_TYPE', $message_details ['commu
nication_type'] );
128:         }
129:         if (issetandnotempty ( 'message_type', $message_details )) {
130:             $this->db->set ( 'MESSAGE_TYPE', $message_details ['message_typ
e'] );
131:         }
132:         $this->db->insert ( 'SMS_HISTORY' );
133:     }
134: }
135: ?>
```

payment_types.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Payment_types extends CI_Model {
5:     function __construct() {
6:         parent::__construct ();
7:     }
8:     function get_all_payment_types() {
9:         $this->db->select ( 'ID, NAME, DESCRIPTION' );
10:        $this->db->from ( 'PAYMENT_TYPES' );
11:        $query = $this->db->get ();
12:        if ($query->num_rows () > 0) {
13:            $queryresults = array ();
14:            foreach ( $query->result_array () as $row ) {
15:                $queryresult = array ();
16:                $queryresult ['id'] = $row ['ID'];
17:                $queryresult ['name'] = $row ['NAME'];
18:                $queryresult ['description'] = $row ['DESCRIPTION'];
19:                array_push ( $queryresults, $queryresult );
20:            }
21:            return $queryresults;
22:        }
23:        return NULL;
24:    }
25:    function get_payment_type_details($payment_type_id) {
26:        $this->load->helper ( 'variable' );
27:
28:        $this->db->select ( 'ID, NAME, DESCRIPTION' );
29:        $this->db->from ( 'PAYMENT_TYPES' );
30:        if (issetandnotempty ( $payment_type_id )) {
31:            $this->db->where ( 'ID', $payment_type_id );
32:        }
33:        $this->db->limit ( 1 );
34:        $query = $this->db->get ();
35:        if ($query->num_rows () > 0) {
36:            $queryresult = array ();
37:            $row = $query->row_array ();
38:            $queryresult ['id'] = $row ['ID'];
39:            $queryresult ['name'] = $row ['NAME'];
40:            $queryresult ['description'] = $row ['DESCRIPTION'];
41:            return $queryresult;
42:        }
43:        return NULL;
44:    }
45:    function update_payment_type($payment_type_id, $payment_type_details) {
46:        $data = array (
47:            'NAME' => $payment_type_details ['name'],
48:            'DESCRIPTION' => $payment_type_details ['description']
49:        );
50:        $this->db->where ( 'ID', $payment_type_id );
51:        $this->db->update ( 'PAYMENT_TYPES', $data );
52:    }
53: }
54: ?>
```


regions.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Regions extends CI_Model {
5:     function __construct() {
6:         parent::__construct ();
7:         $this->load->database ();
8:     }
9:     function get_all_regions() {
10:         $this->db->select ( 'ID, NAME, DESCRIPTION' );
11:         $this->db->from ( 'REGIONS' );
12:         $query = $this->db->get ();
13:         if ($query->num_rows () > 0) {
14:             $queryresults = array ();
15:             foreach ( $query->result_array () as $row ) {
16:                 $queryresult = array ();
17:                 $queryresult ['id'] = $row ['ID'];
18:                 $queryresult ['name'] = $row ['NAME'];
19:                 $queryresult ['description'] = $row ['DESCRIPTION'];
20:                 array_push ( $queryresults, $queryresult );
21:             }
22:             return $queryresults;
23:         }
24:         return NULL;
25:     }
26:     function get_region_details($region_id) {
27:         $this->load->helper ( 'variable' );
28:
29:         $this->db->select ( 'ID, NAME, DESCRIPTION' );
30:         $this->db->from ( 'REGIONS' );
31:         if (issetandnotempty ( $region_id )) {
32:             $this->db->where ( 'ID', $region_id );
33:         }
34:         $this->db->limit ( 1 );
35:         $query = $this->db->get ();
36:         if ($query->num_rows () > 0) {
37:             $queryresult = array ();
38:             $row = $query->row_array ();
39:             $queryresult ['id'] = $row ['ID'];
40:             $queryresult ['name'] = $row ['NAME'];
41:             $queryresult ['description'] = $row ['DESCRIPTION'];
42:             return $queryresult;
43:         }
44:         return NULL;
45:     }
46:     function update_region($region_id, $region_details) {
47:         $data = array (
48:             'NAME' => $region_details ['name'],
49:             'DESCRIPTION' => $region_details ['description']
50:         );
51:         $this->db->where ( 'ID', $region_id );
52:         $this->db->update ( 'REGIONS', $data );
53:     }
54: }
55: ?>
```

sms_configuration.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Sms_configuration extends CI_Model {
5:     function __construct() {
6:         parent::__construct ();
7:         $this->load->database ();
8:     }
9:     function get_gsm_configurations() {
10:        $this->load->helper ( 'variable' );
11:
12:        $this->db->select ( 'PORT, USERNAME, PASSWORD' );
13:        $this->db->from ( 'SMS_CONFIGURATION' );
14:        $this->db->limit ( 1 );
15:        $query = $this->db->get ();
16:        if ($query->num_rows () > 0) {
17:            $queryresult = array ();
18:            $row = $query->row_array ();
19:            $queryresult ['port'] = $row ['PORT'];
20:            $queryresult ['username'] = $row ['USERNAME'];
21:            $queryresult ['password'] = $row ['PASSWORD'];
22:            return $queryresult;
23:        }
24:        return NULL;
25:    }
26:    function update_gsm_configurations($port, $username, $password) {
27:        $data = array (
28:            'USERNAME' => $username,
29:            'PASSWORD' => $password
30:        );
31:        $this->db->where ( 'PORT', $port );
32:        $this->db->update ( 'SMS_CONFIGURATION', $data );
33:    }
34: }
35: ?>
```

priviledge.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Priviledge extends CI_Model {
5:     function __construct() {
6:         parent::__construct ();
7:         $this->load->database ();
8:     }
9:     function get_all_priviledge() {
10:         $this->db->select ( 'ID, NAME, DESCRIPTION' );
11:         $this->db->from ( 'PRIVILEGE' );
12:         $query = $this->db->get ();
13:         if ($query->num_rows () > 0) {
14:             $queryresults = array ();
15:             foreach ( $query->result_array () as $row ) {
16:                 $queryresult = array ();
17:                 $queryresult ['id'] = $row ['ID'];
18:                 $queryresult ['name'] = $row ['NAME'];
19:                 $queryresult ['description'] = $row ['DESCRIPTION'];
20:                 array_push ( $queryresults, $queryresult );
21:             }
22:             return $queryresults;
23:         }
24:         return NULL;
25:     }
26:     function get_priviledge_details($priviledge_id) {
27:         $this->load->helper ( 'variable' );
28:
29:         $this->db->select ( 'ID, NAME, DESCRIPTION' );
30:         $this->db->from ( 'PRIVILEGE' );
31:         if (issetandnotempty ( $priviledge_id )) {
32:             $this->db->where ( 'ID', $priviledge_id );
33:         }
34:         $this->db->limit ( 1 );
35:         $query = $this->db->get ();
36:         if ($query->num_rows () > 0) {
37:             $queryresult = array ();
38:             $row = $query->row_array ();
39:             $queryresult ['id'] = $row ['ID'];
40:             $queryresult ['name'] = $row ['NAME'];
41:             $queryresult ['description'] = $row ['DESCRIPTION'];
42:             return $queryresult;
43:         }
44:         return NULL;
45:     }
46:     function update_priviledge_type($priviledge_id, $priviledge_details) {
47:         $data = array (
48:             'NAME' => $priviledge_details ['name'],
49:             'DESCRIPTION' => $priviledge_details ['description']
50:         );
51:         $this->db->where ( 'ID', $priviledge_id );
52:         $this->db->update ( 'PRIVILEGE', $data );
53:     }
54: }
55: ?>
```

partner_companies.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Partner_companies extends CI_Model {
5:     function __construct() {
6:         parent::__construct ();
7:         $this->load->database ();
8:     }
9:     function count_companies() {
10:         $this->db->select ( 'ID' );
11:         $this->db->from ( 'PARTNER_COMPANIES' );
12:         $query = $this->db->get ();
13:         return $query->num_rows ();
14:     }
15:     function companies_exist() {
16:         $this->db->select ( 'ID' );
17:         $this->db->from ( 'PARTNER_COMPANIES' );
18:         $query = $this->db->get ();
19:         if ($query->num_rows () > 0) {
20:             return TRUE;
21:         }
22:         return FALSE;
23:     }
24:     function get_all_partner_companies() {
25:         $this->db->select ( 'ID, ACRONYM, NAME, ADDRESS, DESCRIPTION, IS_ACTIVE
' );
26:         $this->db->from ( 'PARTNER_COMPANIES' );
27:         $query = $this->db->get ();
28:         if ($query->num_rows () > 0) {
29:             $queryresults = array ();
30:             foreach ( $query->result_array () as $row ) {
31:                 $queryresult = array ();
32:                 $queryresult ['partner_company_id'] = $row ['ID'];
33:                 $queryresult ['acronym'] = $row ['ACRONYM'];
34:                 $queryresult ['name'] = $row ['NAME'];
35:                 $queryresult ['address'] = $row ['ADDRESS'];
36:                 $queryresult ['description'] = $row ['DESCRIPTION'];
37:                 $queryresult ['is_active'] = $row ['IS_ACTIVE'];
38:                 array_push ( $queryresults, $queryresult );
39:             }
40:             return $queryresults;
41:         }
42:         return NULL;
43:     }
44:     function get_active_partner_companies() {
45:         $this->db->select ( 'ID, ACRONYM, NAME, ADDRESS, DESCRIPTION, IS_ACTIVE
' );
46:         $this->db->from ( 'PARTNER_COMPANIES' );
47:         $this->db->where ( 'IS_ACTIVE', '1' );
48:         $query = $this->db->get ();
49:         if ($query->num_rows () > 0) {
50:             $queryresults = array ();
51:             foreach ( $query->result_array () as $row ) {
52:                 $queryresult = array ();
53:                 $queryresult ['partner_company_id'] = $row ['ID'];
54:                 $queryresult ['acronym'] = $row ['ACRONYM'];
55:                 $queryresult ['name'] = $row ['NAME'];
56:                 $queryresult ['address'] = $row ['ADDRESS'];
57:                 $queryresult ['description'] = $row ['DESCRIPTION'];
58:                 $queryresult ['is_active'] = $row ['IS_ACTIVE'];
59:                 array_push ( $queryresults, $queryresult );
60:             }
61:             return $queryresults;
```

partner_companies.php

```
62:         }
63:         return NULL;
64:     }
65:     function get_company_details($partner_company_id) {
66:         $this->load->helper ( 'variable' );
67:
68:         $this->db->select ( 'ID, ACRONYM, NAME, ADDRESS, DESCRIPTION, IS_ACTIVE
' );
69:         $this->db->from ( 'PARTNER_COMPANIES' );
70:         if (issetandnotempty ( $partner_company_id )) {
71:             $this->db->where ( 'ID', $partner_company_id );
72:         }
73:         $this->db->limit ( 1 );
74:         $query = $this->db->get ();
75:         if ($query->num_rows () > 0) {
76:             $queryresult = array ();
77:             $row = $query->row_array ();
78:             $queryresult ['partner_company_id'] = $row ['ID'];
79:             $queryresult ['acronym'] = $row ['ACRONYM'];
80:             $queryresult ['name'] = $row ['NAME'];
81:             $queryresult ['address'] = $row ['ADDRESS'];
82:             $queryresult ['description'] = $row ['DESCRIPTION'];
83:             $queryresult ['is_active'] = $row ['IS_ACTIVE'];
84:             return $queryresult;
85:         }
86:         return NULL;
87:     }
88:     function update_company_details($partner_company_id, $partner_company_details)
{
89:         if (issetandnotempty ( 'acronym', $partner_company_details )) {
90:             $this->db->set ( 'ACRONYM', $partner_company_details ['acronym'
] );
91:         }
92:         if (issetandnotempty ( 'name', $partner_company_details )) {
93:             $this->db->set ( 'NAME', $partner_company_details ['name'] );
94:         }
95:         if (issetandnotempty ( 'address', $partner_company_details )) {
96:             $this->db->set ( 'ADDRESS', $partner_company_details ['address'
] );
97:         }
98:         if (issetandnotempty ( 'description', $partner_company_details )) {
99:             $this->db->set ( 'DESCRIPTION', $partner_company_details ['desc
ription'] );
100:         }
101:         if (issetandnotempty ( 'is_active', $partner_company_details )) {
102:             $this->db->set ( 'IS_ACTIVE', $partner_company_details ['is_act
ive'] );
103:         }
104:         $this->db->where ( 'ID', $partner_company_id );
105:         $this->db->update ( 'PARTNER_COMPANIES' );
106:     }
107:     function save_company_details($partner_company_details) {
108:         $this->load->helper ( 'variable' );
109:
110:         if (issetandnotempty ( 'acronym', $partner_company_details )) {
111:             $this->db->set ( 'ACRONYM', $partner_company_details ['acronym'
] );
112:         }
113:         if (issetandnotempty ( 'name', $partner_company_details )) {
114:             $this->db->set ( 'NAME', $partner_company_details ['name'] );
115:         }
116:         if (issetandnotempty ( 'address', $partner_company_details )) {
117:             $this->db->set ( 'ADDRESS', $partner_company_details ['address'
```

partner_companies.php

```
] );
118:         }
119:         if (issetandnotempty ( 'description', $partner_company_details )) {
120:             $this->db->set ( 'DESCRIPTION', $partner_company_details ['desc
ription'] );
121:         }
122:         if (issetandnotempty ( 'is_active', $partner_company_details )) {
123:             $this->db->set ( 'IS_ACTIVE', $partner_company_details ['is_act
ive'] );
124:         }
125:         $this->db->insert ( 'PARTNER_COMPANIES' );
126:     }
127:     function activate_partner_company($partner_company_id) {
128:         $data = array (
129:             'IS_ACTIVE' => '1'
130:         );
131:         $this->db->where ( 'ID', $partner_company_id );
132:         $this->db->update ( 'PARTNER_COMPANIES', $data );
133:     }
134:     function deactivate_partner_company($partner_company_id) {
135:         $data = array (
136:             'IS_ACTIVE' => '0'
137:         );
138:         $this->db->where ( 'ID', $partner_company_id );
139:         $this->db->update ( 'PARTNER_COMPANIES', $data );
140:     }
141: }
142: ?>
```

communication_types.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Communication_types extends CI_Model {
5:     function __construct() {
6:         parent::__construct ();
7:     }
8:     function get_all_communication_types() {
9:         $this->db->select ( 'ID, NAME, DESCRIPTION' );
10:        $this->db->from ( 'COMMUNICATION_TYPES' );
11:        $query = $this->db->get ();
12:        if ($query->num_rows () > 0) {
13:            $queryresults = array ();
14:            foreach ( $query->result_array () as $row ) {
15:                $queryresult = array ();
16:                $queryresult ['id'] = $row ['ID'];
17:                $queryresult ['name'] = $row ['NAME'];
18:                $queryresult ['description'] = $row ['DESCRIPTION'];
19:                array_push ( $queryresults, $queryresult );
20:            }
21:            return $queryresults;
22:        }
23:        return NULL;
24:    }
25:    function get_communication_type_details($communication_type_id) {
26:        $this->load->helper ( 'variable' );
27:
28:        $this->db->select ( 'ID, NAME, DESCRIPTION' );
29:        $this->db->from ( 'COMMUNICATION_TYPES' );
30:        if (issetandnotempty ( $communication_type_id )) {
31:            $this->db->where ( 'ID', $communication_type_id );
32:        }
33:        $this->db->limit ( 1 );
34:        $query = $this->db->get ();
35:        if ($query->num_rows () > 0) {
36:            $queryresult = array ();
37:            $row = $query->row_array ();
38:            $queryresult ['id'] = $row ['ID'];
39:            $queryresult ['name'] = $row ['NAME'];
40:            $queryresult ['description'] = $row ['DESCRIPTION'];
41:            return $queryresult;
42:        }
43:        return NULL;
44:    }
45:    function update_communication_type($communication_type_id, $communication_type_
details) {
46:        $data = array (
47:            'NAME' => $communication_type_details ['name'],
48:            'DESCRIPTION' => $communication_type_details ['descript
ion']
49:        );
50:        $this->db->where ( 'ID', $communication_type_id );
51:        $this->db->update ( 'COMMUNICATION_TYPES', $data );
52:    }
53: }
54: ?>
```

membership_types.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Membership_types extends CI_Model {
5:     function __construct() {
6:         parent::__construct ();
7:         $this->load->database ();
8:     }
9:     function get_all_membership_types() {
10:         $this->db->select ( 'ID, NAME, DESCRIPTION' );
11:         $this->db->from ( 'MEMBERSHIP_TYPES' );
12:         $query = $this->db->get ();
13:         if ($query->num_rows () > 0) {
14:             $queryresults = array ();
15:             foreach ( $query->result_array () as $row ) {
16:                 $queryresult = array ();
17:                 $queryresult ['id'] = $row ['ID'];
18:                 $queryresult ['name'] = $row ['NAME'];
19:                 $queryresult ['description'] = $row ['DESCRIPTION'];
20:                 array_push ( $queryresults, $queryresult );
21:             }
22:             return $queryresults;
23:         }
24:         return NULL;
25:     }
26:     function get_membership_type_details($membership_type_id) {
27:         $this->load->helper ( 'variable' );
28:
29:         $this->db->select ( 'ID, NAME, DESCRIPTION' );
30:         $this->db->from ( 'MEMBERSHIP_TYPES' );
31:         if (issetandnotempty ( $membership_type_id )) {
32:             $this->db->where ( 'ID', $membership_type_id );
33:         }
34:         $this->db->limit ( 1 );
35:         $query = $this->db->get ();
36:         if ($query->num_rows () > 0) {
37:             $queryresult = array ();
38:             $row = $query->row_array ();
39:             $queryresult ['id'] = $row ['ID'];
40:             $queryresult ['name'] = $row ['NAME'];
41:             $queryresult ['description'] = $row ['DESCRIPTION'];
42:             return $queryresult;
43:         }
44:         return NULL;
45:     }
46:     function update_membership_type($membership_type_id, $membership_type_details)
47:     {
48:         $data = array (
49:             'NAME' => $membership_type_details ['name'],
50:             'DESCRIPTION' => $membership_type_details ['description']
51:         );
52:         $this->db->where ( 'ID', $membership_type_id );
53:         $this->db->update ( 'MEMBERSHIP_TYPES', $data );
54:     }
55: ?>
```


company_details.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Company_details extends CI_Model {
5:     function __construct() {
6:         parent::__construct ();
7:         $this->load->database ();
8:     }
9:     function get_company_details() {
10:        $this->load->helper ( 'variable' );
11:
12:        $this->db->select ( 'ACRONYM, NAME, SITE' );
13:        $this->db->from ( 'COMPANY_DETAILS' );
14:        $this->db->limit ( 1 );
15:        $query = $this->db->get ();
16:        if ($query->num_rows () > 0) {
17:            $queryresult = array ();
18:            $row = $query->row_array ();
19:            $queryresult ['acronym'] = $row ['ACRONYM'];
20:            $queryresult ['name'] = $row ['NAME'];
21:            $queryresult ['site'] = $row ['SITE'];
22:            return $queryresult;
23:        }
24:        return NULL;
25:    }
26:    function update_company_details($old_acronym, $acronym, $name, $site) {
27:        $this->load->helper ( 'variable' );
28:
29:        if (issetandnotempty ( $acronym )) {
30:            $this->db->set ( 'ACRONYM', $acronym );
31:        }
32:        if (issetandnotempty ( $name )) {
33:            $this->db->set ( 'NAME', $name );
34:        }
35:        if (issetandnotempty ( $site )) {
36:            $this->db->set ( 'SITE', $site );
37:        }
38:        $this->db->where ( 'ACRONYM', $old_acronym );
39:        $this->db->update ( 'COMPANY_DETAILS' );
40:    }
41:    function save_member($acronym, $name, $site) {
42:        $this->load->helper ( 'variable' );
43:
44:        if (issetandnotempty ( $acronym )) {
45:            $this->db->set ( 'ACRONYM', $acronym );
46:        }
47:        if (issetandnotempty ( $name )) {
48:            $this->db->set ( 'NAME', $name );
49:        }
50:        if (issetandnotempty ( $site )) {
51:            $this->db->set ( 'SITE', $site );
52:        }
53:        $this->db->insert ( 'COMPANY_DETAILS' );
54:    }
55: }
56: ?>
```

partner_companies_personnel.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Partner_companies_personnel extends CI_Model {
5:     function count_active_personnel($company_id) {
6:         $this->db->select ( 'ID' );
7:         $this->db->from ( 'PARTNER_COMPANIES_PERSONNEL' );
8:         $this->db->where ( 'PARTNER_COMPANY_ID', $company_id );
9:         $this->db->where ( 'IS_ACTIVE', '1' );
10:        $query = $this->db->get ();
11:        return $query->num_rows;
12:    }
13:    function __construct() {
14:        parent::__construct ();
15:        $this->load->database ();
16:    }
17:    function personnel_exists($company_id) {
18:        $this->db->select ( 'ID' );
19:        $this->db->from ( 'PARTNER_COMPANIES_PERSONNEL' );
20:        if (issetandnotempty ( $company_id )) {
21:            $this->db->where_in ( 'PARTNER_COMPANY_ID', $company_id );
22:            $this->db->where ( 'IS_ACTIVE', '1' );
23:            $query = $this->db->get ();
24:            if ($query->num_rows () > 0) {
25:                return TRUE;
26:            }
27:        }
28:        return FALSE;
29:    }
30:    function get_all_personnel() {
31:        $this->db->select ( 'ID, FIRST_NAME, MIDDLE_NAME, LAST_NAME, CELLPHONE_
NUMBER, EMAIL_ADDRESS, PARTNER_COMPANY_ID, REGION_RESPONSIBILITY_ID, IS_ACTIVE' );
32:        $this->db->from ( 'PARTNER_COMPANIES_PERSONNEL' );
33:        $query = $this->db->get ();
34:        if ($query->num_rows () > 0) {
35:            $queryresults = array ();
36:            foreach ( $query->result_array () as $row ) {
37:                $queryresult = array ();
38:                $queryresult ['id'] = $row ['ID'];
39:                $queryresult ['first_name'] = $row ['FIRST_NAME'];
40:                $queryresult ['middle_name'] = $row ['MIDDLE_NAME'];
41:                $queryresult ['last_name'] = $row ['LAST_NAME'];
42:                $queryresult ['cellphone_number'] = $row ['CELLPHONE_NU
MBER'];
43:                $queryresult ['email_address'] = $row ['EMAIL_ADDRESS']
;
44:                $queryresult ['partner_company_id'] = $row ['PARTNER_CO
MPANY_ID'];
45:                $queryresult ['region_responsibility_id'] = $row ['REGI
ON_RESPONSIBILITY_ID'];
46:                $queryresult ['is_active'] = $row ['IS_ACTIVE'];
47:                array_push ( $queryresults, $queryresult );
48:            }
49:            return $queryresults;
50:        }
51:        return NULL;
52:    }
53:    function get_all_active_personnel() {
54:        $this->db->select ( 'ID, FIRST_NAME, MIDDLE_NAME, LAST_NAME, CELLPHONE_
NUMBER, EMAIL_ADDRESS, PARTNER_COMPANY_ID, REGION_RESPONSIBILITY_ID, IS_ACTIVE' );
55:        $this->db->from ( 'PARTNER_COMPANIES_PERSONNEL' );
56:        $this->db->where ( 'IS_ACTIVE', '1' );
57:        $query = $this->db->get ();
```

partner_companies_personnel.php

```
58:         if ($query->num_rows () > 0) {
59:             $queryresults = array ();
60:             foreach ( $query->result_array () as $row ) {
61:                 $queryresult = array ();
62:                 $queryresult ['id'] = $row ['ID'];
63:                 $queryresult ['first_name'] = $row ['FIRST_NAME'];
64:                 $queryresult ['middle_name'] = $row ['MIDDLE_NAME'];
65:                 $queryresult ['last_name'] = $row ['LAST_NAME'];
66:                 $queryresult ['cellphone_number'] = $row ['CELLPHONE_NUM
MBER'];
67:                 $queryresult ['email_address'] = $row ['EMAIL_ADDRESS']
;
68:                 $queryresult ['partner_company_id'] = $row ['PARTNER_CO
MPANY_ID'];
69:                 $queryresult ['region_responsibility_id'] = $row ['REGI
ON_RESPONSIBILITY_ID'];
70:                 $queryresult ['is_active'] = $row ['IS_ACTIVE'];
71:                 array_push ( $queryresults, $queryresult );
72:             }
73:             return $queryresults;
74:         }
75:         return NULL;
76:     }
77:     function get_all_active_personnel_of_company_id($company_id) {
78:         $this->db->select ( 'ID, FIRST_NAME, MIDDLE_NAME, LAST_NAME, CELLPHONE_
NUMBER, EMAIL_ADDRESS, PARTNER_COMPANY_ID, REGION_RESPONSIBILITY_ID, IS_ACTIVE' );
79:         $this->db->from ( 'PARTNER_COMPANIES_PERSONNEL' );
80:         if (issetandnotempty ( $company_id )) {
81:             $this->db->where_in ( 'PARTNER_COMPANY_ID', $company_id );
82:         }
83:         $this->db->where ( 'IS_ACTIVE', '1' );
84:         $query = $this->db->get ();
85:         if ($query->num_rows () > 0) {
86:             $queryresults = array ();
87:             foreach ( $query->result_array () as $row ) {
88:                 $queryresult = array ();
89:                 $queryresult ['id'] = $row ['ID'];
90:                 $queryresult ['first_name'] = $row ['FIRST_NAME'];
91:                 $queryresult ['middle_name'] = $row ['MIDDLE_NAME'];
92:                 $queryresult ['last_name'] = $row ['LAST_NAME'];
93:                 $queryresult ['cellphone_number'] = $row ['CELLPHONE_NUM
MBER'];
94:                 $queryresult ['email_address'] = $row ['EMAIL_ADDRESS']
;
95:                 $queryresult ['partner_company_id'] = $row ['PARTNER_CO
MPANY_ID'];
96:                 $queryresult ['region_responsibility_id'] = $row ['REGI
ON_RESPONSIBILITY_ID'];
97:                 $queryresult ['is_active'] = $row ['IS_ACTIVE'];
98:                 array_push ( $queryresults, $queryresult );
99:             }
100:            return $queryresults;
101:        }
102:        return NULL;
103:    }
104:    function get_all_personnel_of_company_id($company_id) {
105:        $this->db->select ( 'ID, FIRST_NAME, MIDDLE_NAME, LAST_NAME, CELLPHONE_
NUMBER, EMAIL_ADDRESS, PARTNER_COMPANY_ID, REGION_RESPONSIBILITY_ID, IS_ACTIVE' );
106:        $this->db->from ( 'PARTNER_COMPANIES_PERSONNEL' );
107:        if (issetandnotempty ( $company_id )) {
108:            $this->db->where_in ( 'PARTNER_COMPANY_ID', $company_id );
109:        }
110:        $query = $this->db->get ();
```

partner_companies_personnel.php

```
111:         if ($query->num_rows () > 0) {
112:             $queryresults = array ();
113:             foreach ( $query->result_array () as $row ) {
114:                 $queryresult = array ();
115:                 $queryresult ['id'] = $row ['ID'];
116:                 $queryresult ['first_name'] = $row ['FIRST_NAME'];
117:                 $queryresult ['middle_name'] = $row ['MIDDLE_NAME'];
118:                 $queryresult ['last_name'] = $row ['LAST_NAME'];
119:                 $queryresult ['cellphone_number'] = $row ['CELLPHONE_NUM
MBER'];
120:                 $queryresult ['email_address'] = $row ['EMAIL_ADDRESS']
;
121:                 $queryresult ['partner_company_id'] = $row ['PARTNER_CO
MPANY_ID'];
122:                 $queryresult ['region_responsibility_id'] = $row ['REGI
ON_RESPONSIBILITY_ID'];
123:                 $queryresult ['is_active'] = $row ['IS_ACTIVE'];
124:                 array_push ( $queryresults, $queryresult );
125:             }
126:             return $queryresults;
127:         }
128:         return NULL;
129:     }
130:     function get_all_active_regional_personnel_of_company_id($company_id, $region_i
d) {
131:         $this->db->select ( 'ID, FIRST_NAME, MIDDLE_NAME, LAST_NAME, CELLPHONE_
NUMBER, EMAIL_ADDRESS, PARTNER_COMPANY_ID, REGION_RESPONSIBILITY_ID, IS_ACTIVE' );
132:         $this->db->from ( 'PARTNER_COMPANIES_PERSONNEL' );
133:         if (issetandnotempty ( $company_id )) {
134:             $this->db->where_in ( 'PARTNER_COMPANY_ID', $company_id );
135:         }
136:         if (issetandnotempty ( $region_id )) {
137:             $this->db->where_in ( 'REGION_RESPONSIBILITY_ID', $region_id );
138:         }
139:         $this->db->where ( 'IS_ACTIVE', '1' );
140:         $query = $this->db->get ();
141:         if ($query->num_rows () > 0) {
142:             $queryresults = array ();
143:             foreach ( $query->result_array () as $row ) {
144:                 $queryresult = array ();
145:                 $queryresult ['id'] = $row ['ID'];
146:                 $queryresult ['first_name'] = $row ['FIRST_NAME'];
147:                 $queryresult ['middle_name'] = $row ['MIDDLE_NAME'];
148:                 $queryresult ['last_name'] = $row ['LAST_NAME'];
149:                 $queryresult ['cellphone_number'] = $row ['CELLPHONE_NUM
MBER'];
150:                 $queryresult ['email_address'] = $row ['EMAIL_ADDRESS']
;
151:                 $queryresult ['partner_company_id'] = $row ['PARTNER_CO
MPANY_ID'];
152:                 $queryresult ['region_responsibility_id'] = $row ['REGI
ON_RESPONSIBILITY_ID'];
153:                 $queryresult ['is_active'] = $row ['IS_ACTIVE'];
154:                 array_push ( $queryresults, $queryresult );
155:             }
156:             return $queryresults;
157:         }
158:         return NULL;
159:     }
160:     function get_personnel_details($personnel_id) {
161:         $this->db->select ( 'ID, FIRST_NAME, MIDDLE_NAME, LAST_NAME, CELLPHONE_
NUMBER, EMAIL_ADDRESS, PARTNER_COMPANY_ID, REGION_RESPONSIBILITY_ID, IS_ACTIVE' );
162:         $this->db->from ( 'PARTNER_COMPANIES_PERSONNEL' );
```

partner_companies_personnel.php

```
163:         if (issetandnotempty ( $personnel_id )) {
164:             $this->db->where ( 'ID', $personnel_id );
165:         }
166:         $query = $this->db->get ();
167:         if ($query->num_rows () > 0) {
168:             $queryresult = array ();
169:             $row = $query->row_array ();
170:             $queryresult ['id'] = $row ['ID'];
171:             $queryresult ['first_name'] = $row ['FIRST_NAME'];
172:             $queryresult ['middle_name'] = $row ['MIDDLE_NAME'];
173:             $queryresult ['last_name'] = $row ['LAST_NAME'];
174:             $queryresult ['cellphone_number'] = $row ['CELLPHONE_NUMBER'];
175:             $queryresult ['email_address'] = $row ['EMAIL_ADDRESS'];
176:             $queryresult ['partner_company_id'] = $row ['PARTNER_COMPANY_ID
'];
177:             $queryresult ['region_responsibility_id'] = $row ['REGION_RESPO
NSIBILITY_ID'];
178:             $queryresult ['is_active'] = $row ['IS_ACTIVE'];
179:             return $queryresult;
180:         }
181:         return NULL;
182:     }
183:     function update_personnel($personnel_id, $personnel_details) {
184:         if (issetandnotempty ( 'first_name', $personnel_details )) {
185:             $this->db->set ( 'FIRST_NAME', $personnel_details ['first_name'
]);
186:         }
187:         if (issetandnotempty ( 'middle_name', $personnel_details )) {
188:             $this->db->set ( 'MIDDLE_NAME', $personnel_details ['middle_nam
e'] );
189:         }
190:         if (issetandnotempty ( 'last_name', $personnel_details )) {
191:             $this->db->set ( 'LAST_NAME', $personnel_details ['last_name'
]);
192:         }
193:         if (issetandnotempty ( 'cellphone_number', $personnel_details )) {
194:             $this->db->set ( 'CELLPHONE_NUMBER', $personnel_details ['cellp
hone_number'] );
195:         }
196:         if (issetandnotempty ( 'email_address', $personnel_details )) {
197:             $this->db->set ( 'EMAIL_ADDRESS', $personnel_details ['email_ad
dress'] );
198:         }
199:         if (issetandnotempty ( 'partner_company_id', $personnel_details )) {
200:             $this->db->set ( 'PARTNER_COMPANY_ID', $personnel_details ['par
tner_company_id'] );
201:         }
202:         if (issetandnotempty ( 'region_responsibility_id', $personnel_details
) ) {
203:             $this->db->set ( 'REGION_RESPONSIBILITY_ID', $personnel_details
['region_responsibility_id'] );
204:         }
205:         if (issetandnotempty ( 'is_active', $personnel_details )) {
206:             $this->db->set ( 'IS_ACTIVE', $personnel_details ['is_active']
);
207:         }
208:         $this->db->where ( 'ID', $personnel_id );
209:         $this->db->update ( 'PARTNER_COMPANIES_PERSONNEL' );
210:     }
211:     function save_personnel($personnel_details) {
212:         if (issetandnotempty ( 'first_name', $personnel_details )) {
213:             $this->db->set ( 'FIRST_NAME', $personnel_details ['first_name'
]);
];
```

partner_companies_personnel.php

```
214:         }
215:         if (issetandnotempty ( 'middle_name', $personnel_details )) {
216:             $this->db->set ( 'MIDDLE_NAME', $personnel_details ['middle_nam
e'] );
217:         }
218:         if (issetandnotempty ( 'last_name', $personnel_details )) {
219:             $this->db->set ( 'LAST_NAME', $personnel_details ['last_name']
);
220:         }
221:         if (issetandnotempty ( 'cellphone_number', $personnel_details )) {
222:             $this->db->set ( 'CELLPHONE_NUMBER', $personnel_details ['cellp
hone_number'] );
223:         }
224:         if (issetandnotempty ( 'email_address', $personnel_details )) {
225:             $this->db->set ( 'EMAIL_ADDRESS', $personnel_details ['email_ad
dress'] );
226:         }
227:         if (issetandnotempty ( 'partner_company_id', $personnel_details )) {
228:             $this->db->set ( 'PARTNER_COMPANY_ID', $personnel_details ['par
tner_company_id'] );
229:         }
230:         if (issetandnotempty ( 'region_responsibility_id', $personnel_details
) ) {
231:             $this->db->set ( 'REGION_RESPONSIBILITY_ID', $personnel_details
['region_responsibility_id'] );
232:         }
233:         if (issetandnotempty ( 'is_active', $personnel_details )) {
234:             $this->db->set ( 'IS_ACTIVE', $personnel_details ['is_active']
);
235:         }
236:         $this->db->insert ( 'PARTNER_COMPANIES_PERSONNEL' );
237:     }
238:     function activate_personnel($personnel_id) {
239:         $data = array (
240:             'IS_ACTIVE' => '1'
241:         );
242:         $this->db->where ( 'ID', $personnel_id );
243:         $this->db->update ( 'PARTNER_COMPANIES_PERSONNEL', $data );
244:     }
245:     function deactivate_personnel($personnel_id) {
246:         $data = array (
247:             'IS_ACTIVE' => '0'
248:         );
249:         $this->db->where ( 'ID', $personnel_id );
250:         $this->db->update ( 'PARTNER_COMPANIES_PERSONNEL', $data );
251:     }
252: }
253: ?>
```

users.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Users extends CI_Model {
5:     function __construct() {
6:         parent::__construct ();
7:         $this->load->database ();
8:     }
9:     function count_users() {
10:         $this->db->select ( 'ID' );
11:         $this->db->from ( 'USERS' );
12:         $query = $this->db->get ();
13:         return $query->num_rows ();
14:     }
15:     function get_all_users() {
16:         $this->db->select ( 'ID, FIRST_NAME, MIDDLE_NAME, LAST_NAME, CELLPHONE_
NUMBER, EMAIL_ADDRESS, USERNAME, PASSWORD, PRIVILEGE_ID, IS_ACTIVE' );
17:         $this->db->from ( 'USERS' );
18:         $query = $this->db->get ();
19:         if ($query->num_rows () > 0) {
20:             $queryresults = array ();
21:             foreach ( $query->result_array () as $row ) {
22:                 $queryresult = array ();
23:                 $queryresult ['id'] = $row ['ID'];
24:                 $queryresult ['first_name'] = $row ['FIRST_NAME'];
25:                 $queryresult ['middle_name'] = $row ['MIDDLE_NAME'];
26:                 $queryresult ['last_name'] = $row ['LAST_NAME'];
27:                 $queryresult ['cellphone_number'] = $row ['CELLPHONE_NU
MBER'];
28:                 $queryresult ['email_address'] = $row ['EMAIL_ADDRESS']
;
29:                 $queryresult ['username'] = $row ['USERNAME'];
30:                 $queryresult ['password'] = $row ['PASSWORD'];
31:                 $queryresult ['priviledge_id'] = $row ['PRIVILEGE_ID']
;
32:                 $queryresult ['is_active'] = $row ['IS_ACTIVE'];
33:                 array_push ( $queryresults, $queryresult );
34:             }
35:             return $queryresults;
36:         }
37:         return NULL;
38:     }
39:     function get_all_active_users() {
40:         $this->db->select ( 'ID, FIRST_NAME, MIDDLE_NAME, LAST_NAME, CELLPHONE_
NUMBER, EMAIL_ADDRESS, USERNAME, PASSWORD, PRIVILEGE_ID, IS_ACTIVE' );
41:         $this->db->from ( 'USERS' );
42:         $this->db->where ( 'IS_ACTIVE', '1' );
43:         $query = $this->db->get ();
44:         if ($query->num_rows () > 0) {
45:             $queryresults = array ();
46:             foreach ( $query->result_array () as $row ) {
47:                 $queryresult = array ();
48:                 $queryresult ['id'] = $row ['ID'];
49:                 $queryresult ['first_name'] = $row ['FIRST_NAME'];
50:                 $queryresult ['middle_name'] = $row ['MIDDLE_NAME'];
51:                 $queryresult ['last_name'] = $row ['LAST_NAME'];
52:                 $queryresult ['cellphone_number'] = $row ['CELLPHONE_NU
MBER'];
53:                 $queryresult ['email_address'] = $row ['EMAIL_ADDRESS']
;
54:                 $queryresult ['username'] = $row ['USERNAME'];
55:                 $queryresult ['password'] = $row ['PASSWORD'];
56:                 $queryresult ['priviledge_id'] = $row ['PRIVILEGE_ID']
```

users.php

```
;  
57:             $queryresult ['is_active'] = $row ['IS_ACTIVE'];  
58:             array_push ( $queryresults, $queryresult );  
59:         }  
60:         return $queryresults;  
61:     }  
62:     return NULL;  
63: }  
64: function get_all_user_with_priviledge($priviledge_id) {  
65:     $this->load->helper ( 'variable' );  
66:  
67:     $this->db->select ( 'ID, FIRST_NAME, MIDDLE_NAME, LAST_NAME, CELLPHONE_  
NUMBER, EMAIL_ADDRESS, USERNAME, PASSWORD, PRIVILEGE_ID, IS_ACTIVE' );  
68:     $this->db->from ( 'USERS' );  
69:     if (issetandnotempty ( $priviledge_id )) {  
70:         $this->db->where_in ( 'PRIVILEGE_ID', $priviledge_id );  
71:     }  
72:     $query = $this->db->get ();  
73:     if ($query->num_rows () > 0) {  
74:         $queryresults = array ();  
75:         foreach ( $query->result_array () as $row ) {  
76:             $queryresult = array ();  
77:             $queryresult ['id'] = $row ['ID'];  
78:             $queryresult ['first_name'] = $row ['FIRST_NAME'];  
79:             $queryresult ['middle_name'] = $row ['MIDDLE_NAME'];  
80:             $queryresult ['last_name'] = $row ['LAST_NAME'];  
81:             $queryresult ['cellphone_number'] = $row ['CELLPHONE_NU  
MBER'];  
82:             $queryresult ['email_address'] = $row ['EMAIL_ADDRESS']  
;  
83:             $queryresult ['username'] = $row ['USERNAME'];  
84:             $queryresult ['password'] = $row ['PASSWORD'];  
85:             $queryresult ['priviledge_id'] = $row ['PRIVILEGE_ID']  
;  
86:             $queryresult ['is_active'] = $row ['IS_ACTIVE'];  
87:             array_push ( $queryresults, $queryresult );  
88:         }  
89:         return $queryresults;  
90:     }  
91:     return NULL;  
92: }  
93: function get_user_details($user_id) {  
94:     $this->load->helper ( 'variable' );  
95:  
96:     $this->db->select ( 'ID, FIRST_NAME, MIDDLE_NAME, LAST_NAME, CELLPHONE_  
NUMBER, EMAIL_ADDRESS, USERNAME, PASSWORD, PRIVILEGE_ID, IS_ACTIVE' );  
97:     $this->db->from ( 'USERS' );  
98:     if (issetandnotempty ( $user_id )) {  
99:         $this->db->where ( 'ID', $user_id );  
100:     }  
101:     $this->db->limit ( 1 );  
102:     $query = $this->db->get ();  
103:     if ($query->num_rows () > 0) {  
104:         $queryresult = array ();  
105:         $row = $query->row_array ();  
106:         $queryresult ['id'] = $row ['ID'];  
107:         $queryresult ['first_name'] = $row ['FIRST_NAME'];  
108:         $queryresult ['middle_name'] = $row ['MIDDLE_NAME'];  
109:         $queryresult ['last_name'] = $row ['LAST_NAME'];  
110:         $queryresult ['cellphone_number'] = $row ['CELLPHONE_NUMBER'];  
111:         $queryresult ['email_address'] = $row ['EMAIL_ADDRESS'];  
112:         $queryresult ['username'] = $row ['USERNAME'];  
113:         $queryresult ['password'] = $row ['PASSWORD'];
```


users.php

```
114:             $queryresult ['priviledge_id'] = $row ['PRIVILEGE_ID'];
115:             $queryresult ['is_active'] = $row ['IS_ACTIVE'];
116:             return $queryresult;
117:         }
118:     return NULL;
119: }
120: function get_user_details_with_userna($username) {
121:     $this->load->helper ( 'variable' );
122:
123:     $this->db->select ( 'ID, FIRST_NAME, MIDDLE_NAME, LAST_NAME, CELLPHONE_
NUMBER, EMAIL_ADDRESS, USERNAME, PASSWORD, PRIVILEGE_ID, IS_ACTIVE' );
124:     $this->db->from ( 'USERS' );
125:     if (issetandnotempty ( $username )) {
126:         $this->db->where ( 'USERNAME', $username );
127:     }
128:     $this->db->limit ( 1 );
129:     $query = $this->db->get ();
130:     if ( $query->num_rows () > 0 ) {
131:         $queryresult = array ();
132:         $row = $query->row_array ();
133:         $queryresult ['id'] = $row ['ID'];
134:         $queryresult ['first_name'] = $row ['FIRST_NAME'];
135:         $queryresult ['middle_name'] = $row ['MIDDLE_NAME'];
136:         $queryresult ['last_name'] = $row ['LAST_NAME'];
137:         $queryresult ['cellphone_number'] = $row ['CELLPHONE_NUMBER'];
138:         $queryresult ['email_address'] = $row ['EMAIL_ADDRESS'];
139:         $queryresult ['username'] = $row ['USERNAME'];
140:         $queryresult ['password'] = $row ['PASSWORD'];
141:         $queryresult ['priviledge_id'] = $row ['PRIVILEGE_ID'];
142:         $queryresult ['is_active'] = $row ['IS_ACTIVE'];
143:         return $queryresult;
144:     }
145:     return NULL;
146: }
147: function update_user($user_id, $user_details) {
148:     if (issetandnotempty ( 'first_name', $user_details )) {
149:         $this->db->set ( 'FIRST_NAME', $user_details ['first_name'] );
150:     }
151:     if (issetandnotempty ( 'middle_name', $user_details )) {
152:         $this->db->set ( 'MIDDLE_NAME', $user_details ['middle_name'] )
;
153:     }
154:     if (issetandnotempty ( 'last_name', $user_details )) {
155:         $this->db->set ( 'LAST_NAME', $user_details ['last_name'] );
156:     }
157:     if (issetandnotempty ( 'cellphone_number', $user_details )) {
158:         $this->db->set ( 'CELLPHONE_NUMBER', $user_details ['cellphone_
number'] );
159:     }
160:     if (issetandnotempty ( 'email_address', $user_details )) {
161:         $this->db->set ( 'EMAIL_ADDRESS', $user_details ['email_address
'] );
162:     }
163:     if (issetandnotempty ( 'username', $user_details )) {
164:         $this->db->set ( 'USERNAME', $user_details ['username'] );
165:     }
166:     if (issetandnotempty ( 'password', $user_details )) {
167:         $this->db->set ( 'PASSWORD', $user_details ['password'] );
168:     }
169:     if (issetandnotempty ( 'priviledge_id', $user_details )) {
170:         $this->db->set ( 'PRIVILEGE_ID', $user_details ['priviledge_id
'] );
171:     }
```

users.php

```
172:         if (issetandnotempty ( 'is_active', $user_details )) {
173:             $this->db->set ( 'IS_ACTIVE', $user_details ['is_active'] );
174:         }
175:         $this->db->where ( 'ID', $user_id );
176:         $this->db->update ( 'USERS' );
177:     }
178:     function save_user($user_details) {
179:         $this->load->helper ( 'variable' );
180:
181:         if (issetandnotempty ( $user_details ['first_name'] )) {
182:             $this->db->set ( 'FIRST_NAME', $user_details ['first_name'] );
183:         }
184:         if (issetandnotempty ( $user_details ['middle_name'] )) {
185:             $this->db->set ( 'MIDDLE_NAME', $user_details ['middle_name'] )
;
186:         }
187:         if (issetandnotempty ( $user_details ['last_name'] )) {
188:             $this->db->set ( 'LAST_NAME', $user_details ['last_name'] );
189:         }
190:         if (issetandnotempty ( $user_details ['cellphone_number'] )) {
191:             $this->db->set ( 'CELLPHONE_NUMBER', $user_details ['cellphone_
number'] );
192:         }
193:         if (issetandnotempty ( $user_details ['email_address'] )) {
194:             $this->db->set ( 'EMAIL_ADDRESS', $user_details ['email_address
'] );
195:         }
196:         if (issetandnotempty ( $user_details ['username'] )) {
197:             $this->db->set ( 'USERNAME', $user_details ['username'] );
198:         }
199:         if (issetandnotempty ( $user_details ['password'] )) {
200:             $this->db->set ( 'PASSWORD', $user_details ['password'] );
201:         }
202:         if (issetandnotempty ( $user_details ['priviledge_id'] )) {
203:             $this->db->set ( 'PRIVILEGE_ID', $user_details ['priviledge_id
'] );
204:         }
205:         if (issetandnotempty ( $user_details ['is_active'] )) {
206:             $this->db->set ( 'IS_ACTIVE', $user_details ['is_active'] );
207:         }
208:         $this->db->insert ( 'USERS' );
209:     }
210:     function activate_user($user_id) {
211:         $data = array (
212:             'IS_ACTIVE' => '1'
213:         );
214:         $this->db->where ( 'ID', $user_id );
215:         $this->db->update ( 'USERS', $data );
216:     }
217:     function deactivate_user($user_id) {
218:         $data = array (
219:             'IS_ACTIVE' => '0'
220:         );
221:         $this->db->where ( 'ID', $user_id );
222:         $this->db->update ( 'USERS', $data );
223:     }
224:     function get_user_details_with_credentials($username, $password) {
225:         $this->load->helper ( 'variable' );
226:
227:         $this->db->select ( 'ID, USERNAME, PASSWORD' );
228:         $this->db->from ( 'USERS' );
229:         if (issetandnotempty ( $username )) {
230:             $this->db->where ( 'USERNAME', $username );
```

users.php

```
231:         }
232:         if (issetandnotempty ( $password )) {
233:             $this->db->where ( 'PASSWORD', hash ( "sha256", $password ) );
234:         }
235:         $this->db->limit ( 1 );
236:         $query = $this->db->get ();
237:         if ($query->num_rows () > 0) {
238:             $queryresult = array ();
239:             $row = $query->row_array ();
240:             $queryresult ['id'] = $row ['ID'];
241:             $queryresult ['username'] = $row ['USERNAME'];
242:             $queryresult ['password'] = $row ['PASSWORD'];
243:             return $queryresult;
244:         }
245:         return NULL;
246:     }
247: }
248: ?>
```

message_types.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4: class Message_types extends CI_Model {
5:     function __construct() {
6:         parent::__construct ();
7:     }
8:     function get_all_message_types() {
9:         $this->db->select ( 'ID, NAME, DESCRIPTION' );
10:        $this->db->from ( 'MESSAGE_TYPES' );
11:        $query = $this->db->get ();
12:        if ($query->num_rows () > 0) {
13:            $queryresults = array ();
14:            foreach ( $query->result_array () as $row ) {
15:                $queryresult = array ();
16:                $queryresult ['id'] = $row ['ID'];
17:                $queryresult ['name'] = $row ['NAME'];
18:                $queryresult ['description'] = $row ['DESCRIPTION'];
19:                array_push ( $queryresults, $queryresult );
20:            }
21:            return $queryresults;
22:        }
23:        return NULL;
24:    }
25:    function get_message_type_details($message_type_id) {
26:        $this->load->helper ( 'variable' );
27:
28:        $this->db->select ( 'ID, NAME, DESCRIPTION' );
29:        $this->db->from ( 'MESSAGE_TYPES' );
30:        if (issetandnotempty ( $message_type_id )) {
31:            $this->db->where ( 'ID', $message_type_id );
32:        }
33:        $this->db->limit ( 1 );
34:        $query = $this->db->get ();
35:        if ($query->num_rows () > 0) {
36:            $queryresult = array ();
37:            $row = $query->row_array ();
38:            $queryresult ['id'] = $row ['ID'];
39:            $queryresult ['name'] = $row ['NAME'];
40:            $queryresult ['description'] = $row ['DESCRIPTION'];
41:            return $queryresult;
42:        }
43:        return NULL;
44:    }
45:    function update_message_type($message_type_id, $message_type_details) {
46:        $data = array (
47:            'NAME' => $message_type_details ['name'],
48:            'DESCRIPTION' => $message_type_details ['description']
49:        );
50:        $this->db->where ( 'ID', $message_type_id );
51:        $this->db->update ( 'MESSAGE_TYPES', $data );
52:    }
53: }
54: ?>
```

index.html

```
1: <html>
2: <head>
3: <title>403 Forbidden</title>
4: </head>
5: <body>
6:
7:     <p>Directory access is forbidden.</p>
8:
9: </body>
10: </html>
```

error_db.php

```
1: <!DOCTYPE html>
2: <html lang="en">
3: <head>
4: <title>Database Error</title>
5: <style type="text/css">
6: ::selection {
7:     background-color: #E13300;
8:     color: white;
9: }
10:
11: ::moz-selection {
12:     background-color: #E13300;
13:     color: white;
14: }
15:
16: ::webkit-selection {
17:     background-color: #E13300;
18:     color: white;
19: }
20:
21: body {
22:     background-color: #fff;
23:     margin: 40px;
24:     font: 13px/20px normal Helvetica, Arial, sans-serif;
25:     color: #4F5155;
26: }
27:
28: a {
29:     color: #003399;
30:     background-color: transparent;
31:     font-weight: normal;
32: }
33:
34: h1 {
35:     color: #444;
36:     background-color: transparent;
37:     border-bottom: 1px solid #D0D0D0;
38:     font-size: 19px;
39:     font-weight: normal;
40:     margin: 0 0 14px 0;
41:     padding: 14px 15px 10px 15px;
42: }
43:
44: code {
45:     font-family: Consolas, Monaco, Courier New, Courier, monospace;
46:     font-size: 12px;
47:     background-color: #f9f9f9;
48:     border: 1px solid #D0D0D0;
49:     color: #002166;
50:     display: block;
51:     margin: 14px 0 14px 0;
52:     padding: 12px 10px 12px 10px;
53: }
54:
55: #container {
56:     margin: 10px;
57:     border: 1px solid #D0D0D0;
58:     -webkit-box-shadow: 0 0 8px #D0D0D0;
59: }
60:
61: p {
62:     margin: 12px 15px 12px 15px;
63: }
```

error_db.php

```
64: </style>
65: </head>
66: <body>
67:     <div id="container">
68:         <h1><?php echo $heading; ?></h1>
69:         <?php echo $message; ?>
70:     </div>
71: </body>
72: </html>
```

error_general.php

```
1: <!DOCTYPE html>
2: <html lang="en">
3: <head>
4: <title>Error</title>
5: <style type="text/css">
6: ::selection {
7:     background-color: #E13300;
8:     color: white;
9: }
10:
11: ::moz-selection {
12:     background-color: #E13300;
13:     color: white;
14: }
15:
16: ::webkit-selection {
17:     background-color: #E13300;
18:     color: white;
19: }
20:
21: body {
22:     background-color: #fff;
23:     margin: 40px;
24:     font: 13px/20px normal Helvetica, Arial, sans-serif;
25:     color: #4F5155;
26: }
27:
28: a {
29:     color: #003399;
30:     background-color: transparent;
31:     font-weight: normal;
32: }
33:
34: h1 {
35:     color: #444;
36:     background-color: transparent;
37:     border-bottom: 1px solid #D0D0D0;
38:     font-size: 19px;
39:     font-weight: normal;
40:     margin: 0 0 14px 0;
41:     padding: 14px 15px 10px 15px;
42: }
43:
44: code {
45:     font-family: Consolas, Monaco, Courier New, Courier, monospace;
46:     font-size: 12px;
47:     background-color: #f9f9f9;
48:     border: 1px solid #D0D0D0;
49:     color: #002166;
50:     display: block;
51:     margin: 14px 0 14px 0;
52:     padding: 12px 10px 12px 10px;
53: }
54:
55: #container {
56:     margin: 10px;
57:     border: 1px solid #D0D0D0;
58:     -webkit-box-shadow: 0 0 8px #D0D0D0;
59: }
60:
61: p {
62:     margin: 12px 15px 12px 15px;
63: }
```


error_general.php

```
64: </style>
65: </head>
66: <body>
67:     <div id="container">
68:         <h1><?php echo $heading; ?></h1>
69:         <?php echo $message; ?>
70:     </div>
71: </body>
72: </html>
```

error_php.php

```
1: <div
2:     style="border: 1px solid #990000; padding-left: 20px; margin: 0 0 10px 0;">
3:
4:     <h4>A PHP Error was encountered</h4>
5:
6:     <p>Severity: <?php echo $severity; ?></p>
7:     <p>Message: <?php echo $message; ?></p>
8:     <p>Filename: <?php echo $filepath; ?></p>
9:     <p>Line Number: <?php echo $line; ?></p>
10:
11: </div>
```

error_404.php

```
1: <!DOCTYPE html>
2: <html lang="en">
3: <head>
4: <title>404 Page Not Found</title>
5: <style type="text/css">
6: ::selection {
7:     background-color: #E13300;
8:     color: white;
9: }
10:
11: ::moz-selection {
12:     background-color: #E13300;
13:     color: white;
14: }
15:
16: ::webkit-selection {
17:     background-color: #E13300;
18:     color: white;
19: }
20:
21: body {
22:     background-color: #fff;
23:     margin: 40px;
24:     font: 13px/20px normal Helvetica, Arial, sans-serif;
25:     color: #4F5155;
26: }
27:
28: a {
29:     color: #003399;
30:     background-color: transparent;
31:     font-weight: normal;
32: }
33:
34: h1 {
35:     color: #444;
36:     background-color: transparent;
37:     border-bottom: 1px solid #D0D0D0;
38:     font-size: 19px;
39:     font-weight: normal;
40:     margin: 0 0 14px 0;
41:     padding: 14px 15px 10px 15px;
42: }
43:
44: code {
45:     font-family: Consolas, Monaco, Courier New, Courier, monospace;
46:     font-size: 12px;
47:     background-color: #f9f9f9;
48:     border: 1px solid #D0D0D0;
49:     color: #002166;
50:     display: block;
51:     margin: 14px 0 14px 0;
52:     padding: 12px 10px 12px 10px;
53: }
54:
55: #container {
56:     margin: 10px;
57:     border: 1px solid #D0D0D0;
58:     -webkit-box-shadow: 0 0 8px #D0D0D0;
59: }
60:
61: p {
62:     margin: 12px 15px 12px 15px;
63: }
```

error_404.php

```
64: </style>
65: </head>
66: <body>
67:     <div id="container">
68:         <h1><?php echo $heading; ?></h1>
69:         <?php echo $message; ?>
70:     </div>
71: </body>
72: </html>
```

list.php

```
1: <html>
2: <head>
3: <title><?php echo $this->session->userdata ( 'company_acronym' ) . ' - Partner'?></titl
e>
4: <?php
5: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/normalize.css' );
6: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/foundation.min.css' );
7: echo css_tag ( 'resources/foundation-icons/foundation-icons/foundation-icons.css' );
8: echo css_tag ( 'resources/custom/css/sizing.css' );
9: echo css_tag ( 'resources/datatables-foundation/dataTables.foundation.css' );
10: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/modernizr.js' );
11: ?>
12: </head>
13: <body>
14:     <?php
15:     $data ['page_set'] = 'companies';
16:     $this->load->view ( 'navigation/navigationbar_user', $data );
17:     ?>
18:     <div class="row">
19:     <?php echo br(1);?>
20:     </div>
21:     <div class="row">
22:         <div class="small-10 small-centered columns">
23:             <h3>
24:                 <strong>company profiles</strong>
25:             </h3>
26:             <p>a collation of a quick summary about our partner companies.
you
27:                 may do queries with the list through the search field.<
/p>
28:             </div>
29:         </div>
30:         <div class="row">
31:             <div class="small-10 small-centered columns">
32:                 <table>
33:                     <thead>
34:                         <tr>
35:                             <th></th>
36:                             <th>acronym</th>
37:                             <th>name</th>
38:                             <th>services</th>
39:                             <th>personnel</th>
40:                             <th>active</th>
41:                         </tr>
42:                     </thead>
43:                     <tbody>
44:                         <?php
45:                         foreach ( $partner_company_list as $partner_com
pany_element ) {
46:                             ?>
47:                             <tr>
48:                                 <td><?php
49:                                 echo anchor ( $this->config->base_url (
'partner/update/' . $partner_company_element ['partner_company_id'] ), '
50:                                     <i class="fi-widget"></i>' );
51:                                 ?></td>
52:                                 <td><?php echo $partner_company_element
['acronym']; ?></td>
53:                                 <td><?php echo anchor('dashboard/partne
r/' . $partner_company_element ['partner_company_id'], $partner_company_element ['name']); ?></td>
>
54:                                 <td><?php echo anchor ( $this->config->
base_url ( 'dashboard/services/' . $partner_company_element ['partner_company_id'] ), '<i clas
```

list.php

```
s="fi-paint-bucket"></i> services ( '.$partner_company_element['active_service_count'].')' );?>
</td>
55:                                     <td><?php echo anchor ( $this->config->
base_url ( 'dashboard/personnel/' . $partner_company_element ['partner_company_id'] ), ' <i cla
ss="fi-torso-business"></i> personnel ( '.$partner_company_element ['active_personnel_count'].')
' );?></td>
56:                                     <td
57:                                     bgcolor="<?php echo ( $partner_
company_element ['is_active'] == 'active' ? '#77DD77' : '#FDFD96' ) ?>"><strong><?php echo anc
hor ( $this->config->base_url ( 'partner/' . ($partner_company_element ['is_active'] === 'acti
ve' ? 'deactivate' : 'activate') . '/' . $partner_company_element ['partner_company_id'] ), ($
partner_company_element ['is_active'] === 'active' ? '<i class="fi-minus"></i> deactivate' : '
<i class="fi-plus"></i> activate') );?></strong></td>
58:                                     </tr>
59:                                     <?php
60:                                     }
61:                                     ?>
62:                                 </tbody>
63:                             </table>
64:                         </div>
65:         </div>
66:         <?php
67:         echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
68:         // echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/jquery.js' );
69:         echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
70:         echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
71:         echo js_tag ( 'resources/datatables-foundation/dataTables.foundation.min.js' );
72:         ?>
73:         <script type="text/javascript">
74:         $(document).foundation();
75:         $(document).ready( function () {
76:             $('table').DataTable();
77:         } );
78:         </script>
79: </body>
80: </html>
```


registration.php

```
or'
51:                                     ) : '');
52:                                     echo form_input ( $acronym );
53:                                     ?>
54:                                     <?php
55:                                     if (issetandnotempty ( form_err
or ( 'acronym' ) ) ) {
56:                                     echo form_error ( 'acro
nym', '<small class="error">', '</small>' );
57:                                     }
58:                                     ?>
59:                                     </div>
60:                                     <div class="small-8 columns">
61:                                     <label
62:                                     <?php echo (issetandnot
empty ( form_error ( 'name' ) ) ? 'class="error"' : '' )?>>
63:                                     Name </label>
64:                                     <?php
65:                                     $name = array (
66:                                     'type' => 'text
',
67:                                     'name' => 'name
',
68:                                     'id' => 'name',
69:                                     'value' => set_
value ( 'name' ),
70:                                     'maxlength' =>
'255',
71:                                     'placeholder' =
> 'Name'
72:                                     );
73:                                     (issetandnotempty ( form_error
( 'name' ) ) ? $name += array (
74:                                     'class' => 'err
or'
75:                                     ) : '');
76:                                     echo form_input ( $name );
77:                                     ?>
78:                                     <?php
79:                                     if (issetandnotempty ( form_err
or ( 'name' ) ) ) {
80:                                     echo form_error ( 'name
', '<small class="error">', '</small>' );
81:                                     }
82:                                     ?>
83:                                     </div>
84:                                     </div>
85:                                     <div class="row">
86:                                     <div class="small-12 columns">
87:                                     <label
88:                                     <?php echo (issetandnot
empty ( form_error ( 'address' ) ) ? 'class="error"' : '' )?>>
89:                                     Address </label>
90:                                     <?php
91:                                     $address = array (
92:                                     'type' => 'text
',
93:                                     'name' => 'addr
ess',
94:                                     'id' => 'addres
s',
95:                                     'value' => set_
value ( 'address' ),
```


registration.php

```
187:
188:
189:
190:     <?php
191:     echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
192:     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
193:     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
194:     ?>
195:     <script>
196:     $(document).foundation();
197:     </script>
198:
199: </body>
200: </html>
```


update.php

```
100:                                     <?php
101:                                     $address = array (
102:                                         'type' => 'text
',
103:                                         'name' => 'addr
ess',
104:                                         'id' => 'addres
s',
105:                                         'value' => (iss
etandnotempty ( form_error ( 'address' ) ) ? set_value ( 'address' ) : $address),
106:                                         'maxlength' =>
'15',
107:                                         'placeholder' =
> 'Address'
108:                                     );
109:                                     (issetandnotempty ( form_error
( 'address' ) ) ? $address += array (
110:                                         'class' => 'err
or'
111:                                         ) : 'address');
112:                                     echo form_input ( $address );
113:                                     ?>
114:                                     <?php
115:                                     if (issetandnotempty ( form_err
or ( 'address' ) )) {
116:                                         echo form_error ( 'addr
ess', '<small class="error">', '</small>' );
117:                                     }
118:                                     ?>
119:                                     </div>
120:                                     </div>
121:                                     <div class="row">
122:                                         <div class="small-12 columns">
123:                                             <label
124:                                             <?php echo (issetandnot
empty ( form_error ( 'description' ) ) ? 'class="error"' : '' )?>>
125:                                             Description </label>
126:                                     <?php
127:                                     $description = array (
128:                                         'type' => 'text
',
129:                                         'name' => 'desc
ription',
130:                                         'id' => 'descri
ption',
131:                                         'value' => (iss
etandnotempty ( form_error ( 'description' ) ) ? set_value ( 'description' ) : $description),
132:                                         'maxlength' =>
'15',
133:                                         'placeholder' =
> 'description'
134:                                     );
135:                                     (issetandnotempty ( form_error
( 'description' ) ) ? $description += array (
136:                                         'class' => 'err
or'
137:                                         ) : '');
138:                                     echo form_textarea ( $descripti
on );
139:                                     ?>
140:                                     <?php
141:                                     if (issetandnotempty ( form_err
or ( 'description' ) )) {
```

update.php

```
142:                                                                 echo form_error ( 'desc
ription', '<small class="error">', '</small>' );
143:                                                                 }
144:                                                                 ?>
145:                                                                 </div>
146:                                                                 </div>
147:                                                                 <div class="row">
148:                                                                 <div class="small-12 columns">
149:                                                                 <label> Active </label>
150:                                                                 <div class="row">
151:                                                                 <div class="small-6 col
umns">
152:                                                                 <label>
153:                                                                 <?php
154:                                                                 $active = array
(
155:                                                                 'name' => 'is_active',
156:                                                                 'id' => 'is_active',
157:                                                                 'value' => 'active',
158:                                                                 'checked' => ($is_active === '1' ? TRUE : FALSE),
159:                                                                 'style' => 'margin:10px'
160:                                                                 );
161:                                                                 echo form_radio
( $active ) . ' active';
162:                                                                 ?>
163:                                                                 </label>
164:                                                                 </div>
165:                                                                 <div class="small-6 col
umns">
166:                                                                 <label>
167:                                                                 <?php
168:                                                                 $inactive = arr
ay (
169:                                                                 'name' => 'is_active',
170:                                                                 'id' => 'is_active',
171:                                                                 'value' => 'inactive',
172:                                                                 'checked' => ($is_active === '0' ? TRUE : FALSE),
173:                                                                 'style' => 'margin:10px'
174:                                                                 );
175:                                                                 echo form_radio
( $inactive ) . 'inactive';
176:                                                                 ?>
177:                                                                 </label>
178:                                                                 </div>
179:                                                                 </div>
180:                                                                 </div>
181:                                                                 </div>
182:                                                                 <div class="row">
183:                                                                 <div class="small-12 columns"><?php
184:                                                                 $submit = array (
185:                                                                 'name' => 'register',
186:                                                                 'class' => 'tiny button
radius span-column'
```

update.php

```
187:                                     );
188:                                     echo form_submit ( $submit, 'register'
);
189:                                     ?></div>
190:                                     </div>
191:                                     <?php
192:                                     echo form_close ();
193:                                     ?>
194:                                     </div>
195:                                     </div>
196:                                     </div>
197: </div>
198:
199:
200:
201:
202: <?php
203: echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
204: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
205: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
206: ?>
207: <script>
208: $(document).foundation();
209: </script>
210:
211: </body>
212: </html>
```


index.html

```
1: <html>
2: <head>
3: <title>403 Forbidden</title>
4: </head>
5: <body>
6:
7:     <p>Directory access is forbidden.</p>
8:
9: </body>
10: </html>
```

list.php

```
1: <html>
2: <head>
3: <title><?php echo $this->session->userdata ( 'company_acronym' ) . ' - User'?></title>
4: <?php
5: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/normalize.css' );
6: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/foundation.min.css' );
7: echo css_tag ( 'resources/foundation-icons/foundation-icons/foundation-icons.css' );
8: echo css_tag ( 'resources/custom/css/sizing.css' );
9: echo css_tag ( 'resources/datatables-foundation/dataTables.foundation.css' );
10: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/modernizr.js' );
11: ?>
12: </head>
13: <body>
14:     <?php
15:         $data ['page_set'] = 'system';
16:         $this->load->view ( 'navigation/navigationbar_user', $data );
17:     ?>
18:     <div class="row">
19:         <?php echo br(1);?>
20:     </div>
21:     <div class="row">
22:         <div class="small-10 small-centered columns">
23:             <h3>
24:                 <strong>user profiles</strong>
25:             </h3>
26:             <p>a collation of a quick summary about registered users of the
27:                 system. you may do queries with the list through the se
arch field.</p>
28:         </div>
29:     </div>
30:     <div class="row">
31:         <div class="small-10 small-centered columns">
32:             <table>
33:                 <thead>
34:                     <tr>
35:                         <th></th>
36:                         <th>username</th>
37:                         <th>name</th>
38:                         <th>priviledge</th>
39:                         <th>active</th>
40:                     </tr>
41:                 </thead>
42:                 <tbody>
43:                     <?php
44:                         foreach ( $user_list as $user_element ) {
45:                             ?>
46:                             <tr>
47:                                 <td><?php echo anchor ( $this->config->
base_url ( 'index.php/user/update/' . $user_element ['id']), '<i class="fi-wiget"></i>' );?><
/td>
48:                                 <td><?php echo $user_element ['username
']; ?></td>
49:                                 <td><?php echo $user_element ['name'];
?></td>
50:                                 <td><?php echo $user_element ['priviled
ge']; ?></td>
51:                                 <td>
52:                                     bgcolor="<?php echo ( $user_ele
ment ['is_active'] == 'active' ? '#77DD77' : '#DFD96' ) ?>"><strong><?php echo anchor ( $this
->config->base_url ( 'index.php/user/' . ($user_element ['is_active'] == 'active' ? 'deactiva
te' : 'activate') . '/' . $user_element ['id'] ), ($user_element ['is_active'] == 'active' ?
'<i class="fi-minus"></i> deactivate' : '<i class="fi-plus"></i> activate' ) );?></strong></td>
53:                             </tr>
```

list.php

```
54:                                     <?php
55:                                     }
56:                                     ?>
57:                                 </tbody>
58:                             </table>
59:                         </div>
60: </div>
61: <?php
62: echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
63: // echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/jquery.js' );
64: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
65: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
66: echo js_tag ( 'resources/datatables-foundation/dataTables.foundation.min.js' );
67: ?>
68: <script type="text/javascript">
69: $(document).foundation();
70: $(document).ready( function () {
71:     $('table').DataTable();
72: } );
73: </script>
74: </body>
75: </html>
```

registration.php

```
1: <html>
2: <head>
3: <title><?php echo $this->session->userdata ( 'company_acronym' ) . ' - User'?></title>
4: <?php
5: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/normalize.css' );
6: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/foundation.min.css' );
7: echo css_tag ( 'resources/foundation-icons/foundation-icons/foundation-icons.css' );
8: echo css_tag ( 'resources/custom/css/sizing.css' );
9: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/modernizr.js' );
10: ?>
11: </head>
12: <body>
13:     <?php
14:     $data [ 'page_set' ] = 'system';
15:     $this->load->view ( 'navigation/navigationbar_user', $data );
16:     ?>
17:     <div class="row">
18:     <?php echo br(1);?>
19:     </div>
20:     <div class="row">
21:         <div class="small-12 small-centered columns">
22:             <div class="row">
23:                 <div class="large-3 small-12 columns text-justify">
24:                     <h4>
25:                         <strong>user registration</strong>
26:                         <p>please provide the latest details wi
th regards to our customer.
27:                             the membership pin, cellphone n
umber and email address are the
28:                             most important fields for this
form.</p>
29:                     </h4>
30:                 </div>
31:                 <div class="large-9 small-12 columns panel radius">
32:                     <?php
33:                     echo form_open ( 'user/process_registration' );
34:                     ?>
35:                 <div class="row">
36:                     <div class="small-4 columns">
37:                         <label
38:                             <?php echo (issetandnot
empty ( form_error ( 'first_name' ) ) ? 'class="error"' : '' )?>
39:                             First Name </label>
40:                         <?php
41:                         $first_name = array (
42:                             'type' => 'text
',
43:                             'name' => 'firs
t_name',
44:                             'id' => 'first_
name',
45:                             'value' => set_
value ( 'first_name' ),
46:                             'maxlength' =>
'15',
47:                             'placeholder' =
> 'Juan'
48:                         );
49:                         (issetandnotempty ( form_error
( 'first_name' ) ) ? $first_name += array (
50:                             'class' => 'err
or'
```

registration.php

```
52:                                     ) : '');
53:                                     echo form_input ( $first_name )
;
54:                                     ?>
55:                                     <?php
56:                                     if (issetandnotempty ( form_err
or ( 'first_name' ) )) {
57:                                         echo form_error ( 'firs
t_name', '<small class="error">', '</small>' );
58:                                     }
59:                                     ?>
60:                                     </div>
61:                                     <div class="small-4 columns">
62:                                         <label
63:                                             <?php echo (issetandnot
empty ( form_error ( 'middle_name' ) ) ? 'class="error"' : '')?>>
64:                                             Middle Name </label>
65:                                             <?php
66:                                             $middle_name = array (
67:                                                 'type' => 'text
',
68:                                                 'name' => 'midd
le_name',
69:                                                 'id' => 'middle
_name',
70:                                                 'value' => set_
value ( 'middle_name' ),
71:                                                 'maxlength' =>
'15',
72:                                                 'placeholder' =
> 'Marino'
73:                                             );
74:                                             (issetandnotempty ( form_error
( 'middle_name' ) ) ? $middle_name += array (
75:                                                 'class' => 'err
or'
76:                                             ) : '');
77:                                             echo form_input ( $middle_name
);
78:                                             ?>
79:                                             <?php
80:                                             if (issetandnotempty ( form_err
or ( 'middle_name' ) )) {
81:                                                 echo form_error ( 'midd
le_name', '<small class="error">', '</small>' );
82:                                             }
83:                                             ?>
84:                                     </div>
85:                                     <div class="small-4 columns">
86:                                         <label
87:                                             <?php echo (issetandnot
empty ( form_error ( 'last_name' ) ) ? 'class="error"' : '')?>>
88:                                             Last Name </label>
89:                                             <?php
90:                                             $last_name = array (
91:                                                 'type' => 'text
',
92:                                                 'name' => 'last
_name',
93:                                                 'id' => 'last_n
ame',
94:                                                 'value' => set_
value ( 'last_name' ),
```


registration.php

```
<small class="error">', '</small>' );
185:                                     }
186:                                     ?>
187:                                     </div>
188:                                     </div>
189:                                     <div class="row">
190:                                         <div class="small-12 columns">
191:                                             <label
192:                                                 <?php echo (issetandnot
empty ( form_error ( 'password' ) ) ? 'class="error"' : '' )?>>
193:                                                 Password </label>
194:                                         <?php
195:                                         $password = array (
196:                                             'type' => 'password',
197:                                             'name' => 'password',
198:                                             'id' => 'password',
199:                                             'value' => set_value (
'password' ),
200:                                             'maxlength' => '65',
201:                                             'placeholder' => 'Passw
ord'
202:                                         );
203:                                         (issetandnotempty ( form_error ( 'passw
ord' ) ) ? $password += array (
204:                                             'class' => 'error'
205:                                         ) : '');
206:                                         echo form_input ( $password );
207:                                     ?>
208:                                     <?php
209:                                     if (issetandnotempty ( form_error ( 'pa
ssword' ) )) {
210:                                         echo form_error ( 'password', '
<small class="error">', '</small>' );
211:                                     }
212:                                     ?>
213:                                     </div>
214:                                     </div>
215:                                     <div class="row">
216:                                         <div class="small-12 columns">
217:                                             <label
218:                                                 <?php echo (issetandnot
empty ( form_error ( 'retyped_password' ) ) ? 'class="error"' : '' )?>>
219:                                                 Retyped Password </labe
l>
220:                                         <?php
221:                                         $retyped_password = array (
222:                                             'type' => 'password',
223:                                             'name' => 'retyped_pass
word',
224:                                             'id' => 'retyped_passwo
rd',
225:                                             'value' => set_value (
'retyped_password' ),
226:                                             'maxlength' => '65',
227:                                             'placeholder' => 'Retyp
ed Password'
228:                                         );
229:                                         (issetandnotempty ( form_error ( 'retp
ed_password' ) ) ? $retyped_password += array (
230:                                             'class' => 'error'
231:                                         ) : '');
232:                                         echo form_input ( $retyped_password );
233:                                     ?>
```


registration.php

```
234:                                     <?php
235:                                     if (issetandnotempty ( form_error ( 're
typed_password' ) )) {
236:                                     echo form_error ( 'retyped_pass
word', '<small class="error">', '</small>' );
237:                                     }
238:                                     ?>
239:                                     </div>
240:                                 </div>
241:                                 <div class="row">
242:                                     <div class="small-12 columns">
243:                                         <label> Priviledge </label>
244:                                         <?php
245:                                         $priviledge_id = $priviledge_id;
246:                                         echo form_dropdown ( 'priviledge_id', $
priviledge_id );
247:                                     ?></div>
248:                                 </div>
249:
250:
251:                                 <div class="row">
252:                                     <div class="small-12 columns">
253:                                         <label> Active </label>
254:                                         <div class="row">
255:                                             <div class="small-6 col
umns">
256:                                                 <label>
257:                                                 <?php
258:                                                 $active = array
(
259:
260:                                     'name' => 'is_active',
261:                                     'id' => 'is_active',
262:                                     'value' => 'active',
263:                                     'checked' => TRUE,
264:                                     'style' => 'margin:10px'
265:                                     );
266:                                     echo form_radio
( $active ) . ' active';
267:                                     ?>
268:                                     </label>
269:                                 </div>
270:                                     <div class="small-6 col
umns">
271:                                         <label>
272:                                         <?php
273:                                         $inactive = arr
ay (
274:                                     'name' => 'is_active',
275:                                     'id' => 'is_active',
276:                                     'value' => 'inactive',
277:                                     'style' => 'margin:10px'
278:                                     );
279:                                     echo form_radio
```

registration.php

```
( $inactive ) . ' inactive';
280:
281:                                     ?>
282:                                     </label>
283:                                     </div>
284:                                     </div>
285: </div>
286: <div class="row">
287:     <div class="small-12 columns"><?php
288:         $submit = array (
289:             'name' => 'register',
290:             'class' => 'tiny button
radius span-column'
291:         );
292:         echo form_submit ( $submit, 'register'
);
293:     ?></div>
294: </div>
295: <?php
296: echo form_close ();
297: ?>
298: </div>
299: </div>
300: </div>
301: </div>
302:
303:
304:
305:
306: <?php
307: echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
308: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
309: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
310: ?>
311: <script>
312: $(document).foundation();
313: </script>
314:
315: </body>
316: </html>
```


update.php

```
98:                                     <?php
99:                                     $last_name = array (
100:                                     'type' => 'text
',
101:                                     'name' => 'last
_name',
102:                                     'id' => 'last_n
ame',
103:                                     'value' => (iss
etandnotempty ( form_error ( 'last_name' ) ) ? set_value ( 'last_name' ) : $last_name),
104:                                     'maxlength' =>
'15',
105:                                     'placeholder' =
> 'dela Cruz'
106:                                     );
107:                                     (issetandnotempty ( form_error
( 'last_name' ) ) ? $last_name += array (
108:                                     'class' => 'err
or'
109:                                     ) : '');
110:                                     echo form_input ( $last_name );
111:                                     ?>
112:                                     <?php
113:                                     if (issetandnotempty ( form_err
or ( 'last_name' ) )) {
114:                                     echo form_error ( 'last
_name', '<small class="error">', '</small>' );
115:                                     }
116:                                     ?>
117:                                     </div>
118:                                     </div>
119:
120:                                     <div class="row">
121:                                     <div class="small-12 columns">
122:                                     <label
123:                                     <?php echo (issetandnot
empty ( form_error ( 'cellphone_number' ) ) ? 'class="error"' : '')?>
124:                                     Cellphone Number </labe
l>
125:                                     <?php
126:                                     $cellphone_number = array (
127:                                     'type' => 'text
',
128:                                     'name' => 'cell
phone_number',
129:                                     'id' => 'cellph
one_number',
130:                                     'value' => (iss
etandnotempty ( form_error ( 'cellphone_number' ) ) ? set_value ( 'cellphone_number' ) : $cell
phone_number),
131:                                     'maxlength' =>
'11',
132:                                     'placeholder' =
> '09XXXXXXXXX'
133:                                     );
134:                                     (issetandnotempty ( form_error
( 'cellphone_number' ) ) ? $cellphone_number += array (
135:                                     'class' => 'err
or'
136:                                     ) : '');
137:                                     echo form_input ( $cellphone_nu
mber );
138:                                     ?>
```


update.php

```
232:
233:
234:
235:     <?php
236:     echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
237:     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
238:     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
239:     ?>
240:     <script>
241:     $(document).foundation();
242:     </script>
243:
244: </body>
245: </html>
```


login.php

```
1: <html>
2: <head>
3: <title><?php echo $this->session->userdata ( 'company_acronym' ) . ' - Login'?></title>
4: <?php
5: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/normalize.css' );
6: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/foundation.min.css' );
7: echo css_tag ( 'resources/foundation-icons/foundation-icons/foundation-icons.css' );
8: echo css_tag ( 'resources/custom/css/sizing.css' );
9: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/modernizr.js' );
10: ?>
11: </head>
12: <body>
13:     <?php
14:     $data [ 'page_set' ] = 'login';
15:     $this->load->view ( 'navigation/navigationbar', $data );
16:     ?>
17:     <div class="row">
18:     <?php echo br(1);?>
19:     </div>
20:     <div class="row">
21:         <div class="small-10 small-centered columns">
22:             <div class="row">
23:                 <div class="small-12 large-8 columns">
24:                     <h1>Welcome Back!</h1>
25:                     <p>Thank you very much for the care and support
you have provided
26:                         to our company and its members in the p
ast. We are very honored to
27:                         have you in our team.</p>
28:                     <p>If you have any feedbacks/ suggestions with
this system of ours
29:                         feel free to contact your superior. Eve
ry opinion of yours matter
30:                         to us.</p>
31:
32:                     </div>
33:                     <div class="small-12 large-4 columns">
34:                         <div class="panel radius">
35:                             <?php
36:                             echo form_open ( 'login/process_login'
);
37:                             ?>
38:                             <div class="row">
39:                                 <div class="small-12 columns">
40:                                     <?php
41:                                     $username = array (
42:                                         'type'
=> 'text',
43:                                         'name'
=> 'username',
44:                                         'id' =>
'username',
45:                                         'maxlen'
=> '20',
46:                                         'placeh
older' => 'username',
47:                                         'class'
=> 'radius'
48:                                     );
49:                                     echo form_input ( $user
name );
50:                                     ?>
51:                                 </div>
```

login.php

```
52:                                     </div>
53:                                     <div class="row">
54:                                         <div class="small-12 columns">
55:                                             <div class="row">
56:                                                 <div class="sma
11-9 columns">
57:                                                                 <?php
58:                                                                 $passwo
rd = array (
59:     'type' => 'password',
60:     'name' => 'password',
61:     'id' => 'password',
62:     'maxlength' => '20',
63:     'placeholder' => 'password',
64:     'class' => 'radius'
65:                                     );
66:                                     echo fo
rm_input ( $password );
67:                                     ?>
68:                                     </div>
69:                                     <div class="sma
11-3 columns">
70:                                                                 <?php
71:                                                                 $submit
= array (
72:     'name' => 'login',
73:     'class' => 'tiny button radius span-column'
74:                                     );
75:                                     echo fo
rm_submit ( $submit, 'login' );
76:                                     ?>
77:                                     </div>
78:                                     </div>
79:                                     </div>
80:                                     </div>
81:                                     <?php
82:                                     echo form_close ();
83:                                     ?>
84:                                     </div>
85:                                     </div>
86:                                     </div>
87:                                     </div>
88:                                     </div>
89:
90:
91:     <?php
92:     echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
93:     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
94:     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
95:     ?>
96:     <script>
97:     $(document).foundation();
98:     </script>
99:
100: </body>
```

login.php

101: </html>

history.php

```
1: <html>
2: <head>
3: <title><?php echo $this->session->userdata ( 'company_acronym' ) . ' - Messaging'?></ti
tle>
4: <?php
5: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/normalize.css' );
6: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/foundation.min.css' );
7: echo css_tag ( 'resources/foundation-icons/foundation-icons/foundation-icons.css' );
8: echo css_tag ( 'resources/custom/css/sizing.css' );
9: echo css_tag ( 'resources/datatables-foundation/dataTables.foundation.css' );
10: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/modernizr.js' );
11: ?>
12:
13:
14: </head>
15: <body>
16:     <?php
17:         $data [ 'page_set' ] = 'system';
18:         $this->load->view ( 'navigation/navigationbar_user', $data );
19:         ?>
20:         <div class="row">
21:             <?php echo br(1);?>
22:         </div>
23:         <div class="row">
24:             <div class="small-10 small-centered columns">
25:                 <h3>
26:                     <strong>sms history</strong>
27:                 </h3>
28:                 <p>log of all the messages that came in and out of the system.<
/p>
29:             </div>
30:         </div>
31:         <div class="row">
32:             <div class="small-10 small-centered columns">
33:                 <table>
34:                     <thead>
35:                         <tr>
36:                             <th>message</th>
37:                             <th>sender</th>
38:                             <th>receiver</th>
39:                             <th>communication</th>
40:                             <th>message</th>
41:                             <th>timestamp</th>
42:                         </tr>
43:                     </thead>
44:                     <tbody>
45:                         <?php
46:                         foreach ( $message_list as $message_element ) {
47:                             ?>
48:                             <tr>
49:                                 <td><?php echo $message_element [ 'messa
ge' ]; ?></td>
50:                                 <td><?php echo $message_element [ 'sende
r_number' ]; ?></td>
51:                                 <td><?php echo $message_element [ 'recei
ver_number' ]; ?></td>
52:                                 <td><?php echo $message_element [ 'commu
nication_type' ]; ?></td>
53:                                 <td><?php echo $message_element [ 'messa
ge_type' ]; ?></td>
54:                                 <td><?php echo $message_element [ 'date'
]; ?></td>
55:                             </tr>
```

history.php

```
56:                                     <?php
57:                                     }
58:                                     ?>
59:                                     </tbody>
60:                                 </table>
61:                    </div>
62: </div>
63: <?php
64: echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
65: // echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/jquery.js' );
66: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
67: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
68: echo js_tag ( 'resources/datatables-foundation/dataTables.foundation.min.js' );
69: ?>
70: <script type="text/javascript">
71: $(document).foundation();
72: $(document).ready( function () {
73:     $('table').DataTable();
74: } );
75: </script>
76: </body>
77: </html>
```


advisory.php

```
49:
50:     <?php
}
51:
echo form_checkbox ( 'home_region[]', $home_region_element ['id'], TRUE );
52:
echo $home_region_element ['name'];
53:
echo br ();
54:
$row ++;
55:
if (($row % 6) == 0 || $row == $size) {
56:     ?>
57:
58:         </div>
59:     <?php
}
60:
61:     }
62:     </div>
63: </div>
64: <dd class="accordion-navigation
65: ">
66:     <a href="#panel2b" clas
s="text-center"><strong><i
67:         class="
fi-torsos-all"></i> membership</strong></a>
68:     <div id="panel2b" class
="content">
69:         <div class="row
">
70:
71:             <?php
0;
72:             $row =
73:             $size =
74:             foreach
( $membership as $membership_element ) {
75:
76:         <div class="small-6 columns">
77:
78:     <?php
}
79:
echo form_checkbox ( 'membership[]', $membership_element ['id'], TRUE );
80:
echo $membership_element ['name'];
81:
echo br ();
82:
$row ++;
83:
```

advisory.php

```
if (($row % 3) == 0 || $row == $size) {
    84:     ?>
    85:         </div>
    86:
    87:     <?php
}
    88:                                     }
    89:                                     ?>
    90:                                     </div>
    91:                                     </div>
    92:                                     </dd>
    93:                                     <dd class="accordion-navigation
">
    94:                                     <a href="#panel3b" clas
s="text-center"><strong><i
    95:                                     class="
fi-shopping-cart"></i> payment mode</strong></a>
    96:                                     <div id="panel3b" class
="content">
    97:                                     <div class="row
">
    98:                                     <div cl
ass="small-12 columns">
    99:
    <?php
    100:
    $row = 0;
    101:
    $size = count ( $payment_type );
    102:
    foreach ( $payment_type as $payment_type_element ) {
    103:
    104:         if ((( $row + 1 ) % 3) == 1) {
    105:             ?>
    106:                 <div class="small-6 columns">
    107:                     <?php
    108:                 }
    109:         echo form_checkbox ( 'payment_type[]', $payment_type_element ['id'], TRUE );
    110:         echo $payment_type_element ['name'];
    111:         echo br ();
    112:         if (($row % 3) == 0 || $row == $size) {
    113:             ?>
    114:                 </div>
    115:             <?php
    116:         }
    }
    117:
```


advisory.php

```
?>
118:                                                                                               </div>
119:                                                                                               </div>
120:                                                                                               </div>
121:                                                                                               </dd>
122:                                                                                               </dl>
123:                                                                                               </div>
124: <?php echo br();?>
125: <div class="row">
126:     <div class="small-12 columns">
127:         <?php
128:             $advisory = array (
129:                 'name' => 'advisory',
130:                 'id' => 'advisory',
131:                 'maxlength' =>
132:                     '135',
133:                 'rows' => '3',
134:                 'placeholder' =>
135:                     'Advisory'
136:             );
137:             echo form_textarea ( $advisory );
138:         </div>
139:     <div class="row">
140:         <div class="small-12 columns"><?php
141:             $submit = array (
142:                 'name' => 'refer',
143:                 'class' => 'tiny button'
144:             );
145:             echo form_submit ( $submit, 'refer' );
146:         ?></div>
147:     </div>
148: <?php
149:     echo form_close ();
150: ?>
151: </div>
152: </div>
153: </div>
154: </div>
155:
156:
157:
158:
159: <?php
160: echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
161: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
162: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
163: ?>
164: <script>
165: $(document).foundation();
166: </script>
167:
168: </body>
169: </html>
```


service.php

```
104:                                     echo form_close ();
105:                                     ?>
106:                                     </div>
107:                                 </div>
108:         </div>
109: </div>
110:
111:
112:
113:
114: <?php
115: echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
116: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
117: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
118: ?>
119: <script>
120: $(document).foundation();
121: </script>
122:
123: </body>
124: </html>
```

referral.php

```
1: <html>
2: <head>
3: <title><?php echo $this->session->userdata ( 'company_acronym' ) . ' - Messaging'?></ti
tle>
4: <?php
5: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/normalize.css' );
6: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/foundation.min.css' );
7: echo css_tag ( 'resources/foundation-icons/foundation-icons/foundation-icons.css' );
8: echo css_tag ( 'resources/custom/css/sizing.css' );
9: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/modernizr.js' );
10: ?>
11: </head>
12: <body>
13:     <?php
14:         $data ['page_set'] = 'customers';
15:         $this->load->view ( 'navigation/navigationbar_user', $data );
16:         ?>
17:         <div class="row">
18:             <?php echo br(1);?>
19:         </div>
20:         <div class="row">
21:             <div class="small-12 small-centered columns">
22:                 <div class="row">
23:                     <div class="large-3 small-12 columns text-justify">
24:                         <h4>
25:                             <strong>customer referrals</strong>
26:                             <p>provide welfare services offered by
our partner companies to
27:                                     our customer.</p>
28:                         </h4>
29:                     </div>
30:                     <div class="large-9 small-12 columns panel radius">
31:                         <?php
32:                             echo form_open ( 'messaging/process_referral_ge
t_service' );
33:                             ?>
34:                             <div class="row">
35:                                 <div class="small-12 columns">
36:                                     <label
37:                                         <?php echo (issetandnot
empty ( form_error ( '' ) ) ? 'class="error" : '' )?>>
38:                                         PIN Number </label>
39:                                         <?php
40:                                         $pin = array (
41:                                             'type' => 'text
',
42:                                             'name' => 'pin'
',
43:                                             'id' => 'pin',
44:                                             'value' => set_
value ( 'pin' ),
45:                                             'maxlength' =>
'15',
46:                                             'placeholder' =
> 'PIN'
47:                                         );
48:                                         (issetandnotempty ( form_error
( 'pin' ) ) ? $pin += array (
49:                                             'class' => 'err
or'
50:                                             ) : '');
51:                                         echo form_input ( $pin );
52:                                         ?>
```

referral.php

```
53:                                     <?php
54:                                     if (issetandnotempty ( form_err
or ( 'pin' ) )) {
55:                                     echo form_error ( 'pin'
, '<small class="error">', '</small>' );
56:                                     }
57:                                     ?>
58:                                     </div>
59:                                     </div>
60:                                     <div class="row">
61:                                         <div class="small-12 columns">
62:                                             <div class="row collapse prefix
-radius">
63:                                                 <div class="small-3 col
umns">
64:                                                     <span class="pr
efix">partner company</span>
65:                                                 </div>
66:                                                 <div class="small-9 col
umns">
67:                                                     <?php
68:                                                     $partner_compan
y_id = $partner_company_id;
69:                                                     echo form_dropd
own ( 'partner_company_id', $partner_company_id );
70:                                                     ?>
71:                                                 </div>
72:                                             </div>
73:                                         </div>
74:                                     </div>
75:                                     <div class="row">
76:                                         <div class="small-12 columns"><?php
77:                                         $submit = array (
78:                                             'name' => 'proceed',
79:                                             'class' => 'tiny button
radius span-column'
80:                                         );
81:                                         echo form_submit ( $submit, 'select ser
vices' );
82:                                         ?></div>
83:                                     </div>
84:                                     <?php
85:                                     echo form_close ();
86:                                     ?>
87:                                     </div>
88:                                     </div>
89:                                     </div>
90:                                     </div>
91:
92:
93:
94:
95:                                     <?php
96:                                     echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
97:                                     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
98:                                     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
99:                                     ?>
100:                                     <script>
101:                                     $(document).foundation();
102:                                     </script>
103:
104: </body>
105: </html>
```

list.php

```
1: <html>
2: <head>
3: <title><?php echo $this->session->userdata ( 'company_acronym' ) . ' - Member'?></title
>
4: <?php
5: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/normalize.css' );
6: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/foundation.min.css' );
7: echo css_tag ( 'resources/foundation-icons/foundation-icons/foundation-icons.css' );
8: echo css_tag ( 'resources/custom/css/sizing.css' );
9: echo css_tag ( 'resources/datatables-foundation/dataTables.foundation.css' );
10: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/modernizr.js' );
11: ?>
12: </head>
13: <body>
14:     <?php
15:     $data ['page_set'] = 'customers';
16:     $this->load->view ( 'navigation/navigationbar_user', $data );
17:     ?>
18:     <div class="row">
19:     <?php echo br(1);?>
20:     </div>
21:     <div class="row">
22:         <div class="small-10 small-centered columns">
23:             <h3>
24:                 <strong>customer profiles</strong>
25:             </h3>
26:             <p>a collation of a quick summary about our customers. you may
do
27:                 queries with the list through the search field.</p>
28:         </div>
29:     </div>
30:     <div class="row">
31:         <div class="small-10 small-centered columns">
32:             <table>
33:                 <thead>
34:                     <tr>
35:                         <th></th>
36:                         <th>member's pin</th>
37:                         <th>name</th>
38:                         <th>membership</th>
39:                         <th>payment mode</th>
40:                         <th>notifiable</th>
41:                     </tr>
42:                 </thead>
43:                 <tbody>
44:                     <?php
45:                     foreach ( $member_list as $member_element ) {
46:                         ?>
47:                         <tr>
48:                             <td><?php echo anchor ( $this->config->
base_url ( 'index.php/member/update/' . $member_element ['pin'] ), '<i class="fi-widget"></i>'
);?></td>
49:                             <td><?php echo $member_element ['pin'];
?></td>
50:                             <td><?php echo $member_element ['name']
; ?></td>
51:                             <td><?php echo $member_element ['member
ship_type']; ?></td>
52:                             <td><?php echo $member_element ['paymen
t_type']; ?></td>
53:                             <td
54:                                 bgcolor="<?php echo ( $member_e
lement ['is_notifiable'] == 'notifiable' ? '#77DD77' : '#FDFD96' ) ?>"><strong><?php echo anch
```

list.php

```
or ( $this->config->base_url ( 'index.php/member/' . ($member_element ['is_notifiable'] === 'n
otifiable' ? 'unnotify' : 'notify') . '/' . $member_element ['pin'] ), ($member_element ['is_n
otifiable'] === 'notifiable' ? '<i class="fi-minus"></i> unnotify' : '<i class="fi-plus"></i>
notify') );?></strong></td>
55:                                     </tr>
56:                                     <?php
57:                                     }
58:                                     ?>
59:                                     </tbody>
60:                                 </table>
61:         </div>
62:     </div>
63:     <?php
64:     echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
65:     // echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/jquery.js' );
66:     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
67:     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
68:     echo js_tag ( 'resources/datatables-foundation/dataTables.foundation.min.js' );
69:     ?>
70:     <script type="text/javascript">
71:     $(document).foundation();
72:     $(document).ready( function () {
73:         $('table').DataTable();
74:     } );
75:     </script>
76: </body>
77: </html>
```


registration.php

```
le_name',
    98:                                     'id' => 'middle
_name',
    99:                                     'value' => set_
value ( 'middle_name' ),
    100:                                    'maxlength' =>
'15',
    101:                                    'placeholder' =
> 'Marino'
    102:                                    );
    103:                                    (issetandnotempty ( form_error
    104:                                    'class' => 'err
or'
    105:                                    ) : '');
    106:                                    echo form_input ( $middle_name
);
    107:                                    ?>
    108:                                    <?php
    109:                                    if (issetandnotempty ( form_err
or ( 'middle_name' ) ) ) {
    110:                                        echo form_error ( 'midd
le_name', '<small class="error">', '</small>' );
    111:                                        }
    112:                                        ?>
    113:                                        </div>
    114:                                    <div class="small-4 columns">
    115:
    116:                                        <label
    117:                                            <?php echo (issetandnot
empty ( form_error ( 'last_name' ) ) ? 'class="error"' : '' )?>>
    118:                                                Last Name </label>
    119:                                        <?php
    120:                                        $last_name = array (
    121:                                            'type' => 'text
',
    122:                                            'name' => 'last
_name',
    123:                                            'id' => 'last_n
ame',
    124:                                            'value' => set_
value ( 'last_name' ),
    125:                                            'maxlength' =>
'15',
    126:                                            'placeholder' =
> 'de la Cruz'
    127:                                            );
    128:                                            (issetandnotempty ( form_error
    129:                                            'class' => 'err
or'
    130:                                            ) : '');
    131:                                            echo form_input ( $last_name );
    132:                                            ?>
    133:                                            <?php
    134:                                            if (issetandnotempty ( form_err
or ( 'last_name' ) ) ) {
    135:                                                echo form_error ( 'last
_name', '<small class="error">', '</small>' );
    136:                                                }
    137:                                                ?>
    138:                                            </div>
    139:                                        </div>
```

registration.php

```
140:
141:         <div class="row">
142:             <div class="small-12 columns">
143:
144:                 <label
145:                     <?php echo (issetandnot
empty ( form_error ( 'cellphone_number' ) ) ? 'class="error"' : '' )?>>
146:                     Cellphone Number </label>
147:
148:                     <?php
149:                         $cellphone_number = array (
150:                             'type' => 'text
',
151:                             'name' => 'cell
phone_number',
152:                             'id' => 'cellph
one_number',
153:                             'value' => set_
value ( 'cellphone_number' ),
154:                             'maxlength' =>
'11',
155:                             'placeholder' =
> '09XXXXXXXXXX'
156:                         );
157:                     (issetandnotempty ( form_error
( 'cellphone_number' ) ) ? $cellphone_number += array (
158:                         'class' => 'err
or'
159:                     ) : '');
160:                     echo form_input ( $cellphone_nu
mber );
161:                 <?>
162:                 <?php
163:                     if (issetandnotempty ( form_err
or ( 'cellphone_number' ) ) ) {
164:                         echo form_error ( 'cell
phone_number', '<small class="error">', '</small>' );
165:                     }
166:                 <?>
167:             </div>
168:         </div>
169:         <div class="row">
170:             <div class="small-12 columns">
171:
172:                 <label
173:                     <?php echo (issetandnot
empty ( form_error ( 'email_address' ) ) ? 'class="error"' : '' )?>>
174:                     Email Address </label>
175:
176:                     <?php
177:                         $email_address = array (
178:                             'type' => 'text
',
179:                             'name' => 'emai
l_address',
180:                             'id' => 'email_
address',
181:                             'value' => set_
value ( 'email_address' ),
182:                             'maxlength' =>
'255',
183:                             'placeholder' =
> 'username@email.com'
184:                         );
```


registration.php

```
271:                                     <div class="row">
272:                                         <div class="small-12 columns"><?php
273:                                             $submit = array (
274:                                                 'name' => 'register',
275:                                                 'class' => 'tiny button
radius span-column'
276:                                             );
277:                                         echo form_submit ( $submit, 'register'
);
278:                                     ?></div>
279:                                     </div>
280:                                     <?php
281:                                     echo form_close ();
282:                                     ?>
283:                                     </div>
284:                                     </div>
285:                                     </div>
286:                                     </div>
287:
288:
289:
290:
291:     <?php
292:     echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
293:     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
294:     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
295:     ?>
296:     <script>
297:     $(document).foundation();
298:     </script>
299:
300: </body>
301: </html>
```

upload.php

```
1: <html>
2: <head>
3: <title><?php echo $this->session->userdata ( 'company_acronym' ) . ' - Member'?></title
>
4: <?php
5: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/normalize.css' );
6: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/foundation.min.css' );
7: echo css_tag ( 'resources/foundation-icons/foundation-icons/foundation-icons.css' );
8: echo css_tag ( 'resources/custom/css/sizing.css' );
9: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/modernizr.js' );
10: ?>
11: </head>
12: <body>
13:     <?php
14:         $data ['page_set'] = 'customers';
15:         $this->load->view ( 'navigation/navigationbar_user', $data );
16:         ?>
17:         <div class="row">
18:             <?php echo br(1);?>
19:         </div>
20:         <div class="row">
21:             <div class="small-12 small-centered columns">
22:                 <div class="row">
23:                     <div class="large-3 small-12 columns text-justify">
24:                         <h4>
25:                             <strong>customer upload</strong>
26:                             <p>register new customers via csv file
generated by preexisting
27:                                 infrastructures.</p>
28:                         </h4>
29:                     </div>
30:                     <div class="large-9 small-12 columns panel radius">
31:                         <?php
32:                         echo form_open_multipart ( 'member/process_uplo
ad' );
33:                         ?>
34:                         <div class="row">
35:                             <div class="small-12 columns">
36:                                 <?php
37:                                 $mhc_members_archive_file = array (
38:                                     'type' => 'file',
39:                                     'name' => 'userfile',
40:                                     'id' => 'userfile',
41:                                     'size' => '50',
42:                                     'style' => 'width:50%'
43:                                 );
44:                                 echo form_input ( $mhc_members_archive
_file );
45:                                 ?>
46:                             </div>
47:                         </div>
48:                         <div class="row">
49:                             <div class="small-12 columns"><?php
50:                             $submit = array (
51:                                 'name' => 'upload',
52:                                 'class' => 'tiny button
radius span-column'
53:                             );
54:                             echo form_submit ( $submit, 'upload' );
55:                             ?></div>
56:                         </div>
57:                         <?php
58:                         echo form_close ();
```


upload.php

```
59:                                     ?>
60:                                     </div>
61:                                 </div>
62:                            </div>
63:                    </div>
64:
65:
66:
67:
68:    <?php
69:    echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
70:    echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
71:    echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
72:    ?>
73:    <script>
74:    $(document).foundation();
75:    </script>
76:
77: </body>
78: </html>
```


update.php

```
141:                                     <div class="row">
142:                                         <div class="small-12 columns">
143:
144:                                             <label
145: <?php echo (issetandnot
empty ( form_error ( 'cellphone_number' ) ) ? 'class="error"' : '' )?>>
146:                                             Cellphone Number </label>
147:
148:                                             <?php
149: $cellphone_number = array (
150:                                     'type' => 'text
',
151:                                     'name' => 'cell
phone_number',
152:                                     'id' => 'cellph
one_number',
153:                                     'value' => (iss
etandnotempty ( form_error ( 'cellphone_number' ) ) ? set_value ( 'cellphone_number' ) : $cell
phone_number),
154:                                     'maxlength' =>
'11',
155:                                     'placeholder' =
> '09XXXXXXXXXX'
156:                                     );
157: (issetandnotempty ( form_error
( 'cellphone_number' ) ) ? $cellphone_number += array (
158:                                     'class' => 'err
or'
159:                                     ) : '');
160: echo form_input ( $cellphone_nu
mber );
161:
162: <?php
163: if (issetandnotempty ( form_err
or ( 'cellphone_number' ) ) ) {
164:                                     echo form_error ( 'cell
phone_number', '<small class="error">', '</small>' );
165:                                     }
166:                                     ?>
167:                                     </div>
168:                                     <div class="row">
169:                                         <div class="small-12 columns">
170:
171:                                             <label
172: <?php echo (issetandnot
empty ( form_error ( 'email_address' ) ) ? 'class="error"' : '' )?>>
173:                                             Email Address </label>
174:
175:                                             <?php
176: $email_address = array (
177:                                     'type' => 'text
',
178:                                     'name' => 'emai
l_address',
179:                                     'id' => 'email_
address',
180:                                     'value' => (iss
etandnotempty ( form_error ( 'email_address' ) ) ? set_value ( 'email_addr
ess'),
181:                                     'maxlength' =>
'255',
182:                                     'placeholder' =
> 'username@email.com'
```


update.php

```
umns">
227:                                     <span class="pr
efix">payment</span>
228:                                     </div>
229:                                     <div class="small-8 col
umns">
230:                                     <?php
231:                                     $payment_mode =
$payment_mode;
232:                                     echo form_dropd
own ( 'payment_mode', $payment_mode, $selected_payment_mode );
233:                                     ?>
234:                                     </div>
235:                                     </div>
236:                                     </div>
237:                                     </div>
238:                                     <div class="row">
239:                                         <div class="small-12 columns">
240:                                             <div class="row">
241:                                                 <div class="small-6 col
umns">
242:                                                     <label>
243:                                                     <?php
244:                                                     $notifiable = a
rray (
245: 'name' => 'is_notifiable',
246: 'id' => 'is_notifiable',
247: 'value' => 'notifiable',
248: 'checked' => ($is_notifiable === '1' ? TRUE : FALSE),
249: 'style' => 'margin:10px'
250:                                     );
251:                                     echo form_radio
( $notifiable ) . ' notifiable';
252:                                     ?>
253:                                     </label>
254:                                     </div>
255:                                     <div class="small-6 col
umns">
256:                                         <label>
257:                                         <?php
258:                                         $non_notifiable
= array (
259: 'name' => 'is_notifiable',
260: 'id' => 'is_notifiable',
261: 'value' => 'non_notifiable',
262: 'checked' => ($is_notifiable === '0' ? TRUE : FALSE),
263: 'style' => 'margin:10px'
264:                                     );
265:                                     echo form_radio
( $non_notifiable ) . ' non-notifiable';
266:                                     ?>
267:                                     </label>
268:                                     </div>
```

update.php

```
269:                                     </div>
270:                                     </div>
271: </div>
272: <div class="row">
273:     <div class="small-12 columns"><?php
274:         $submit = array (
275:             'name' => 'register',
276:             'class' => 'tiny button
radius span-column'
277:         );
278:         echo form_submit ( $submit, 'register'
);
279:     ?></div>
280: </div>
281: <?php
282: echo form_close ();
283: ?>
284: </div>
285: </div>
286: </div>
287: </div>
288:
289:
290:
291:
292: <?php
293: echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
294: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
295: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
296: ?>
297: <script>
298: $(document).foundation();
299: </script>
300:
301: </body>
302: </html>
```


navigationbar_user.php

```
1: <nav class="top-bar" data-topbar role="navigation">
2:     <ul class="title-area">
3:         <li class="name">
4:             <h1>
5:                 <?php
6:                     echo anchor_popup ( 'http://' . $this->session->userdata ( 'com
pany_site' ), $this->session->userdata ( 'company_acronym' ) );
7:                 ?>
8:             </h1>
9:         </li>
10:        <li class="toggle-topbar menu-icon"><a href="#"><span></span></a></li>
11:    </ul>
12:
13:    <section class="top-bar-section">
14:        <ul class="left">
15:            <li
16:                class="<?php if ($page_set == 'customers') { echo 'acti
ve'; }?> has-dropdown">
17:                <?php
18:                echo anchor ( 'dashboard', '<i class="fi-torsos-all"></
i> customers' );
19:                ?>
20:                <ul class="dropdown">
21:                    <li><?php
22:                        echo anchor ( 'dashboard/profiles', '<i class="
fi-torsos-female-male"></i> profiles' );
23:                    ?></li>
24:                    <li><?php
25:                        echo anchor ( 'dashboard/messaging', '<i class=
"fi-social-tumblr"></i> messaging' );
26:                    ?></li>
27:                </ul>
28:            </li>
29:            <li
30:                class="<?php if ($page_set == 'companies') { echo 'acti
ve'; }?> has-dropdown">
31:                <?php
32:                echo anchor ( 'dashboard', '<i class="fi-home"></i> com
panies' );
33:                ?>
34:                <ul class="dropdown">
35:                    <li>
36:                        <?php
37:                        echo anchor ( 'dashboard/companies', '<i class=
"fi-align-left"></i> companies' );
38:                    ?>
39:                    </li>
40:                </ul>
41:            </li>
42:            <li
43:                class="<?php if ($page_set == 'system') { echo 'active'
; }?> has-dropdown">
44:                <?php
45:                echo anchor ( 'dashboard', '<i class="fi-book-bookmark"
></i> system' );
46:                ?>
47:                <ul class="dropdown">
48:                    <li>
49:                        <?php
50:                        echo anchor ( 'dashboard/accounts', '<i class="
fi-results-demographics"></i> accounts' );
51:                    ?>
52:                    </li>
```

navigationbar_user.php

```
53:                                     </ul>
54:                                     </li>
55:     </ul>
56:
57:
58:     <ul class="right">
59:         <li class="has-dropdown">
60:             <?php
61:                 echo anchor ( 'user/update/' . $this->session->userdata ( 'user
_id' ), '<i class="fi-torso"></i> ' . $this->session->userdata ( 'first_name' ) );
62:                 ?>
63:                 <ul class="dropdown">
64:                     <li><?php
65:                         echo anchor ( 'user/update/' . $this->session->
userdata ( 'user_id' ), '<i class="fi-torso"></i> ' . $this->session->userdata ( 'priviledge_n
ame' ) );
66:                         ?></li>
67:                         <li><?php
68:                             echo anchor ( 'user/update/' . $this->session->
userdata ( 'user_id' ), '<i class="fi-widget"></i> update' );
69:                             ?></li>
70:                             <li><?php
71:                                 echo anchor ( 'login/logout', '<i class="fi-wid
get"></i> logout' );
72:                                 ?></li>
73:                                     </ul>
74:                                     </li>
75:     </ul>
76: </section>
77: </nav>
```

navigationbar.php

```
1: <nav class="top-bar" data-topbar role="navigation">
2:     <ul class="title-area">
3:         <li class="name">
4:             <h1>
5:                 <?php
6:                     echo anchor_popup ( 'http://' . $this->session->userdata ( 'com
pany_site' ), $this->session->userdata ( 'company_acronym' ) );
7:                 ?>
8:             </h1>
9:         </li>
10:        <li class="toggle-topbar menu-icon"><a href="#"><span></span></a></li>
11:    </ul>
12:
13:    <section class="top-bar-section">
14:        <ul class="left">
15:            <li <?php if ( $page_set == 'home' ) { echo 'class="active"'; }?>
>
16:                <?php
17:                    echo anchor ( '', ' <i class="fi-home"></i> home' );
18:                ?>
19:            </li>
20:            <li <?php if ( $page_set == 'about' ) { echo 'class="active"'; }?
>>
21:                <?php
22:                    echo anchor ( '', ' <i class="fi-comment-quotes"></i> ab
out' );
23:                ?>
24:            </li>
25:        </ul>
26:
27:        <ul class="right">
28:            <li <?php if ( $page_set == 'login' ) { echo 'class="active"'; }?
>>
29:                <?php
30:                    echo anchor ( 'login', ' <i class="fi-torso"></i> login'
);
31:                ?>
32:            </li>
33:        </ul>
34:    </section>
35: </nav>
```


personnel.php

```
' . $partner_company_id, 'lists' );
56:
57:                                ?></strong>
                                <p>list personnel belonging to
<?php echo $partner_company_name?></p>
58:                                </div>
59:                                </div>
60:                                </div>
61:                                </div>
62:                                </div>
63:                                </div>
64:                                <?php
65:                                echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
66:                                echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
67:                                echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
68:                                ?>
69:                                <script>
70:                                $(document).foundation();
71:                                </script>
72:
73: </body>
74: </html>
```

partner.php

```
1: <html>
2: <head>
3: <title><?php echo $this->session->userdata ( 'company_acronym' ) . ' - Partner'?></titl
e>
4: <?php
5: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/normalize.css' );
6: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/foundation.min.css' );
7: echo css_tag ( 'resources/foundation-icons/foundation-icons/foundation-icons.css' );
8: echo css_tag ( 'resources/custom/css/sizing.css' );
9: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/modernizr.js' );
10: ?>
11: </head>
12: <body>
13:     <?php
14:         $data ['page_set'] = 'companies';
15:         $this->load->view ( 'navigation/navigationbar_user', $data );
16:         ?>
17:         <div class="row">
18:             <?php echo br(1);?>
19:         </div>
20:         <div class="row">
21:             <div class="small-10 small-centered columns">
22:                 <h3>
23:                     <strong><?php echo $partner_company_name?> details</str
ong>
24:                 </h3>
25:                 <p>provids a brief overview of how you could manage <?php echo
$partner_company_name?'s details.</p>
26:             </div>
27:         </div>
28:         <div class="row">
29:             <div class="small-10 small-centered columns">
30:                 <div class="row">
31:                     <div class="small-11 small-offset-1 columns">
32:                         <div class="row">
33:                             <div class="small-1 small-text-center c
olumns">
34:                                 <h1>
35:                                     <i class="fi-home"></i>
36:                                 </h1>
37:                             </div>
38:                             <div class="small-11 columns">
39:                                 <strong><?php
40:                                     echo anchor ( 'dashboard/servic
es/' . $partner_company_id, 'services' );
41:                                     ?></strong>
42:                                 <p>provides a summary of the se
rvices <?php echo $partner_company_name?> offers.</p>
43:                             </div>
44:                         </div>
45:                     <div class="row">
46:                         <div class="small-1 small-text-center c
olumns">
47:                             <h1>
48:                                 <i class="fi-list"></i>
49:                             </h1>
50:                         </div>
51:                         <div class="small-11 columns">
52:                             <strong><?php
53:                                 echo anchor ( 'dashboard/person
nel/' . $partner_company_id, 'personnel' );
54:                                 ?></strong>
55:                             <p>provides a summary of the <?>
```

partner.php

```
php echo $partner_company_name?'>'s personnel.</p>
56:                                     </div>
57:                                     </div>
58:                                     </div>
59:                                     </div>
60:                                     </div>
61: </div>
62: <?php
63: echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
64: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
65: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
66: ?>
67: <script>
68: $(document).foundation();
69: </script>
70:
71: </body>
72: </html>
```

dashboard.php

```
1: <html>
2: <head>
3: <title><?php
4: echo $this->session->userdata ( 'company_acronym' );
5: ?></title>
6: <?php
7: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/normalize.css' );
8: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/foundation.min.css' );
9: echo css_tag ( 'resources/foundation-icons/foundation-icons/foundation-icons.css' );
10: echo css_tag ( 'resources/custom/css/sizing.css' );
11: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/modernizr.js' );
12: ?>
13: </head>
14: <body>
15:     <?php
16:     $data ['page_set'] = 'dashboard';
17:     $this->load->view ( 'navigation/navigationbar_user', $data );
18:     ?>
19:     <div class="row">
20:     <?php echo br(1);?>
21:     </div>
22:     <div class="row">
23:         <div class="small-10 small-centered columns">
24:             <h3>
25:                 <strong>my capabilities</strong>
26:             </h3>
27:             <p>listed here are the things you may perform with the system.
as a
28:                 trusted employee of ours, we look forward to the action
s that you
29:                 will make that we hope is geared towards enhancing our
relationship
30:                 with our customers and our partner companies.</p>
31:         </div>
32:     </div>
33:     <div class="row">
34:         <div class="small-10 small-centered columns">
35:             <div class="row">
36:                 <div class="large-4 small-12 columns">
37:                     <ul class="pricing-table">
38:                         <li class="title">customers</li>
39:                         <li class="price"><h1>
40:                             <i class="fi-torsos-all
"></i>
41:                                 </h1></li>
42:                         <li class="description">have a new cust
omer? does one need an
43:                             update to their profile? or som
e message needs to be sent to
44:                             them?</li>
45:                         <li class="bullet-item">
46:                             <?php
47:                             echo anchor ( 'dashboard/profil
es', '<i class="fi-torsos-female-male"></i> profiles' );
48:                             ?>
49:                             </li>
50:                         <li class="bullet-item">
51:                             <?php
52:                             echo anchor ( 'dashboard/messag
ing', '<i class="fi-social-tumblr"></i> messaging' );
53:                             ?></li>
54:                         <li class="bullet-item"><?php echo $mem
ber_count.' member/s';?></li>
```


dashboard.php

```
55:                                     </ul>
56:                                 </div>
57:                                 <div class="large-4 small-12 columns">
58:                                     <ul class="pricing-table">
59:                                         <li class="title">companies</li>
60:                                         <li class="price"><h1>
61:                                             <i class="fi-home"></i>
62:                                         </h1></li>
63:                                         <li class="description">yey! a new part
ner company? does one has a
64:                                             new service we could offer to o
ur customers?</li>
65:                                         <li class="bullet-item">
66:                                             <?php
67:                                             echo anchor ( 'dashboard/compan
ies', '<i class="fi-align-left"></i> companies' );
68:                                             ?>
69:                                         </li>
70:                                         <li class="bullet-item"><?php echo $com
pany_count.' partner/s'?></li>
71:                                     </ul>
72:                                 </div>
73:                                 <div class="large-4 small-12 columns">
74:                                     <ul class="pricing-table">
75:                                         <li class="title"><large>system</large>
</li>
76:                                         <li class="price"><h1>
77:                                             <i class="fi-book-bookm
ark"></i>
78:                                         </h1></li>
79:                                         <li class="description">need to add a n
ew helping hand to our
80:                                             system? does one needs some twe
aking to be done?</li>
81:                                         <li class="bullet-item">
82:                                             <?php
83:                                             echo anchor ( 'dashboard/accoun
ts', '<i class="fi-results-demographics"></i> accounts' );
84:                                             ?>
85:                                         </li>
86:                                         <li class="bullet-item"><?php echo $use
r_count.' user/s'?></li>
87:                                     </ul>
88:                                 </div>
89:                             </div>
90:                         </div>
91:                 </div>
92:
93:
94:                 <?php
95:                 echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
96:                 echo js_tag ( 'resources/foundation-5.5.2.philhealth/vendor/fastclick.js' );
97:                 echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
98:                 ?>
99:                 <script>
100:                 $(document).foundation();
101:                 </script>
102:
103: </body>
104: </html>
```

messaging.php

```
1: <html>
2: <head>
3: <title><?php echo $this->session->userdata ( 'company_acronym' ) . ' - Messaging'?></ti
tle>
4: <?php
5: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/normalize.css' );
6: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/foundation.min.css' );
7: echo css_tag ( 'resources/foundation-icons/foundation-icons/foundation-icons.css' );
8: echo css_tag ( 'resources/custom/css/sizing.css' );
9: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/modernizr.js' );
10: ?>
11: </head>
12: <body>
13:     <?php
14:         $data ['page_set'] = 'customers';
15:         $this->load->view ( 'navigation/navigationbar_user', $data );
16:     ?>
17:     <div class="row">
18:         <?php echo br(1);?>
19:     </div>
20:     <div class="row">
21:         <div class="small-10 small-centered columns">
22:             <h3>
23:                 <strong>customer messaging</strong>
24:             </h3>
25:             <p>empowers you to generate referrals and broadcast advisories
to our
26:                 customers. you may see a full recap of all the messages
sent and
27:                 received by the system through the history tab.</p>
28:         </div>
29:     </div>
30:     <div class="row">
31:         <div class="small-10 small-centered columns">
32:             <div class="row">
33:                 <div class="small-11 small-offset-1 columns">
34:                     <div class="row">
35:                         <div class="small-1 small-text-center c
columns">
36:                             <h1>
37:                                 <i class="fi-link"></i>
38:                             </h1>
39:                         </div>
40:                         <div class="small-11 columns">
41:                             <strong><?php
42:                                 echo anchor ( 'messaging/referr
al', 'referral' );
43:                             ?></strong>
44:                             <p>generate a new referral for
a customer. referrals can be
45:                                 availed starting from t
he day of request generation.</p>
46:                         </div>
47:                     </div>
48:                 <div class="row">
49:                     <div class="small-1 small-text-center c
columns">
50:                         <h1>
51:                             <i class="fi-megaphone"
></i>
52:                         </h1>
53:                     </div>
54:                 <div class="small-11 columns">
```

messaging.php

```
55:                                     <strong><?php
56:                                     echo anchor ( 'messaging/adviso
ry', 'advisory' );
57:                                     ?></strong>
58:                                     <p>broadcast up-to-date and rel
evant advisories to our customers
59:                                     with varying home regio
ns, membership and payment types.</p>
60:                                     </div>
61:                                     </div>
62:                                     <div class="row">
63:                                     <div class="small-11 small-text-center c
olumns">
64:                                     <h1>
65:                                     <i class="fi-list"></i>
66:                                     </h1>
67:                                     </div>
68:                                     <div class="small-11 columns">
69:                                     <strong><?php
70:                                     echo anchor ( 'messaging/histor
y', 'history' );
71:                                     ?></strong>
72:                                     <p>have a clear overview of wha
t messages the system has sent and
73:                                     received through SMS in
the past.</p>
74:                                     </div>
75:                                     </div>
76:                                     </div>
77:                                     </div>
78:                                     </div>
79:                                     </div>
80:                                     <?php
81:                                     echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
82:                                     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
83:                                     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
84:                                     ?>
85:                                     <script>
86:                                     $(document).foundation();
87:                                     </script>
88:
89: </body>
90: </html>
```


companies.php

```
55:                                     </h1>
56:                                     </div>
57:                                     <div class="small-11 columns">
58:                                         <strong><?php
59:                                             echo anchor ( 'partner/lists',
'lists' );
60:                                             ?></strong>
61:                                         <p>view a tabulated archive of
all the partner companies
62:                                             registered into our sys
tem.</p>
63:                                     </div>
64:                                     </div>
65:                                     </div>
66:                                     </div>
67:                                     </div>
68:     </div>
69:     <?php
70:     echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
71:     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
72:     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
73:     ?>
74:     <script>
75:     $(document).foundation();
76:     </script>
77:
78: </body>
79: </html>
```

profiles.php

```
1: <html>
2: <head>
3: <title><?php echo $this->session->userdata ( 'company_acronym' ) . ' - Profiles'?></tit
le>
4: <?php
5: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/normalize.css' );
6: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/foundation.min.css' );
7: echo css_tag ( 'resources/foundation-icons/foundation-icons/foundation-icons.css' );
8: echo css_tag ( 'resources/custom/css/sizing.css' );
9: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/modernizr.js' );
10: ?>
11: </head>
12: <body>
13:     <?php
14:         $data ['page_set'] = 'customers';
15:         $this->load->view ( 'navigation/navigationbar_user', $data );
16:         ?>
17:         <div class="row">
18:             <?php echo br(1);?>
19:         </div>
20:         <div class="row">
21:             <div class="small-10 small-centered columns">
22:                 <h3>
23:                     <strong>customer profiles <?php echo '('.$member_count.
')'?></strong>
24:                 </h3>
25:                 <p>enables you to add a new customer into the system either man
ually
26:                     or through a csv file exported from preexisting infrast
ructures of
27:                     our company. you may also edit existing customers in ou
r database
28:                     through the lists provided.</p>
29:             </div>
30:         </div>
31:         <div class="row">
32:             <div class="small-10 small-centered columns">
33:                 <div class="row">
34:                     <div class="small-11 small-offset-1 columns">
35:                         <?php if(is_system_administrator()) {?>
36:                         <div class="row">
37:                             <div class="small-1 small-text-center c
olumns">
38:                                 <h1>
39:                                     <i class="fi-torso"></i
>
40:                                 </h1>
41:                             </div>
42:                         <div class="small-11 columns">
43:                             <strong><?php
44:                                 echo anchor ( 'member/register', 'regis
ter' );
45:                             ?></strong>
46:                             <p>add a new customer into the
system. manually input their
47:                                 details in a form provi
ded.</p>
48:                         </div>
49:                     </div>
50:                 <div class="row">
51:                     <div class="small-1 small-text-center c
olumns">
52:                         <h1>
```

profiles.php

```
53:                                     <i class="fi-upload"></i>
i>
54:                                     </h1>
55:                                     </div>
56:                                     <div class="small-11 columns">
57:                                         <strong><?php
58:                                         echo anchor ( 'member/upload', 'upload'
);
59:                                         ?></strong>
60:                                         <p>automatically add a new cust
omer/s into the system through a
61:                                         csv file formatted by o
ur preexisting infrastructures.</p>
62:                                     </div>
63:                                     </div>
64:                                     <?php }?>
65:                                     <div class="row">
66:                                         <div class="small-1 small-text-center c
olumns">
67:                                             <h1>
68:                                             <i class="fi-list"></i>
69:                                             </h1>
70:                                         </div>
71:                                         <div class="small-11 columns">
72:                                             <strong><?php
73:                                             echo anchor ( 'member/lists', '
lists' );
74:                                             ?></strong>
75:                                             <p>view a tabulated archive of
all the users registered into our
76:                                             system.</p>
77:                                         </div>
78:                                     </div>
79:                                     </div>
80:                                     </div>
81:                                     </div>
82:                                     </div>
83:                                     <?php
84:                                     echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
85:                                     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
86:                                     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
87:                                     ?>
88:                                     <script>
89:                                     $(document).foundation();
90:                                     </script>
91:
92: </body>
93: </html>
```


accounts.php

```
54:                                     </h1>
55:                                     </div>
56:                                     <div class="small-11 columns">
57:                                         <strong><?php
58:                                             echo anchor ( 'user/lists', 'li
sts' );
59:                                         ?></strong>
60:                                     <p>view a tabulated archive of
all the users of our system.</p>
61:                                     </div>
62:                                     </div>
63:                                     </div>
64:                                     </div>
65:                                     </div>
66:                                     </div>
67:                                     <?php
68:                                     echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
69:                                     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
70:                                     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
71:                                     ?>
72:                                     <script>
73:                                     $(document).foundation();
74:                                     </script>
75: </body>
76: </html>
```


services.php

```
. $partner_company_id, 'lists' );
56:
57:                                ?></strong>
p echo $partner_company_name?></p>                                <p>list services belonging <?ph
58:                                </div>
59:                                </div>
60:                                </div>
61:                                </div>
62:                                </div>
63:                                </div>
64:                                <?php
65:                                echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
66:                                echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
67:                                echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
68:                                ?>
69:                                <script>
70:                                $(document).foundation();
71:                                </script>
72:
73: </body>
74: </html>
```


list.php

```
51:                                     ?>
52:                                     </tbody>
53:                                 </table>
54:                             </div>
55: </div>
56: <?php
57: echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
58: // echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/jquery.js' );
59: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
60: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
61: echo js_tag ( 'resources/datatables-foundation/dataTables.foundation.min.js' );
62: ?>
63: <script type="text/javascript">
64: $(document).foundation();
65: $(document).ready( function () {
66:     $('table').DataTable();
67: } );
68: </script>
69: </body>
70: </html>
```

registration.php

```
1: <html>
2: <head>
3: <title><?php echo $this->session->userdata ( 'company_acronym' ) . ' - Personnel'?></ti
tle>
4: <?php
5: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/normalize.css' );
6: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/foundation.min.css' );
7: echo css_tag ( 'resources/foundation-icons/foundation-icons/foundation-icons.css' );
8: echo css_tag ( 'resources/custom/css/sizing.css' );
9: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/modernizr.js' );
10: ?>
11: </head>
12: <body>
13:     <?php
14:         $data ['page_set'] = 'companies';
15:         $this->load->view ( 'navigation/navigationbar_user', $data );
16:         ?>
17:         <div class="row">
18:             <?php echo br(1);?>
19:         </div>
20:         <div class="row">
21:             <div class="small-12 small-centered columns">
22:                 <div class="row">
23:                     <div class="large-3 small-12 columns text-justify">
24:                         <h4>
25:                             <strong>service registration</strong>
26:                             <p>register a new service to our system
.</p>
27:                         </h4>
28:                     </div>
29:                     <div class="large-9 small-12 columns panel radius">
30:                         <?php
31:                             echo form_open ( 'service/process_registration'
);
32:                             ?>
33:                             <div class="row">
34:                                 <div class="small-12 columns">
35:
36:
37:
38:                                     <label
39:                                         <?php echo (issetandnot
empty ( form_error ( '' ) ) ? 'class="error" : '' )?>>
40:                                         Service </label>
41:                                     <?php
42:                                         $name = array (
43:                                             'type' => 'text
',
44:                                             'name' => 'name
',
45:                                             'id' => 'name',
46:                                             'value' => set_
value ( 'name' ),
47:                                             'maxlength' =>
'50',
48:                                             'placeholder' =
> 'Service'
49:                                         );
50:                                         (issetandnotempty ( form_error
( 'name' ) ) ? $name += array (
51:                                             'class' => 'err
or'
52:                                         ) : '' );
```


registration.php

```
100:                                     <div class="small-6 col
umns">
101:                                     <label>
102:                                     <?php
103:                                     $available = ar
ray (
104:                                     'name' => 'is_available',
105:                                     'id' => 'is_available',
106:                                     'value' => 'available',
107:                                     'checked' => TRUE,
108:                                     'style' => 'margin:10px'
109:                                     );
110:                                     echo form_radio
( $available ) . ' available';
111:                                     ?>
112:                                     </label>
113:                                     </div>
114:                                     <div class="small-6 col
umns">
115:                                     <label>
116:                                     <?php
117:                                     $inavailable =
array (
118:                                     'name' => 'is_available',
119:                                     'id' => 'is_available',
120:                                     'value' => 'inavailable',
121:                                     'style' => 'margin:10px'
122:                                     );
123:                                     echo form_radio
( $inavailable ) . ' inavailable';
124:                                     ?>
125:                                     </label>
126:                                     </div>
127:                                     </div>
128:                                     </div>
129:                                     </div>
130:                                     <div class="row">
131:                                     <div class="small-12 columns"><?php
132:                                     $submit = array (
133:                                     'name' => 'register',
134:                                     'class' => 'tiny button
radius span-column'
135:                                     );
136:                                     echo form_submit ( $submit, 'register'
);
137:                                     ?></div>
138:                                     </div>
139:                                     <?php
140:                                     echo form_close ();
141:                                     ?>
142:                                     </div>
143:                                     </div>
144:                                     </div>
145:                                     </div>
```


registration.php

```
146:
147:
148:
149:
150:     <?php
151:     echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
152:     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
153:     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
154:     ?>
155:     <script>
156:     $(document).foundation();
157:     </script>
158:
159: </body>
160: </html>
```

update.php

```
1: <html>
2: <head>
3: <title><?php echo $this->session->userdata ( 'company_acronym' ) . ' - Personnel'?></ti
tle>
4: <?php
5: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/normalize.css' );
6: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/foundation.min.css' );
7: echo css_tag ( 'resources/foundation-icons/foundation-icons/foundation-icons.css' );
8: echo css_tag ( 'resources/custom/css/sizing.css' );
9: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/modernizr.js' );
10: ?>
11: </head>
12: <body>
13:     <?php
14:     $data ['page_set'] = 'companies';
15:     $this->load->view ( 'navigation/navigationbar_user', $data );
16:     ?>
17:     <div class="row">
18:     <?php echo br(1);?>
19:     </div>
20:     <div class="row">
21:         <div class="small-12 small-centered columns">
22:             <div class="row">
23:                 <div class="large-3 small-12 columns text-justify">
24:                     <h4>
25:                         <strong>service registration</strong>
26:                         <p>register a new service to our system
.</p>
27:                     </h4>
28:                 </div>
29:                 <div class="large-9 small-12 columns panel radius">
30:                     <?php
31:                     echo form_open ( 'service/process_update' );
32:                     $service_id = array (
33:                         'type' => 'hidden',
34:                         'name' => 'service_id',
35:                         'value' => $service_id,
36:                         'id' => 'service_id',
37:                         'maxlength' => '12',
38:                         'size' => '50',
39:                         'style' => 'width:50%',
40:                         'placeholder' => 'MHICSRV00001'
41:                     );
42:                     echo form_input ( $service_id );
43:                     ?>
44:                     <div class="row">
45:                         <div class="small-12 columns">
46:
47:
48:
49:                             <label
50:                                 <?php echo (issetandnot
empty ( form_error ( '' ) ) ? 'class="error" : '' )?>>
51:                                 Service </label>
52:                             <?php
53:                             $name = array (
54:                                 'type' => 'text
',
55:                                 'name' => 'name
',
56:                                 'id' => 'name',
57:                                 'value' => (iss
```


update.php

```
radius span-column'
147:                                     );
148:                                     echo form_submit ( $submit, 'register'
);
149:                                     ?></div>
150:                                     </div>
151:                                     <?php
152:                                     echo form_close ();
153:                                     ?>
154:                                     </div>
155:                                     </div>
156:                                     </div>
157: </div>
158:
159:
160:
161:
162: <?php
163: echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
164: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
165: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
166: ?>
167: <script>
168: $(document).foundation();
169: </script>
170:
171: </body>
172: </html>
```

list.php

```
1: <html>
2: <head>
3: <title><?php echo $this->session->userdata ( 'company_acronym' ) . ' - Personnel'?></ti
tle>
4: <?php
5: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/normalize.css' );
6: echo css_tag ( 'resources/foundation-5.5.2.philhealth/css/foundation.min.css' );
7: echo css_tag ( 'resources/foundation-icons/foundation-icons/foundation-icons.css' );
8: echo css_tag ( 'resources/custom/css/sizing.css' );
9: echo css_tag ( 'resources/datatables-foundation/dataTables.foundation.css' );
10: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/modernizr.js' );
11: ?>
12: </head>
13: <body>
14:     <?php
15:         $data ['page_set'] = 'customers';
16:         $this->load->view ( 'navigation/navigationbar_user', $data );
17:     ?>
18:     <div class="row">
19:         <?php echo br(1);?>
20:     </div>
21:     <div class="row">
22:         <div class="small-10 small-centered columns">
23:             <h3>
24:                 <strong><?php echo anchor($this->config->base_url('/dash
board/partner/' . $partner_company_id), $partner_company_name);?'>'s personnel list</strong>
25:             </h3>
26:             <p>a collation of a quick summary of personnel of the <?php ech
o $partner_company_name;?>, <?php echo anchor($this->config->base_url('personnel/register/' . $p
artner_company_id), 'register')?> a new one?</p>
27:         </div>
28:     </div>
29:     <div class="row">
30:         <div class="small-10 small-centered columns">
31:             <table>
32:                 <thead>
33:                     <tr>
34:                         <th></th>
35:                         <th>name</th>
36:                         <th>region covered</th>
37:                         <th>active</th>
38:                     </tr>
39:                 </thead>
40:                 <tbody>
41:                     <?php
42:                     foreach ( $partner_company_personnel_list as $p
artner_company_personnel_element ) {
43:                         ?>
44:                         <tr>
45:                             <td><?php echo anchor ( $this->config->
base_url ( 'personnel/update/' . $partner_company_personnel_element ['personnel_id'] ), '<i cl
ass="fi-widget"></i>' );?></td>
46:                             <td><?php echo $partner_company_personn
el_element ['name']; ?></td>
47:                             <td><?php echo $partner_company_personn
el_element ['region_responsibility_id']; ?></td>
48:                             <td>
49:                                 bgcolor="<?php echo ( $partner_
company_personnel_element ['is_active'] == 'active' ? '#77DD77' : '#FDFD96' ) ?>"><strong><?ph
p echo anchor ( $this->config->base_url ( 'personnel/' . ($partner_company_personnel_element ['i
s_active'] == 'active' ? 'deactivate' : 'activate') . '/' . $partner_company_personnel_elemen
t ['personnel_id'] ), ($partner_company_personnel_element ['is_active'] == 'active' ? 'deacti
vate' : 'activate') );?></strong></td>
```

list.php

```
50:         </tr>
51:         <?php
52:         }
53:     ?>
54:     </tbody>
55: </table>
56: </div>
57: </div>
58: <?php
59: echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
60: // echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/jquery.js' );
61: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
62: echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
63: echo js_tag ( 'resources/datatables-foundation/dataTables.foundation.min.js' );
64: ?>
65: <script type="text/javascript">
66: $(document).foundation();
67: $(document).ready( function () {
68:     $('table').DataTable();
69: } );
70: </script>
71: </body>
72: </html>
```


registration.php

```
138:                                     </div>
139:                               </div>
140:                               <div class="row">
141:                                   <div class="small-12 columns">
142:                                       <label
143:                                           <?php echo (issetandnot
empty ( form_error ( 'email_address' ) ) ? 'class="error"' : '' )?>>
144:                                           Email Address </label>
145:                                           <?php
146:                                           $email_address = array (
147:                                               'type' => 'text
',
148:                                               'name' => 'email_
l_address',
149:                                               'id' => 'email_
address',
150:                                               'value' => set_
value ( 'email_address' ),
151:                                               'maxlength' =>
'255',
152:                                               'placeholder' =
> 'username@email.com'
153:                                           );
154:                                           (issetandnotempty ( form_error
( 'email_address' ) ) ? $email_address += array (
155:                                               'class' => 'err
or'
156:                                           ) : '');
157:                                           echo form_input ( $email_addres
s );
158:                                           ?>
159:                                           <?php
160:                                           if (issetandnotempty ( form_err
or ( 'email_address' ) )) {
161:                                               echo form_error ( 'email_
l_address', '<small class="error">', '</small>' );
162:                                           }
163:                                           ?>
164:                                       </div>
165:                               </div>
166:                               <div class="row">
167:                                   <div class="small-12 columns">
168:                                       <label> Partner Company </label>
>
169:                                           <?php
170:                                           $partner_compan
y_id = $partner_company_id;
171:                                           echo form_dropd
own ( 'partner_company_id', $partner_company_id, $selected_partner_company_id );
172:                                           ?>
173:
174:                                       </div>
175:                               </div>
176:                               <div class="row">
177:                                   <div class="small-12 columns">
178:                                       <label>Region Covered</label>
179:                                       <?php
180:                                       $region_respons
ibility_id = $region_responsibility_id;
181:                                       echo form_dropd
own ( 'region_responsibility_id', $region_responsibility_id );
182:                                       ?>
183:                                   </div>
```


registration.php

```
230:                                     ?>
231:                                     </div>
232:                                 </div>
233:                             </div>
234:                         </div>
235:
236:
237:
238:
239:     <?php
240:     echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
241:     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
242:     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
243:     ?>
244:     <script>
245:     $(document).foundation();
246:     </script>
247:
248: </body>
249: </html>
```


update.php

```
etandnotempty ( form_error ( 'first_name' ) ) ? set_value ( 'first_name' ) : $first_name),
56:                                                                 'maxlength' =>
'15',
57:                                                                 'placeholder' =
> 'Juan'
58:                                                                 );
59:                                                                 (issetandnotempty ( form_error
( 'first_name' ) ) ? $first_name += array (
60:                                                                 'class' => 'err
or'
61:                                                                 ) : '');
62:                                                                 echo form_input ( $first_name )
;
63:                                                                 ?>
64:                                                                 <?php
65:                                                                 if (issetandnotempty ( form_err
or ( 'first_name' ) )) {
66:                                                                 echo form_error ( 'firs
t_name', '<small class="error">', '</small>' );
67:                                                                 }
68:                                                                 ?>
69:                                                                 </div>
70:                                                                 <div class="small-4 columns">
71:                                                                 <label
72:                                                                 <?php echo (issetandnot
empty ( form_error ( 'middle_name' ) ) ? 'class="error"' : '')?>>
73:                                                                 Middle Name </label>
74:                                                                 <?php
75:                                                                 $middle_name = array (
76:                                                                 'type' => 'text
',
77:                                                                 'name' => 'midd
le_name',
78:                                                                 'id' => 'middle
_name',
79:                                                                 'value' => (iss
etandnotempty ( form_error ( 'middle_name' ) ) ? set_value ( 'middle_name' ) : $middle_name),
80:                                                                 'maxlength' =>
'15',
81:                                                                 'placeholder' =
> 'Marino'
82:                                                                 );
83:                                                                 (issetandnotempty ( form_error
( 'middle_name' ) ) ? $middle_name += array (
84:                                                                 'class' => 'err
or'
85:                                                                 ) : '');
86:                                                                 echo form_input ( $middle_name
);
87:                                                                 ?>
88:                                                                 <?php
89:                                                                 if (issetandnotempty ( form_err
or ( 'middle_name' ) )) {
90:                                                                 echo form_error ( 'midd
le_name', '<small class="error">', '</small>' );
91:                                                                 }
92:                                                                 ?>
93:                                                                 </div>
94:                                                                 <div class="small-4 columns">
95:                                                                 <label
96:                                                                 <?php echo (issetandnot
empty ( form_error ( 'last_name' ) ) ? 'class="error"' : '')?>>
```

update.php

```
98:                                                                                               Last Name </label>
99:                                                                                               <?php
100:                                                                                              $last_name = array (
101:                                                                                                  'type' => 'text
',
102:                                                                                                  'name' => 'last
_name',
103:                                                                                                  'id' => 'last_n
ame',
104:                                                                                                  'value' => (iss
etandnotempty ( form_error ( 'last_name' ) ) ? set_value ( 'last_name' ) : $last_name),
105:                                                                                                  'maxlength' =>
'15',
106:                                                                                                  'placeholder' =
> 'dela Cruz'
107:                                                                                               );
108:                                                                                               (issetandnotempty ( form_error
( 'last_name' ) ) ? $last_name += array (
109:                                                                                                   'class' => 'err
or'
110:                                                                                                   ) : '');
111:                                                                                               echo form_input ( $last_name );
112:                                                                                               ?>
113:                                                                                               <?php
114:                                                                                               if (issetandnotempty ( form_err
or ( 'last_name' ) )) {
115:                                                                                                   echo form_error ( 'last
_name', '<small class="error">', '</small>' );
116:                                                                                               }
117:                                                                                               ?>
118:                                                                                               </div>
119:                                                                                               </div>
120:
121:                                                                                               <div class="row">
122:                                                                                                   <div class="small-12 columns">
123:                                                                                                       <label
124:                                                                                                           <?php echo (issetandnot
empty ( form_error ( 'cellphone_number' ) ) ? 'class="error"' : '')?>>
125:                                                                                                               Cellphone Number </labe
l>
126:                                                                                               <?php
127:                                                                                               $cellphone_number = array (
128:                                                                                                   'type' => 'text
',
129:                                                                                                   'name' => 'cell
phone_number',
130:                                                                                                   'id' => 'cellph
one_number',
131:                                                                                                   'value' => (iss
etandnotempty ( form_error ( 'cellphone_number' ) ) ? set_value ( 'cellphone_number' ) : $cell
phone_number),
132:                                                                                                   'maxlength' =>
'11',
133:                                                                                                   'placeholder' =
> '09XXXXXXXX'
134:                                                                                                   );
135:                                                                                                   (issetandnotempty ( form_error
( 'cellphone_number' ) ) ? $cellphone_number += array (
136:                                                                                                       'class' => 'err
or'
137:                                                                                                       ) : '');
138:                                                                                                   echo form_input ( $cellphone_nu
mber );
```


update.php

```
184:                                     <label> Covered Region </label>
185:                                     <?php
186:                                     $region_respons
ibility_id = $region_responsibility_id;
187:                                     echo form_dropd
own ( 'region_responsibility_id', $region_responsibility_id, $selected_region_responsibility_i
d );
188:                                     ?>
189:                                     </div>
190:                               </div>
191:                               <div class="row">
192:                                 <div class="small-12 columns">
193:                                   <label> Active</label>
194:                                   <div class="row">
195:                                     <div class="small-6 col
umns">
196:                                       <label>
197:                                       <?php
198:                                       $active = array
(
199:                                         'name' => 'is_active',
200:                                         'id' => 'is_active',
201:                                         'value' => 'active',
202:                                         'checked' => ($is_active === 'active' ? TRUE : FALSE),
203:                                         'style' => 'margin:10px'
204:                                       );
205:                                       echo form_radio
( $active ) . ' active';
206:                                       ?>
207:                                       </label>
208:                                     </div>
209:                                     <div class="small-6 col
umns">
210:                                       <label>
211:                                       <?php
212:                                       $inactive = arr
ay (
213:                                         'name' => 'is_active',
214:                                         'id' => 'is_active',
215:                                         'value' => 'inactive',
216:                                         'checked' => ($is_active === 'inactive' ? TRUE : FALSE),
217:                                         'style' => 'margin:10px'
218:                                       );
219:                                       echo form_radio
( $inactive ) . ' inactive';
220:                                       ?>
221:                                       </label>
222:                                     </div>
223:                                   </div>
224:                                 </div>
225:                               </div>
226:                               <div class="row">
227:                                 <div class="small-12 columns"><?php
```

update.php

```
228:                                     $submit = array (
229:                                     'name' => 'register',
230:                                     'class' => 'tiny button
radius span-column'
231:                                     );
232:                                     echo form_submit ( $submit, 'register'
);
233:                                     ?></div>
234:                                     </div>
235:                                     <?php
236:                                     echo form_close ();
237:                                     ?>
238:                                     </div>
239:                                     </div>
240:                                     </div>
241:                                     </div>
242:
243:
244:
245:
246:                                     <?php
247:                                     echo js_tag ( 'resources/jquery/jquery-2.1.4.min.js' );
248:                                     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/vendor/fastclick.js' );
249:                                     echo js_tag ( 'resources/foundation-5.5.2.philhealth/js/foundation.min.js' );
250:                                     ?>
251:                                     <script>
252:                                     $(document).foundation();
253:                                     </script>
254:
255: </body>
256: </html>
```

index.html

```
1: <html>
2: <head>
3: <title>403 Forbidden</title>
4: </head>
5: <body>
6:
7:     <p>Directory access is forbidden.</p>
8:
9: </body>
10: </html>
```

.htaccess

1: Deny from all

index.html

```
1: <html>
2: <head>
3: <title>403 Forbidden</title>
4: </head>
5: <body>
6:
7:     <p>Directory access is forbidden.</p>
8:
9: </body>
10: </html>
```

index.html

```
1: <html>
2: <head>
3: <title>403 Forbidden</title>
4: </head>
5: <body>
6:
7:     <p>Directory access is forbidden.</p>
8:
9: </body>
10: </html>
```

hooks.php

```
1: <?php
2:
3: if (! defined ( 'BASEPATH' ))
4:     exit ( 'No direct script access allowed' );
5: /*
6: | -----
7: | Hooks
8: | -----
9: | This file lets you define "hooks" to extend CI without hacking the core
10: | files. Please see the user guide for info:
11: |
12: |     http://codeigniter.com/user\_guide/general/hooks.html
13: |
14: */
15:
16:
17:
18: /* End of file hooks.php */
19: /* Location: ./application/config/hooks.php */
```


profiler.php

```
1: <?php
2:
3: if (! defined ( 'BASEPATH' ))
4:     exit ( 'No direct script access allowed' );
5: /*
6: | -----
7: | Profiler Sections
8: | -----
9: | This file lets you determine whether or not various sections of Profiler
10: | data are displayed when the Profiler is enabled.
11: | Please see the user guide for info:
12: |
13: |     http://codeigniter.com/user\_guide/general/profiling.html
14: |
15: */
16:
17:
18:
19: /* End of file profiler.php */
20: /* Location: ./application/config/profiler.php */
```

autoload.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4:     /*
5:  * | -----
6:  * | AUTO-LOADER
7:  * | -----
8:  * | This file specifies which systems should be loaded by default.
9:  * |
10: * | In order to keep the framework as light-weight as possible only the
11: * | absolute minimal resources are loaded by default. For example,
12: * | the database is not connected to automatically since no assumption
13: * | is made regarding whether you intend to use it. This file lets
14: * | you globally define which systems you would like loaded with every
15: * | request.
16: * |
17: * | -----
18: * | Instructions
19: * | -----
20: * |
21: * | These are the things you can load automatically:
22: * |
23: * | 1. Packages
24: * | 2. Libraries
25: * | 3. Helper files
26: * | 4. Custom config files
27: * | 5. Language files
28: * | 6. Models
29: * |
30: */
31:
32: /*
33: * | -----
34: * | Auto-load Packages
35: * | -----
36: * | Prototype:
37: * |
38: * | $autoload['packages'] = array(APPPATH.'third_party', '/usr/local/shared');
39: * |
40: */
41:
42: $autoload ['packages'] = array ();
43:
44: /*
45: * | -----
46: * | Auto-load Libraries
47: * | -----
48: * | These are the classes located in the system/libraries folder
49: * | or in your application/libraries folder.
50: * |
51: * | Prototype:
52: * |
53: * | $autoload['libraries'] = array('database', 'session', 'xmlrpc');
54: */
55:
56: $autoload ['libraries'] = array (
57:     'session',
58:     'database',
59:     'form_validation'
60: );
61:
62: /*
63: * | -----
```

autoload.php

```
64:  * | Auto-load Helper Files
65:  * | -----
66:  * | Prototype:
67:  * |
68:  * | $autoload['helper'] = array('url', 'file');
69:  */
70:
71: $autoload ['helper'] = array (
72:     'variable',
73:     'import',
74:     'html',
75:     'url',
76:     'form',
77:     'session',
78:     'form_custom_validation'
79: );
80:
81: /*
82:  * | -----
83:  * | Auto-load Config files
84:  * | -----
85:  * | Prototype:
86:  * |
87:  * | $autoload['config'] = array('config1', 'config2');
88:  * |
89:  * | NOTE: This item is intended for use ONLY if you have created custom
90:  * | config files. Otherwise, leave it blank.
91:  * |
92:  */
93:
94: $autoload ['config'] = array ();
95:
96: /*
97:  * | -----
98:  * | Auto-load Language files
99:  * | -----
100:  * | Prototype:
101:  * |
102:  * | $autoload['language'] = array('lang1', 'lang2');
103:  * |
104:  * | NOTE: Do not include the "_lang" part of your file. For example
105:  * | "codeigniter_lang.php" would be referenced as array('codeigniter');
106:  * |
107:  */
108:
109: $autoload ['language'] = array ();
110:
111: /*
112:  * | -----
113:  * | Auto-load Models
114:  * | -----
115:  * | Prototype:
116:  * |
117:  * | $autoload['model'] = array('model1', 'model2');
118:  * |
119:  */
120:
121: $autoload ['model'] = array ();
122:
123:
124: /* End of file autoload.php */
125: /* Location: ./application/config/autoload.php */
```

constants.php

```
1: <?php
2:
3: if (! defined ( 'BASEPATH' ))
4:     exit ( 'No direct script access allowed' );
5:
6:     /*
7:  * |-----
8:  * | File and Directory Modes
9:  * |-----
10: * |
11: * | These prefs are used when checking and setting modes when working
12: * | with the file system. The defaults are fine on servers with proper
13: * | security, but you may wish (or even need) to change the values in
14: * | certain environments (Apache running a separate process for each
15: * | user, PHP under CGI with Apache suEXEC, etc.). Octal values should
16: * | always be used to set the mode correctly.
17: * |
18: */
19: define ( 'FILE_READ_MODE', 0644 );
20: define ( 'FILE_WRITE_MODE', 0666 );
21: define ( 'DIR_READ_MODE', 0755 );
22: define ( 'DIR_WRITE_MODE', 0777 );
23:
24: /*
25: * |-----
26: * | File Stream Modes
27: * |-----
28: * |
29: * | These modes are used when working with fopen()/popen()
30: * |
31: */
32:
33: define ( 'FOPEN_READ', 'rb' );
34: define ( 'FOPEN_READ_WRITE', 'r+b' );
35: define ( 'FOPEN_WRITE_CREATE_DESTRUCTIVE', 'wb' ); // truncates existing file data, use
with care
36: define ( 'FOPEN_READ_WRITE_CREATE_DESTRUCTIVE', 'w+b' ); // truncates existing file dat
a, use with care
37: define ( 'FOPEN_WRITE_CREATE', 'ab' );
38: define ( 'FOPEN_READ_WRITE_CREATE', 'a+b' );
39: define ( 'FOPEN_WRITE_CREATE_STRICT', 'xb' );
40: define ( 'FOPEN_READ_WRITE_CREATE_STRICT', 'x+b' );
41:
42:
43: /* End of file constants.php */
44: /* Location: ./application/config/constants.php */
```

doctypes.php

```
1: <?php
2:
3: if (! defined ( 'BASEPATH' ))
4:     exit ( 'No direct script access allowed' );
5:
6: $_doctypes = array (
7:     'xhtml11' => '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://
/www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">',
8:     'xhtml1-strict' => '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict
//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">',
9:     'xhtml1-trans' => '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transit
ional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">',
10:    'xhtml1-frame' => '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset
t//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">',
11:    'html5' => '<!DOCTYPE html>',
12:    'html4-strict' => '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "h
ttp://www.w3.org/TR/html4/strict.dtd">',
13:    'html4-trans' => '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transiti
onal//EN" "http://www.w3.org/TR/html4/loose.dtd">',
14:    'html4-frame' => '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset
//EN" "http://www.w3.org/TR/html4/frameset.dtd">'
15: );
16:
17: /* End of file doctypes.php */
18: /* Location: ./application/config/doctypes.php */
```

config.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4:
5:     /*
6:  * -----
7:  * Base Site URL
8:  * -----
9:  *
10: * URL to your CodeIgniter root. Typically this will be your base URL,
11: * WITH a trailing slash:
12: *
13: * http://example.com/
14: *
15: * If this is not set then CodeIgniter will try to guess the protocol, domain
16: * and path to your installation. However, you should always configure this
17: * explicitly and never rely on auto-guessing, especially in production
18: * environments.
19: *
20: */
21: $config ['base_url'] = '';
22:
23: /*
24: * -----
25: * Index File
26: * -----
27: *
28: * Typically this will be your index.php file, unless you've renamed it to
29: * something else. If you are using mod_rewrite to remove the page set this
30: * variable so that it is blank.
31: *
32: */
33: $config ['index_page'] = '';
34:
35: /*
36: * -----
37: * URI PROTOCOL
38: * -----
39: *
40: * This item determines which server global should be used to retrieve the
41: * URI string. The default setting of 'AUTO' works for most servers.
42: * If your links do not seem to work, try one of the other delicious flavors:
43: *
44: * 'AUTO' Default - auto detects
45: * 'PATH_INFO' Uses the PATH_INFO
46: * 'QUERY_STRING' Uses the QUERY_STRING
47: * 'REQUEST_URI' Uses the REQUEST_URI
48: * 'ORIG_PATH_INFO' Uses the ORIG_PATH_INFO
49: *
50: */
51: $config ['uri_protocol'] = 'REQUEST_URI';
52:
53: /*
54: * -----
55: * URL suffix
56: * -----
57: *
58: * This option allows you to add a suffix to all URLs generated by CodeIgniter.
59: * For more information please see the user guide:
60: *
61: * http://codeigniter.com/user_guide/general/urls.html
62: */
63:
```

config.php

```
64: $config ['url_suffix'] = '';
65:
66: /*
67: * |-----
68: * | Default Language
69: * |-----
70: * |
71: * | This determines which set of language files should be used. Make sure
72: * | there is an available translation if you intend to use something other
73: * | than english.
74: * |
75: */
76: $config ['language'] = 'english';
77:
78: /*
79: * |-----
80: * | Default Character Set
81: * |-----
82: * |
83: * | This determines which character set is used by default in various methods
84: * | that require a character set to be provided.
85: * |
86: */
87: $config ['charset'] = 'UTF-8';
88:
89: /*
90: * |-----
91: * | Enable/Disable System Hooks
92: * |-----
93: * |
94: * | If you would like to use the 'hooks' feature you must enable it by
95: * | setting this variable to TRUE (boolean). See the user guide for details.
96: * |
97: */
98: $config ['enable_hooks'] = FALSE;
99:
100: /*
101: * |-----
102: * | Class Extension Prefix
103: * |-----
104: * |
105: * | This item allows you to set the filename/classname prefix when extending
106: * | native libraries. For more information please see the user guide:
107: * |
108: * | http://codeigniter.com/user\_guide/general/core\_classes.html
109: * | http://codeigniter.com/user\_guide/general/creating\_libraries.html
110: * |
111: */
112: $config ['subclass_prefix'] = 'MY_';
113:
114: /*
115: * |-----
116: * | Allowed URL Characters
117: * |-----
118: * |
119: * | This lets you specify with a regular expression which characters are permitted
120: * | within your URLs. When someone tries to submit a URL with disallowed
121: * | characters they will get a warning message.
122: * |
123: * | As a security measure you are STRONGLY encouraged to restrict URLs to
124: * | as few characters as possible. By default only these are allowed: a-z 0-9~%.:_-
125: * |
126: * | Leave blank to allow all characters -- but only if you are insane.
```

config.php

```
127: * |
128: * | DO NOT CHANGE THIS UNLESS YOU FULLY UNDERSTAND THE REPERCUSSIONS!!
129: * |
130: */
131: $config ['permitted_uri_chars'] = 'a-z 0-9~%.:_\-';
132:
133: /*
134: * |-----
135: * | Enable Query Strings
136: * |-----
137: * |
138: * | By default CodeIgniter uses search-engine friendly segment based URLs:
139: * | example.com/who/what/where/
140: * |
141: * | By default CodeIgniter enables access to the $_GET array. If for some
142: * | reason you would like to disable it, set 'allow_get_array' to FALSE.
143: * |
144: * | You can optionally enable standard query string based URLs:
145: * | example.com?who=me&what=something&where=here
146: * |
147: * | Options are: TRUE or FALSE (boolean)
148: * |
149: * | The other items let you set the query string 'words' that will
150: * | invoke your controllers and its functions:
151: * | example.com/index.php?c=controller&m=function
152: * |
153: * | Please note that some of the helpers won't work as expected when
154: * | this feature is enabled, since CodeIgniter is designed primarily to
155: * | use segment based URLs.
156: * |
157: */
158: $config ['allow_get_array'] = TRUE;
159: $config ['enable_query_strings'] = FALSE;
160: $config ['controller_trigger'] = 'c';
161: $config ['function_trigger'] = 'm';
162: $config ['directory_trigger'] = 'd'; // experimental not currently in use
163:
164: /*
165: * |-----
166: * | Error Logging Threshold
167: * |-----
168: * |
169: * | If you have enabled error logging, you can set an error threshold to
170: * | determine what gets logged. Threshold options are:
171: * | You can enable error logging by setting a threshold over zero. The
172: * | threshold determines what gets logged. Threshold options are:
173: * |
174: * | 0 = Disables logging, Error logging TURNED OFF
175: * | 1 = Error Messages (including PHP errors)
176: * | 2 = Debug Messages
177: * | 3 = Informational Messages
178: * | 4 = All Messages
179: * |
180: * | For a live site you'll usually only enable Errors (1) to be logged otherwise
181: * | your log files will fill up very fast.
182: * |
183: */
184: $config ['log_threshold'] = 0;
185:
186: /*
187: * |-----
188: * | Error Logging Directory Path
189: * |-----
```


config.php

```
190: * |
191: * | Leave this BLANK unless you would like to set something other than the default
192: * | application/logs/ folder. Use a full server path with trailing slash.
193: * |
194: */
195: $config ['log_path'] = '';
196:
197: /*
198: * |-----
199: * | Date Format for Logs
200: * |-----
201: * |
202: * | Each item that is logged has an associated date. You can use PHP date
203: * | codes to set your own date formatting
204: * |
205: */
206: $config ['log_date_format'] = 'Y-m-d H:i:s';
207:
208: /*
209: * |-----
210: * | Cache Directory Path
211: * |-----
212: * |
213: * | Leave this BLANK unless you would like to set something other than the default
214: * | system/cache/ folder. Use a full server path with trailing slash.
215: * |
216: */
217: $config ['cache_path'] = '';
218:
219: /*
220: * |-----
221: * | Encryption Key
222: * |-----
223: * |
224: * | If you use the Encryption class or the Session class you
225: * | MUST set an encryption key. See the user guide for info.
226: * |
227: */
228: $config ['encryption_key'] = '7ua3ZMtxp8@u$R!uR$kdyaXR&@TrWW4URPzpd4QMVSDe3PkrHz%hfzhht
hSk4srxvamaQ&xrs5D#bMsvt7$u%S2d!Ekwfe5GE&^TwNCQNMJhzq8VVRqw6Z4VsYxM%jtFc!qaZT*AQnTY%%!MjU4qNjKB
uZT!EE5P*hyGmYz7m4VpdKep9hyFGxmw@*qd6Kvvn^mEQyM$#tRSZ#bswc44vbfjWJuBQ2Z5YbydXKECaqtdAeE&wfm9jd
MF&H8BcjydA';
229:
230: /*
231: * |-----
232: * | Session Variables
233: * |-----
234: * |
235: * | 'sess_cookie_name' = the name you want for the cookie
236: * | 'sess_expiration' = the number of SECONDS you want the session to last.
237: * | by default sessions last 7200 seconds (two hours). Set to zero for no expiration.
238: * | 'sess_expire_on_close' = Whether to cause the session to expire automatically
239: * | when the browser window is closed
240: * | 'sess_encrypt_cookie' = Whether to encrypt the cookie
241: * | 'sess_use_database' = Whether to save the session data to a database
242: * | 'sess_table_name' = The name of the session database table
243: * | 'sess_match_ip' = Whether to match the user's IP address when reading the session
data
244: * | 'sess_match_useragent' = Whether to match the User Agent when reading the session
data
245: * | 'sess_time_to_update' = how many seconds between CI refreshing Session Information
246: * |
247: */
```

config.php

```
248: $config ['sess_cookie_name'] = 'ghic_crm_t';
249: $config ['sess_expiration'] = 7200;
250: $config ['sess_expire_on_close'] = TRUE;
251: $config ['sess_encrypt_cookie'] = TRUE;
252: $config ['sess_use_database'] = FALSE;
253: $config ['sess_table_name'] = 'ghic_crm_t';
254: $config ['sess_match_ip'] = TRUE;
255: $config ['sess_match_useragent'] = TRUE;
256: $config ['sess_time_to_update'] = 300;
257:
258: /*
259: * |-----
260: * | Cookie Related Variables
261: * |-----
262: *
263: * | 'cookie_prefix' = Set a prefix if you need to avoid collisions
264: * | 'cookie_domain' = Set to .your-domain.com for site-wide cookies
265: * | 'cookie_path' = Typically will be a forward slash
266: * | 'cookie_secure' = Cookies will only be set if a secure HTTPS connection exists.
267: * |
268: */
269: $config ['cookie_prefix'] = "";
270: $config ['cookie_domain'] = "";
271: $config ['cookie_path'] = "/";
272: $config ['cookie_secure'] = FALSE;
273:
274: /*
275: * |-----
276: * | Global XSS Filtering
277: * |-----
278: *
279: * | Determines whether the XSS filter is always active when GET, POST or
280: * | COOKIE data is encountered
281: * |
282: */
283: $config ['global_xss_filtering'] = TRUE;
284:
285: /*
286: * |-----
287: * | Cross Site Request Forgery
288: * |-----
289: * | Enables a CSRF cookie token to be set. When set to TRUE, token will be
290: * | checked on a submitted form. If you are accepting user data, it is strongly
291: * | recommended CSRF protection be enabled.
292: *
293: * | 'csrf_token_name' = The token name
294: * | 'csrf_cookie_name' = The cookie name
295: * | 'csrf_expire' = The number in seconds the token should expire.
296: */
297: $config ['csrf_protection'] = TRUE;
298: $config ['csrf_token_name'] = 'ghic_crm_t_csrf';
299: $config ['csrf_cookie_name'] = 'ghic_crm_t_csrf_cookie';
300: $config ['csrf_expire'] = 7200;
301:
302: /*
303: * |-----
304: * | Output Compression
305: * |-----
306: *
307: * | Enables Gzip output compression for faster page loads. When enabled,
308: * | the output class will test whether your server supports Gzip.
309: * | Even if it does, however, not all browsers support compression
310: * | so enable only if you are reasonably sure your visitors can handle it.
```

config.php

```
311: * |
312: * | VERY IMPORTANT: If you are getting a blank page when compression is enabled it
313: * | means you are prematurely outputting something to your browser. It could
314: * | even be a line of whitespace at the end of one of your scripts. For
315: * | compression to work, nothing can be sent before the output buffer is called
316: * | by the output class. Do not 'echo' any values with compression enabled.
317: * |
318: */
319: $config ['compress_output'] = FALSE;
320:
321: /*
322: * | -----
323: * | Master Time Reference
324: * | -----
325: * |
326: * | Options are 'local' or 'gmt'. This pref tells the system whether to use
327: * | your server's local time as the master 'now' reference, or convert it to
328: * | GMT. See the 'date helper' page of the user guide for information
329: * | regarding date handling.
330: * |
331: */
332: $config ['time_reference'] = 'local';
333:
334: /*
335: * | -----
336: * | Rewrite PHP Short Tags
337: * | -----
338: * |
339: * | If your PHP installation does not have short tag support enabled CI
340: * | can rewrite the tags on-the-fly, enabling you to utilize that syntax
341: * | in your view files. Options are TRUE or FALSE (boolean)
342: * |
343: */
344: $config ['rewrite_short_tags'] = FALSE;
345:
346: /*
347: * | -----
348: * | Reverse Proxy IPs
349: * | -----
350: * |
351: * | If your server is behind a reverse proxy, you must whitelist the proxy IP
352: * | addresses from which CodeIgniter should trust the HTTP_X_FORWARDED_FOR
353: * | header in order to properly identify the visitor's IP address.
354: * | Comma-delimited, e.g. '10.0.1.200,10.0.1.201'
355: * |
356: */
357: $config ['proxy_ips'] = '';
358:
359:
360: /* End of file config.php */
361: /* Location: ./application/config/config.php */
```

migration.php

```
1: <?php
2:
3: defined ( 'BASEPATH' ) or exit ( 'No direct script access allowed' );
4: /*
5: * |-----
6: * | Enable/Disable Migrations
7: * |-----
8: * |
9: * | Migrations are disabled by default but should be enabled
10: * | whenever you intend to do a schema migration.
11: * |
12: */
13: $config [ 'migration_enabled' ] = FALSE;
14:
15: /*
16: * |-----
17: * | Migrations version
18: * |-----
19: * |
20: * | This is used to set migration version that the file system should be on.
21: * | If you run $this->migration->latest() this is the version that schema will
22: * | be upgraded / downgraded to.
23: * |
24: */
25: $config [ 'migration_version' ] = 0;
26:
27: /*
28: * |-----
29: * | Migrations Path
30: * |-----
31: * |
32: * | Path to your migrations folder.
33: * | Typically, it will be within your application path.
34: * | Also, writing permission is required within the migrations path.
35: * |
36: */
37: $config [ 'migration_path' ] = APPPATH . 'migrations/';
38:
39:
40: /* End of file migration.php */
41: /* Location: ./application/config/migration.php */
```

routes.php

```
1: <?php
2: if (! defined ( 'BASEPATH' ))
3:     exit ( 'No direct script access allowed' );
4:     /*
5:  * | -----
6:  * | URI ROUTING
7:  * | -----
8:  * | This file lets you re-map URI requests to specific controller functions.
9:  * |
10: * | Typically there is a one-to-one relationship between a URL string
11: * | and its corresponding controller class/method. The segments in a
12: * | URL normally follow this pattern:
13: * |
14: * | example.com/class/method/id/
15: * |
16: * | In some instances, however, you may want to remap this relationship
17: * | so that a different class/function is called than the one
18: * | corresponding to the URL.
19: * |
20: * | Please see the user guide for complete details:
21: * |
22: * | http://codeigniter.com/user\_guide/general/routing.html
23: * |
24: * | -----
25: * | RESERVED ROUTES
26: * | -----
27: * |
28: * | There are two reserved routes:
29: * |
30: * | $route['default_controller'] = 'welcome';
31: * |
32: * | This route indicates which controller class should be loaded if the
33: * | URI contains no data. In the above example, the "welcome" class
34: * | would be loaded.
35: * |
36: * | $route['404_override'] = 'errors/page_missing';
37: * |
38: * | This route will tell the Router what URI segments to use if those provided
39: * | in the URL cannot be matched to a valid route.
40: * |
41: */
42:
43: $route [ 'default_controller' ] = "login";
44: $route [ '404_override' ] = '';
45:
46:
47: /* End of file routes.php */
48: /* Location: ./application/config/routes.php */
```

user_agents.php

```
1: <?php
2:
3: if (! defined ( 'BASEPATH' ))
4:     exit ( 'No direct script access allowed' );
5:     /*
6:  * | -----
7:  * | USER AGENT TYPES
8:  * | -----
9:  * | This file contains four arrays of user agent data. It is used by the
10: * | User Agent Class to help identify browser, platform, robot, and
11: * | mobile device data. The array keys are used to identify the device
12: * | and the array values are used to set the actual name of the item.
13: * |
14: */
15:
16: $platforms = array (
17:     'windows nt 6.0' => 'Windows Longhorn',
18:     'windows nt 5.2' => 'Windows 2003',
19:     'windows nt 5.0' => 'Windows 2000',
20:     'windows nt 5.1' => 'Windows XP',
21:     'windows nt 4.0' => 'Windows NT 4.0',
22:     'winnt4.0' => 'Windows NT 4.0',
23:     'winnt 4.0' => 'Windows NT',
24:     'winnt' => 'Windows NT',
25:     'windows 98' => 'Windows 98',
26:     'win98' => 'Windows 98',
27:     'windows 95' => 'Windows 95',
28:     'win95' => 'Windows 95',
29:     'windows' => 'Unknown Windows OS',
30:     'os x' => 'Mac OS X',
31:     'ppc mac' => 'Power PC Mac',
32:     'freebsd' => 'FreeBSD',
33:     'ppc' => 'Macintosh',
34:     'linux' => 'Linux',
35:     'debian' => 'Debian',
36:     'sunos' => 'Sun Solaris',
37:     'beos' => 'BeOS',
38:     'apachebench' => 'ApacheBench',
39:     'aix' => 'AIX',
40:     'irix' => 'Irix',
41:     'osf' => 'DEC OSF',
42:     'hp-ux' => 'HP-UX',
43:     'netbsd' => 'NetBSD',
44:     'bsdi' => 'BSDi',
45:     'openbsd' => 'OpenBSD',
46:     'gnu' => 'GNU/Linux',
47:     'unix' => 'Unknown Unix OS'
48: );
49:
50: // The order of this array should NOT be changed. Many browsers return
51: // multiple browser types so we want to identify the sub-type first.
52: $browsers = array (
53:     'Flock' => 'Flock',
54:     'Chrome' => 'Chrome',
55:     'Opera' => 'Opera',
56:     'MSIE' => 'Internet Explorer',
57:     'Internet Explorer' => 'Internet Explorer',
58:     'Shiira' => 'Shiira',
59:     'Firefox' => 'Firefox',
60:     'Chimera' => 'Chimera',
61:     'Phoenix' => 'Phoenix',
62:     'Firebird' => 'Firebird',
63:     'Camino' => 'Camino',
```

user_agents.php

```
64:         'Netscape' => 'Netscape',
65:         'OmniWeb' => 'OmniWeb',
66:         'Safari' => 'Safari',
67:         'Mozilla' => 'Mozilla',
68:         'Konqueror' => 'Konqueror',
69:         'icab' => 'iCab',
70:         'Lynx' => 'Lynx',
71:         'Links' => 'Links',
72:         'hotjava' => 'HotJava',
73:         'amaya' => 'Amaya',
74:         'IBrowse' => 'IBrowse'
75: );
76:
77: $mobiles = array (
78:     // legacy array, old values commented out
79:     'mobileexplorer' => 'Mobile Explorer',
80:     // 'openwave' => 'Open Wave',
81:     // 'opera mini' => 'Opera Mini',
82:     // 'operamini' => 'Opera Mini',
83:     // 'elaine' => 'Palm',
84:     'palmsource' => 'Palm',
85:     // 'digital paths' => 'Palm',
86:     // 'avantgo' => 'Avantgo',
87:     // 'xiino' => 'Xiino',
88:     'palmscape' => 'Palmscape',
89:     // 'nokia' => 'Nokia',
90:     // 'ericsson' => 'Ericsson',
91:     // 'blackberry' => 'BlackBerry',
92:     // 'motorola' => 'Motorola'
93:
94:     // Phones and Manufacturers
95:     'motorola' => "Motorola",
96:     'nokia' => "Nokia",
97:     'palm' => "Palm",
98:     'iphone' => "Apple iPhone",
99:     'ipad' => "iPad",
100:    'ipod' => "Apple iPod Touch",
101:    'sony' => "Sony Ericsson",
102:    'ericsson' => "Sony Ericsson",
103:    'blackberry' => "BlackBerry",
104:    'cocoon' => "O2 Cocoon",
105:    'blazer' => "Treo",
106:    'lg' => "LG",
107:    'amoi' => "Amoi",
108:    'xda' => "XDA",
109:    'mda' => "MDA",
110:    'vario' => "Vario",
111:    'htc' => "HTC",
112:    'samsung' => "Samsung",
113:    'sharp' => "Sharp",
114:    'sie-' => "Siemens",
115:    'alcatel' => "Alcatel",
116:    'benq' => "BenQ",
117:    'ipaq' => "HP iPaq",
118:    'mot-' => "Motorola",
119:    'playstation portable' => "PlayStation Portable",
120:    'hiptop' => "Danger Hiptop",
121:    'nec-' => "NEC",
122:    'panasonic' => "Panasonic",
123:    'philips' => "Philips",
124:    'sagem' => "Sagem",
125:    'sanyo' => "Sanyo",
126:    'spv' => "SPV",
```

user_agents.php

```
127:         'zte' => "ZTE",
128:         'sendo' => "Sendo",
129:
130:         // Operating Systems
131:         'symbian' => "Symbian",
132:         'SymbianOS' => "SymbianOS",
133:         'elaine' => "Palm",
134:         'palm' => "Palm",
135:         'series60' => "Symbian S60",
136:         'windows ce' => "Windows CE",
137:
138:         // Browsers
139:         'obigo' => "Obigo",
140:         'netfront' => "Netfront Browser",
141:         'openwave' => "Openwave Browser",
142:         'mobileexplorer' => "Mobile Explorer",
143:         'operamini' => "Opera Mini",
144:         'opera mini' => "Opera Mini",
145:
146:         // Other
147:         'digital paths' => "Digital Paths",
148:         'avantgo' => "AvantGo",
149:         'xiino' => "Xiino",
150:         'novarra' => "Novarra Transcoder",
151:         'vodafone' => "Vodafone",
152:         'docomo' => "NTT DoCoMo",
153:         'o2' => "O2",
154:
155:         // Fallback
156:         'mobile' => "Generic Mobile",
157:         'wireless' => "Generic Mobile",
158:         'j2me' => "Generic Mobile",
159:         'midp' => "Generic Mobile",
160:         'cldc' => "Generic Mobile",
161:         'up.link' => "Generic Mobile",
162:         'up.browser' => "Generic Mobile",
163:         'smartphone' => "Generic Mobile",
164:         'cellphone' => "Generic Mobile"
165:     );
166:
167: // There are hundreds of bots but these are the most common.
168: $robots = array (
169:     'googlebot' => 'Googlebot',
170:     'msnbot' => 'MSNBot',
171:     'slurp' => 'Inktomi Slurp',
172:     'yahoo' => 'Yahoo',
173:     'askjeeves' => 'AskJeeves',
174:     'fastcrawler' => 'FastCrawler',
175:     'infoseek' => 'InfoSeek Robot 1.0',
176:     'lycos' => 'Lycos'
177: );
178:
179: /* End of file user_agents.php */
180: /* Location: ./application/config/user_agents.php */
```


foreign_chars.php

```

1: <?php
2:
3: if (! defined ( 'BASEPATH' ))
4:     exit ( 'No direct script access allowed' );
5:     /*
6:  * | -----
7:  * | Foreign Characters
8:  * | -----
9:  * | This file contains an array of foreign characters for transliteration
10: * | conversion used by the Text helper
11: * |
12: */
13: $foreign_characters = array (
14:     '/Ã|Ã|Ç/' => 'ae',
15:     '/Ã|Å\223/' => 'oe',
16:     '/Ã¼/' => 'ue',
17:     '/Ã\204/' => 'Ae',
18:     '/Ã\234/' => 'Ue',
19:     '/Ã\226/' => 'Oe',
20:     '/Ã\200|Ã\201|Ã\202|Ã\203|Ã\204|Ã\205|Ç°|Ã\200|Ã\202|Ã\204|Ç\215/' => '
A',
21:     '/Ã |Ã;|Ãç|Ãf|Ã¥|Ç»|Ã\201|Ã\203|Ã\205|Ç\216|Âª/' => 'a',
22:     '/Ã\207|Ã\206|Ã\210|Ã\212|Ã\214/' => 'C',
23:     '/Ã§|Ã\207|Ã\211|Ã\213|Ã\215/' => 'c',
24:     '/Ã\220|Ã\216|Ã\220/' => 'D',
25:     '/Ã°|Ã\217|Ã\221/' => 'd',
26:     '/Ã\210|Ã\211|Ã\212|Ã\213|Ã\222|Ã\224|Ã\226|Ã\230|Ã\232/' => 'E',
27:     '/Ã¨|Ã©|Ãª|Ã«|Ã\223|Ã\225|Ã\227|Ã\231|Ã\233/' => 'e',
28:     '/Ã\234|Ã\236|Ã |Ãç/' => 'G',
29:     '/Ã\235|Ã\237|Ã;|Ãf/' => 'g',
30:     '/Ã|Ã|/' => 'H',
31:     '/Ã¥|Ã§/' => 'h',
32:     '/Ã\214|Ã\215|Ã\216|Ã\217|Ã¨|Ãª|Ã¬|Ç\217|Ã®|Ã°/' => 'I',
33:     '/Ã¬|Ã¯|Ã®|Ã¬|Ã©|Ã«|Ã¬|Ç\220|Ã¯|Ã±/' => 'i',
34:     '/Ã´/' => 'J',
35:     '/Ãµ/' => 'j',
36:     '/Ã¶/' => 'K',
37:     '/Ã·/' => 'k',
38:     '/Ã¹|Ã»|Ã½|Ã¿|Ã\201/' => 'L',
39:     '/Ã°|Ã¼|Ã½|Ã\200|Ã\202/' => 'l',
40:     '/Ã\221|Ã\203|Ã\205|Ã\207/' => 'N',
41:     '/Ã±|Ã\204|Ã\206|Ã\210|Ã\211/' => 'n',
42:     '/Ã\222|Ã\223|Ã\224|Ã\225|Ã\214|Ã\216|Ç\221|Ã\220|Æ |Ã\230|Ç¼/' => 'O',
43:     '/Ã²|Ã³|Ã´|Ãµ|Ã\215|Ã\217|Ç\222|Ã\221|Æ;|Ã,|Ç¿|Ã°/' => 'o',
44:     '/Ã\224|Ã\226|Ã\230/' => 'R',
45:     '/Ã\225|Ã\227|Ã\231/' => 'r',
46:     '/Ã\232|Ã\234|Ã\236|Ã /' => 'S',
47:     '/Ã\233|Ã\235|Ã\237|Ã;|Ã¿/' => 's',
48:     '/Ãç|Ã|Ã|/' => 'T',
49:     '/Ãf|Ã¥|Ã§/' => 't',
50:     '/Ã\231|Ã\232|Ã\233|Ã¨|Ãª|Ã¬|Ã®|Ã°|Ã²|Æ¯|Ç\223|Ç\225|Ç\227|Ç\231|Ç\233/
' => 'U',
51:     '/Ã¹|Ã°|Ã»|Ã©|Ã«|Ã¬|Ã¯|Ã±|Ã³|Æ°|Ç\224|Ç\226|Ç\230|Ç\232|Ç\234/' => 'u',
52:     '/Ã\235|Ã,|Ã¶/' => 'Y',
53:     '/Ã½|Ã¿|Ã·/' => 'Y',
54:     '/Ã´/' => 'W',
55:     '/Ãµ/' => 'w',
56:     '/Ã¹|Ã»|Ã½/' => 'Z',
57:     '/Ã°|Ã¼|Ã½/' => 'z',
58:     '/Ã\206|Ç¼/' => 'AE',
59:     '/Ã\237/' => 'ss',
60:     '/Ã²/' => 'IJ',
61:     '/Ã³/' => 'ij',

```

foreign_chars.php

```
62:          '/Å\222/' => 'OE',
63:          '/Æ\222/' => 'f'
64: );
65:
66: /* End of file foreign_chars.php */
67: /* Location: ./application/config/foreign_chars.php */
```

smileys.php

```
1: <?php
2:
3: if (! defined ( 'BASEPATH' ))
4:     exit ( 'No direct script access allowed' );
5:     /*
6:  * | -----
7:  * | SMILEYS
8:  * | -----
9:  * | This file contains an array of smileys for use with the emoticon helper.
10:  * | Individual images can be used to replace multiple simileys. For example:
11:  * | :-) and :) use the same image replacement.
12:  * |
13:  * | Please see user guide for more info:
14:  * | http://codeigniter.com/user\_guide/helpers/smiley\_helper.html
15:  * |
16:  */
17:
18: $smileys = array (
19:
20:     // smiley image name width height alt
21:
22:     ':-)' => array (
23:         'grin.gif',
24:         '19',
25:         '19',
26:         'grin'
27:     ),
28:     ':lol:' => array (
29:         'lol.gif',
30:         '19',
31:         '19',
32:         'LOL'
33:     ),
34:     ':cheese:' => array (
35:         'cheese.gif',
36:         '19',
37:         '19',
38:         'cheese'
39:     ),
40:     ':)' => array (
41:         'smile.gif',
42:         '19',
43:         '19',
44:         'smile'
45:     ),
46:     ';-)' => array (
47:         'wink.gif',
48:         '19',
49:         '19',
50:         'wink'
51:     ),
52:     ';)' => array (
53:         'wink.gif',
54:         '19',
55:         '19',
56:         'wink'
57:     ),
58:     ':smirk:' => array (
59:         'smirk.gif',
60:         '19',
61:         '19',
62:         'smirk'
63:     ),
```

smileys.php

```
64:         ':roll:' => array (
65:             'rolleyes.gif',
66:             '19',
67:             '19',
68:             'rolleyes'
69:         ),
70:         ':-S' => array (
71:             'confused.gif',
72:             '19',
73:             '19',
74:             'confused'
75:         ),
76:         ':wow:' => array (
77:             'surprise.gif',
78:             '19',
79:             '19',
80:             'surprised'
81:         ),
82:         ':bug:' => array (
83:             'bigsurprise.gif',
84:             '19',
85:             '19',
86:             'big surprise'
87:         ),
88:         ':-P' => array (
89:             'tongue_laugh.gif',
90:             '19',
91:             '19',
92:             'tongue laugh'
93:         ),
94:         '%-P' => array (
95:             'tongue_rolleye.gif',
96:             '19',
97:             '19',
98:             'tongue rolleye'
99:         ),
100:        ':;-P' => array (
101:            'tongue_wink.gif',
102:            '19',
103:            '19',
104:            'tongue wink'
105:        ),
106:        ':P' => array (
107:            'raspberry.gif',
108:            '19',
109:            '19',
110:            'raspberry'
111:        ),
112:        ':blank:' => array (
113:            'blank.gif',
114:            '19',
115:            '19',
116:            'blank stare'
117:        ),
118:        ':long:' => array (
119:            'longface.gif',
120:            '19',
121:            '19',
122:            'long face'
123:        ),
124:        ':ohh:' => array (
125:            'ohh.gif',
126:            '19',
```

smileys.php

```
127:             '19',
128:             'ohh'
129:         ),
130:         ':grrr:' => array (
131:             'grrr.gif',
132:             '19',
133:             '19',
134:             'grrr'
135:         ),
136:         ':gulp:' => array (
137:             'gulp.gif',
138:             '19',
139:             '19',
140:             'gulp'
141:         ),
142:         '8-/' => array (
143:             'ohoh.gif',
144:             '19',
145:             '19',
146:             'oh oh'
147:         ),
148:         ':down:' => array (
149:             'downer.gif',
150:             '19',
151:             '19',
152:             'downer'
153:         ),
154:         ':red:' => array (
155:             'embarrassed.gif',
156:             '19',
157:             '19',
158:             'red face'
159:         ),
160:         ':sick:' => array (
161:             'sick.gif',
162:             '19',
163:             '19',
164:             'sick'
165:         ),
166:         ':shut:' => array (
167:             'shuteye.gif',
168:             '19',
169:             '19',
170:             'shut eye'
171:         ),
172:         ':-/ ' => array (
173:             'hmm.gif',
174:             '19',
175:             '19',
176:             'hmmm'
177:         ),
178:         '>:( ' => array (
179:             'mad.gif',
180:             '19',
181:             '19',
182:             'mad'
183:         ),
184:         ':mad:' => array (
185:             'mad.gif',
186:             '19',
187:             '19',
188:             'mad'
189:         ),
```

smileys.php

```
190:         '>:-(' => array (
191:             'angry.gif',
192:             '19',
193:             '19',
194:             'angry'
195:         ),
196:         ':angry:' => array (
197:             'angry.gif',
198:             '19',
199:             '19',
200:             'angry'
201:         ),
202:         ':zip:' => array (
203:             'zip.gif',
204:             '19',
205:             '19',
206:             'zipper'
207:         ),
208:         ':kiss:' => array (
209:             'kiss.gif',
210:             '19',
211:             '19',
212:             'kiss'
213:         ),
214:         ':ahhh:' => array (
215:             'shock.gif',
216:             '19',
217:             '19',
218:             'shock'
219:         ),
220:         ':coolsmile:' => array (
221:             'shade_smile.gif',
222:             '19',
223:             '19',
224:             'cool smile'
225:         ),
226:         ':coolsmirk:' => array (
227:             'shade_smirk.gif',
228:             '19',
229:             '19',
230:             'cool smirk'
231:         ),
232:         ':coolgrin:' => array (
233:             'shade_grin.gif',
234:             '19',
235:             '19',
236:             'cool grin'
237:         ),
238:         ':coolhmm:' => array (
239:             'shade_hmm.gif',
240:             '19',
241:             '19',
242:             'cool hmm'
243:         ),
244:         ':coolmad:' => array (
245:             'shade_mad.gif',
246:             '19',
247:             '19',
248:             'cool mad'
249:         ),
250:         ':coolcheese:' => array (
251:             'shade_cheese.gif',
252:             '19',
```

smileys.php

```
253:             '19',
254:             'cool cheese'
255:         ),
256:         ':vampire:' => array (
257:             'vampire.gif',
258:             '19',
259:             '19',
260:             'vampire'
261:         ),
262:         ':snake:' => array (
263:             'snake.gif',
264:             '19',
265:             '19',
266:             'snake'
267:         ),
268:         ':exclaim:' => array (
269:             'exclaim.gif',
270:             '19',
271:             '19',
272:             'excaim'
273:         ),
274:         ':question:' => array (
275:             'question.gif',
276:             '19',
277:             '19',
278:             'question'
279:         )
280: ) // no comma after last item
281:
282: ;
283:
284: /* End of file smileys.php */
285: /* Location: ./application/config/smileys.php */
```

mimes.php

```
1: <?php
2:
3: if (! defined ( 'BASEPATH' ))
4:     exit ( 'No direct script access allowed' );
5:     /*
6:  * | -----
7:  * | MIME TYPES
8:  * | -----
9:  * | This file contains an array of mime types. It is used by the
10: * | Upload class to help identify allowed file types.
11: * |
12: */
13:
14: $mimes = array (
15:     'hqx' => 'application/mac-binhex40',
16:     'cpt' => 'application/mac-compactpro',
17:     'csv' => array (
18:         'text/x-comma-separated-values',
19:         'text/comma-separated-values',
20:         'application/octet-stream',
21:         'application/vnd.ms-excel',
22:         'application/x-csv',
23:         'text/x-csv',
24:         'text/csv',
25:         'application/csv',
26:         'application/excel',
27:         'application/vnd.msexcel',
28:         'text/plain'
29:     ),
30:     'bin' => 'application/macbinary',
31:     'dms' => 'application/octet-stream',
32:     'lha' => 'application/octet-stream',
33:     'lzh' => 'application/octet-stream',
34:     'exe' => array (
35:         'application/octet-stream',
36:         'application/x-msdownload'
37:     ),
38:     'class' => 'application/octet-stream',
39:     'psd' => 'application/x-photoshop',
40:     'so' => 'application/octet-stream',
41:     'sea' => 'application/octet-stream',
42:     'dll' => 'application/octet-stream',
43:     'oda' => 'application/oda',
44:     'pdf' => array (
45:         'application/pdf',
46:         'application/x-download'
47:     ),
48:     'ai' => 'application/postscript',
49:     'eps' => 'application/postscript',
50:     'ps' => 'application/postscript',
51:     'smi' => 'application/smil',
52:     'smil' => 'application/smil',
53:     'mif' => 'application/vnd.mif',
54:     'xls' => array (
55:         'application/excel',
56:         'application/vnd.ms-excel',
57:         'application/msexcel'
58:     ),
59:     'ppt' => array (
60:         'application/powerpoint',
61:         'application/vnd.ms-powerpoint'
62:     ),
63:     'wbxml' => 'application/wbxml',
```


mimes.php

```
64:         'wmlc' => 'application/wmlc',
65:         'dcr' => 'application/x-director',
66:         'dir' => 'application/x-director',
67:         'dXR' => 'application/x-director',
68:         'dvi' => 'application/x-dvi',
69:         'gtar' => 'application/x-gtar',
70:         'gz' => 'application/x-gzip',
71:         'php' => 'application/x-httpd-php',
72:         'php4' => 'application/x-httpd-php',
73:         'php3' => 'application/x-httpd-php',
74:         'phtml' => 'application/x-httpd-php',
75:         'phps' => 'application/x-httpd-php-source',
76:         'js' => 'application/x-javascript',
77:         'swf' => 'application/x-shockwave-flash',
78:         'sit' => 'application/x-stuffit',
79:         'tar' => 'application/x-tar',
80:         'tgz' => array (
81:             'application/x-tar',
82:             'application/x-gzip-compressed'
83:         ),
84:         'xhtml' => 'application/xhtml+xml',
85:         'xht' => 'application/xhtml+xml',
86:         'zip' => array (
87:             'application/x-zip',
88:             'application/zip',
89:             'application/x-zip-compressed'
90:         ),
91:         'mid' => 'audio/midi',
92:         'midi' => 'audio/midi',
93:         'mpga' => 'audio/mpeg',
94:         'mp2' => 'audio/mpeg',
95:         'mp3' => array (
96:             'audio/mpeg',
97:             'audio/mpg',
98:             'audio/mpeg3',
99:             'audio/mp3'
100:        ),
101:        'aif' => 'audio/x-aiff',
102:        'aiff' => 'audio/x-aiff',
103:        'aifc' => 'audio/x-aiff',
104:        'ram' => 'audio/x-pn-realaudio',
105:        'rm' => 'audio/x-pn-realaudio',
106:        'rpm' => 'audio/x-pn-realaudio-plugin',
107:        'ra' => 'audio/x-realaudio',
108:        'rv' => 'video/vnd.rn-realvideo',
109:        'wav' => array (
110:            'audio/x-wav',
111:            'audio/wave',
112:            'audio/wav'
113:        ),
114:        'bmp' => array (
115:            'image/bmp',
116:            'image/x-windows-bmp'
117:        ),
118:        'gif' => 'image/gif',
119:        'jpeg' => array (
120:            'image/jpeg',
121:            'image/pjpeg'
122:        ),
123:        'jpg' => array (
124:            'image/jpeg',
125:            'image/pjpeg'
126:        ),
```

mimes.php

```
127:         'jpe' => array (
128:             'image/jpeg',
129:             'image/pjpeg'
130:         ),
131:         'png' => array (
132:             'image/png',
133:             'image/x-png'
134:         ),
135:         'tiff' => 'image/tiff',
136:         'tif' => 'image/tiff',
137:         'css' => 'text/css',
138:         'html' => 'text/html',
139:         'htm' => 'text/html',
140:         'shtml' => 'text/html',
141:         'txt' => 'text/plain',
142:         'text' => 'text/plain',
143:         'log' => array (
144:             'text/plain',
145:             'text/x-log'
146:         ),
147:         'rtx' => 'text/richtext',
148:         'rtf' => 'text/rtf',
149:         'xml' => 'text/xml',
150:         'xsl' => 'text/xml',
151:         'mpeg' => 'video/mpeg',
152:         'mpg' => 'video/mpeg',
153:         'mpe' => 'video/mpeg',
154:         'qt' => 'video/quicktime',
155:         'mov' => 'video/quicktime',
156:         'avi' => 'video/x-msvideo',
157:         'movie' => 'video/x-sgi-movie',
158:         'doc' => 'application/msword',
159:         'docx' => array (
160:             'application/vnd.openxmlformats-officedocument.wordproc
essingml.document',
161:             'application/zip'
162:         ),
163:         'xlsx' => array (
164:             'application/vnd.openxmlformats-officedocument.spreadsh
eetml.sheet',
165:             'application/zip'
166:         ),
167:         'word' => array (
168:             'application/msword',
169:             'application/octet-stream'
170:         ),
171:         'xl' => 'application/excel',
172:         'eml' => 'message/rfc822',
173:         'json' => array (
174:             'application/json',
175:             'text/json'
176:         )
177: );
178:
179:
180: /* End of file mimes.php */
181: /* Location: ./application/config/mimes.php */
```

database.php

```
1: <?php
2:
3: if (! defined ( 'BASEPATH' ))
4:     exit ( 'No direct script access allowed' );
5:     /*
6:  * | -----
7:  * | DATABASE CONNECTIVITY SETTINGS
8:  * | -----
9:  * | This file will contain the settings needed to access your database.
10: * |
11: * | For complete instructions please consult the 'Database Connection'
12: * | page of the User Guide.
13: * |
14: * | -----
15: * | EXPLANATION OF VARIABLES
16: * | -----
17: * |
18: * | ['hostname'] The hostname of your database server.
19: * | ['username'] The username used to connect to the database
20: * | ['password'] The password used to connect to the database
21: * | ['database'] The name of the database you want to connect to
22: * | ['dbdriver'] The database type. ie: mysql. Currently supported:
23: * | mysql, mysqli, postgre, odbc, mssql, sqlite, oci8
24: * | ['dbprefix'] You can add an optional prefix, which will be added
25: * | to the table name when using the Active Record class
26: * | ['pconnect'] TRUE/FALSE - Whether to use a persistent connection
27: * | ['db_debug'] TRUE/FALSE - Whether database errors should be displayed.
28: * | ['cache_on'] TRUE/FALSE - Enables/disables query caching
29: * | ['cachedir'] The path to the folder where cache files should be stored
30: * | ['char_set'] The character set used in communicating with the database
31: * | ['dbcollat'] The character collation used in communicating with the database
32: * | NOTE: For MySQL and MySQLi databases, this setting is only used
33: * | as a backup if your server is running PHP < 5.2.3 or MySQL < 5.0.7
34: * | (and in table creation queries made with DB Forge).
35: * | There is an incompatibility in PHP with mysql_real_escape_string() which
36: * | can make your site vulnerable to SQL injection if you are using a
37: * | multi-byte character set and are running versions lower than these.
38: * | Sites using Latin-1 or UTF-8 database character set and collation are unaffected.
39: * | ['swap_pre'] A default table prefix that should be swapped with the dbprefix
40: * | ['autoinit'] Whether or not to automatically initialize the database.
41: * | ['stricton'] TRUE/FALSE - forces 'Strict Mode' connections
42: * | - good for ensuring strict SQL while developing
43: * |
44: * | The $active_group variable lets you choose which connection group to
45: * | make active. By default there is only one group (the 'default' group).
46: * |
47: * | The $active_record variables lets you determine whether or not to load
48: * | the active record class
49: */
50:
51: $active_group = 'default';
52: $active_record = TRUE;
53:
54: $db ['default'] ['hostname'] = 'localhost';
55: $db ['default'] ['username'] = 'root';
56: $db ['default'] ['password'] = '';
57: $db ['default'] ['database'] = 'MHIC_CRM';
58: $db ['default'] ['dbdriver'] = 'mysql';
59: $db ['default'] ['dbprefix'] = '';
60: $db ['default'] ['pconnect'] = TRUE;
61: $db ['default'] ['db_debug'] = TRUE;
62: $db ['default'] ['cache_on'] = FALSE;
63: $db ['default'] ['cachedir'] = '';
```

database.php

```
64: $db ['default'] ['char_set'] = 'utf8';
65: $db ['default'] ['dbcollat'] = 'utf8_general_ci';
66: $db ['default'] ['swap_pre'] = '';
67: $db ['default'] ['autoinit'] = TRUE;
68: $db ['default'] ['stricton'] = FALSE;
69:
70:
71: /* End of file database.php */
72: /* Location: ./application/config/database.php */
```

index.html

```
1: <html>
2: <head>
3: <title>403 Forbidden</title>
4: </head>
5: <body>
6:
7:     <p>Directory access is forbidden.</p>
8:
9: </body>
10: </html>
```

index.html

```
1: <html>
2: <head>
3: <title>403 Forbidden</title>
4: </head>
5: <body>
6:
7:     <p>Directory access is forbidden.</p>
8:
9: </body>
10: </html>
```

index.html

```
1: <html>
2: <head>
3: <title>403 Forbidden</title>
4: </head>
5: <body>
6:
7:     <p>Directory access is forbidden.</p>
8:
9: </body>
10: </html>
```

index.html

```
1: <html>
2: <head>
3: <title>403 Forbidden</title>
4: </head>
5: <body>
6:
7:     <p>Directory access is forbidden.</p>
8:
9: </body>
10: </html>
```


toc.html

107:
108:
109:
Supported Features
110:
Application Flow
111:
112:
-View-Controller
113:
Architectural Goals
114:
115:
116:
117:
118:
Introduction
119:
120:
121:
122:
123:
124:
125:
126:
127:
128:
129:
130:
131:
132:
133:
134:
135:
136:
137:
138:
139:
140:
141:
142:
143:
144:
145:
146:

```
<li><a href="./overview/cheatsheets.htm
    Cheatsheets</a></li>
<li><a href="./overview/features.html">
<li><a href="./overview/appflow.html">A
    Chart</a></li>
<li><a href="./overview/mvc.html">Model
<li><a href="./overview/goals.html">Arc

</ul>
<h3>Tutorial</h3>
<ul>
<li><a href="./tutorial/index.html">Int
<li><a href="./tutorial/static_pages.ht
<li><a href="./tutorial/news_section.ht
<li><a href="./tutorial/create_news_ite
    news items</a></li>
<li><a href="./tutorial/conclusion.html

</ul>
</td>
<td valign="top" width="25%">

<h3>General Topics</h3>
<ul>
<li><a href="./general/urls.html">CodeI
<li><a href="./general/controllers.html
<li><a href="./general/reserved_names.h
    Names</a></li>
<li><a href="./general/views.html">View
<li><a href="./general/models.html">Mod
<li><a href="./general/helpers.html">He
<li><a href="./general/libraries.html">
    Libraries</a></li>
<li><a href="./general/creating_librari
    Your Own Libraries</a><
<li><a href="./general/drivers.html">Us
    Drivers</a></li>
<li><a href="./general/creating_drivers
    Your Own Drivers</a></l
<li><a href="./general/core_classes.htm
```

```

toc.html
    147:
    148:
s - Extending the
    149:
    150:
>Auto-loading
    151:
    152:
.html">Common
    153:
    154:
I Routing</a></li>
    155:
or Handling</a></li>
    156:
ching</a></li>
    157:
Profiling Your
    158:
    159:
g via the CLI</a></li>
    160:
ml">Managing
    161:
    162:
l">Handling
    163:
a></li>
    164:
html">Alternative
    165:
    166:
ecurity</a></li>
    167:
>PHP Style Guide</a></li>
    168:
iting
    169:
    170:
    171:
    172:
    173:
    174:
    175:
om/">Community
    176:
    177:
/CodeIgniter/wiki">Community
    178:
    179:
    180:
    181:
    182:
    183:
    184:
    185:
    186:
">Benchmarking
    187:
    188:
>Calendar Class</a></li>
    189:
t Class</a></li>

```

```

Classes</a></li>
<li><a href="./general/hooks.html">Hook
    Core</a></li>
<li><a href="./general/autoloader.html"
    Resources</a></li>
<li><a href="./general/common_functions
    Functions</a></li>
<li><a href="./general/routing.html">UR
<li><a href="./general/errors.html">Err
<li><a href="./general/caching.html">Ca
<li><a href="./general/profiling.html">
    Application</a></li>
<li><a href="./general/cli.html">Runnin
<li><a href="./general/managing_apps.ht
    Applications</a></li>
<li><a href="./general/environments.htm
    Multiple Environments</
<li><a href="./general/alternative_php.
    PHP Syntax</a></li>
<li><a href="./general/security.html">S
<li><a href="./general/styleguide.html"
<li><a href="./doc_style/index.html">Wr
    Documentation</a></li>
</ul>
<h3>Additional Resources</h3>
<ul>
<li><a href="http://forum.codeigniter.c
    Forums</a></li>
<li><a href="https://github.com/bcit-ci
    Wiki</a></li>
</ul>
</td>
<td valign="top" width="25%">
<h3>Class Reference</h3>
<ul>
<li><a href="./libraries/benchmark.html
    Class</a></li>
<li><a href="./libraries/calendar.html"
<li><a href="./libraries/cart.html">Car

```

toc.html

190:
onfig Class
191:
ail Class
192:
l">Encryption
193:
194:
.html">File
195:
>
196:
n.html">Form
197:
i>
198:
Class
199:
ML Table Class
200:
>Image
201:

202:
put Class
203:
l">Javascript
204:
205:
oader Class
206:
>Language Class
207:
>Migration Class
208:
utput Class
209:
l">Pagination
210:
211:
>Security Class
212:
>Session Class
213:
>Trackback Class
214:
emplate Parser
215:
216:
l">Typography
217:
218:
tml">Unit Testing
219:
220:
Class
221:
l">User Agent
222:
223:
ML-RPC Class
224:
Encoding Class

C
Em
<a href="./libraries/encryption.htm
Class
<a href="./libraries/file_uploading
Uploading Class
<a href="./libraries/form_validatio
Validation Class
FTP
HT
<a href="./libraries/image_lib.html
Manipulation Class<
In
<a href="./libraries/javascript.htm
Class
L
<a href="./libraries/language.html"
<a href="./libraries/migration.html
O
<a href="./libraries/pagination.htm
Class
<a href="./libraries/security.html"
<a href="./libraries/sessions.html"
<a href="./libraries/trackback.html
T
Class
<a href="./libraries/typography.htm
Class
<a href="./libraries/unit_testing.h
Class
URI
<a href="./libraries/user_agent.htm
Class
X
Zip

toc.html

```
225:                                     </ul>
226:
227:
228:                                     </td>
229:                                     <td valign="top" width="25%">
230:
231:                                     <h3>Driver Reference</h3>
232:                                     <ul>
233:                                         <li><a href="./libraries/caching.html">
Caching Class</a></li>
234:                                         <li><a href="./database/index.html">Dat
abase Class</a></li>
235:                                         <li><a href="./libraries/javascript.htm
l">Javascript
236:                                             Class</a></li>
237:                                     </ul>
238:
239:                                     <h3>Helper Reference</h3>
240:                                     <ul>
241:                                         <li><a href="./helpers/array_helper.htm
l">Array Helper</a></li>
242:                                         <li><a href="./helpers/captcha_helper.h
tml">CAPTCHA
243:                                             Helper</a></li>
244:                                         <li><a href="./helpers/cookie_helper.ht
ml">Cookie Helper</a></li>
245:                                         <li><a href="./helpers/date_helper.html
">Date Helper</a></li>
246:                                         <li><a href="./helpers/directory_helper
.html">Directory
247:                                             Helper</a></li>
248:                                         <li><a href="./helpers/download_helper.
html">Download
249:                                             Helper</a></li>
250:                                         <li><a href="./helpers/email_helper.htm
l">Email Helper</a></li>
251:                                         <li><a href="./helpers/file_helper.html
">File Helper</a></li>
252:                                         <li><a href="./helpers/form_helper.html
">Form Helper</a></li>
253:                                         <li><a href="./helpers/html_helper.html
">HTML Helper</a></li>
254:                                         <li><a href="./helpers/inflector_helper
.html">Inflector
255:                                             Helper</a></li>
256:                                         <li><a href="./helpers/language_helper.
html">Language
257:                                             Helper</a></li>
258:                                         <li><a href="./helpers/number_helper.ht
ml">Number Helper</a></li>
259:                                         <li><a href="./helpers/path_helper.html
">Path Helper</a></li>
260:                                         <li><a href="./helpers/security_helper.
html">Security
261:                                             Helper</a></li>
262:                                         <li><a href="./helpers/smiley_helper.ht
ml">Smiley Helper</a></li>
263:                                         <li><a href="./helpers/string_helper.ht
ml">String Helper</a></li>
264:                                         <li><a href="./helpers/text_helper.html
">Text Helper</a></li>
265:                                         <li><a href="./helpers/typography_helpe
r.html">Typography
```

toc.html

```
266:                                     Helper</a></li>
267:                                     <li><a href="./helpers/url_helper.html"
>URL Helper</a></li>
268:                                     <li><a href="./helpers/xml_helper.html"
>XML Helper</a></li>
269:                                     </ul>
270:
271:
272:
273:
274:                                     </td>
275:                                     </tr>
276:                                 </table>
277:
278: </div>
279: <!-- END CONTENT -->
280:
281:
282: <div id="footer">
283:     <p>
284:         <a href="#top">Top of Page</a>
285:     </p>
286:     <p>
287:         <a href="http://codeigniter.com">CodeIgniter</a> &nbsp;&#x2013;&nbsp;
&nbsp;&#x2013;&nbsp;
288:         Copyright &#x2013; 2006 - 2014 &nbsp;&#x2013;&nbsp;&nbsp;&nbsp;<a
289:             href="http://ellislab.com/">EllisLab, Inc.</a> &nbsp;&#x2013;&nbsp;&#x2013;&nbsp;
&nbsp;&#x2013;&nbsp;
290:         Copyright &#x2013; 2014 - 2015 &nbsp;&#x2013;&nbsp;&nbsp;&nbsp;<a
291:             href="http://bcit.ca/">British Columbia Institute of Te
chnology</a>
292:     </p>
293: </div>
294:
295:
296: </body>
297: </html>
```


index.html

```
58:                                 value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;   <input
59:                                 type="text" class="input" style="width:
    200px;" name="q" id="q"
60:                                 size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
61:                                 class="submit" name="sa" value="Go" />
62:                                 </form></td>
63:                                 </tr>
64:          </table>
65:
66:
67:
68:          <br clear="all" />
69:
70:          <div class="center">
71:              
73:          </div>
74:
75:
76:          <!-- START CONTENT -->
77:          <div id="content">
78:
79:
80:
81:              <h2>Welcome to CodeIgniter</h2>
82:
83:              <p>CodeIgniter is an Application Development Framework - a toolkit
84:                  - for people who build web sites using PHP. Its goal is to enab
le you
85:                  to develop projects much faster than you could if you were writ
ing
86:                  code from scratch, by providing a rich set of libraries for com
monly
87:                  needed tasks, as well as a simple interface and logical structu
re to
88:                  access these libraries. CodeIgniter lets you creatively focus o
n your
89:                  project by minimizing the amount of code needed for a given tas
k.</p>
90:
91:
92:              <h2>Who is CodeIgniter For?</h2>
93:
94:              <p>CodeIgniter is right for you if:</p>
95:
96:              <ul>
97:                  <li>You want a framework with a small footprint.</li>
98:                  <li>You need exceptional performance.</li>
99:                  <li>You need broad compatibility with standard hosting accounts
100:                     that run a variety of PHP versions and configurations.<
/li>
101:                  <li>You want a framework that requires nearly zero
102:                     configuration.</li>
103:                  <li>You want a framework that does not require you to use the
104:                     command line.</li>
105:                  <li>You want a framework that does not require you to adhere to
106:                     restrictive coding rules.</li>
107:                  <li>You do not want to be forced to learn a templating language
108:                     (although a template parser is optionally available if
you desire
109:                     one).</li>
```

index.html

```
110:                 <li>You eschew complexity, favoring simple solutions.</li>
111:                 <li>You need clear, thorough documentation.</li>
112:             </ul>
113:
114:
115:         </div>
116:         <!-- END CONTENT -->
117:
118:
119:         <div id="footer">
120:             <p>
121:                 <a href="#top">Top of Page</a>
122:             </p>
123:             <p>
124:                 <a href="http://codeigniter.com">CodeIgniter</a> &nbsp;&#x2022;
&nbsp;&#x2022;
125:                 Copyright &#x2019; 2006 - 2014 &nbsp;&#x2022;&nbsp;&#x2022; <a
126:                     href="http://ellislab.com/">EllisLab, Inc.</a> &nbsp;&#x2022;&#x2022;
&#x2022;&nbsp;&#x2022;
127:                 Copyright &#x2019; 2014 - 2015 &nbsp;&#x2022;&nbsp;&#x2022; <a
128:                     href="http://bcit.ca/">British Columbia Institute of Te
chnology</a>
129:             </p>
130:         </div>
131:
132:
133:
134: </body>
135: </html>
```


index.html

```
109: Copyright © 2006 - 2014 · EllisLab, Inc. · &
110: EllisLab, Inc. · &
iddot;·&
111: Copyright © 2014 - 2015 · British Columbia Institute of Te
112: British Columbia Institute of Te
chnology·&
113: British Columbia Institute of Te·&
114: British Columbia Institute of Te·&
115: British Columbia Institute of Te·&
116: British Columbia Institute of Te·&
117: British Columbia Institute of Te·&
```


template.html

```
56:
57:     <br clear="all" />
58:
59:
60:     <!-- START CONTENT -->
61:     <div id="content">
62:
63:
64:         <h1>Foo Class</h1>
65:
66:         <p>Brief description of Foo Class. If it extends a native
67:             CodeIgniter class, please link to the class in the CodeIgniter
68:             documents here.</p>
69:
70:         <p class="important">
71:             <strong>Important:</strong>&nbsp; This is an important note wit
72:                 <kbd>EMPHASIS</kbd>
73:                 .
74:         </p>
75:
76:         <p>Features:</p>
77:
78:         <ul>
79:             <li>Foo</li>
80:             <li>Bar</li>
81:         </ul>
82:
83:         <h2>Usage Heading</h2>
84:
85:         <p>
86:             Within a text string,
87:             <var>highlight variables</var>
88:             using
89:             <var>&lt;var&gt;&lt;/var&gt;</var></var>
90:             tags, and
91:             <dfn>highlight code</dfn>
92:             using the
93:             <dfn>&lt;dfn&gt;&lt;/dfn&gt;</dfn></dfn>
94:             tags.
95:         </p>
96:
97:         <h3>Sub-heading</h3>
98:
99:         <p>
100:             Put code examples within
101:             <dfn>&lt;code&gt;&lt;/code&gt;</dfn>
102:             tags:
103:         </p>
104:
105:         <code>
106:             $this->load->library('foo');<br /> <br /> $this->foo->bar('bat'
107:         </code>
108:
109:
110:         <h2>Table Preferences</h2>
111:
112:         <p>Use tables where appropriate for long lists of preferences.</p>
113:
114:
115:         <table cellpadding="0" cellspacing="1" border="0" style="width: 100%"
116:             class="tableborder">
```


captcha_helper.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>CAPTCHA Helper : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype_lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28: <!-- START NAVIGATION -->
29: <div id="nav">
30: <div id="nav_inner">
31: <script type="text/javascript">create_menu('../');</script>
32: </div>
33: </div>
34: <div id="nav2">
35: <a name="top"></a><a href="javascript:void(0);"
36: onclick="myHeight.toggle();"></a>
40: </div>
41: <div id="masthead">
42: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43: <tr>
44: <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45: <td id="breadcrumb_right"><a href="../toc.html">Table of
46: Contents Page</a></td>
47: </tr>
48: </table>
49: </div>
50: <!-- END NAVIGATION -->
51:
52: <!-- START BREADCRUMB -->
53: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54: <tr>
55: <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
56: er
57: <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="../index.
html">User Guide
58: <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp; CAPTCHA Helper</td>
59: </tr>
60: </table>
61: </div>
62: </body>
63: </html>
```

captcha_helper.html

```
59:         <td id="searchbox"><form method="get"
60:             action="http://www.google.com/search">
61:             <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                 value="codeigniter.com/user_guide/" />S
earch User Guide<input
63:         type="text" class="input" style="width:
200px;" name="q" id="q"
64:         size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:         class="submit" name="sa" value="Go" />
66:             </form></td>
67:         </tr>
68:     </table>
69:     <!-- END BREADCRUMB -->
70:
71:     <br clear="all" />
72:
73:
74:     <!-- START CONTENT -->
75:     <div id="content">
76:
77:
78:         <h1>CAPTCHA Helper</h1>
79:
80:         <p>The CAPTCHA Helper file contains functions that assist in
81:             creating CAPTCHA images.</p>
82:
83:
84:         <h2>Loading this Helper</h2>
85:
86:         <p>This helper is loaded using the following code:</p>
87:         <code>$this->load->helper('captcha');</code>
88:
89:         <p>The following functions are available:</p>
90:
91:         <h2>
92:             create_captcha(
93:                 <var>$data</var>
94:             )
95:         </h2>
96:
97:         <p>Takes an array of information to generate the CAPTCHA as input
98:             and creates the image to your specifications, returning an arra
y of
99:             associative data about the image.</p>
100:
101:         <code>
102:             [array]<br /> ( <br /> &nbsp;&nbsp;&nbsp;'image' => IMAGE TAG<br />
103:             &nbsp;&nbsp;&nbsp;'time' => TIMESTAMP (in microtime)<br />
104:             &nbsp;&nbsp;&nbsp;'word' => CAPTCHA WORD<br /> )
105:         </code>
106:
107:         <p>
108:             The "image" is the actual image tag:
109:             <code>&lt;img
110:                 src=&quot;http://example.com/captcha/12345.jpg&quot;;
111:                 width=&quot;140&quot;; height=&quot;50&quot;; /&gt;</code>
112:
113:         </p>
114:         <p>The "time" is the micro timestamp used as the image name
115:             without the file extension. It will be a number like this:
```

captcha_helper.html

```

116:                1139612155.3422</p>
117:
118:        <p>The "word" is the word that appears in the captcha image, which
119:                if not supplied to the function, will be a random string.</p>
120:
121:        <h3>Using the CAPTCHA helper</h3>
122:
123:        <p>Once loaded you can generate a captcha like this:</p>
124:
125:        <code>
126:                $vals = array(<br /> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;'word' => 'Random
127:                word',<br /> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;'img_path' => './captcha/'
128:                &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;'img_url' => 'http://example.com/captch
129:                &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;'font_path' => './path/to/fonts/texb.tt
130:                &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;'img_width' => '150',<br />
131:                &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;'img_height' => 30,<br />
132:                &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;'expiration' => 7200<br />
133:                &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;);<br /> <br /> $cap =
134:                create_captcha($vals);<br /> echo $cap['image'];
135:        </code>
136:
137:        <ul>
138:                <li>The captcha function requires the GD image library.</li>
139:                <li>Only the img_path and img_url are required.</li>
140:                <li>If a "word" is not supplied, the function will generate a
141:                        random ASCII string. You might put together your own wo
142:                        rd library
143:                        that you can draw randomly from.</li>
144:                <li>If you do not specify a path to a TRUE TYPE font, the nativ
145:                        e
146:                        ugly GD font will be used.</li>
147:                <li>The "captcha" folder must be writable (666, or 777)</li>
148:                <li>The "expiration" (in seconds) signifies how long an image
149:                        ed. The
150:                        will remain in the captcha folder before it will be del
151:                        eted. The
152:                        default is two hours.</li>
153:        </ul>
154:        <h3>Adding a Database</h3>
155:
156:        <p>
157:                In order for the captcha function to prevent someone from submi
158:                tting,
159:                you will need to add the information returned from
160:                <code>create_captcha()</code>
161:                function to your database. Then, when the data from the form is
162:                submitted by the user you will need to verify that the data exi
163:                sts in
164:                the database and has not expired.
165:        </p>
166:
167:        <p>Here is a table prototype:</p>
168:
169:        <code>
170:                CREATE TABLE captcha (<br /> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;captcha_id bigint(13) unsign
171:                ed
172:                NOT NULL auto_increment,<br /> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;captcha_time int(10) unsig
173:                ned
174:                NOT NULL,<br /> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;ip_address varchar(16) default '0' NOT NU
175:                LL,<br />

```

captcha_helper.html

```
168:          &nbsp;word varchar(20) NOT NULL,<br /> &nbsp;PRIMARY KEY
169:          'captcha_id' ('captcha_id'),<br /> &nbsp;KEY 'word' ('word')<br
/>
170:          );
171:      </code>
172:
173:      <p>Here is an example of usage with a database. On the page where
174:      the CAPTCHA will be shown you'll have something like this:</p>
175:
176:      <code>
177:          $this-&gt;load-&gt;helper(&#x27;captcha&#x27;);<br /> $vals = a
rray(<br />
178:          &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&#x27;img_path&#x27; =&gt;
179:          &#x27;./captcha/&#x27;,<br />
180:          &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&#x27;img_url&#x27; =&gt;
181:          &#x27;http://example.com/captcha/&#x27;<br />
182:          &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&#x27;);<br /> <br /> $cap =
183:          create_captcha($vals);<br /> <br /> $data = array(<br />
184:          &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&#x27;captcha_time&#x27; =&gt;
185:          $cap[&#x27;time&#x27;],<br />
186:          &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&#x27;ip_address&#x27; =&gt;
187:          $this-&gt;input-&gt;ip_address(),<br />
188:          &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&#x27;word&#x27; =&gt; $cap[&#x27;word&
&#x27;];<br />
189:          &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&#x27;);<br /> <br /> $query =
190:          $this-&gt;db-&gt;insert_string(&#x27;captcha&#x27;,&nbsp; $data);<br
/>
191:          $this-&gt;db-&gt;query($query);<br /> <br /> echo &#x27;Submit
the
192:          word you see below:&#x27;;<br /> echo $cap[&#x27;image&#x27;];<
br />
193:          echo &#x27;&lt;input type=&quot;text&quot; name=&quot;captcha&q
uot;
194:          value=&quot;&quot; /&gt;&#x27;;
195:      </code>
196:
197:      <p>Then, on the page that accepts the submission you'll have
198:      something like this:</p>
199:
200:      <code>
201:          // First, delete old captchas<br /> $expiration = time()-7200;
//
202:          Two hour limit<br /> $this-&gt;db-&gt;query(&quot;DELETE FROM
203:          captcha WHERE captcha_time &lt; &quot;.$expiration); <br /> <br
/>
204:          // Then see if a captcha exists:<br /> $sql = &quot;SELECT COUN
T(*)
205:          AS count FROM captcha WHERE word = ? AND ip_address = ? AND
206:          captcha_time &gt; ?&quot;;<br /> $binds =
207:          array($_POST[&#x27;captcha&#x27;], $this-&gt;input-&gt;ip_adre
ss(),
208:          $expiration);<br /> $query = $this-&gt;db-&gt;query($sql, $bind
s);<br />
209:          $row = $query-&gt;row();<br /> <br /> if ($row-&gt;count == 0)<
br />
210:          {<br /> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;echo &quot;You must submit the
word
211:          that appears in the image&quot;;<br /> }
212:      </code>
213:
214:  </div>
215:  <!-- END CONTENT -->
216:
```


file_helper.html

```
60:                <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
61:                value="codeigniter.com/user_guide/" />S
earch User Guide<input
62:                type="text" class="input" style="width:
200px;" name="q" id="q"
63:                size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
64:                class="submit" name="sa" value="Go" />
65:                </form></td>
66:            </tr>
67:        </table>
68:        <!-- END BREADCRUMB -->
69:
70:        <br clear="all" />
71:
72:
73:        <!-- START CONTENT -->
74:        <div id="content">
75:
76:
77:            <h1>File Helper</h1>
78:
79:            <p>The File Helper file contains functions that assist in working
80:                with files.</p>
81:
82:
83:            <h2>Loading this Helper</h2>
84:
85:            <p>This helper is loaded using the following code:</p>
86:            <code>$this->load->helper('file');</code>
87:
88:            <p>The following functions are available:</p>
89:
90:            <h2>
91:                read_file('
92:                    <var>path</var>
93:                ')
94:            </h2>
95:
96:            <p>Returns the data contained in the file specified in the path.
97:                Example:</p>
98:
99:            <code>$string = read_file('./path/to/file.php');</code>
100:
101:            <p>The path can be a relative or full server path. Returns FALSE
102:                (boolean) on failure.</p>
103:
104:            <p class="important">
105:                <strong>Note:</strong> The path is relative to your main site
106:                index.php file, NOT your controller or view files. CodeIgniter
uses a
107:                front controller so paths are always relative to the main site
index.
108:            </p>
109:
110:            <p>If your server is running an open_basedir restriction this
111:                function might not work if you are trying to access a file abov
e the
112:                calling script.</p>
113:
114:            <h2>
115:                write_file('
```


file_helper.html

```
be
172:         deleted as well. Example:
173:     </p>
174:
175:     <code>delete_files('./path/to/directory/', TRUE);</code>
176:
177:     <p class="important">
178:         <strong>Note:</strong> The files must be writable or owned by t
he
179:         system in order to be deleted.
180:     </p>
181:
182:     <h2>
183:         get_filenames('
184:         <var>path/to/directory/</var>
185:         ')
186:     </h2>
187:
188:     <p>Takes a server path as input and returns an array containing
189:         the names of all files contained within it. The file path can
190:         optionally be added to the file names by setting the second par
ameter
191:         to TRUE.</p>
192:
193:     <h2>
194:         get_dir_file_info('
195:         <var>path/to/directory/</var>
196:         ',
197:         <kbd>$stop_level_only</kbd>
198:         = TRUE)
199:     </h2>
200:
201:     <p>
202:         Reads the specified directory and builds an array containing th
e
203:         filenames, filesize, dates, and permissions. Sub-folders contai
ned
204:         within the specified path are only read if forced by sending th
e
205:         second parameter,
206:         <kbd>$stop_level_only</kbd>
207:         to
208:         <samp>FALSE</samp>
209:         , as this can be an intensive operation.
210:     </p>
211:
212:     <h2>
213:         get_file_info('
214:         <var>path/to/file</var>
215:         ',
216:         <kbd>$file_information</kbd>
217:         )
218:     </h2>
219:
220:     <p>Given a file and path, returns the name, path, size, date
221:         modified. Second parameter allows you to explicitly declare wha
t
222:         information you want returned; options are: name, server_path,
size,
223:         date, readable, writable, executable, fileperms. Returns FALSE
if the
224:         file cannot be found.</p>
225:
```

file_helper.html

```
226: <p class="important">
227: <strong>Note:</strong> The &quot;writable&quot; uses the PHP fu
nction
228: is_writable() which is known to have issues on the IIS webserve
r.
229: Consider using fileperms instead, which returns information fro
m
230: PHP's fileperms() function.
231: </p>
232: <h2>
233: get_mime_by_extension('
234: <var>file</var>
235: ')
236: </h2>
237:
238: <p>Translates a file extension into a mime type based on
239: config/mimes.php. Returns FALSE if it can't determine the type,
or
240: open the mime config file.</p>
241: <p>
242: <code>
243: $file = &quot;somefile.png&quot;;<br /> echo $file . '
244: mime type of ' . get_mime_by_extension($file);
245: </code>
246: </p>
247: <p class="critical">
248: <strong>Note:</strong> This is not an accurate way of determini
ng
249: file mime types, and is here strictly as a convenience. It shou
ld not
250: be used for security.
251: </p>
252:
253: <h2>
254: symbolic_permissions(
255: <kbd>$perms</kbd>
256: )
257: </h2>
258:
259: <p>
260: Takes numeric permissions (such as is returned by
261: <kbd>fileperms()</kbd>
262: and returns standard symbolic notation of file permissions.
263: </p>
264:
265: <code>
266: echo symbolic_permissions(fileperms('./index.php'));<br /> <br
/>
267: // -rw-r--r--
268: </code>
269:
270: <h2>
271: octal_permissions(
272: <kbd>$perms</kbd>
273: )
274: </h2>
275:
276: <p>
277: Takes numeric permissions (such as is returned by
278: <kbd>fileperms()</kbd>
279: and returns a three character octal notation of file permission
s.
```


directory_helper.html

```
td>
60:           <td id="searchbox"><form method="get"
61:               action="http://www.google.com/search">
62:               <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
63:                   value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;   <input
64:                   type="text" class="input" style="width:
200px;" name="q" id="q"
65:                   size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
66:                   class="submit" name="sa" value="Go" />
67:               </form></td>
68:           </tr>
69: </table>
70: <!-- END BREADCRUMB -->
71:
72: <br clear="all" />
73:
74:
75: <!-- START CONTENT -->
76: <div id="content">
77:
78:         <h1>Directory Helper</h1>
79:
80:         <p>The Directory Helper file contains functions that assist in
81:             working with directories.</p>
82:
83:
84:         <h2>Loading this Helper</h2>
85:
86:         <p>This helper is loaded using the following code:</p>
87:         <code>$this->load->helper('directory');</code>
88:
89:         <p>The following functions are available:</p>
90:
91:         <h2>
92:             directory_map(
93:                 <var>source directory</var>
94:                 '
95:             )
96:         </h2>
97:
98:         <p>This function reads the directory path specified in the first
99:             parameter and builds an array representation of it and all its
100:            contained files. Example:</p>
101:
102:         <code>$map = directory_map('./mydirectory/');</code>
103:
104:         <p class="important">
105:             <strong>Note:</strong> Paths are almost always relative to your
106:            index.php file.
107:         </p>
108:
109:         <p>Sub-folders contained within the directory will be mapped as
110:            well. If you wish to control the recursion depth, you can do so
111:            the second parameter (integer). A depth of 1 will only map the
112:            level directory:</p>
113:
114:         <code>$map = directory_map('./mydirectory/', 1);</code>
```



```

smiley_helper.html
60:         action="http://www.google.com/search">
61:         <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:         value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp; <input
63:         type="text" class="input" style="width:
200px;" name="q" id="q"
64:         size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:         class="submit" name="sa" value="Go" />
66:         </form></td>
67:         </tr>
68:     </table>
69:     <!-- END BREADCRUMB -->
70:
71:     <br clear="all" />
72:
73:     <!-- START CONTENT -->
74:     <div id="content">
75:
76:
77:
78:         <h1>Smiley Helper</h1>
79:
80:         <p>The Smiley Helper file contains functions that let you manage
81:             smileys (emoticons).</p>
82:
83:
84:         <h2>Loading this Helper</h2>
85:
86:         <p>This helper is loaded using the following code:</p>
87:         <code>$this->load->helper('smiley');</code>
88:
89:         <h2>Overview</h2>
90:
91:         <p>
92:             The Smiley helper has a renderer that takes plain text smileys
, like
93:             <dfn>:-)</dfn>
94:             and turns them into a image representation, like 
97:         </p>
98:
99:         <p>It also lets you display a set of smiley images that when
100:             clicked will be inserted into a form field. For example, if you
have
101:             a blog that allows user commenting you can show the smileys nex
t to
102:             the comment form. Your users can click a desired smiley and wit
h the
103:             help of some JavaScript it will be placed into the form field.<
/p>
104:
105:
106:
107:         <h2>Clickable Smileys Tutorial</h2>
108:
109:         <p>Here is an example demonstrating how you might create a set of
110:             clickable smileys next to a form field. This example requires t
hat
111:             you first download and install the smiley images, then create a

```

smiley_helper.html

```
112:         controller and the View as described.</p>
113:
114:     <p class="important">
115:         <strong>Important:</strong> Before you begin, please <a
116:             href="http://ellislab.com/asset/ci_download_files/smiley
ys.zip">download
117:             the smiley images</a> and put them in a publicly access
ible place on
118:             your server. This helper also assumes you have the smiley repla
cement
119:             array located at
120:             <dfn>application/config/smileys.php</dfn>
121:     </p>
122:
123:
124:     <h3>The Controller</h3>
125:
126:     <p>
127:         In your
128:         <dfn>application/controllers/</dfn>
129:         folder, create a file called
130:         <kbd>smileys.php</kbd>
131:         and place the code below in it.
132:     </p>
133:
134:     <p>
135:         <strong>Important:</strong> Change the URL in the
136:         <dfn>get_clickable_smileys()</dfn>
137:         function below so that it points to your
138:         <dfn>smiley</dfn>
139:         folder.
140:     </p>
141:
142:     <p>
143:         You'll notice that in addition to the smiley helper we are usin
g the
144:         <a href="../../../libraries/table.html">Table Class</a>.
145:     </p>
146:
147:     <textarea class="textarea" style="width: 100%" cols="50" rows="25">
148: &lt;?php
149:
150: class Smileys extends CI_Controller {
151:
152:     function __construct()
153:     {
154:         parent::__construct();
155:     }
156:
157:     function index()
158:     {
159:         $this->load->helper('smiley');
160:         $this->load->library('table');
161:
162:         $image_array = get_clickable_smileys('http://example.com/images/smileys
/', 'comments');
163:
164:         $col_array = $this->table->make_columns($image_array, 8);
165:
166:         $data['smiley_table'] = $this->table->generate($col_array);
167:
168:         $this->load->view('smiley_view', $data);
169:     }
}
```

smiley_helper.html

```
170:
171: }
172: ?&gt;
173: </textarea>
174:
175:         <p>
176:             In your
177:             <dfn>application/views/</dfn>
178:             folder, create a file called
179:             <kbd>smiley_view.php</kbd>
180:             and place this code in it:
181:         </p>
182:
183:         <textarea class="textarea" style="width: 100%" cols="50" rows="20">
184: &lt;/html>
185: &lt;/head>
186: &lt;/title>Smileys&lt;/title>
187:
188: &lt;?php echo smiley_js(); ?>
189:
190: &lt;/head>
191: &lt;/body>
192:
193: &lt;form name="blog">
194: &lt;textarea name="comments" id="comments" cols="40" rows="4">&lt;/textarea>
195: &lt;/form>
196:
197: &lt;p>Click to insert a smiley!&lt;/p>
198:
199: &lt;?php echo $smiley_table; ?>
200:
201: &lt;/body>
202: &lt;/html>
203: </textarea>
204:
205:
206:         <p>
207:             When you have created the above controller and view, load it by
208:             visiting
209:             <dfn>http://www.example.com/index.php/smileys/</dfn>
210:         </p>
211:
212:
213:         <h3>Field Aliases</h3>
214:
215:         <p>When making changes to a view it can be inconvenient to have
216:             the field id in the controller. To work around this, you can gi
ve
217:             your smiley links a generic name that will be tied to a specifi
c id
218:             in your view.</p>
219:         <code>$image_array =
220:             get_smiley_links("http://example.com/images/smileys/",
221:                 "comment_textarea_alias");</code>
222:
223:         <p>To map the alias to the field id, pass them both into the
224:             smiley_js function:</p>
225:         <code>$image_array = smiley_js("comment_textarea_alias",
226:             "comments");</code>
227:
228:
229:         <h1>Function Reference</h1>
230:
```


smiley_helper.html

```
281: Copyright © 2006 - 2014 · <a
282: href="http://ellislab.com/">EllisLab, Inc.</a> · &
iddot;&nbsp;
283: Copyright © 2014 - 2015 · <a
284: href="http://bcit.ca/">British Columbia Institute of Te
chnology</a>
285: </p>
286: </div>
287:
288: </body>
289: </html>
```


string_helper.html

```
60:                                     action="http://www.google.com/search">
61:                                     <input type="hidden" name="as_sitestearch" id="a
s_sitestearch"
62:                                     value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;  <input
63:                                     type="text" class="input" style="width:
200px;" name="q" id="q"
64:                                     size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                                     class="submit" name="sa" value="Go" />
66:                                     </form></td>
67:                                     </tr>
68:                               </table>
69:                               <!-- END BREADCRUMB -->
70:
71:                               <br clear="all" />
72:
73:                               <!-- START CONTENT -->
74:                               <div id="content">
75:
76:
77:
78:                               <h1>String Helper</h1>
79:
80:                               <p>The String Helper file contains functions that assist in
81:                                     working with strings.</p>
82:
83:
84:                               <h2>Loading this Helper</h2>
85:
86:                               <p>This helper is loaded using the following code:</p>
87:                               <code>$this->load->helper('string');</code>
88:
89:                               <p>The following functions are available:</p>
90:
91:                               <h2>random_string()</h2>
92:
93:                               <p>Generates a random string based on the type and length you
94:                                     specify. Useful for creating passwords or generating random has
hes.</p>
95:
96:                               <p>The first parameter specifies the type of string, the second
97:                                     parameter specifies the length. The following choices are avail
able:</p>
98:
99:                               alpha, alnum, numeric, nozero, unique, md5, encrypt and sha1
100:                               <ul>
101:                                   <li><strong>alpha</strong>:&nbsp;  A string with lower and
102:                                           uppercase letters only.</li>
103:                                   <li><strong>alnum</strong>:&nbsp;  Alpha-numeric string with
104:                                           lower and uppercase characters.</li>
105:                                   <li><strong>numeric</strong>:&nbsp;  Numeric string.</li>
106:                                   <li><strong>nozero</strong>:&nbsp;  Numeric string with no zeros
.</li>
107:                                   <li><strong>unique</strong>:&nbsp;  Encrypted with MD5 and
108:                                           uniqid(). Note: The length parameter is not available f
or this type.
109:                                           Returns a fixed length 32 character string.</li>
110:                                   <li><strong>sha1</strong>:&nbsp;  An encrypted random number
111:                                           based on <code>do_hash()</code> from the <a
112:                                           href="security_helper.html">security helper</a>.</li>
113:                               </ul>
114:
```


string_helper.html

```
175:         &quot;;http://example.com/index.php&quot;;
176:     </code>
177: <h2>trim_slashes()</h2>
178: <p>
179:     Removes any leading/trailing slashes from a string. Example:<br />
180:     <code>
181:         $string = &quot;;/this/that/theother/&quot;;<br /> echo
182:         trim_slashes($string); // results in this/that/theother
183:     </code>
184: </p>
185:
186:
187: <h2>reduce_multiples()</h2>
188: <p>Reduces multiple instances of a particular character occuring
189:     directly after each other. Example:</p>
190: <code>
191:     $string="Fred, Bill,, Joe, Jimmy";<br />
192:     $string=reduce_multiples($string,","); //results in "Fred, Bill
193:     , Joe,
194:     Jimmy"
195: </code>
196: <p>
197:     The function accepts the following parameters:
198:     <code>reduce_multiples(string: text to search in, string:
199:         character to reduce, boolean: whether to remove the cha
200:         racter from
201:         the front and end of the string)</code>
202:
203:     The first parameter contains the string in which you want to re
204:     duce
205:     the multiplies. The second parameter contains the character you
206:     want
207:     to have reduced. The third parameter is FALSE by default; if se
208:     t to
209:     TRUE it will remove occurrences of the character at the beginnin
210:     g and
211:     the end of the string. Example:
212:     <code>
213:         $string=",Fred, Bill,, Joe, Jimmy,";<br />
214:         $string=reduce_multiples($string, ",", TRUE); //result
215:         s in "Fred,
216:         Bill, Joe, Jimmy"
217:     </code>
218: </p>
219:
220: <h2>quotes_to_entities()</h2>
221: <p>Converts single and double quotes in a string to the
222:     corresponding HTML entities. Example:</p>
223: <code>
224:     $string="Joe's \"dinner\"";<br />
225:     $string=quotes_to_entities($string); //results in "Joe&#39;
226:     s
227:     &quot;dinner&quot;"
228: </code>
229:
230: <h2>strip_quotes()</h2>
231: <p>Removes single and double quotes from a string. Example:</p>
232: <code>
233:     $string="Joe's \"dinner\"";<br /> $string=strip_quotes($string)
234:     ;
235:     //results in "Joes dinner"
```


text_helper.html

```
60:                                     action="http://www.google.com/search">
61:                                     <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                                     value="codeigniter.com/user_guide/" />S
earch User Guide<input
63:                                     type="text" class="input" style="width:
200px;" name="q" id="q"
64:                                     size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                                     class="submit" name="sa" value="Go" />
66:                                     </form></td>
67:                                     </tr>
68:                                     </table>
69:                                     <!-- END BREADCRUMB -->
70:
71:                                     <br clear="all" />
72:
73:
74:                                     <!-- START CONTENT -->
75:                                     <div id="content">
76:
77:
78:                                     <h1>Text Helper</h1>
79:
80:                                     <p>The Text Helper file contains functions that assist in working
81:                                     with text.</p>
82:
83:
84:                                     <h2>Loading this Helper</h2>
85:
86:                                     <p>This helper is loaded using the following code:</p>
87:                                     <code>$this->load->helper('text');</code>
88:
89:                                     <p>The following functions are available:</p>
90:
91:
92:                                     <h2>word_limiter()</h2>
93:
94:                                     <p>
95:                                     Truncates a string to the number of <strong>words</strong> spec
ified.
96:                                     Example:
97:                                     </p>
98:
99:                                     <code>
100:                                     $string = "Here is a nice text string consisting of eleven word
s."<br />
101:                                     <br /> $string = word_limiter($string, 4);<br />
102:                                     <br /> // Returns: Here is a nice&#8230;
103:                                     </code>
104:
105:                                     <p>The third parameter is an optional suffix added to the string.
106:                                     By default it adds an ellipsis.</p>
107:
108:
109:                                     <h2>character_limiter()</h2>
110:
111:                                     <p>
112:                                     Truncates a string to the number of <strong>characters</strong>
113:                                     specified. It maintains the integrity of words so the character
count
114:                                     may be slightly more or less than what you specify. Example:
115:                                     </p>
```

text_helper.html

```
116:
117:     <code>
118:         $string = "Here is a nice text string consisting of eleven word
s."; <br />
119:         <br /> $string = character_limiter($string, 20); <br />
120:         <br /> // Returns: Here is a nice text string&#8230;
121:     </code>
122:
123:     <p>The third parameter is an optional suffix added to the string,
124:         if undeclared this helper uses an ellipsis.</p>
125:
126:
127:
128:     <h2>ascii_to_entities()</h2>
129:
130:     <p>Converts ASCII values to character entities, including high
131:         ASCII and MS Word characters that can cause problems when used
in a
132:         web page, so that they can be shown consistently regardless of
133:         browser settings or stored reliably in a database. There is som
e
134:         dependence on your server's supported character sets, so it may
not
135:         be 100% reliable in all cases, but for the most part it should
136:         correctly identify characters outside the normal range (like ac
cented
137:         characters). Example:</p>
138:
139:     <code>$string = ascii_to_entities($string);</code>
140:
141:
142:     <h2>entities_to_ascii()</h2>
143:
144:     <p>This function does the opposite of the previous one; it turns
145:         character entities back into ASCII.</p>
146:
147:     <h2>convert_accented_characters()</h2>
148:
149:     <p>Transliterates high ASCII characters to low ASCII equivalents,
150:         useful when non-English characters need to be used where only
151:         standard ASCII characters are safely used, for instance, in URL
s.</p>
152:
153:     <code>$string = convert_accented_characters($string);</code>
154:
155:     <p>
156:         This function uses a companion config file
157:         <dfn>application/config/foreign_chars.php</dfn>
158:         to define the to and from array for transliteration.
159:     </p>
160:
161:     <h2>word_censor()</h2>
162:
163:     <p>Enables you to censor words within a text string. The first
164:         parameter will contain the original string. The second will con
tain
165:         an array of words which you disallow. The third (optional) para
meter
166:         can contain a replacement value for the words. If not specified
they
167:         are replaced with pound signs: #####. Example:</p>
168:
169:     <code>
```

text_helper.html

```
170:             $disallowed = array('darn', 'shucks', 'golly', 'phooey');<br />
<br />
171:             $string = word_censor($string, $disallowed, 'Beep!');
172:         </code>
173:
174:
175:         <h2>highlight_code()</h2>
176:
177:         <p>Colorizes a string of code (PHP, HTML, etc.). Example:</p>
178:
179:         <code>$string = highlight_code($string);</code>
180:
181:         <p>The function uses PHP's highlight_string() function, so the
182:             colors used are the ones specified in your php.ini file.</p>
183:
184:
185:         <h2>highlight_phrase()</h2>
186:
187:         <p>Will highlight a phrase within a text string. The first
188:             parameter will contain the original string, the second will con-
tain
189:             the phrase you wish to highlight. The third and fourth paramete-
rs
190:             will contain the opening/closing HTML tags you would like the p-
hrase
191:             wrapped in. Example:</p>
192:
193:         <code>
194:             $string = "Here is a nice text string about nothing in particul-
ar."<br />
195:             <br /> $string = highlight_phrase($string, "nice text", '&lt;sp-
an
196:                 style="color:#990000">', '&lt;/span>');
197:         </code>
198:
199:         <p>The above text returns:</p>
200:
201:         <p>
202:             Here is a <span style="color: #990000">nice text</span> string
about
203:             nothing in particular.
204:         </p>
205:
206:
207:
208:         <h2>word_wrap()</h2>
209:
210:         <p>
211:             Wraps text at the specified <strong>character</strong> count wh-
ile
212:             maintaining complete words. Example:
213:         </p>
214:
215:         <code>
216:             $string = "Here is a simple string of text that will help us
217:             demonstrate this function."<br /> <br /> echo word_wrap($strin-
g,
218:             25);<br /> <br /> // Would produce:<br /> <br /> Here is a simp-
le
219:             string<br /> of text that will help<br /> us demonstrate this<b-
r />
220:             function
221:         </code>
```


array_helper.html

```

1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xh
tml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Array Helper : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype.lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28:     <!-- START NAVIGATION -->
29:     <div id="nav">
30:         <div id="nav_inner">
31:             <script type="text/javascript">create_menu('../');</script>
32:         </div>
33:     </div>
34:     <div id="nav2">
35:         <a name="top"></a><a href="javascript:void(0);"
36:         onclick="myHeight.toggle();"></a>
40:     </div>
41:     <div id="masthead">
42:         <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43:             <tr>
44:                 <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45:                 <td id="breadcrumb_right"><a href="../toc.html">Table o
46:                                     Contents Page</a></td>
47:             </tr>
48:         </table>
49:     </div>
50:     <!-- END NAVIGATION -->
51:
52:     <!-- START BREADCRUMB -->
53:     <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54:         <tr>
55:             <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
56:             er
57:             <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="../index.
58:             html">User Guide
59:             <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;Array Helper</td>
60:             <td id="searchbox"><form method="get"

```


inflector_helper.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml
tml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Inflector Helper : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12:       href='../userguide.css' />
13:
14: <script type="text/javascript" src=" ../nav/nav.js"></script>
15: <script type="text/javascript" src=" ../nav/prototype-lite.js"></script>
16: <script type="text/javascript" src=" ../nav/moo.fx.js"></script>
17: <script type="text/javascript" src=" ../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28:     <!-- START NAVIGATION -->
29:     <div id="nav">
30:         <div id="nav_inner">
31:             <script type="text/javascript">create_menu('../');</script>
32:         </div>
33:     </div>
34:     <div id="nav2">
35:         <a name="top"></a><a href="javascript:void(0);"
36:             onclick="myHeight.toggle();"></a>
40:     </div>
41:     <div id="masthead">
42:         <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43:             <tr>
44:                 <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45:                 <td id="breadcrumb_right"><a href=" ../toc.html">Table o
46:                                     Contents Page</a></td>
47:             </tr>
48:         </table>
49:     </div>
50:     <!-- END NAVIGATION -->
51:
52:     <!-- START BREADCRUMB -->
53:     <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54:         <tr>
55:             <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
er
57:             Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href=" ../index.
html">User Guide
58:             Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp; Inflector Helper</
td>
```

inflector_helper.html

```

59:          <td id="searchbox"><form method="get"
60:            action="http://www.google.com/search">
61:            <input type="hidden" name="as_sitesearch" id="as_
s_sitesearch"
62:              value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;  <input
63:            type="text" class="input" style="width:
200px;" name="q" id="q"
64:            size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:            class="submit" name="sa" value="Go" />
66:          </form></td>
67:        </tr>
68:      </table>
69:      <!-- END BREADCRUMB -->
70:
71:      <br clear="all" />
72:
73:
74:      <!-- START CONTENT -->
75:      <div id="content">
76:
77:
78:          <h1>Inflector Helper</h1>
79:
80:          <p>The Inflector Helper file contains functions that permits you
81:            to change words to plural, singular, camel case, etc.</p>
82:
83:
84:          <h2>Loading this Helper</h2>
85:
86:          <p>This helper is loaded using the following code:</p>
87:          <code>$this->load->helper('inflector');</code>
88:
89:          <p>The following functions are available:</p>
90:
91:
92:          <h2>singular()</h2>
93:
94:          <p>Changes a plural word to singular. Example:</p>
95:
96:          <code>
97:            $word = "dogs";<br /> echo singular($word); // Returns "dog"
98:          </code>
99:
100:
101:          <h2>plural()</h2>
102:
103:          <p>Changes a singular word to plural. Example:</p>
104:
105:          <code>
106:            $word = "dog";<br /> echo plural($word); // Returns "dogs"
107:          </code>
108:
109:
110:          <p>To force a word to end with &quot;es&quot;; use a second
111:            &quot;true&quot;; argument.</p>
112:          <code>
113:            $word = &quot;pass&quot;;<br /> echo plural($word, TRUE); // Re
turns
114:            &quot;passes&quot;;
115:          </code>
116:

```


inflector_helper.html

iddot;

175:

Copyright © 2014 - 2015 · <a

176:

href="http://bcit.ca/">British Columbia Institute of Te

chnology

177:

</p>

178:

</div>

179:

180: </body>

181: </html>

date_helper.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Date Helper : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12:     href='../userguide.css' />
13:
14: <script type="text/javascript" src="../../nav/nav.js"></script>
15: <script type="text/javascript" src="../../nav/prototype.light.js"></script>
16: <script type="text/javascript" src="../../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28:     <!-- START NAVIGATION -->
29:     <div id="nav">
30:         <div id="nav_inner">
31:             <script type="text/javascript">create_menu('../');</script>
32:         </div>
33:     </div>
34:     <div id="nav2">
35:         <a name="top"></a><a href="javascript:void(0);"
36:         onclick="myHeight.toggle();"></a>
40:     </div>
41:     <div id="masthead">
42:         <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43:             <tr>
44:                 <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45:                 <td id="breadcrumb_right"><a href="../../toc.html">Table of
46:                                     Contents Page</a></td>
47:             </tr>
48:         </table>
49:     </div>
50:     <!-- END NAVIGATION -->
51:
52:     <!-- START BREADCRUMB -->
53:     <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54:         <tr>
55:
56:             <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgniter
57:                                     Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="../../index.html">User Guide
58:                                     Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;Date Helper</td>
59:             <td id="searchbox"><form method="get"
f
```

date_helper.html

```
60:                                     action="http://www.google.com/search">
61:                                     <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                                     value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;   <input
63:                                     type="text" class="input" style="width:
200px;" name="q" id="q"
64:                                     size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                                     class="submit" name="sa" value="Go" />
66:                                     </form></td>
67:                                     </tr>
68:     </table>
69:     <!-- END BREADCRUMB -->
70:
71:     <br clear="all" />
72:
73:
74:     <!-- START CONTENT -->
75:     <div id="content">
76:
77:
78:         <h1>Date Helper</h1>
79:
80:         <p>The Date Helper file contains functions that help you work with
81:             dates.</p>
82:
83:
84:         <h2>Loading this Helper</h2>
85:
86:         <p>This helper is loaded using the following code:</p>
87:         <code>$this->load->helper('date');</code>
88:
89:
90:         <p>The following functions are available:</p>
91:
92:         <h2>now()</h2>
93:
94:         <p>Returns the current time as a Unix timestamp, referenced either
95:             to your server's local time or GMT, based on the "time referenc
e"
96:             setting in your config file. If you do not intend to set your m
aster
97:             time reference to GMT (which you'll typically do if you run a s
ite
98:             that lets each user set their own timezone settings) there is n
o
99:             benefit to using this function over PHP's time() function.</p>
100:
101:
102:
103:
104:         <h2>mdate()</h2>
105:
106:         <p>
107:             This function is identical to PHPs <a href="http://www.php.net/
date">date()</a>
108:             function, except that it lets you use MySQL style date codes, w
here
109:             each code letter is preceded with a percent sign: %Y %m %d etc.
110:
111:         </p>
112:         <p>The benefit of doing dates this way is that you don't have to
```

date_helper.html

```
113:          worry about escaping any characters that are not date codes, as
you
114:          would normally have to do with the date() function. Example:</p>
>
115:
116:          <code>
117:          $datestring = "Year: %Y Month: %m Day: %d - %h:%i %a";<br /> $t
ime =
118:          time();<br /> <br /> echo mdate($datestring, $time);
119:          </code>
120:
121:          <p>If a timestamp is not included in the second parameter the
122:          current time will be used.</p>
123:
124:
125:          <h2>standard_date()</h2>
126:
127:          <p>Lets you generate a date string in one of several standardized
128:          formats. Example:</p>
129:
130:          <code>
131:          $format = 'DATE_RFC822';<br /> $time = time();<br /> <br /> ech
o
132:          standard_date($format, $time);
133:          </code>
134:
135:          <p>The first parameter must contain the format, the second
136:          parameter must contain the date as a Unix timestamp.</p>
137:
138:          <p>Supported formats:</p>
139:
140:          <table cellpadding="0" cellspacing="1" border="0" style="width: 100%"
141:          class="tableborder">
142:          <tr>
143:              <th>Constant</th>
144:              <th>Description</th>
145:              <th>Example</th>
146:          </tr>
147:          <tr>
148:              <td>DATE_ATOM</td>
149:              <td>Atom</td>
150:              <td>2005-08-15T16:13:03+0000</td>
151:          </tr>
152:          <tr>
153:              <td>DATE_COOKIE</td>
154:              <td>HTTP Cookies</td>
155:              <td>Sun, 14 Aug 2005 16:13:03 UTC</td>
156:          </tr>
157:          <tr>
158:              <td>DATE_ISO8601</td>
159:              <td>ISO-8601</td>
160:              <td>2005-08-14T16:13:03+00:00</td>
161:          </tr>
162:          <tr>
163:              <td>DATE_RFC822</td>
164:              <td>RFC 822</td>
165:              <td>Sun, 14 Aug 05 16:13:03 UTC</td>
166:          </tr>
167:          <tr>
168:              <td>DATE_RFC850</td>
169:              <td>RFC 850</td>
170:              <td>Sunday, 14-Aug-05 16:13:03 UTC</td>
171:          </tr>
```

date_helper.html

```
172:         <tr>
173:             <td>DATE_RFC1036</td>
174:             <td>RFC 1036</td>
175:             <td>Sunday, 14-Aug-05 16:13:03 UTC</td>
176:         </tr>
177:         <tr>
178:             <td>DATE_RFC1123</td>
179:             <td>RFC 1123</td>
180:             <td>Sun, 14 Aug 2005 16:13:03 UTC</td>
181:         </tr>
182:         <tr>
183:             <td>DATE_RFC2822</td>
184:             <td>RFC 2822</td>
185:             <td>Sun, 14 Aug 2005 16:13:03 +0000</td>
186:         </tr>
187:         <tr>
188:             <td>DATE_RSS</td>
189:             <td>RSS</td>
190:             <td>Sun, 14 Aug 2005 16:13:03 UTC</td>
191:         </tr>
192:         <tr>
193:             <td>DATE_W3C</td>
194:             <td>World Wide Web Consortium</td>
195:             <td>2005-08-14T16:13:03+0000</td>
196:         </tr>
197:     </table>
198:
199:     <h2>local_to_gmt()</h2>
200:
201:     <p>Takes a Unix timestamp as input and returns it as GMT. Example:</p>
202:
203:     <code>
204:         $now = time();<br /> <br /> $gmt = local_to_gmt($now);
205:     </code>
206:
207:
208:     <h2>gmt_to_local()</h2>
209:
210:     <p>Takes a Unix timestamp (referenced to GMT) as input, and
211:     converts it to a localized timestamp based on the timezone and
212:     Daylight Saving time submitted. Example:</p>
213:
214:     <code>
215:         $timestamp = '1140153693';<br /> $timezone = 'UM8';<br />
216:         $daylight_saving = TRUE;<br /> <br /> echo gmt_to_local($timest
amp,
217:         $timezone, $daylight_saving);
218:     </code>
219:
220:     <p>
221:         <strong>Note:</strong> For a list of timezones see the referenc
e at
222:         the bottom of this page.
223:     </p>
224:
225:     <h2>mysql_to_unix()</h2>
226:
227:     <p>Takes a MySQL Timestamp as input and returns it as Unix.
228:     Example:</p>
229:
230:     <code>
231:         $mysql = '20061124092345';<br /> <br /> $unix =
232:         mysql_to_unix($mysql);
```

date_helper.html

```
233:         </code>
234:
235:
236:         <h2>unix_to_human()</h2>
237:
238:         <p>Takes a Unix timestamp as input and returns it in a human
239:             readable format with this prototype:</p>
240:
241:         <code>YYYY-MM-DD HH:MM:SS AM/PM</code>
242:
243:         <p>This can be useful if you need to display a date in a form
244:             field for submission.</p>
245:
246:         <p>The time can be formatted with or without seconds, and it can
247:             be set to European or US format. If only the timestamp is submi
tted
248:             it will return the time without seconds formatted for the U.S.
249:             Examples:</p>
250:
251:         <code>
252:             $now = time();<br /> <br /> echo unix_to_human($now); // U.S. t
ime,
253:             no seconds<br /> <br /> echo unix_to_human($now, TRUE, 'us'); /
/
254:             U.S. time with seconds<br /> <br /> echo unix_to_human($now, TR
UE,
255:             'eu'); // Euro time with seconds
256:         </code>
257:
258:
259:         <h2>human_to_unix()</h2>
260:
261:         <p>The opposite of the above function. Takes a "human" time as
262:             input and returns it as Unix. This function is useful if you ac
cept
263:             "human" formatted dates submitted via a form. Returns FALSE (bo
olean)
264:             if the date string passed to it is not formatted as indicated a
bove.
265:             Example:</p>
266:
267:         <code>
268:             $now = time();<br /> <br /> $human = unix_to_human($now);<br />
<br />
269:             $unix = human_to_unix($human);
270:         </code>
271:
272:
273:
274:
275:
276:         <h2>timespan()</h2>
277:
278:         <p>Formats a unix timestamp so that is appears similar to this:</p>
279:
280:         <code>1 Year, 10 Months, 2 Weeks, 5 Days, 10 Hours, 16 Minutes</code>
281:
282:         <p>The first parameter must contain a Unix timestamp. The second
283:             parameter must contain a timestamp that is greater than the fir
st
284:             timestamp. If the second parameter empty, the current time will
be
285:             used. The most common purpose for this function is to show how
```

date_helper.html

```
much
286:         time has elapsed from some point in time in the past to now. Ex
ample:</p>
287:
288:         <code>
289:         $post_date = '1079621429';<br /> $now = time();<br /> <br /> ec
ho
290:         timespan($post_date, $now);
291:         </code>
292:
293:         <p class="important">
294:         <strong>Note:</strong> The text generated by this function is f
ound
295:         in the following language file:
296:         language/&lt;your_lang&gt;/date_lang.php
297:         </p>
298:
299:
300:         <h2>days_in_month()</h2>
301:
302:         <p>Returns the number of days in a given month/year. Takes leap
303:         years into account. Example:</p>
304:         <code>echo days_in_month(06, 2005);</code>
305:
306:         <p>If the second parameter is empty, the current year will be
307:         used.</p>
308:         <h2>timezones()</h2>
309:         <p>Takes a timezone reference (for a list of valid timezones, see
310:         the &quot;Timezone Reference&quot; below) and returns the numbe
r of
311:         hours offset from UTC.</p>
312:         <p>
313:         <code>echo timezones('UM5');</code>
314:         </p>
315:         <p>This function is useful when used with timezone_menu().</p>
316:         <h2>timezone_menu()</h2>
317:         <p>Generates a pull-down menu of timezones, like this one:</p>
318:
319:         <form action="#">
320:             <select name="timezones">
321:                 <option value='UM12'>(UTC - 12:00) Enitwetok, Kwajalien
</option>
322:                 <option value='UM11'>(UTC - 11:00) Nome, Midway Island,
323:                 Samoa</option>
324:                 <option value='UM10'>(UTC - 10:00) Hawaii</option>
325:                 <option value='UM9'>(UTC - 9:00) Alaska</option>
326:                 <option value='UM8'>(UTC - 8:00) Pacific Time</option>
327:                 <option value='UM7'>(UTC - 7:00) Mountain Time</option>
328:                 <option value='UM6'>(UTC - 6:00) Central Time, Mexico C
ity</option>
329:                 <option value='UM5'>(UTC - 5:00) Eastern Time, Bogota,
330:                 Lima, Quito</option>
331:                 <option value='UM4'>(UTC - 4:00) Atlantic Time, Caracas
, La
332:                 Paz</option>
333:                 <option value='UM25'>(UTC - 3:30) Newfoundland</option>
334:                 <option value='UM3'>(UTC - 3:00) Brazil, Buenos Aires,
335:                 Georgetown, Falkland Is.</option>
336:                 <option value='UM2'>(UTC - 2:00) Mid-Atlantic, Ascensio
n
337:                 Is., St Helena</option>
338:                 <option value='UM1'>(UTC - 1:00) Azores, Cape Verde Isl
ands</option>
```

date_helper.html

```
339:         <option value='UTC' selected='selected'>(UTC) Casablanc
a,
340:             Dublin, Edinburgh, London, Lisbon, Monrovia</op
tion>
341:         <option value='UP1'>(UTC + 1:00) Berlin, Brussels,
342:             Copenhagen, Madrid, Paris, Rome</option>
343:         <option value='UP2'>(UTC + 2:00) Kaliningrad, South Afr
ica,
344:             Warsaw</option>
345:         <option value='UP3'>(UTC + 3:00) Baghdad, Riyadh, Mosco
w,
346:             Nairobi</option>
347:         <option value='UP25'>(UTC + 3:30) Tehran</option>
348:         <option value='UP4'>(UTC + 4:00) Adu Dhabi, Baku, Musca
t,
349:             Tbilisi</option>
350:         <option value='UP35'>(UTC + 4:30) Kabul</option>
351:         <option value='UP5'>(UTC + 5:00) Islamabad, Karachi,
352:             Tashkent</option>
353:         <option value='UP45'>(UTC + 5:30) Bombay, Calcutta, Mad
ras,
354:             New Delhi</option>
355:         <option value='UP6'>(UTC + 6:00) Almaty, Colomba, Dhaka
</option>
356:         <option value='UP7'>(UTC + 7:00) Bangkok, Hanoi, Jakart
a</option>
357:         <option value='UP8'>(UTC + 8:00) Beijing, Hong Kong, Pe
rth,
358:             Singapore, Taipei</option>
359:         <option value='UP9'>(UTC + 9:00) Osaka, Sapporo, Seoul,
360:             Tokyo, Yakutsk</option>
361:         <option value='UP85'>(UTC + 9:30) Adelaide, Darwin</opt
ion>
362:         <option value='UP10'>(UTC + 10:00) Melbourne, Papua New
363:             Guinea, Sydney, Vladivostok</option>
364:         <option value='UP11'>(UTC + 11:00) Srednekolymsk, New
365:             Caledonia, Solomon Islands</option>
366:         <option value='UP12'>(UTC + 12:00) Auckland, Wellington
,
367:             Fiji, Marshall Island</option>
368:     </select>
369: </form>
370:
371: <p>This menu is useful if you run a membership site in which your
372:     users are allowed to set their local timezone value.</p>
373:
374: <p>The first parameter lets you set the "selected" state of the
375:     menu. For example, to set Pacific time as the default you will
do
376:     this:</p>
377:
378: <code>echo timezone_menu('UM8');</code>
379:
380: <p>Please see the timezone reference below to see the values of
381:     this menu.</p>
382:
383: <p>The second parameter lets you set a CSS class name for the
384:     menu.</p>
385:
386: <p class="important">
387:     <strong>Note:</strong> The text contained in the menu is found
in the
388:     following language file: language/&lt;your_lang&gt;/date_lang.p
```

date_helper.html

```
hp
389:         </p>
390:
391:
392:
393:         <h2>Timezone Reference</h2>
394:
395:         <p>The following table indicates each timezone and its location.</p>
396:
397:         <table cellpadding="0" cellspacing="1" border="0" style="width: 100%"
398:             class="tableborder">
399:             <tr>
400:                 <th>Time Zone</th>
401:                 <th>Location</th>
402:             </tr>
403:             <tr>
404:
405:                 <td class="td">UM12</td>
406:                 <td class="td">(UTC - 12:00) Enitwetok, Kwajalien</td>
407:             </tr>
408:             <tr>
409:                 <td class="td">UM11</td>
410:                 <td class="td">(UTC - 11:00) Nome, Midway Island, Samoa
411: </td>
412:             </tr>
413:             <tr>
414:                 <td class="td">UM10</td>
415:                 <td class="td">(UTC - 10:00) Hawaii</td>
416:             </tr>
417:             <tr>
418:                 <td class="td">UM9</td>
419:                 <td class="td">(UTC - 9:00) Alaska</td>
420:             </tr>
421:             <tr>
422:                 <td class="td">UM8</td>
423:                 <td class="td">(UTC - 8:00) Pacific Time</td>
424:             </tr>
425:             <tr>
426:                 <td class="td">UM7</td>
427:                 <td class="td">(UTC - 7:00) Mountain Time</td>
428:             </tr>
429:             <tr>
430:                 <td class="td">UM6</td>
431:                 <td class="td">(UTC - 6:00) Central Time, Mexico City</
432: td>
433:             </tr>
434:             <tr>
435:                 <td class="td">UM5</td>
436:                 <td class="td">(UTC - 5:00) Eastern Time, Bogota, Lima,
437: Quito</td>
438:             </tr>
439:             <tr>
440:                 <td class="td">UM4</td>
441:                 <td class="td">(UTC - 4:00) Atlantic Time, Caracas, La
442: Paz</td>
443:             </tr>
444:             <tr>
445:                 <td class="td">UM25</td>
446:                 <td class="td">(UTC - 3:30) Newfoundland</td>
447:             </tr>
448:             <tr>
449:                 <td class="td">UM3</td>
450:                 <td class="td">(UTC - 3:00) Brazil, Buenos Aires, Georg
```


date_helper.html

```
etown,
 447:                Falkland Is.</td>
448:                </tr>
449:                <tr>
450:                <td class="td">UM2</td>
451:                <td class="td">(UTC - 2:00) Mid-Atlantic, Ascension Is.
, St
452:                Helena</td>
453:                </tr>
454:                <tr>
455:                <td class="td">UM1</td>
456:                <td class="td">(UTC - 1:00) Azores, Cape Verde Islands<
/td>
457:                </tr>
458:                <tr>
459:                <td class="td">UTC</td>
460:                <td class="td">(UTC) Casablanca, Dublin, Edinburgh, Lon
don,
461:                Lisbon, Monrovia</td>
462:                </tr>
463:                <tr>
464:                <td class="td">UP1</td>
465:                <td class="td">(UTC + 1:00) Berlin, Brussels, Copenhage
n,
466:                Madrid, Paris, Rome</td>
467:                </tr>
468:                <tr>
469:                <td class="td">UP2</td>
470:                <td class="td">(UTC + 2:00) Kaliningrad, South Africa,
Warsaw</td>
471:                </tr>
472:                <tr>
473:                <td class="td">UP3</td>
474:                <td class="td">(UTC + 3:00) Baghdad, Riyadh, Moscow, Na
irobi</td>
475:                </tr>
476:                <tr>
477:                <td class="td">UP25</td>
478:                <td class="td">(UTC + 3:30) Tehran</td>
479:                </tr>
480:                <tr>
481:                <td class="td">UP4</td>
482:                <td class="td">(UTC + 4:00) Adu Dhabi, Baku, Muscat, Tb
ilisi</td>
483:                </tr>
484:                <tr>
485:                <td class="td">UP35</td>
486:                <td class="td">(UTC + 4:30) Kabul</td>
487:                </tr>
488:                <tr>
489:                <td class="td">UP5</td>
490:                <td class="td">(UTC + 5:00) Islamabad, Karachi, Tashken
t</td>
491:                </tr>
492:                <tr>
493:                <td class="td">UP45</td>
494:                <td class="td">(UTC + 5:30) Bombay, Calcutta, Madras, N
ew Delhi</td>
495:                </tr>
496:                <tr>
497:                <td class="td">UP6</td>
498:                <td class="td">(UTC + 6:00) Almaty, Colomba, Dhaka</td>
499:                </tr>
```


date_helper.html

```
553: Copyright © 2014 - 2015 · British Columbia Institute of Te
554: chnology
```

555: [British Columbia Institute of Technology](http://bcit.ca/)

```
556: </div>
557: </p>
558: </body>
559: </html>
```


url_helper.html

```
118:         site_url($segments);
119:     </code>
120:
121:
122:     <h2>base_url()</h2>
123:     <p>Returns your site base URL, as specified in your config file.
124:         Example:</p>
125:     <code>echo base_url();</code>
126:
127:     <p>
128:         This function returns the same thing as site_url, without the
129:         <dfn>index_page</dfn>
130:         or
131:         <dfn>url_suffix</dfn>
132:         being appended.
133:     </p>
134:
135:     <p>Also like site_url, you can supply segments as a string or an
136:         array. Here is a string example:</p>
137:
138:     <code>echo base_url("blog/post/123");</code>
139:
140:     <p>The above example would return something like:
141:         http://example.com/blog/post/123</p>
142:
143:     <p>This is useful because unlike site_url(), you can supply a
144:         string to a file, such as an image or stylesheet. For example:<
/p>
145:
146:     <code>echo base_url("images/icons/edit.png");</code>
147:
148:     <p>This would give you something like:
149:         http://example.com/images/icons/edit.png</p>
150:
151:
152:     <h2>current_url()</h2>
153:     <p>Returns the full URL (including segments) of the page being
154:         currently viewed.</p>
155:
156:
157:     <h2>uri_string()</h2>
158:     <p>Returns the URI segments of any page that contains this
159:         function. For example, if your URL was this:</p>
160:     <code>http://some-site.com/blog/comments/123</code>
161:
162:     <p>The function would return:</p>
163:     <code>/blog/comments/123</code>
164:
165:
166:     <h2>index_page()</h2>
167:     <p>Returns your site "index" page, as specified in your config
168:         file. Example:</p>
169:     <code>echo index_page();</code>
170:
171:
172:
173:     <h2>anchor()</h2>
174:
175:     <p>Creates a standard HTML anchor link based on your local site
176:         URL:</p>
177:
178:     <code>&lt;a href="http://example.com">Click Here&lt;/a></code>
179:
```

url_helper.html

```
180: <p>The tag has three optional parameters:</p>
181:
182: <code>
183:     anchor(
184:         <var>uri segments</var>
185:         ,
186:         <var>text</var>
187:         ,
188:         <var>attributes</var>
189:     )
190: </code>
191:
192: <p>
193:     The first parameter can contain any segments you wish appended
to the
194:     URL. As with the
195:     <dfn>site_url()</dfn>
196:     function above, segments can be a string or an array.
197: </p>
198:
199: <p>
200:     <strong>Note:</strong>&nbsp; If you are building links that are
201:     internal to your application do not include the base URL
202:     (http://...). This will be added automatically from the informa
tion
203:     specified in your config file. Include only the URI segments yo
u wish
204:     appended to the URL.
205: </p>
206:
207: <p>The second segment is the text you would like the link to say.
208:     If you leave it blank, the URL will be used.</p>
209:
210: <p>The third parameter can contain a list of attributes you would
211:     like added to the link. The attributes can be a simple string o
r an
212:     associative array.</p>
213:
214: <p>Here are some examples:</p>
215:
216: <code>echo anchor('news/local/123', 'My News', 'title="News
217:     title"');</code>
218:
219: <p>Would produce: &lt;a
220:     href="http://example.com/index.php/news/local/123" title="News
221:     title">My News&lt;/a></p>
222:
223: <code>echo anchor('news/local/123', 'My News', array('title' =>
224:     'The best news!'));</code>
225:
226: <p>Would produce: &lt;a
227:     href="http://example.com/index.php/news/local/123" title="The b
est
228:     news!">My News&lt;/a></p>
229:
230:
231: <h2>anchor_popup()</h2>
232:
233: <p>
234:     Nearly identical to the
235:     <dfn>anchor()</dfn>
236:     function except that it opens the URL in a new window. You can
237:     specify JavaScript window attributes in the third parameter to
```


url_helper.html

```
289:         <p>Identical to the above function except it writes an obfuscated
290:         version of the mailto tag using ordinal numbers written with
291:         JavaScript to help prevent the email address from being harvest
ed by
292:         spam bots.</p>
293:
294:
295:         <h2>auto_link()</h2>
296:
297:         <p>Automatically turns URLs and email addresses contained in a
298:         string into links. Example:</p>
299:
300:         <code>$string = auto_link($string);</code>
301:
302:         <p>The second parameter determines whether URLs and emails are
303:         converted or just one or the other. Default behavior is both if
the
304:         parameter is not specified. Email links are encoded as safe_mai
lto()
305:         as shown above.</p>
306:
307:         <p>Converts only URLs:</p>
308:         <code>$string = auto_link($string, 'url');</code>
309:
310:         <p>Converts only Email addresses:</p>
311:         <code>$string = auto_link($string, 'email');</code>
312:
313:         <p>The third parameter determines whether links are shown in a new
314:         window. The value can be TRUE or FALSE (boolean):</p>
315:         <code>$string = auto_link($string, 'both', TRUE);</code>
316:
317:
318:         <h2>url_title()</h2>
319:         <p>Takes a string as input and creates a human-friendly URL
320:         string. This is useful if, for example, you have a blog in whic
h
321:         you'd like to use the title of your entries in the URL. Example
:</p>
322:
323:         <code>
324:         $title = "What's wrong with CSS?";<br /> <br /> $url_title =
325:         url_title($title);<br /> <br /> // Produces: Whats-wrong-with-C
SS
326:         </code>
327:
328:
329:         <p>The second parameter determines the word delimiter. By default
330:         dashes are used.</p>
331:
332:         <code>
333:         $title = "What's wrong with CSS?";<br /> <br /> $url_title =
334:         url_title($title, '_');<br /> <br /> // Produces:
335:         Whats_wrong_with_CSS
336:         </code>
337:
338:         <p>
339:         The third parameter determines whether or not lowercase charact
ers
340:         are forced. By default they are not. Options are boolean
341:         <dfn>TRUE</dfn>
342:         /
343:         <dfn>FALSE</dfn>
344:         :
```


form_helper.html

```
60:         action="http://www.google.com/search">
61:         <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:           value="codeigniter.com/user_guide/" />S
earch User Guide<input
63:         type="text" class="input" style="width:
200px;" name="q" id="q"
64:         size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:         class="submit" name="sa" value="Go" />
66:     </form></td>
67: </tr>
68: </table>
69: <!-- END BREADCRUMB -->
70:
71: <br clear="all" />
72:
73:
74: <!-- START CONTENT -->
75: <div id="content">
76:
77:
78:     <h1>Form Helper</h1>
79:
80:     <p>The Form Helper file contains functions that assist in working
81:       with forms.</p>
82:
83:
84:     <h2>Loading this Helper</h2>
85:
86:     <p>This helper is loaded using the following code:</p>
87:     <code>$this->load->helper('form');</code>
88:
89:     <p>The following functions are available:</p>
90:
91:
92:
93:     <h2>form_open()</h2>
94:
95:     <p>
96:       Creates an opening form tag with a base URL <strong>built
97:       from your config preferences</strong>. It will optional
ly let you add form
98:       attributes and hidden input fields, and will always add the att
ribute
99:       <code>accept-charset</code>
100:       based on the charset value in your config file.
101:     </p>
102:
103:     <p>The main benefit of using this tag rather than hard coding your
104:       own HTML is that it permits your site to be more portable in th
e
105:       event your URLs ever change.</p>
106:
107:     <p>Here's a simple example:</p>
108:
109:     <code>echo form_open('email/send');</code>
110:
111:     <p>The above example would create a form that points to your base
112:       URL plus the "email/send" URI segments, like this:</p>
113:
114:     <code>&lt;form method="post" accept-charset="utf-8"
115:       action="http://example.com/index.php/email/send" /></code>
```

form_helper.html

```
116:
117:         <h4>Adding Attributes</h4>
118:
119:         <p>Attributes can be added by passing an associative array to the
120:             second parameter, like this:</p>
121:
122:         <code>
123:             $attributes = array('class' => 'email', 'id' => 'myform');<br /
> <br />
124:             echo form_open('email/send', $attributes);
125:         </code>
126:
127:         <p>The above example would create a form similar to this:</p>
128:
129:         <code>&lt;form method="post" accept-charset="utf-8"
130:             action="http://example.com/index.php/email/send" &nbsp;class="em
ail"
131:             &nbsp;id="myform" /></code>
132:
133:         <h4>Adding Hidden Input Fields</h4>
134:
135:         <p>Hidden fields can be added by passing an associative array to
136:             the third parameter, like this:</p>
137:
138:         <code>
139:             $hidden = array('username' => 'Joe', 'member_id' => '234');<br
/> <br />
140:             echo form_open('email/send', '', $hidden);
141:         </code>
142:
143:         <p>The above example would create a form similar to this:</p>
144:
145:         <code>
146:             &lt;form method="post" accept-charset="utf-8"
147:             action="http://example.com/index.php/email/send"><br /> &lt;input
t
148:             type="hidden" name="username" value="Joe" /><br /> &lt;input
149:             type="hidden" name="member_id" value="234" />
150:         </code>
151:
152:
153:         <h2>form_open_multipart()</h2>
154:
155:         <p>
156:             This function is absolutely identical to the
157:             <dfn>form_open()</dfn>
158:             tag above except that it adds a multipart attribute, which is
159:             necessary if you would like to use the form to upload files wit
h.
160:
161:         </p>
162:
163:         <h2>form_hidden()</h2>
164:
165:         <p>Lets you generate hidden input fields. You can either submit a
166:             name/value string to create one field:</p>
167:
168:         <code>
169:             form_hidden('username', 'johndoe');<br /> <br /> // Would produ
ce:<br />
170:             <br /> &lt;input type="hidden" name="username" value="johndoe"
/>
171:
```


form_helper.html

```
322: Lets you create a standard multiselect field. The first paramet
er
323: will contain the name of the field, the second parameter will c
ontain
324: an associative array of options, and the third parameter will c
ontain
325: the value or values you wish to be selected. The parameter usag
e is
326: identical to using
327: <kbd>form_dropdown()</kbd>
328: above, except of course that the name of the field will need to
use
329: POST array syntax, e.g.
330: <samp>foo[]</samp>
331: .
332: </p>
333:
334:
335: <h2>form_fieldset()</h2>
336:
337: <p>Lets you generate fieldset/legend fields.</p>
338: <code>
339: echo form_fieldset('Address Information');<br /> echo
340: &quot;&lt;p&gt;fieldset content here&lt;/p&gt;\n&quot;;<br /> e
cho
341: form_fieldset_close(); <br /> <br /> // Produces<br />
342: &lt;fieldset&gt; <br /> &lt;legend&gt;Address
343: Information&lt;/legend&gt; <br /> &lt;p&gt;form content
344: here&lt;/p&gt; <br /> &lt;/fieldset&gt;
345: </code>
346: <p>Similar to other functions, you can submit an associative array
347: in the second parameter if you prefer to set additional attribu
tes.</p>
348: <p>
349: <code>
350: $attributes = array('id' =&gt; 'address_info', 'class'
=&gt;
351: 'address_info');<br /> echo form_fieldset('Address Info
rmation',
352: $attributes);<br /> echo &quot;&lt;p&gt;fieldset conten
t
353: here&lt;/p&gt;\n&quot;;<br /> echo form_fieldset_close(
); <br /> <br />
354: // Produces<br /> &lt;fieldset id=&quot;address_info&qu
ot;
355: class=&quot;address_info&quot;&gt; <br /> &lt;legend&gt;
;Address
356: Information&lt;/legend&gt; <br /> &lt;p&gt;form content
357: here&lt;/p&gt; <br /> &lt;/fieldset&gt;
358: </code>
359: </p>
360: <h2>form_fieldset_close()</h2>
361: <p>Produces a closing &lt;/fieldset&gt; tag. The only advantage to
362: using this function is it permits you to pass data to it which
will
363: be added below the tag. For example:</p>
364: <code>
365: $string = &quot;&lt;/div&gt;&lt;/div&gt;&quot;;<br /> <br /> ec
ho
366: form_fieldset_close($string);<br /> <br /> // Would produce:<br
/>
367: &lt;/fieldset&gt;<br /> &lt;/div&gt;&lt;/div&gt;
368: </code>
```


form_helper.html

```
</>
473:         echo form_button($data);<br /> <br /> // Would produce:<br />
474:         &lt;button name="button" id="button" value="true"
475:         type="reset"&gt;Reset&lt;/button&gt;;
476:     </code>
477:
478:     If you would like your form to contain some additional data, like
479:     JavaScript, you can pass it as a string in the third parameter:
480:     <code>
481:         $js = 'onClick="some_function()";<br />
482:         <br /> echo form_button('mybutton', 'Click Me', $js);
483:     </code>
484:
485:
486:     <h2>form_close()</h2>
487:
488:     <p>Produces a closing &lt;/form> tag. The only advantage to using
489:     this function is it permits you to pass data to it which will be
e
490:     added below the tag. For example:</p>
491:
492:     <code>
493:         $string = "&lt;/div>&lt;/div>";<br /> <br /> echo
494:         form_close($string);<br /> <br /> // Would produce:<br /> <br />
>
495:         &lt;/form><br /> &lt;/div>&lt;/div>
496:     </code>
497:
498:
499:
500:
501:
502:     <h2>form_prep()</h2>
503:
504:     <p>Allows you to safely use HTML and characters such as quotes
505:     within form elements without breaking out of the form. Consider
this
506:     example:</p>
507:
508:     <code>
509:         $string = 'Here is a string containing <strong>"quoted"</strong>
>
510:         text.';<br /> <br /> &lt;input type="text" name="myform" value=
=
511:         <var>$string</var>
512:         " />
513:     </code>
514:
515:     <p>Since the above string contains a set of quotes it will cause
516:     the form to break. The form_prep function converts HTML so that
it
517:     can be used safely:</p>
518:
519:     <code>
520:         &lt;input type="text" name="myform" value="
521:         <var>&lt;?php echo form_prep($string); ?></var>
522:         " />
523:     </code>
524:
525:     <p class="important">
526:         <strong>Note:</strong> If you use any of the form helper functi
ons
527:         listed in this page the form values will be prepped automatical
```


language_helper.html

```

1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xh
tml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Language Helper : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype_lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28: <!-- START NAVIGATION -->
29: <div id="nav">
30: <div id="nav_inner">
31: <script type="text/javascript">create_menu('../');</script>
32: </div>
33: </div>
34: <div id="nav2">
35: <a name="top"></a><a href="javascript:void(0);"
36: onclick="myHeight.toggle();"></a>
40: </div>
41: <div id="masthead">
42: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43: <tr>
44: <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45: <td id="breadcrumb_right"><a href="../toc.html">Table of
46: Contents Page</a></td>
47: </tr>
48: </table>
49: </div>
50: <!-- END NAVIGATION -->
51:
52:
53: <!-- START BREADCRUMB -->
54:
55: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
56: <tr>
57: <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
58: er <a href="../index.html">User Guide
59: <a href="http://codeigniter.com/">CodeIgnit

```


language_helper.html

```
d>
60:                 <td id="searchbox"><form method="get"
61:                   action="http://www.google.com/search">
62:                   <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
63:                   value="codeigniter.com/user_guide/" />S
earch User Guide<input
64:                   type="text" class="input" style="width:
200px;" name="q" id="q"
65:                   size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
66:                   class="submit" name="sa" value="Go" />
67:                 </form></td>
68:             </tr>
69:         </table>
70:         <!-- END BREADCRUMB -->
71:         <br clear="all" />
72:         <!-- START CONTENT -->
73:         <div id="content">
74:
75:             <h1>Language Helper</h1>
76:
77:             <p>The Language Helper file contains functions that assist in
78:                 working with language files.</p>
79:
80:             <h2>Loading this Helper</h2>
81:
82:             <p>This helper is loaded using the following code:</p>
83:             <code>$this->load->helper('language');</code>
84:
85:             <p>The following functions are available:</p>
86:
87:             <h2>
88:                 lang('
89:                 <var>language line</var>
90:                 ', '
91:                 <var>element id</var>
92:                 ')
93:             </h2>
94:
95:             <p>
96:                 This function returns a line of text from a loaded language fil
97:                 e with
98:                 simplified syntax that may be more desirable for view files tha
99:                 n
100:                 calling
101:                 <code>$this->lang->line()</code>
102:                 . The optional second parameter will also output a form label f
103:                 or
104:                 you. Example:
105:
106:             <p>
107:
108:             <code>
109:                 echo lang('
110:                 <samp>language_key</samp>
111:                 ', '
112:                 <samp>form_item_id</samp>
113:                 ');<br /> // becomes &lt;label
114:
```


html_helper.html

```
60:         action="http://www.google.com/search">
61:         <input type="hidden" name="as_sitesearch" id="as_sitesearch"
62:         value="codeigniter.com/user_guide/" />
search User Guide<input type="text" class="input" style="width:
63: 200px;" name="q" id="q"
64: > <input type="submit"
65: class="submit" name="sa" value="Go" />
66:         </form></td>
67:     </tr>
68: </table>
69: <!-- END BREADCRUMB -->
70:
71: <br clear="all" />
72:
73: <!-- START CONTENT -->
74: <div id="content">
75:
76:
77:
78:     <h1>HTML Helper</h1>
79:
80:     <p>The HTML Helper file contains functions that assist in working
81:     with HTML.</p>
82:
83:     <ul>
84:         <li><a href="#br">br()</a></li>
85:         <li><a href="#heading">heading()</a></li>
86:         <li><a href="#img">img()</a></li>
87:         <li><a href="#link_tag">link_tag()</a></li>
88:         <li><a href="#nbs">nbs()</a></li>
89:         <li><a href="#ol_and_ul">ol() and ul()</a></li>
90:         <li><a href="#meta">meta()</a></li>
91:         <li><a href="#doctype">doctype()</a></li>
92:     </ul>
93:
94:     <h2>Loading this Helper</h2>
95:
96:     <p>This helper is loaded using the following code:</p>
97:     <code>$this->load->helper('html');</code>
98:
99:     <p>The following functions are available:</p>
100:
101:     <h2>
102:         <a name="br"></a>br()
103:     </h2>
104:     <p>Generates line break tags (&lt;br />) based on the number you
105:     submit. Example:</p>
106:     <code>echo br(3);</code>
107:     <p>The above would produce: &lt;br />&lt;br />&lt;br /></p>
108:
109:     <h2>
110:         <a name="heading"></a>heading()
111:     </h2>
112:     <p>Lets you create HTML &lt;h1> tags. The first parameter will
113:     contain the data, the second the size of the heading. Example:<
/p>
114:     <code>echo heading('Welcome!', 3);</code>
115:     <p>The above would produce: &lt;h3>Welcome!&lt;/h3></p>
116:
117:     <p>Additionally, in order to add attributes to the heading tag
```

html_helper.html

```
118:         such as HTML classes, ids or inline styles, a third parameter i
s
119:         available.</p>
120: <code>echo heading('Welcome!', 3, 'class="pink"')</code>
121: <p>The above code produces: <h3 class="pink">Welcome!</h3></p>
122:
123:
124: <h2>
125:     <a name="img"></a>img()
126: </h2>
127: <p>Lets you create HTML <img /> tags. The first parameter
128:     contains the image source. Example:</p>
129: <code>
130:     echo img('images/picture.jpg');<br /> // gives 
132: </code>
133: <p>There is an optional second parameter that is a TRUE/FALSE
134:     value that specifies if the src should have the page specified
by
135:     $config['index_page'] added to the address it creates. Presumab
ly,
136:     this would be if you were using a media controller.</p>
137: <p>
138:     <code>
139:         echo img('images/picture.jpg', TRUE);<br /> // gives 
142:     </code>
143: </p>
144: <p>Additionally, an associative array can be passed to the img()
145:     function for complete control over all attributes and values. If an
f an
146:     alt attribute is not provided, CodeIgniter will generate an empty
ty
147:     string.</p>
148: <p>
149:     <code>
150:         $image_properties = array(<br />
151:             <code> <br />
152:             <code> 'images/picture.jpg', <br />
153:             <code> <br />
154:             <code> 'Me, demonstrating how to eat 4 slices of pizza a
155:             <code> <br />
156:             <code> 'post_images', <br />
157:             <code> <br />
158:             <code> '200', <br />
159:             <code> <br />
160:             <code> '200', <br />
161:             <code> <br />
162:             <code> 'That was quite a night', <br />
163:             <code> <br />
164:             <code> 'lightbox', <br /> );<br /> <br /> img($image_prop
erties);<br />
```


html_helper.html

```
</p>
401:
402:         <code>
403:         echo doctype();<br /> // &lt;!DOCTYPE html PUBLIC "-//W3C//DTD
XHTML
404:         1.0 Strict//EN"
405:         "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"&gt;<br /> <
br />
406:         echo doctype('html4-trans');<br /> // &lt;!DOCTYPE HTML PUBLIC
407:         "-//W3C//DTD HTML 4.01//EN"
408:         "http://www.w3.org/TR/html4/strict.dtd"&gt;;
409:         </code>
410:
411:         <p>
412:         The following is a list of doctype choices. These are configura
ble,
413:         and pulled from
414:         <samp>application/config/doctypes.php</samp>
415:         </p>
416:
417:         <table cellpadding="0" cellspacing="1" border="0" style="width: 100%"
418:         class="tableborder">
419:         <tr>
420:             <th>Doctype</th>
421:             <th>Option</th>
422:             <th>Result</th>
423:         </tr>
424:         <tr>
425:             <td class="td">XHTML 1.1</td>
426:             <td class="td">doctype('xhtml11')</td>
427:             <td class="td">&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XH
TML
428:                 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtm
l11.dtd"&gt;</td>
429:         </tr>
430:         <tr>
431:             <td class="td">XHTML 1.0 Strict</td>
432:             <td class="td">doctype('xhtml1-strict')</td>
433:             <td class="td">&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XH
TML 1.0
434:                 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xh
tml1-strict.dtd"&gt;</td>
435:         </tr>
436:         <tr>
437:             <td class="td">XHTML 1.0 Transitional</td>
438:             <td class="td">doctype('xhtml1-trans')</td>
439:             <td class="td">&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XH
TML 1.0
440:                 Transitional//EN"
441:                 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transit
ional.dtd"&gt;</td>
442:         </tr>
443:         <tr>
444:             <td class="td">XHTML 1.0 Frameset</td>
445:             <td class="td">doctype('xhtml1-frame')</td>
446:             <td class="td">&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XH
TML 1.0
447:                 Frameset//EN"
448:                 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset
.t.dtd"&gt;</td>
449:         </tr>
450:         <tr>
451:             <td class="td">HTML 5</td>
```


typography_helper.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Typography Helper : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../../nav/nav.js"></script>
15: <script type="text/javascript" src="../../nav/prototype_lite.js"></script>
16: <script type="text/javascript" src="../../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28: <!-- START NAVIGATION -->
29: <div id="nav">
30: <div id="nav_inner">
31: <script type="text/javascript">create_menu('../');</script>
32: </div>
33: </div>
34: <div id="nav2">
35: <a name="top"></a><a href="javascript:void(0);"
36: onclick="myHeight.toggle();"></a>
40: </div>
41: <div id="masthead">
42: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43: <tr>
44: <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45: <td id="breadcrumb_right"><a href="../../toc.html">Table of
46: Contents Page</a></td>
47: </tr>
48: </table>
49: </div>
50: <!-- END NAVIGATION -->
51:
52: <!-- START BREADCRUMB -->
53: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54: <tr>
55: <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
56: er
57: <a href="../../index.html">User Guide
58: <a href="../../typography_helper.html">Typography Helper</td>
```

typography_helper.html

```
59:                 <td id="searchbox"><form method="get"
60:                   action="http://www.google.com/search">
61:                     <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                       value="codeigniter.com/user_guide/" />S
earch User Guide<input
63:                       type="text" class="input" style="width:
200px;" name="q" id="q"
64:                       size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                       class="submit" name="sa" value="Go" />
66:                 </form></td>
67:             </tr>
68:         </table>
69:         <!-- END BREADCRUMB -->
70:
71:         <br clear="all" />
72:
73:
74:         <!-- START CONTENT -->
75:         <div id="content">
76:
77:
78:             <h1>Typography Helper</h1>
79:
80:             <p>The Typography Helper file contains functions that help your
81:                 format text in semantically relevant ways.</p>
82:
83:
84:             <h2>Loading this Helper</h2>
85:
86:             <p>This helper is loaded using the following code:</p>
87:             <code>$this->load->helper('typography');</code>
88:
89:             <p>The following functions are available:</p>
90:
91:
92:             <h2>auto_typography()</h2>
93:
94:             <p>
95:                 Formats text so that it is semantically and typographically cor
rect
96:                 HTML. Please see the <a href="../libraries/typography.html">Typ
ography
97:                 Class</a> for more info.
98:             </p>
99:
100:            <p>Usage example:</p>
101:
102:            <code>$string = auto_typography($string);</code>
103:
104:            <p>
105:                <strong>Note:</strong> Typographic formatting can be processor
106:                intensive, particularly if you have a lot of content being form
atted.
107:                If you choose to use this function you may want to consider <a
108:                href="../general/caching.html">caching</a> your pages.
109:            </p>
110:
111:
112:            <h2>nl2br_except_pre()</h2>
113:
114:            <p>
```


path_helper.html

```

60:                                     <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
61:                                     value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;   <input
62:                                     type="text" class="input" style="width:
200px;" name="q" id="q"
63:                                     size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
64:                                     class="submit" name="sa" value="Go" />
65:                                     </form></td>
66:                                     </tr>
67:     </table>
68:     <!-- END BREADCRUMB -->
69:
70:     <br clear="all" />
71:
72:
73:     <!-- START CONTENT -->
74:     <div id="content">
75:
76:
77:         <h1>Path Helper</h1>
78:
79:         <p>The Path Helper file contains functions that permits you to
80:             work with file paths on the server.</p>
81:
82:
83:         <h2>Loading this Helper</h2>
84:
85:         <p>This helper is loaded using the following code:</p>
86:         <code>$this->load->helper('path');</code>
87:
88:         <p>The following functions are available:</p>
89:
90:
91:         <h2>set_realpath()</h2>
92:
93:         <p>Checks to see if the path exists. This function will return a
94:             server path without symbolic links or relative directory struct
ures.
95:             An optional second argument will cause an error to be triggered
if
96:             the path cannot be resolved.</p>
97:
98:         <code>
99:             $directory = '/etc/passwd';<br /> echo set_realpath($directory)
;<br />
100:             // returns &quot;/etc/passwd&quot;;<br /> <br />
101:             $non_existent_directory = '/path/to/nowhere';<br /> echo
102:             set_realpath($non_existent_directory, TRUE);<br /> // returns a
n <strong>error</strong>,
103:             as the path could not be resolved <br />
104:             <br /> echo set_realpath($non_existent_directory, FALSE);<br />
//
105:             returns &quot;/path/to/nowhere&quot;;
106:
107:
108:
109:         </code>
110:         <h2>&nbsp;  </h2>
111:     </div>
112:     <!-- END CONTENT -->
113:

```


cookie_helper.html

```
60:                                     action="http://www.google.com/search">
61:                                     <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                                     value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;   <input
63:                                     type="text" class="input" style="width:
200px;" name="q" id="q"
64:                                     size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                                     class="submit" name="sa" value="Go" />
66:                                     </form></td>
67:                                     </tr>
68:     </table>
69:     <!-- END BREADCRUMB -->
70:
71:     <br clear="all" />
72:
73:
74:     <!-- START CONTENT -->
75:     <div id="content">
76:
77:
78:         <h1>Cookie Helper</h1>
79:
80:         <p>The Cookie Helper file contains functions that assist in
81:             working with cookies.</p>
82:
83:
84:         <h2>Loading this Helper</h2>
85:
86:         <p>This helper is loaded using the following code:</p>
87:         <code>$this->load->helper('cookie');</code>
88:
89:         <p>The following functions are available:</p>
90:
91:         <h2>set_cookie()</h2>
92:
93:         <p>
94:             This helper function gives you view file friendly syntax to set
95:             browser cookies. Refer to the <a href="../../libraries/input.html"
>Input
96:             class</a> for a description of use, as this function is
an alias to
97:             $this->input->set_cookie().
98:         </p>
99:
100:        <h2>get_cookie()</h2>
101:
102:        <p>
103:            This helper function gives you view file friendly syntax to get
104:            browser cookies. Refer to the <a href="../../libraries/input.html"
>Input
105:            class</a> for a description of use, as this function is
an alias to
106:            $this->input->cookie().
107:        </p>
108:
109:
110:        <h2>delete_cookie()</h2>
111:
112:        <p>Lets you delete a cookie. Unless you've set a custom path or
113:            other values, only the name of the cookie is needed:</p>
114:
```


security_helper.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml
html1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Security Helper : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototypelite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28:     <!-- START NAVIGATION -->
29:     <div id="nav">
30:         <div id="nav_inner">
31:             <script type="text/javascript">create_menu('../');</script>
32:         </div>
33:     </div>
34:     <div id="nav2">
35:         <a name="top"></a><a href="javascript:void(0);"
36:         onclick="myHeight.toggle();"></a>
40:     </div>
41:     <div id="masthead">
42:         <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43:             <tr>
44:                 <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45:                 <td id="breadcrumb_right"><a href="../toc.html">Table o
46:                                     Contents Page</a></td>
47:             </tr>
48:         </table>
49:     </div>
50:     <!-- END NAVIGATION -->
51:
52:     <!-- START BREADCRUMB -->
53:     <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54:         <tr>
55:             <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
56: er
57:             <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="../index.
html">User Guide
58:             <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;Security Helper</t
d>
```

security_helper.html

```
59:           <td id="searchbox"><form method="get"
60:                     action="http://www.google.com/search">
61:                     <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                               value="codeigniter.com/user_guide/" />S
earch User Guide<input
63:       type="text" class="input" style="width:
200px;" name="q" id="q"
64:       size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:       class="submit" name="sa" value="Go" />
66:           </form></td>
67:       </tr>
68: </table>
69: <!-- END BREADCRUMB -->
70:
71: <br clear="all" />
72:
73:
74: <!-- START CONTENT -->
75: <div id="content">
76:
77:
78:         <h1>Security Helper</h1>
79:
80:         <p>The Security Helper file contains security related functions.</p>
81:
82:
83:         <h2>Loading this Helper</h2>
84:
85:         <p>This helper is loaded using the following code:</p>
86:         <code>$this->load->helper('security');</code>
87:
88:         <p>The following functions are available:</p>
89:
90:
91:         <h2>xss_clean()</h2>
92:
93:         <p>
94:             Provides Cross Site Script Hack filtering. This function is an
alias
95:             to the one in the <a href="../libraries/input.html">Input class
</a>.
96:             More info can be found there.
97:         </p>
98:
99:
100:        <h2>sanitize_filename()</h2>
101:
102:        <p>
103:            Provides protection against directory traversal. This function
is an
104:            alias to the one in the <a href="../libraries/security.html">Se
curity
105:            class</a>. More info can be found there.
106:        </p>
107:
108:
109:        <h2>do_hash()</h2>
110:
111:        <p>Permits you to create SHA1 or MD5 one way hashes suitable for
112:            encrypting passwords. Will create SHA1 by default. Examples:</p>
```

>

security_helper.html

170: </div>

171:

172: </body>

173: </html>

number_helper.html

```
60:                                     action="http://www.google.com/search">
61:                                     <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                                     value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp; <input
63:                                     type="text" class="input" style="width:
200px;" name="q" id="q"
64:                                     size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                                     class="submit" name="sa" value="Go" />
66:                                     </form></td>
67:                                     </tr>
68: </table>
69: <!-- END BREADCRUMB -->
70:
71: <br clear="all" />
72:
73:
74: <!-- START CONTENT -->
75: <div id="content">
76:
77:
78:     <h1>Number Helper</h1>
79:
80:     <p>The Number Helper file contains functions that help you work
81:         with numeric data.</p>
82:
83:
84:     <h2>Loading this Helper</h2>
85:
86:     <p>This helper is loaded using the following code:</p>
87:     <code>$this->load->helper('number');</code>
88:
89:     <p>The following functions are available:</p>
90:
91:
92:     <h2>byte_format()</h2>
93:
94:     <p>Formats a numbers as bytes, based on size, and adds the
95:         appropriate suffix. Examples:</p>
96:
97:     <code>
98:         echo byte_format(456); // Returns 456 Bytes<br /> echo
99:         byte_format(4567); // Returns 4.5 KB<br /> echo byte_format(456
78);
100:         // Returns 44.6 KB<br /> echo byte_format(456789); // Returns 4
47.8
101:         KB<br /> echo byte_format(3456789); // Returns 3.3 MB<br /> ech
o
102:         byte_format(12345678912345); // Returns 1.8 GB<br /> echo
103:         byte_format(123456789123456789); // Returns 11,228.3 TB
104:     </code>
105:
106:     <p>An optional second parameter allows you to set the precision of
107:         the result.</p>
108:
109:     <code> echo byte_format(45678, 2); // Returns 44.61 KB </code>
110:
111:     <p class="important">
112:         <strong>Note:</strong> The text generated by this function is f
ound
113:         in the following language file: language/
114:         <your_lang>/number_lang.php
```


moo.fx.js

```
1: /*
2: moo.fx, simple effects library built with prototype.js (http://prototype.conio.net).
3: by Valerio Proietti (http://mad4milk.net) MIT-style LICENSE.
4: for more info (http://moofx.mad4milk.net).
5: 10/24/2005
6: v(1.0.2)
7: */
8:
9: //base
10: var fx = new Object();
11: fx.Base = function(){};
12: fx.Base.prototype = {
13:     setOptions: function(options) {
14:         this.options = {
15:             duration: 500,
16:             onComplete: ''
17:         }
18:         Object.extend(this.options, options || {});
19:     },
20:
21:     go: function() {
22:         this.duration = this.options.duration;
23:         this.startTime = (new Date).getTime();
24:         this.timer = setInterval (this.step.bind(this), 13);
25:     },
26:
27:     step: function() {
28:         var time = (new Date).getTime();
29:         var Tpos = (time - this.startTime) / (this.duration);
30:         if (time >= this.duration+this.startTime) {
31:             this.now = this.to;
32:             clearInterval (this.timer);
33:             this.timer = null;
34:             if (this.options.onComplete) setTimeout(this.options.onComplete
35: .bind(this), 10);
36:         }
37:         else {
38:             this.now = ((-Math.cos(Tpos*Math.PI)/2) + 0.5) * (this.to-this.
39: from) + this.from;
40:             //this time-position, sinoidal transition thing is from script.
41:         }
42:         this.increase();
43:     },
44:
45:     custom: function(from, to) {
46:         if (this.timer != null) return;
47:         this.from = from;
48:         this.to = to;
49:         this.go();
50:     },
51:
52:     hide: function() {
53:         this.now = 0;
54:         this.increase();
55:     },
56:
57:     clearTimer: function() {
58:         clearInterval(this.timer);
59:         this.timer = null;
60:     }
61: }
```

moo.fx.js

```
61: //stretchers
62: fx.Layout = Class.create();
63: fx.Layout.prototype = Object.extend(new fx.Base(), {
64:     initialize: function(el, options) {
65:         this.el = $(el);
66:         this.el.style.overflow = "hidden";
67:         this.el.iniWidth = this.el.offsetWidth;
68:         this.el.iniHeight = this.el.offsetHeight;
69:         this.setOptions(options);
70:     }
71: });
72:
73: fx.Height = Class.create();
74: Object.extend(Object.extend(fx.Height.prototype, fx.Layout.prototype), {
75:     increase: function() {
76:         this.el.style.height = this.now + "px";
77:     },
78:
79:     toggle: function() {
80:         if (this.el.offsetHeight > 0) this.custom(this.el.offsetHeight, 0);
81:         else this.custom(0, this.el.scrollHeight);
82:     }
83: });
```

hacks.txt

```
1: I did the following hack in moo.fx.js:
2:
3: At line 79 in the toggle: function() function, I added:
4:
5: document.getElementById('nav').style.display = 'block';
6:
7: -- Rick Ellis
8:
9:
10: Also removed fx.Opacity and fx.Height from moo.fx.js -- Pascal
```


user_guide_menu.js

```
1: window.onload = function() {  
2:     myHeight = new fx.Height('nav', {duration: 400});  
3:     myHeight.hide();  
4: }
```

```

nav.js
1: function create_menu(basepath)
2: {
3:     var base = (basepath == 'null') ? '' : basepath;
4:
5:     document.write(
6:         '<table cellpadding="0" cellspacing="0" border="0" style="width:98%"><
tr>' +
7:         '<td class="td" valign="top">' +
8:
9:         '<ul>' +
10:        '<li><a href="'+base+'index.html">User Guide Home</a></li>' +
11:        '<li><a href="'+base+'toc.html">Table of Contents Page</a></li>' +
12:        '</ul>' +
13:
14:        '<h3>Basic Info</h3>' +
15:        '<ul>' +
16:            '<li><a href="'+base+'general/requirements.html">Server Require
ments</a></li>' +
17:            '<li><a href="'+base+'license.html">License Agreement</a></li>'
+
18:            '<li><a href="'+base+'changelog.html">Change Log</a></li>' +
19:            '<li><a href="'+base+'general/credits.html">Credits</a></li>' +
20:            '</ul>' +
21:
22:            '<h3>Installation</h3>' +
23:            '<ul>' +
24:                '<li><a href="'+base+'installation/downloads.html">Downloading
CodeIgniter</a></li>' +
25:                '<li><a href="'+base+'installation/index.html">Installation Ins
tructions</a></li>' +
26:                '<li><a href="'+base+'installation/upgrading.html">Upgrading fr
om a Previous Version</a></li>' +
27:                '<li><a href="'+base+'installation/troubleshooting.html">Troubl
eshooting</a></li>' +
28:                '</ul>' +
29:
30:                '<h3>Introduction</h3>' +
31:                '<ul>' +
32:                    '<li><a href="'+base+'overview/getting_started.html">Getting St
arted</a></li>' +
33:                    '<li><a href="'+base+'overview/at_a_glance.html">CodeIgniter at
a Glance</a></li>' +
34:                    '<li><a href="'+base+'overview/cheatsheets.html">CodeIgniter Ch
eatsheets</a></li>' +
35:                    '<li><a href="'+base+'overview/features.html">Supported Feature
s</a></li>' +
36:                    '<li><a href="'+base+'overview/appflow.html">Application Flow C
hart</a></li>' +
37:                    '<li><a href="'+base+'overview/mvc.html">Model-View-Controller<
/a></li>' +
38:                    '<li><a href="'+base+'overview/goals.html">Architectural Goals<
/a></li>' +
39:                    '</ul>' +
40:
41:                '<h3>Tutorial</h3>' +
42:                '<ul>' +
43:                    '<li><a href="'+base+'tutorial/index.html">Introduction</a></li
>' +
44:                    '<li><a href="'+base+'tutorial/static_pages.html">Static pages<
/a></li>' +
45:                    '<li><a href="'+base+'tutorial/news_section.html">News section<
/a></li>' +
46:                    '<li><a href="'+base+'tutorial/create_news_items.html">Create n

```

```

nav.js
ews items</a></li>' +
47:         '<li><a href="' + base + 'tutorial/conclusion.html">Conclusion</a><
/li>' +
48:         '</ul>' +
49:
50:         '</td><td class="td_sep" valign="top">' +
51:
52:         '<h3>General Topics</h3>' +
53:         '<ul>' +
54:         '<li><a href="' + base + 'general/urls.html">CodeIgniter URLs</a></
li>' +
55:         '<li><a href="' + base + 'general/controllers.html">Controllers</a>
</li>' +
56:         '<li><a href="' + base + 'general/reserved_names.html">Reserved Nam
es</a></li>' +
57:         '<li><a href="' + base + 'general/views.html">Views</a></li>' +
58:         '<li><a href="' + base + 'general/models.html">Models</a></li>' +
59:         '<li><a href="' + base + 'general/helpers.html">Helpers</a></li>' +
60:         '<li><a href="' + base + 'general/libraries.html">Using CodeIgniter
Libraries</a></li>' +
61:         '<li><a href="' + base + 'general/creating_libraries.html">Creating
Your Own Libraries</a></li>' +
62:         '<li><a href="' + base + 'general/drivers.html">Using CodeIgniter D
rivers</a></li>' +
63:         '<li><a href="' + base + 'general/creating_drivers.html">Creating Y
our Own Drivers</a></li>' +
64:         '<li><a href="' + base + 'general/core_classes.html">Creating Core
Classes</a></li>' +
65:         '<li><a href="' + base + 'general/hooks.html">Hooks - Extending the
Core</a></li>' +
66:         '<li><a href="' + base + 'general/autoloader.html">Auto-loading Res
ources</a></li>' +
67:         '<li><a href="' + base + 'general/common_functions.html">Common Fun
ctions</a></li>' +
68:         '<li><a href="' + base + 'general/routing.html">URI Routing</a></li
>' +
69:         '<li><a href="' + base + 'general/errors.html">Error Handling</a></
li>' +
70:         '<li><a href="' + base + 'general/caching.html">Caching</a></li>' +
71:         '<li><a href="' + base + 'general/profiling.html">Profiling Your Ap
plication</a></li>' +
72:         '<li><a href="' + base + 'general/cli.html">Running via the CLI</a>
</li>' +
73:         '<li><a href="' + base + 'general/managing_apps.html">Managing Appl
ications</a></li>' +
74:         '<li><a href="' + base + 'general/environments.html">Handling Multi
ple Environments</a></li>' +
75:         '<li><a href="' + base + 'general/alternative_php.html">Alternative
PHP Syntax</a></li>' +
76:         '<li><a href="' + base + 'general/security.html">Security</a></li>'
+
77:         '<li><a href="' + base + 'general/styleguide.html">PHP Style Guide<
/a></li>' +
78:         '<li><a href="' + base + 'doc_style/index.html">Writing Documentati
on</a></li>' +
79:         '</ul>' +
80:
81:         '<h3>Additional Resources</h3>' +
82:         '<ul>' +
83:         '<li><a href="http://forum.codeigniter.com/">Community Forums</a></li>'
+
84:         '<li><a href="https://github.com/bcit-ci/CodeIgniter/wiki">Community Wi
ki</a></li>' +

```

```

nav.js
85:         '</ul>' +
86:
87:         '</td><td class="td_sep" valign="top">' +
88:
89:         '<h3>Class Reference</h3>' +
90:         '<ul>' +
91:         '<li><a href="' + base + 'libraries/benchmark.html">Benchmarking Class</a><
/li>' +
92:         '<li><a href="' + base + 'libraries/calendar.html">Calendar Class</a></li>'
+
93:         '<li><a href="' + base + 'libraries/cart.html">Cart Class</a></li>' +
94:         '<li><a href="' + base + 'libraries/config.html">Config Class</a></li>' +
95:         '<li><a href="' + base + 'libraries/email.html">Email Class</a></li>' +
96:         '<li><a href="' + base + 'libraries/encryption.html">Encryption Class</a></
li>' +
97:         '<li><a href="' + base + 'libraries/file_uploading.html">File Uploading Cla
ss</a></li>' +
98:         '<li><a href="' + base + 'libraries/form_validation.html">Form Validation C
lass</a></li>' +
99:         '<li><a href="' + base + 'libraries/ftp.html">FTP Class</a></li>' +
100:        '<li><a href="' + base + 'libraries/table.html">HTML Table Class</a></li>'
+
101:        '<li><a href="' + base + 'libraries/image_lib.html">Image Manipulation Clas
s</a></li>' +
102:        '<li><a href="' + base + 'libraries/input.html">Input Class</a></li>' +
103:        '<li><a href="' + base + 'libraries/javascript.html">Javascript Class</a></
li>' +
104:        '<li><a href="' + base + 'libraries/loader.html">Loader Class</a></li>' +
105:        '<li><a href="' + base + 'libraries/language.html">Language Class</a></li>'
+
106:        '<li><a href="' + base + 'libraries/migration.html">Migration Class</a></li
>' +
107:        '<li><a href="' + base + 'libraries/output.html">Output Class</a></li>' +
108:        '<li><a href="' + base + 'libraries/pagination.html">Pagination Class</a></
li>' +
109:        '<li><a href="' + base + 'libraries/security.html">Security Class</a></li>'
+
110:        '<li><a href="' + base + 'libraries/sessions.html">Session Class</a></li>'
+
111:        '<li><a href="' + base + 'libraries/trackback.html">Trackback Class</a></li
>' +
112:        '<li><a href="' + base + 'libraries/parser.html">Template Parser Class</a><
/li>' +
113:        '<li><a href="' + base + 'libraries/typography.html">Typography Class</a></
li>' +
114:        '<li><a href="' + base + 'libraries/unit_testing.html">Unit Testing Class</
a></li>' +
115:        '<li><a href="' + base + 'libraries/uri.html">URI Class</a></li>' +
116:        '<li><a href="' + base + 'libraries/user_agent.html">User Agent Class</a></
li>' +
117:        '<li><a href="' + base + 'libraries/xmlrpc.html">XML-RPC Class</a></li>' +
118:        '<li><a href="' + base + 'libraries/zip.html">Zip Encoding Class</a></li>'
+
119:        '</ul>' +
120:
121:        '</td><td class="td_sep" valign="top">' +
122:
123:        '<h3>Driver Reference</h3>' +
124:        '<ul>' +
125:        '<li><a href="' + base + 'libraries/caching.html">Caching Class</a></li>' +
126:        '<li><a href="' + base + 'database/index.html">Database Class</a></li>' +
127:        '<li><a href="' + base + 'libraries/javascript.html">Javascript Class</a></
li>' +

```

```

nav.js
128:         '</ul>' +
129:
130:         '<h3>Helper Reference</h3>' +
131:         '<ul>' +
132:         '<li><a href="'+base+'helpers/array_helper.html">Array Helper</a></li>'
+
133:         '<li><a href="'+base+'helpers/captcha_helper.html">CAPTCHA Helper</a></li>' +
li>' +
134:         '<li><a href="'+base+'helpers/cookie_helper.html">Cookie Helper</a></li>'
>' +
135:         '<li><a href="'+base+'helpers/date_helper.html">Date Helper</a></li>' +
136:         '<li><a href="'+base+'helpers/directory_helper.html">Directory Helper</
a></li>' +
137:         '<li><a href="'+base+'helpers/download_helper.html">Download Helper</a>'
</li>' +
138:         '<li><a href="'+base+'helpers/email_helper.html">Email Helper</a></li>'
+
139:         '<li><a href="'+base+'helpers/file_helper.html">File Helper</a></li>' +
140:         '<li><a href="'+base+'helpers/form_helper.html">Form Helper</a></li>' +
141:         '<li><a href="'+base+'helpers/html_helper.html">HTML Helper</a></li>' +
142:         '<li><a href="'+base+'helpers/inflector_helper.html">Inflector Helper</
a></li>' +
143:         '<li><a href="'+base+'helpers/language_helper.html">Language Helper</a>'
</li>' +
144:         '<li><a href="'+base+'helpers/number_helper.html">Number Helper</a></li>'
>' +
145:         '<li><a href="'+base+'helpers/path_helper.html">Path Helper</a></li>' +
146:         '<li><a href="'+base+'helpers/security_helper.html">Security Helper</a>'
</li>' +
147:         '<li><a href="'+base+'helpers/smiley_helper.html">Smiley Helper</a></li>'
>' +
148:         '<li><a href="'+base+'helpers/string_helper.html">String Helper</a></li>'
>' +
149:         '<li><a href="'+base+'helpers/text_helper.html">Text Helper</a></li>' +
150:         '<li><a href="'+base+'helpers/typography_helper.html">Typography Helper
</a></li>' +
151:         '<li><a href="'+base+'helpers/url_helper.html">URL Helper</a></li>' +
152:         '<li><a href="'+base+'helpers/xml_helper.html">XML Helper</a></li>' +
153:         '</ul>' +
154:
155:         '</td></tr></table>');
156: }

```

prototype.lite.js

```
1: /* Prototype JavaScript framework
2: * (c) 2005 Sam Stephenson <sam@conio.net>
3: *
4: * Prototype is freely distributable under the terms of an MIT-style license.
5: *
6: * For details, see the Prototype web site: http://prototype.conio.net/
7: *
8: /*-----*/
9:
10:
11: //note: this is a stripped down version of prototype, to be used with moo.fx by mad4mil
k (http://moofx.mad4milk.net).
12:
13: var Class = {
14:   create: function() {
15:     return function() {
16:       this.initialize.apply(this, arguments);
17:     }
18:   }
19: }
20:
21: Object.extend = function(destination, source) {
22:   for (property in source) {
23:     destination[property] = source[property];
24:   }
25:   return destination;
26: }
27:
28: Function.prototype.bind = function(object) {
29:   var __method = this;
30:   return function() {
31:     return __method.apply(object, arguments);
32:   }
33: }
34:
35: function $( ) {
36:   var elements = new Array();
37:
38:   for (var i = 0; i < arguments.length; i++) {
39:     var element = arguments[i];
40:     if (typeof element == 'string')
41:       element = document.getElementById(element);
42:
43:     if (arguments.length == 1)
44:       return element;
45:
46:     elements.push(element);
47:   }
48:
49:   return elements;
50: }
51:
52: //-----
53:
54: document.getElementsByClassName = function(className) {
55:   var children = document.getElementsByTagName('*') || document.all;
56:   var elements = new Array();
57:
58:   for (var i = 0; i < children.length; i++) {
59:     var child = children[i];
60:     var classNames = child.className.split(' ');
61:     for (var j = 0; j < classNames.length; j++) {
62:       if (classNames[j] == className) {
```

prototype.lite.js

```
63:             elements.push(child);
64:             break;
65:         }
66:     }
67: }
68:
69: return elements;
70: }
71:
72: //-----
73:
74: if (!window.Element) {
75:     var Element = new Object();
76: }
77:
78: Object.extend(Element, {
79:     remove: function(element) {
80:         element = $(element);
81:         element.parentNode.removeChild(element);
82:     },
83:
84:     hasClassName: function(element, className) {
85:         element = $(element);
86:         if (!element)
87:             return;
88:         var a = element.className.split(' ');
89:         for (var i = 0; i < a.length; i++) {
90:             if (a[i] == className)
91:                 return true;
92:         }
93:         return false;
94:     },
95:
96:     addClassName: function(element, className) {
97:         element = $(element);
98:         Element.removeClassName(element, className);
99:         element.className += ' ' + className;
100:     },
101:
102:     removeClassName: function(element, className) {
103:         element = $(element);
104:         if (!element)
105:             return;
106:         var newClassName = '';
107:         var a = element.className.split(' ');
108:         for (var i = 0; i < a.length; i++) {
109:             if (a[i] != className) {
110:                 if (i > 0)
111:                     newClassName += ' ';
112:                 newClassName += a[i];
113:             }
114:         }
115:         element.className = newClassName;
116:     },
117:
118:     // removes whitespace-only text node children
119:     cleanWhitespace: function(element) {
120:         element = $(element);
121:         for (var i = 0; i < element.childNodes.length; i++) {
122:             var node = element.childNodes[i];
123:             if (node.nodeType == 3 && !/\S/.test(node.nodeValue))
124:                 Element.remove(node);
125:         }

```

prototype.lite.js

```
126:   }  
127: });
```


forge.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Database Forge Class : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype-lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28: <!-- START NAVIGATION -->
29: <div id="nav">
30:     <div id="nav_inner">
31:         <script type="text/javascript">create_menu('../');</script>
32:     </div>
33: </div>
34: <div id="nav2">
35:     <a name="top"></a><a href="javascript:void(0);"
36:     onclick="myHeight.toggle();"></a>
40: </div>
41: <div id="masthead">
42:     <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43:         <tr>
44:             <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45:             <td id="breadcrumb_right"><a href="../toc.html">Table o
46:
47:                                     Contents Page</a></td>
48:         </tr>
49:     </table>
50: <!-- END NAVIGATION -->
51:
52: <!-- START BREADCRUMB -->
53: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54:     <tr>
55:         <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
56: er
57:                                     Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="../index.
58: html">User Guide
59:                                     Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="index.htm
60: l">Database Library</a>
```

forge.html

```
59:                &nbsp;&#8250;&nbsp;&nbsp;&nbsp; Database Forge Class</td>
60:                <td id="searchbox"><form method="get"
61:                    action="http://www.google.com/search">
62:                    <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
63:                        value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;&nbsp;&nbsp; <input
64:                    type="text" class="input" style="width:
200px;" name="q" id="q"
65:                    size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
66:                    class="submit" name="sa" value="Go" />
67:                </form></td>
68:            </tr>
69:        </table>
70:        <!-- END BREADCRUMB -->
71:
72:
73:        <br clear="all" />
74:
75:
76:        <!-- START CONTENT -->
77:        <div id="content">
78:
79:            <h1>Database Forge Class</h1>
80:
81:            <p>The Database Forge Class contains functions that help you
82:                manage your database.</p>
83:
84:            <h3>Table of Contents</h3>
85:
86:            <ul>
87:                <li><a href="#init">Initializing the Forge Class</a></li>
88:                <li><a href="#create">Creating a Database</a></li>
89:                <li><a href="#drop">Dropping a Database</a></li>
90:                <li><a href="#add_field">Adding Fields</a></li>
91:                <li><a href="#add_key">Adding Keys</a></li>
92:                <li><a href="#create_table">Creating a Table</a></li>
93:                <li><a href="#drop_table">Dropping a Table</a></li>
94:                <li><a href="#rename_table">Renaming a Table</a></li>
95:                <li><a href="#modifying_tables">Modifying a Table</a></li>
96:            </ul>
97:
98:
99:            <h2>
100:                <a name="init"></a>Initializing the Forge Class
101:            </h2>
102:
103:            <p class="important">
104:                <strong>Important:</strong>&nbsp;&nbsp;&nbsp; In order to initialize the Fo
rge
105:                class, your database driver must already be running, since the
forge
106:                class relies on it.
107:            </p>
108:
109:            <p>Load the Forge Class as follows:</p>
110:
111:            <code>$this->load->dbforge()</code>
112:
113:            <p>
114:                Once initialized you will access the functions using the
115:                <dfn>$this->dbforge</dfn>
```


forge.html

```
233: </p>
234: <h3>$this-&gt;dbforge-&gt;add_field()</h3>
235: <p>The add fields function will accept the above array.</p>
236: <h3>Passing strings as fields</h3>
237: <p>If you know exactly how you want a field to be created, you can
238:     pass the string into the field definitions with add_field()</p>
239: <p>
240:     <code>$this-&gt;dbforge-&gt;add_field(&quot;label varchar(100)
241:         NOT NULL DEFAULT 'default label'&quot;);</code>
242: </p>
243: <p class="important">
244:     Note: Multiple calls to
245:     <dfn>add_field()</dfn>
246:     are cumulative.
247: </p>
248: <h3>Creating an id field</h3>
249: <p>There is a special exception for creating id fields. A field
250:     with type id will automatically be assigned as an INT(9)
251:     auto_incrementing Primary Key.</p>
252: <p>
253:     <code>
254:         $this-&gt;dbforge-&gt;add_field('id');<br /> // gives i
d INT(9) NOT
255:         NULL AUTO_INCREMENT
256:     </code>
257: </p>
258: <h2>
259:     <a name="add_key" id="add_key"></a>Adding Keys
260: </h2>
261: <p>
262:     Generally speaking, you'll want your table to have Keys. This i
s
263:     accomplished with
264:     <dfn>$this-&gt;dbforge-&gt;add_key('field')</dfn>
265:     . An optional second parameter set to TRUE will make it a prima
ry
266:     key. Note that
267:     <dfn>add_key()</dfn>
268:     must be followed by a call to
269:     <dfn>create_table()</dfn>
270:     .
271: </p>
272: <p>Multiple column non-primary keys must be sent as an array.
273:     Sample output below is for MySQL.</p>
274: <p>
275:     <code>
276:         $this-&gt;dbforge-&gt;add_key('blog_id', TRUE);<br /> /
/ gives
277:         PRIMARY KEY `blog_id` (`blog_id`)<br /> <br />
278:         $this-&gt;dbforge-&gt;add_key('blog_id', TRUE);<br />
279:         $this-&gt;dbforge-&gt;add_key('site_id', TRUE);<br /> /
/ gives
280:         PRIMARY KEY `blog_id_site_id` (`blog_id`, `site_id`)<br
/> <br />
281:         $this-&gt;dbforge-&gt;add_key('blog_name');<br /> // gi
ves KEY
282:         `blog_name` (`blog_name`)<br /> <br />
283:         $this-&gt;dbforge-&gt;add_key(array('blog_name', 'blog_
label'));<br />
284:         // gives KEY `blog_name_blog_label` (`blog_name`, `blog
_label`)
285:     </code>
286: </p>
```


index.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml
tml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>The Database Class : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12:     href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype.lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28:     <!-- START NAVIGATION -->
29:     <div id="nav">
30:         <div id="nav_inner">
31:             <script type="text/javascript">create_menu('../');</script>
32:         </div>
33:     </div>
34:     <div id="nav2">
35:         <a name="top"></a><a href="javascript:void(0);"
36:             onclick="myHeight.toggle();"></a>
40:     </div>
41:     <div id="masthead">
42:         <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43:             <tr>
44:                 <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45:                 <td id="breadcrumb_right"><a href="../toc.html">Table of
46:                 Contents Page</a></td>
47:             </tr>
48:         </table>
49:     </div>
50:     <!-- END NAVIGATION -->
51:
52:     <!-- START BREADCRUMB -->
53:     <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54:         <tr>
55:             <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
56:             er
57:             <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="../index.
html">User Guide
58:             <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="http://codeigniter.com/">Database Library</
td>
```


call_function.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml
html1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Custom Function Calls : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype.lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28: <!-- START NAVIGATION -->
29: <div id="nav">
30: <div id="nav_inner">
31: <script type="text/javascript">create_menu('../');</script>
32: </div>
33: </div>
34: <div id="nav2">
35: <a name="top"></a><a href="javascript:void(0);"
36: onclick="myHeight.toggle();"></a>
40: </div>
41: <div id="masthead">
42: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43: <tr>
44: <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45: <td id="breadcrumb_right"><a href="../toc.html">Table o
46: Contents Page</a></td>
47: </tr>
48: </table>
49: </div>
50: <!-- END NAVIGATION -->
51:
52: <!-- START BREADCRUMB -->
53: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54: <tr>
55: <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
56: er
57: <a href="../index.html">User Guide
58: <a href="index.htm
59: l">Database Library</a>
```

call_function.html

```
59:                 &nbsp;&#8250;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; Custom Function Calls</td>
60: <td id="searchbox"><form method="get"
61:         action="http://www.google.com/search">
62:         <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
63:                 value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; <input
64:                 type="text" class="input" style="width:
200px;" name="q" id="q"
65:                 size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
66:                 class="submit" name="sa" value="Go" />
67:         </form></td>
68:     </tr>
69: </table>
70: <!-- END BREADCRUMB -->
71:
72:
73: <br clear="all" />
74:
75:
76: <!-- START CONTENT -->
77: <div id="content">
78:
79:         <h1>Custom Function Calls</h1>
80:
81:         <h2>$.this->db->call_function();</h2>
82:
83:         <p>
84:             This function enables you to call PHP database functions that a
re not
85:             natively included in CodeIgniter, in a platform independent man-
ner.
86:             For example, lets say you want to call the
87:             <dfn>mysql_get_client_info()</dfn>
88:             function, which is <strong>not</strong> natively supported by
89:             CodeIgniter. You could do so like this:
90:         </p>
91:
92:         <code>
93:             $.this->db->call_function('
94:             <var>get_client_info</var>
95:             ');
96:         </code>
97:
98:         <p>
99:             You must supply the name of the function, <strong>without</stro-
ng>
100:             the
101:             <var>mysql_</var>
102:             prefix, in the first parameter. The prefix is added automatical-
ly
103:             based on which database driver is currently being used. This pe-
rmits
104:             you to run the same function on different database platforms.
105:             Obviously not all function calls are identical between platform
s, so
106:             there are limits to how useful this function can be in terms of
107:             portability.
108:         </p>
109:
110:         <p>Any parameters needed by the function you are calling will be
111:             added to the second parameter.</p>
```


configuration.html

```
110:         simple one is to permit you to optionally store multiple sets o
f
111:         connection values. If, for example, you run multiple environmen
ts
112:         (development, production, test, etc.) under a single installati
on,
113:         you can set up a connection group for each, then switch between
114:         groups as needed. For example, to set up a "test" environment y
ou
115:         would do this:</p>
116:
117:         <code>
118:         $db['test']['hostname'] = "localhost";<br /> $db['test']['usern
ame']
119:         = "root";<br /> $db['test']['password'] = "";<br />
120:         $db['test']['database'] = "database_name";<br />
121:         $db['test']['dbdriver'] = "mysql";<br /> $db['test']['dbprefix'
] =
122:         "";<br /> $db['test']['pconnect'] = TRUE;<br />
123:         $db['test']['db_debug'] = FALSE;<br /> $db['test']['cache_on']
=
124:         FALSE;<br /> $db['test']['cachedir'] = &quot;&quot;;<br />
125:         $db['test']['char_set'] = "utf8";<br /> $db['test']['dbcollat']
=
126:         "utf8_general_ci";<br /> $db['test']['swap_pre'] = "";<br />
127:         $db['test']['autoinit'] = TRUE;<br /> $db['test']['stricton'] =
128:         FALSE;
129:         </code>
130:
131:
132:         <p>Then, to globally tell the system to use that group you would
133:         set this variable located in the config file:</p>
134:
135:         <code>$active_group = "test";</code>
136:
137:         <p>Note: The name "test" is arbitrary. It can be anything you
138:         want. By default we've used the word "default" for the primary
139:         connection, but it too can be renamed to something more relevan
t to
140:         your project.</p>
141:
142:         <h3>Active Record</h3>
143:
144:         <p>
145:         The <a href="active_record.html">Active Record Class</a> is glo
bally
146:         enabled or disabled by setting the $active_record variable in t
he
147:         database configuration file to TRUE/FALSE (boolean). If you are
not
148:         using the active record class, setting it to FALSE will utilize
fewer
149:         resources when the database classes are initialized.
150:         </p>
151:
152:         <code>$active_record = TRUE;</code>
153:
154:         <p class="important">
155:         <strong>Note:</strong> that some CodeIgniter classes such as Se
ssions
156:         require Active Records be enabled to access certain functionali
ty.
157:         </p>
```

configuration.html

```
158:
159:         <h3>Explanation of Values:</h3>
160:
161:         <ul>
162:             <li><strong>hostname</strong> - The hostname of your database
163:                 server. Often this is "localhost".</li>
164:             <li><strong>username</strong> - The username used to connect to
165:                 the database.</li>
166:             <li><strong>password</strong> - The password used to connect to
167:                 the database.</li>
168:             <li><strong>database</strong> - The name of the database you
169:                 want to connect to.</li>
170:             <li><strong>dbdriver</strong> - The database type. ie: mysql,
171:                 postgres, odbc, etc. Must be specified in lower case.</
li>
172:             <li><strong>dbprefix</strong> - An optional table prefix which
173:                 will added to the table name when running <a
174:                 href="active_record.html">Active Record</a> queries. Th
is permits
175:                 multiple CodeIgniter installations to share one databas
e.</li>
176:             <li><strong>pconnect</strong> - TRUE/FALSE (boolean) - Whether
177:                 to use a persistent connection.</li>
178:             <li><strong>db_debug</strong> - TRUE/FALSE (boolean) - Whether
179:                 database errors should be displayed.</li>
180:             <li><strong>cache_on</strong> - TRUE/FALSE (boolean) - Whether
181:                 database query caching is enabled, see also <a href="ca
ching.html">Database
182:                 Caching Class</a>.</li>
183:             <li><strong>cachedir</strong> - The absolute server path to you
r
184:                 database query cache directory.</li>
185:             <li><strong>char_set</strong> - The character set used in
186:                 communicating with the database.</li>
187:             <li><strong>dbcollat</strong> - The character collation used in
188:                 communicating with the database.
189:             <p class="important">
190:                 <strong>Note:</strong> For MySQL and MySQLi dat
abases, this setting
191:                 is only used as a backup if your server is runn
ing PHP &lt; 5.2.3
192:                 or MySQL &lt; 5.0.7 (and in table creation quer
ies made with DB
193:                 Forge). There is an incompatibility in PHP with
194:                 mysql_real_escape_string() which can make your
site vulnerable to
195:                 SQL injection if you are using a multi-byte cha
racter set and are
196:                 running versions lower than these. Sites using
Latin-1 or UTF-8
197:                 database character set and collation are unaffe
cted.
198:             </p></li>
199:             <li><strong>swap_pre</strong> - A default table prefix that
200:                 should be swapped with <var>dbprefix</var>. This is use
ful for
201:                 distributed applications where you might run manually w
ritten
202:                 queries, and need the prefix to still be customizable b
y the end
203:                 user.</li>
204:             <li><strong>autoinit</strong> - Whether or not to automatically
205:                 connect to the database when the library loads. If set
```


results.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml
tml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Generating Query Results : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype_lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28: <!-- START NAVIGATION -->
29: <div id="nav">
30: <div id="nav_inner">
31: <script type="text/javascript">create_menu('../');</script>
32: </div>
33: </div>
34: <div id="nav2">
35: <a name="top"></a><a href="javascript:void(0);"
36: onclick="myHeight.toggle();"></a>
40: </div>
41: <div id="masthead">
42: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43: <tr>
44: <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45: <td id="breadcrumb_right"><a href="../toc.html">Table of
46: Contents Page</a></td>
47: </tr>
48: </table>
49: </div>
50: <!-- END NAVIGATION -->
51:
52: <!-- START BREADCRUMB -->
53: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54: <tr>
55: <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
56: er
57: <a href="../index.html">User Guide
58: <a href="index.htm
59: l">Database Library</a>
```


results.html

```
228:                               $query->previous_row('array')</strong>
229:         </p>
230:
231:
232:
233:         <h1>Result Helper Functions</h1>
234:
235:
236:         <h2>$query->num_rows()</h2>
237:         <p>
238:             The number of rows returned by the query. Note: In this example
239:
240:             <dfn>$query</dfn>
241:             is the variable that the query result object is assigned to:
242:         </p>
243:         <code>
244:             $query = $this->db->query('SELECT * FROM my_table');<br />
245:             <br /> echo $query->num_rows();
246:         </code>
247:
248:         <h2>$query->num_fields()</h2>
249:         <p>The number of FIELDS (columns) returned by the query. Make sure
250:             to call the function using your query result object:</p>
251:
252:         <code>
253:             $query = $this->db->query('SELECT * FROM my_table');<br />
254:             <br /> echo $query->num_fields();
255:         </code>
256:
257:
258:
259:         <h2>$query->free_result()</h2>
260:         <p>It frees the memory associated with the result and deletes the
261:             result resource ID. Normally PHP frees its memory automatically
at
262:             the end of script execution. However, if you are running a lot
of
263:             queries in a particular script you might want to free the resul
t
264:             after each query result has been generated in order to cut down
on
265:             memory consumptions. Example:</p>
266:
267:         <code>
268:             $query = $this->db->query('SELECT title FROM my_table');<br />
269:             <br /> foreach ($query->result() as $row)<br /> {<br />
270:             &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;echo $row->title;<br /> }<br />
271:             $query->free_result(); // The $query result object will no long
er be
272:             available<br /> <br /> $query2 = $this->db->query('SELECT name
FROM
273:             some_table');<br />
274:             <br /> $row = $query2->row();<br /> echo $row->name;<br />
275:             $query2->free_result(); // The $query2 result object will no lo
nger
276:             be available
277:         </code>
278:
279:
280:
281:
282:
```


transactions.html

```
1">Database Library</a>
60:                               &nbsp;&#8250;&nbsp;&nbsp; Transactions</td>
61:                               <td id="searchbox"><form method="get"
62:                               action="http://www.google.com/search">
63:                               <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
64:                               value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;&nbsp; <input
65:                               type="text" class="input" style="width:
200px;" name="q" id="q"
66:                               size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
67:                               class="submit" name="sa" value="Go" />
68:                               </form></td>
69:                               </tr>
70:                               </table>
71:                               <!-- END BREADCRUMB -->
72:
73:
74:                               <br clear="all" />
75:
76:
77:                               <!-- START CONTENT -->
78:                               <div id="content">
79:
80:
81:                               <h1>Transactions</h1>
82:
83:                               <p>
84:                               CodeIgniter's database abstraction allows you to use
85:                               <dfn>transactions</dfn>
86:                               with databases that support transaction-safe table types. In My
SQL,
87:                               you'll need to be running InnoDB or BDB table types rather than
the
88:                               more common MyISAM. Most other database platforms support
89:                               transactions natively.
90:                               </p>
91:
92:                               <p>If you are not familiar with transactions we recommend you find
93:                               a good online resource to learn about them for your particular
94:                               database. The information below assumes you have a basic
95:                               understanding of transactions.</p>
96:
97:                               <h2>CodeIgniter's Approach to Transactions</h2>
98:
99:                               <p>CodeIgniter utilizes an approach to transactions that is very
100:                               similar to the process used by the popular database class ADODB
.
101:                               We've chosen that approach because it greatly simplifies the pr
ocess
102:                               of running transactions. In most cases all that is required are
two
103:                               lines of code.</p>
104:
105:                               <p>
106:                               Traditionally, transactions have required a fair amount of work
to
107:                               implement since they demand that you to keep track of your quer
ies
108:                               and determine whether to
109:                               <dfn>commit</dfn>
110:                               or
```


transactions.html

```
111:         <dfn>rollback</dfn>
112:         based on the success or failure of your queries. This is partic
ularly
113:         cumbersome with nested queries. In contrast, we've implemented
a
114:         smart transaction system that does all this for you automatical
ly
115:         (you can also manage your transactions manually if you choose t
o, but
116:         there's really no benefit).
117:     </p>
118:
119:     <h2>Running Transactions</h2>
120:
121:     <p>
122:         To run your queries using transactions you will use the
123:         <dfn>$this->db->trans_start()</dfn>
124:         and
125:         <dfn>$this->db->trans_complete()</dfn>
126:         functions as follows:
127:     </p>
128:
129:     <code>
130:         <kbd>$this->db->trans_start();</kbd>
131:         <br /> $this->db->query('AN SQL QUERY...');<br />
132:         $this->db->query('ANOTHER QUERY...');<br /> $this->db->query('A
ND
133:         YET ANOTHER QUERY...');<br />
134:         <kbd>$this->db->trans_complete();</kbd>
135:     </code>
136:
137:     <p>You can run as many queries as you want between the
138:         start/complete functions and they will all be committed or roll
ed
139:         back based on success or failure of any given query.</p>
140:
141:
142:     <h2>Strict Mode</h2>
143:
144:     <p>
145:         By default CodeIgniter runs all transactions in
146:         <dfn>Strict Mode</dfn>
147:         . When strict mode is enabled, if you are running multiple grou
ps of
148:         transactions, if one group fails all groups will be rolled back
. If
149:         strict mode is disabled, each group is treated independently, m
eaning
150:         a failure of one group will not affect any others.
151:     </p>
152:
153:     <p>Strict Mode can be disabled as follows:</p>
154:
155:     <code>$this->db->trans_strict(FALSE);</code>
156:
157:
158:     <h2>Managing Errors</h2>
159:
160:     <p>
161:         If you have error reporting enabled in your
162:         <dfn>config/database.php</dfn>
163:         file you'll see a standard error message if the commit was
164:         unsuccessful. If debugging is turned off, you can manage your o
```


utilities.html

```
436: Copyright © 2014 - 2015 &nbsp;&middot;&nbsp;&nbsp;<a
437: href="http://bcit.ca/">British Columbia Institute of Te
chnology</a>
438: </p>
439: </div>
440:
441: </body>
442: </html>
```

caching.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Database Caching Class : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../../nav/nav.js"></script>
15: <script type="text/javascript" src="../../nav/prototype-lite.js"></script>
16: <script type="text/javascript" src="../../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28: <!-- START NAVIGATION -->
29: <div id="nav">
30: <div id="nav_inner">
31: <script type="text/javascript">create_menu('../');</script>
32: </div>
33: </div>
34: <div id="nav2">
35: <a name="top"></a><a href="javascript:void(0);"
36: onclick="myHeight.toggle();"></a>
40: </div>
41: <div id="masthead">
42: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43: <tr>
44: <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45: <td id="breadcrumb_right"><a href="../../toc.html">Table of
46: Contents Page</a></td>
47: </tr>
48: </table>
49: </div>
50: <!-- END NAVIGATION -->
51:
52: <!-- START BREADCRUMB -->
53: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54: <tr>
55: <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
56: er
57: <a href="../../index.html">User Guide
58: <a href="index.htm
59: l">Database Library</a>
```


caching.html

114:
115: **<p>**CodeIgniter's query caching system happens dynamically when
116: your pages are viewed. When caching is enabled, the first time
a web
117: page is loaded, the query result object will be serialized and
stored
118: in a text file on your server. The next time the page is loaded
the
119: cache file will be used instead of accessing your database. You
r
120: database usage can effectively be reduced to zero for any pages
that
121: have been cached.**</p>**
122:
123: **<p>**
124: Only
125: **<dfn>**read-type**</dfn>**
126: (SELECT) queries can be cached, since these are the only type o
f
127: queries that produce a result.
128: **<dfn>**Write-type**</dfn>**
129: (INSERT, UPDATE, etc.) queries, since they don't generate a res
ult,
130: will not be cached by the system.
131: **</p>**
132:
133: **<p>**Cache files DO NOT expire. Any queries that have been cached
134: will remain cached until you delete them. The caching system pe
rmits
135: you clear caches associated with individual pages, or you can d
elete
136: the entire collection of cache files. Typically you'll want to
use
137: the housekeeping functions described below to delete cache file
s
138: after certain events take place, like when you've added new
139: information to your database.**</p>**
140:
141: **<h2>**Will Caching Improve Your Site's Performance?**</h2>**
142:
143: **<p>**Getting a performance gain as a result of caching depends on
144: many factors. If you have a highly optimized database under ver
y
145: little load, you probably won't see a performance boost. If you
r
146: database is under heavy use you probably will see an improved
147: response, assuming your file-system is not overly taxed. Rememb
er
148: that caching simply changes how your information is retrieved,
149: shifting it from being a database operation to a file-system on
e.**</p>**
150:
151: **<p>**In some clustered server environments, for example, caching may
152: be detrimental since file-system operations are so intense. On
single
153: servers in shared environments, caching will probably be benefi
cial.
154: Unfortunately there is no single answer to the question of whet
her
155: you should cache your database. It really depends on your situa
tion.**</p>**
156:
157: **<h2>**How are Cache Files Stored?**</h2>**

caching.html

```
158:
159: <p>CodeIgniter places the result of EACH query into its own cache
160: file. Sets of cache files are further organized into sub-folder
s
161: corresponding to your controller functions. To be precise, the
162: sub-folders are named identically to the first two segments of
your
163: URI (the controller class name and function name).</p>
164:
165: <p>
166: For example, let's say you have a controller called
167: <dfn>blog</dfn>
168: with a function called
169: <dfn>comments</dfn>
170: that contains three queries. The caching system will create a c
ache
171: folder called
172: <kbd>blog+comments</kbd>
173: , into which it will write three cache files.
174: </p>
175:
176: <p>If you use dynamic queries that change based on information in
177: your URI (when using pagination, for example), each instance of
the
178: query will produce its own cache file. It's possible, therefore
, to
179: end up with many times more cache files than you have queries.<
/p>
180:
181:
182: <h2>Managing your Cache Files</h2>
183:
184: <p>Since cache files do not expire, you'll need to build deletion
185: routines into your application. For example, let's say you have
a
186: blog that allows user commenting. Whenever a new comment is sub
mitted
187: you'll want to delete the cache files associated with the contr
oller
188: function that serves up your comments. You'll find two delete
189: functions described below that help you clear data.</p>
190:
191:
192: <h2>Not All Database Functions Work with Caching</h2>
193:
194: <p>Lastly, we need to point out that the result object that is
195: cached is a simplified version of the full result object. For t
hat
196: reason, some of the query result functions are not available fo
r use.</p>
197:
198: <p>
199: The following functions
200: <kbd>ARE NOT</kbd>
201: available when using a cached result object:
202: </p>
203:
204: <ul>
205: <li>num_fields()</li>
206: <li>field_names()</li>
207: <li>field_data()</li>
208: <li>free_result()</li>
209: </ul>
```


caching.html

```
210:
211: <p>Also, the two database resources (result_id and conn_id) are
212:     not available when caching, since result resources only pertain
to
213:     run-time operations.</p>
214:
215:
216: <br />
217:
218: <h1>Function Reference</h1>
219:
220:
221:
222: <h2>${this->db->cache_on()}&nbsp; / &nbsp; ${this->db->cache_off()}</h2>
223:
224: <p>Manually enables/disables caching. This can be useful if you
225:     want to keep certain queries from being cached. Example:</p>
226:
227: <code>
228:     // Turn caching on<br /> ${this->db->cache_on();<br /> $query =
229:     ${this->db->query("SELECT * FROM mytable");<br /> <br /> // Turn
230:     caching off for this one query<br /> ${this->db->cache_off();<br
/>
231:     $query = ${this->db->query("SELECT * FROM members WHERE member_i
d =
232:     '$current_user'");<br /> <br /> // Turn caching back on<br />
233:     ${this->db->cache_on();<br /> $query = ${this->db->query("SELECT
*
234:     FROM another_table");
235: </code>
236:
237:
238: <h2>${this->db->cache_delete()}</h2>
239:
240: <p>Deletes the cache files associated with a particular page. This
241:     is useful if you need to clear caching after you update your
242:     database.</p>
243:
244: <p>
245:     The caching system saves your cache files to folders that corre
spond
246:     to the URI of the page you are viewing. For example, if you are
247:     viewing a page at
248:     <dfn>example.com/index.php/blog/comments</dfn>
249:     , the caching system will put all cache files associated with i
t in a
250:     folder called
251:     <dfn>blog+comments</dfn>
252:     . To delete those particular cache files you will use:
253: </p>
254:
255: <code>${this->db->cache_delete('blog', 'comments');</code>
256:
257: <p>If you do not use any parameters the current URI will be used
258:     when determining what should be cleared.</p>
259:
260:
261: <h2>${this->db->cache_delete_all()}</h2>
262:
263: <p>Clears all existing cache files. Example:</p>
264:
265: <code>${this->db->cache_delete_all();</code>
266:
```


helpers.html

```
115:
116:
117: <h2>$this->db->last_query();</h2>
118: <p>Returns the last query that was run (the query string, not the
119:     result). Example:</p>
120:
121: <code>
122:     $str = $this->db->last_query();<br /> <br /> // Produces: SELEC
T *
123:     FROM sometable....
124: </code>
125:
126:
127: <p>The following two functions help simplify the process of
128:     writing database INSERTs and UPDATEs.</p>
129:
130:
131: <h2>$this->db->insert_string();</h2>
132: <p>This function simplifies the process of writing database
133:     inserts. It returns a correctly formatted SQL insert string. Ex
ample:</p>
134:
135: <code>
136:     $data = array('name' => $name, 'email' => $email, 'url' => $url
);<br />
137:     <br /> $str = $this->db->insert_string('table_name', $data);
138: </code>
139:
140: <p>The first parameter is the table name, the second is an
141:     associative array with the data to be inserted. The above examp
le
142:     produces:</p>
143: <code>INSERT INTO table_name (name, email, url) VALUES ('Rick',
144:     'rick@example.com', 'example.com')</code>
145:
146: <p class="important">Note: Values are automatically escaped,
147:     producing safer queries.</p>
148:
149:
150:
151: <h2>$this->db->update_string();</h2>
152: <p>This function simplifies the process of writing database
153:     updates. It returns a correctly formatted SQL update string. Ex
ample:</p>
154:
155: <code>
156:     $data = array('name' => $name, 'email' => $email, 'url' => $url
);<br />
157:     <br /> $where = "author_id = 1 AND status = 'active'"; <br />
158:     <br /> $str = $this->db->update_string('table_name', $data, $wh
ere);
159: </code>
160:
161: <p>The first parameter is the table name, the second is an
162:     associative array with the data to be updated, and the third
163:     parameter is the "where" clause. The above example produces:</p>
>
164: <code> UPDATE table_name SET name = 'Rick', email =
165:     'rick@example.com', url = 'example.com' WHERE author_id = 1 AND
166:     status = 'active'</code>
167:
168: <p class="important">Note: Values are automatically escaped,
169:     producing safer queries.</p>
```


table_data.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Table Data : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype_lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28: <!-- START NAVIGATION -->
29: <div id="nav">
30: <div id="nav_inner">
31: <script type="text/javascript">create_menu('../');</script>
32: </div>
33: </div>
34: <div id="nav2">
35: <a name="top"></a><a href="javascript:void(0);"
36: onclick="myHeight.toggle();"></a>
40: </div>
41: <div id="masthead">
42: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43: <tr>
44: <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45: <td id="breadcrumb_right"><a href="../toc.html">Table of
46: Contents Page</a></td>
47: </tr>
48: </table>
49: </div>
50: <!-- END NAVIGATION -->
51:
52: <!-- START BREADCRUMB -->
53: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54: <tr>
55: <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
56: er
57: <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="http://codeigniter.com/user_guide/index.html">User Guide
58: <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="http://codeigniter.com/user_guide/index.html">Database Library</a>
```


connecting.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Connecting to your Database : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototypelite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28: <!-- START NAVIGATION -->
29: <div id="nav">
30: <div id="nav_inner">
31: <script type="text/javascript">create_menu('../');</script>
32: </div>
33: </div>
34: <div id="nav2">
35: <a name="top"></a><a href="javascript:void(0);"
36: onclick="myHeight.toggle();"></a>
40: </div>
41: <div id="masthead">
42: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43: <tr>
44: <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45: <td id="breadcrumb_right"><a href="../toc.html">Table of
46: Contents Page</a></td>
47: </tr>
48: </table>
49: </div>
50: <!-- END NAVIGATION -->
51:
52: <!-- START BREADCRUMB -->
53: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54: <tr>
55: <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
56: er
57: <a href="../index.html">User Guide
58: <a href="index.html">Database Library</a>
```

connecting.html

```
59:          &nbsp;&#8250;&nbsp;&nbsp;&nbsp; Connecting</td>
60:          <td id="searchbox"><form method="get"
61:              action="http://www.google.com/search">
62:              <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
63:                  value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;&nbsp;&nbsp; <input
64:                  type="text" class="input" style="width:
200px;" name="q" id="q"
65:                  size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
66:                  class="submit" name="sa" value="Go" />
67:              </form></td>
68:          </tr>
69:      </table>
70:      <!-- END BREADCRUMB -->
71:
72:
73:      <br clear="all" />
74:
75:
76:      <!-- START CONTENT -->
77:      <div id="content">
78:
79:
80:          <h1>Connecting to your Database</h1>
81:
82:          <p>There are two ways to connect to a database:</p>
83:
84:          <h2>Automatically Connecting</h2>
85:
86:          <p>
87:              The "auto connect" feature will load and instantiate the databa
se
88:              class with every page load. To enable "auto connecting", add th
e word
89:              <var>database</var>
90:              to the library array, as indicated in the following file:
91:          </p>
92:
93:          <p>
94:              <kbd>application/config/autoload.php</kbd>
95:          </p>
96:
97:          <h2>Manually Connecting</h2>
98:
99:          <p>If only some of your pages require database connectivity you
100:              can manually connect to your database by adding this line of co
de in
101:              any function where it is needed, or in your class constructor t
o make
102:              the database available globally in that class.</p>
103:
104:          <code>$this->load->database();</code>
105:
106:          <p class="important">
107:              If the above function does <strong>not</strong> contain any
108:              information in the first parameter it will connect to the group
109:              specified in your database config file. For most people, this i
s the
110:              preferred method of use.
111:          </p>
112:
```


connecting.html

```
113:         <h3>Available Parameters</h3>
114:
115:         <ol>
116:             <li>The database connection values, passed either as an array o
r
117:                 a DSN string.</li>
118:             <li>TRUE/FALSE (boolean). Whether to return the connection ID
119:                 (see Connecting to Multiple Databases below).</li>
120:             <li>TRUE/FALSE (boolean). Whether to enable the Active Record
121:                 class. Set to TRUE by default.</li>
122:         </ol>
123:
124:
125:         <h3>Manually Connecting to a Database</h3>
126:
127:         <p>
128:             The first parameter of this function can <strong>optionally</st
rong>
129:             be used to specify a particular database group from your config
file,
130:             or you can even submit connection values for a database that is
not
131:             specified in your config file. Examples:
132:
133:         </p>
134:         <p>To choose a specific group from your config file you can do
135:             this:</p>
136:
137:         <code>
138:             $this->load->database('
139:             <samp>group_name</samp>
140:             ');
141:         </code>
142:
143:         <p>
144:             Where
145:             <samp>group_name</samp>
146:             is the name of the connection group from your config file.
147:
148:         </p>
149:
150:         <p>To connect manually to a desired database you can pass an array
151:             of values:</p>
152:
153:         <code>
154:             $config['hostname'] = "localhost";<br /> $config['username'] =
155:             "myusername";<br /> $config['password'] = "mypassword";<br />
156:             $config['database'] = "mydatabase";<br /> $config['dbdriver'] =
157:             "mysql";<br /> $config['dbprefix'] = "";<br /> $config['pconnec
t']
158:             = FALSE;<br /> $config['db_debug'] = TRUE;<br />
159:             $config['cache_on'] = FALSE;<br /> $config['cachedir'] = "";<br
/>
160:             $config['char_set'] = "utf8";<br /> $config['dbcollat'] =
161:             "utf8_general_ci";<br /> <br /> $this->load->database(
162:             <samp>$config</samp>
163:             );
164:         </code>
165:
166:         <p>
167:             For information on each of these values please see the <a
168:                 href="configuration.html">configuration page</a>.
169:         </p>
```

connecting.html

```
170:
171: <p>Or you can submit your database values as a Data Source Name.
172:     DSNs must have this prototype:</p>
173:
174: <code>
175:     $dsn = 'dbdriver://username:password@hostname/database';<br />
<br />
176:     $this->load->database(
177:         <samp>$dsn</samp>
178:     );
179: </code>
180:
181: <p>To override default config values when connecting with a DSN
182:     string, add the config variables as a query string.</p>
183:
184: <code>
185:     $dsn =
186:         'dbdriver://username:password@hostname/database?char_set=utf8&a
mp;dbcollat=utf8_general_ci&amp;cache_on=true&amp;cachedir=/path/to/cache';<br />
187:     <br /> $this->load->database(
188:         <samp>$dsn</samp>
189:     );
190: </code>
191:
192: <h2>Connecting to Multiple Databases</h2>
193:
194: <p>If you need to connect to more than one database simultaneously
195:     you can do so as follows:</p>
196:
197:
198: <code>
199:     $DB1 = $this->load->database('group_one', TRUE);<br /> $DB2 =
200:     $this->load->database('group_two', TRUE);
201: </code>
202:
203: <p>Note: Change the words "group_one" and "group_two" to the
204:     specific group names you are connecting to (or you can pass the
205:     connection values as indicated above).</p>
206:
207: <p>By setting the second parameter to TRUE (boolean) the function
208:     will return the database object.</p>
209:
210: <div class="important">
211:     <p>When you connect this way, you will use your object name to
212:         issue commands rather than the syntax used throughout t
his guide. In
213:         other words, rather than issuing commands with:</p>
214:
215:     <p>
216:         $this->db->query();<br />$this->db->result();<br /> etc
...
217:     </p>
218:
219:     <p>You will instead use:</p>
220:
221:     <p>
222:         $DB1->query();<br />$DB1->result();<br /> etc...
223:     </p>
224:
225: </div>
226:
227: <h2>Reconnecting / Keeping the Connection Alive</h2>
228:
```


active_record.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xh
tml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Active Record : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12:       href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype.lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24: </head>
25: <body>
26:
27:     <!-- START NAVIGATION -->
28:     <div id="nav">
29:         <div id="nav_inner">
30:             <script type="text/javascript">create_menu('../');</script>
31:         </div>
32:     </div>
33:     <div id="nav2">
34:         <a name="top"></a><a href="javascript:void(0);"
35:             onclick="myHeight.toggle();"></a>
39:     </div>
40:     <div id="masthead">
41:         <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
42:             <tr>
43:                 <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
44:                 <td id="breadcrumb_right"><a href="../toc.html">Table of
45:                     Contents Page</a></td>
46:             </tr>
47:         </table>
48:     </div>
49:     <!-- END NAVIGATION -->
50:
51:     <!-- START BREADCRUMB -->
52:     <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
53:         <tr>
54:             <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
55:                 er
56:                 <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="../index.
57:                 <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="index.htm
58:                 &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="index.htm">Database Library</a>
59:                 &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="index.htm">Active Record</td>
```

active_record.html

```
59:          <td id="searchbox"><form method="get"
60:              action="http://www.google.com/search">
61:              <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                  value="codeigniter.com/user_guide/" />S
earch User Guide<input
63:                  type="text" class="input" style="width:
200px;" name="q" id="q"
64:                  size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                  class="submit" name="sa" value="Go" />
66:          </form></td>
67:      </tr>
68:  </table>
69:  <!-- END BREADCRUMB -->
70:
71:  <br clear="all" />
72:
73:  <!-- START CONTENT -->
74:  <div id="content">
75:
76:      <h1>Active Record Class</h1>
77:
78:      <p>CodeIgniter uses a modified version of the Active Record
79:          Database Pattern. This pattern allows information to be retriev
ed,
80:          inserted, and updated in your database with minimal scripting.
In
81:          some cases only one or two lines of code are necessary to perfo
rm a
82:          database action. CodeIgniter does not require that each databas
e
83:          table be its own class file. It instead provides a more simplif
ied
84:          interface.</p>
85:
86:      <p>Beyond simplicity, a major benefit to using the Active Record
87:          features is that it allows you to create database independent
88:          applications, since the query syntax is generated by each datab
ase
89:          adapter. It also allows for safer queries, since the values are
90:          escaped automatically by the system.</p>
91:
92:      <p class="important">
93:          <strong>Note:</strong> If you intend to write your own queries
you
94:          can disable this class in your database config file, allowing t
he
95:          core database library and adapter to utilize fewer resources.<b
r />
96:      </p>
97:
98:      <ul>
99:          <li><a href="#select">Selecting Data</a></li>
100:         <li><a href="#insert">Inserting Data</a></li>
101:         <li><a href="#update">Updating Data</a></li>
102:         <li><a href="#delete">Deleting Data</a></li>
103:         <li><a href="#chaining">Method Chaining</a></li>
104:         <li><a href="#caching">Active Record Caching</a></li>
105:      </ul>
106:
107:      <h1>
108:          <a name="select">&nbsp;</a>Selecting Data
```


active_record.html

```
168: <h2>$this->db->select();</h2>
169: <p>Permits you to write the SELECT portion of your query:</p>
170: <p>
171:     <code>
172:         $this->db->select('title, content, date');<br /> <br />
$query =
173:         $this->db->get('mytable');<br /> <br /> // Produces: SE
LECT title,
174:         content, date FROM mytable
175:     </code>
176: </p>
177: <p class="important">
178:     <strong>Note:</strong> If you are selecting all (*) from a tabl
e you
179:     do not need to use this function. When omitted, CodeIgniter ass
umes
180:     you wish to SELECT *
181: </p>
182:
183: <p>$this-&gt;db-&gt;select() accepts an optional second parameter.
184:     If you set it to FALSE, CodeIgniter will not try to protect you
r
185:     field or table names with backticks. This is useful if you need
a
186:     compound select statement.</p>
187: <p>
188:     <code>
189:         $this-&gt;db-&gt;select('(SELECT SUM(payments.amount) F
ROM payments
190:         WHERE payments.invoice_id=4') AS amount_paid', FALSE);
<br />
191:         $query = $this-&gt;db-&gt;get('mytable');<br />
192:     </code>
193: </p>
194: <h2>$this->db->select_max();</h2>
195: <p>Writes a "SELECT MAX(field)" portion for your query. You can
196:     optionally include a second parameter to rename the resulting f
ield.</p>
197: <p>
198:     <code>
199:         $this->db->select_max('age');<br /> $query =
200:         $this->db->get('members');<br /> // Produces: SELECT MA
X(age) as
201:         age FROM members<br /> <br /> $this-&gt;db-&gt;select_m
ax('age',
202:         'member_age');<br /> $query = $this-&gt;db-&gt;get('mem
bers');<br />
203:         // Produces: SELECT MAX(age) as member_age FROM members
204:     </code>
205: </p>
206:
207: <h2>$this->db->select_min();</h2>
208: <p>
209:     Writes a "SELECT MIN(field)" portion for your query. As with
210:     <dfn>select_max()</dfn>
211:     , You can optionally include a second parameter to rename the
212:     resulting field.
213: </p>
214: <p>
215:     <code>
216:         $this->db->select_min('age');<br /> $query =
217:         $this->db->get('members');<br /> // Produces: SELECT MI
N(age) as
```

active_record.html

```
218:         age FROM members
219:     </code>
220: </p>
221:
222: <h2>${this->db->select_avg()};</h2>
223: <p>
224:     Writes a "SELECT AVG(field)" portion for your query. As with
225:     <dfn>select_max()</dfn>
226:     , You can optionally include a second parameter to rename the
227:     resulting field.
228: </p>
229: <p>
230:     <code>
231:         ${this->db->select_avg('age');<br /> $query =
232:         ${this->db->get('members');<br /> // Produces: SELECT AV
G(age) as
233:         age FROM members
234:     </code>
235: </p>
236:
237: <h2>${this->db->select_sum()};</h2>
238: <p>
239:     Writes a "SELECT SUM(field)" portion for your query. As with
240:     <dfn>select_max()</dfn>
241:     , You can optionally include a second parameter to rename the
242:     resulting field.
243: </p>
244: <p>
245:     <code>
246:         ${this->db->select_sum('age');<br /> $query =
247:         ${this->db->get('members');<br /> // Produces: SELECT SU
M(age) as
248:         age FROM members
249:     </code>
250: </p>
251:
252: <h2>${this->db->from()};</h2>
253:
254: <p>Permits you to write the FROM portion of your query:</p>
255:
256: <code>
257:     ${this->db->select('title, content, date');<br />
258:     ${this->db->from('mytable');<br /> <br /> $query = ${this->db->ge
t()<br />
259:     <br /> // Produces: SELECT title, content, date FROM mytable
260: </code>
261:
262: <p class="important">
263:     Note: As shown earlier, the FROM portion of your query can be
264:     specified in the
265:     <dfn>${this->db->get()</dfn>
266:     function, so use whichever method you prefer.
267: </p>
268:
269: <h2>${this->db->join()};</h2>
270:
271: <p>Permits you to write the JOIN portion of your query:</p>
272:
273: <code>
274:     ${this->db->select('*');<br /> ${this->db->from('blogs');<br />
275:     ${this->db->join('comments', 'comments.id = blogs.id');<br /> <b
r />
276:     $query = ${this->db->get()<br /> <br /> // Produces: <br /> //
```


active_record.html

```
277:             SELECT * FROM blogs<br /> // JOIN comments ON comments.id = blo
gs.id<br />
278:             </code>
279:
280:             <p>Multiple function calls can be made if you need several joins
281:                 in one query.</p>
282:
283:             <p>If you need a specific type of JOIN you can specify it via the
284:                 third parameter of the function. Options are: left, right, oute
r,
285:                 inner, left outer, and right outer.</p>
286:
287:             <code>
288:                 $this->db->join('comments', 'comments.id = blogs.id', <strong>'
left'</strong>);<br />
289:                 <br /> // Produces: LEFT JOIN comments ON comments.id = blogs.i
d
290:             </code>
291:
292:
293:
294:
295:
296:             <h2>{$this->db->where()}</h2>
297:             <p>
298:                 This function enables you to set <strong>WHERE</strong> clauses
using
299:                 one of four methods:
300:             </p>
301:
302:             <p class="important">
303:                 <strong>Note:</strong> All values passed to this function are e
scaped
304:                 automatically, producing safer queries.
305:             </p>
306:
307:             <ol>
308:                 <li><strong>Simple key/value method:</strong> <code>
309:                     $this->db->where('name', $name); <br />
310:                     <br /> // Produces: WHERE name = 'Joe'
311:                 </code>
312:
313:                 <p>Notice that the equal sign is added for you.</p>
314:
315:                 <p>
316:                     If you use multiple function calls they will be
chained together
317:                     with
318:                     <var>AND</var>
319:                     between them:
320:                 </p> <code>
321:                     $this->db->where('name', $name);<br /> $this->d
b->where('title',
322:                     $title);<br /> $this->db->where('status', $stat
us); <br />
323:                     <br /> // WHERE name = 'Joe' AND title = 'boss'
AND status =
324:                     'active'
325:                 </code></li>
326:
327:                 <li><strong>Custom key/value method:</strong>
328:
329:                 <p>You can include an operator in the first parameter i
```

active_record.html

```
n order
330:         to control the comparison:</p> <code>
331:         $this->db->where('name !=', $name);<br /> $this
->db->where('id <',
332:         $id); <br />
333:         <br />// Produces: WHERE name != 'Joe' AND id <
45
334:         </code></li>
335:     <li><strong>Associative array method:</strong> <code>
336:         $array = array('name' => $name, 'title' => $tit
le, 'status' =>
337:         $status);<br />
338:         <br /> $this->db->where($array); <br />
339:         <br />// Produces: WHERE name = 'Joe' AND title
= 'boss' AND
340:         status = 'active'
341:     </code>
342:
343:     <p>You can include your own operators using this method
as well:</p>
344:
345:     <code>
346:         $array = array('name !=' => $name, 'id <' => $i
d, 'date >' =>
347:         $date);<br />
348:         <br /> $this->db->where($array);
349:     </code></li>
350: <li><strong>Custom string:</strong>
351:
352:     <p>You can write your own clauses manually:</p> <code>
353:         $where = "name='Joe' AND status='boss' OR statu
s='active'";<br />
354:         <br /> $this->db->where($where);
355:     </code></li>
356: </ol>
357:
358:
359: <p>$this-&gt;db-&gt;where() accepts an optional third parameter.
360:     If you set it to FALSE, CodeIgniter will not try to protect you
r
361:     field or table names with backticks.</p>
362: <p>
363:     <code>
364:         $this-&gt;db-&gt;where('MATCH (field) AGAINST (&quot;va
lue&quot;)',
365:         NULL, FALSE);<br />
366:     </code>
367: </p>
368: <h2>$this->db->or_where();</h2>
369: <p>This function is identical to the one above, except that
370:     multiple instances are joined by OR:</p>
371:
372: <code>
373:     $this->db->where('name !=', $name);<br /> $this->db->or_where('
id
374:     >', $id); <br /> <br />// Produces: WHERE name != 'Joe' OR id >
50
375: </code>
376:
377: <p class="important">Note: or_where() was formerly known as
378:     orwhere(), which has been removed.</p>
379:
380:
```

active_record.html

```
381: <h2>$this->db->where_in();</h2>
382: <p>Generates a WHERE field IN ('item', 'item') SQL query joined
383: with AND if appropriate</p>
384: <p>
385: <code>
386: $names = array('Frank', 'Todd', 'James');<br />
387: $this->db->where_in('username', $names);<br /> // Produces: WHERE
ces: WHERE
388: username IN ('Frank', 'Todd', 'James')
389: </code>
390: </p>
391:
392: <h2>$this->db->or_where_in();</h2>
393: <p>Generates a WHERE field IN ('item', 'item') SQL query joined
394: with OR if appropriate</p>
395: <p>
396: <code>
397: $names = array('Frank', 'Todd', 'James');<br />
398: $this->db->or_where_in('username', $names);<br /> // Produces: OR
roduces: OR
399: username IN ('Frank', 'Todd', 'James')
400: </code>
401: </p>
402:
403: <h2>$this->db->where_not_in();</h2>
404: <p>Generates a WHERE field NOT IN ('item', 'item') SQL query
405: joined with AND if appropriate</p>
406: <p>
407: <code>
408: $names = array('Frank', 'Todd', 'James');<br />
409: $this->db->where_not_in('username', $names);<br /> // Produces:
roduces:
410: WHERE username NOT IN ('Frank', 'Todd', 'James')
411: </code>
412: </p>
413:
414: <h2>$this->db->or_where_not_in();</h2>
415: <p>Generates a WHERE field NOT IN ('item', 'item') SQL query
416: joined with OR if appropriate</p>
417: <p>
418: <code>
419: $names = array('Frank', 'Todd', 'James');<br />
420: $this->db->or_where_not_in('username', $names);<br /> // Produces:
/ Produces:
421: OR username NOT IN ('Frank', 'Todd', 'James')
422: </code>
423: </p>
424:
425: <h2>$this->db->like();</h2>
426: <p>
427: This function enables you to generate <strong>LIKE</strong> clauses,
uses,
428: useful for doing searches.
429: </p>
430:
431: <p class="important">
432: <strong>Note:</strong> All values passed to this function are escaped
scaped
433: automatically.
434: </p>
435:
436:
437: <ol>
```

active_record.html

```
438:         <li><strong>Simple key/value method:</strong> <code>
439:             $this->db->like('title', 'match'); <br />
440:             <br /> // Produces: WHERE title LIKE '%match%'
441:         </code>
442:
443:         <p>
444:             If you use multiple function calls they will be
chained together
445:             with
446:             <var>AND</var>
447:             between them:
448:         </p> <code>
449:             $this->db->like('title', 'match');<br /> $this-
>db->like('body',
450:             'match'); <br />
451:             <br /> // WHERE title LIKE '%match%' AND body L
IKE '%match%'
452:         </code> If you want to control where the wildcard (%) i
s placed, you can
453:         use an optional third argument. Your options are 'befor
e', 'after'
454:         and 'both' (which is the default). <code>
455:             $this->db->like('title', 'match', 'before'); <b
r /> // Produces:
456:             WHERE title LIKE '%match' <br /> <br />
457:             $this->db->like('title', 'match', 'after'
); <br /> //
458:             Produces: WHERE title LIKE 'match%' <br /> <br
 />
459:             $this->db->like('title', 'match', 'both')
; <br /> //
460:             Produces: WHERE title LIKE '%match%'
461:         </code></li> If you do not want to use the wildcard (%)
you can pass to the
462:         optional third argument the option 'none'.
463:
464:         <code>
465:             $this->db->like('title', 'match', 'none'); <br />
//
466:             Produces: WHERE title LIKE 'match'
467:         </code>
468:
469:         <li><strong>Associative array method:</strong> <code>
470:             $array = array('title' => $match, 'page1' => $m
atch, 'page2' =>
471:             $match);<br />
472:             <br /> $this->db->like($array); <br />
473:             <br /> // WHERE title LIKE '%match%' AND page1 L
IKE '%match%' AND
474:             page2 LIKE '%match%'
475:         </code></li>
476:     </ol>
477:
478:
479:     <h2>$this->db->or_like();</h2>
480:     <p>This function is identical to the one above, except that
multiple instances are joined by OR:</p>
481:
482:     <code>
483:         $this->db->like('title', 'match');<br /> $this->db->or_like('bo
dy',
484:         $match); <br /> <br /> // WHERE title LIKE '%match%' OR body LIK
E
```

active_record.html

```
486:           '%match%'
487:         </code>
488:
489:
490:
491:
492:         <p class="important">Note: or_like() was formerly known as
493:           orlike(), which has been removed.</p>
494:         <h2>${this-&gt;db-&gt;not_like()}</h2>
495:         <p>
496:           This function is identical to <strong>like()</strong>, except t
hat it
497:           generates NOT LIKE statements:
498:         </p>
499:         <code>
500:           ${this-&gt;db-&gt;not_like('title', 'match');<br /> <br /> // WH
ERE
501:           title NOT LIKE '%match%'
502:         </code>
503:         <h2>${this-&gt;db-&gt;or_not_like()}</h2>
504:         <p>
505:           This function is identical to <strong>not_like()</strong>, exce
pt
506:           that multiple instances are joined by OR:
507:         </p>
508:         <code>
509:           ${this-&gt;db-&gt;like('title', 'match');<br />
510:           ${this-&gt;db-&gt;or_not_like('body', 'match'); <br /> <br /> //
511:           WHERE title LIKE '%match%' OR body NOT LIKE '%match%'
512:         </code>
513:         <h2>${this->db->group_by()}</h2>
514:         <p>Permits you to write the GROUP BY portion of your query:</p>
515:
516:         <code>
517:           ${this->db->group_by("title"); <br />
518:           <br /> // Produces: GROUP BY title
519:         </code>
520:
521:         <p>You can also pass an array of multiple values as well:</p>
522:
523:         <code>
524:           ${this->db->group_by(array("title", "date")); <br /> <br /> //
525:           Produces: GROUP BY title, date
526:         </code>
527:
528:         <p class="important">Note: group_by() was formerly known as
529:           groupby(), which has been removed.</p>
530:
531:         <h2>
532:           ${this-&gt;db-&gt;distinct()}<br />
533:         </h2>
534:         <p>Adds the &quot;DISTINCT&quot; keyword to a query</p>
535:         <p>
536:           <code>
537:             ${this-&gt;db-&gt;distinct()}<br /> ${this-&gt;db-&gt;get
('table');<br />
538:             <br /> // Produces: SELECT DISTINCT * FROM table
539:           </code>
540:         </p>
541:         <h2>${this->db->having()}</h2>
542:         <p>Permits you to write the HAVING portion of your query. There
543:           are 2 possible syntaxes, 1 argument or 2:</p>
544:
```

active_record.html

```
545:         <code>
546:             $this->db->having('user_id = 45'); <br /> // Produces: HAVING
547:             user_id = 45<br /> <br /> $this-&gt;db-&gt;having('user_id', 45
); <br />
548:             // Produces: HAVING user_id = 45<br /> <br />
549:         </code>
550:
551: <p>You can also pass an array of multiple values as well:</p>
552:
553:
554: <p>
555:         <code>
556:             $this->db->having(array('title = ' => 'My Title', 'id <
=> $id)); <br />
557:             <br /> // Produces: HAVING title = 'My Title', id < 45
558:         </code>
559:     </p>
560: <p>If you are using a database that CodeIgniter escapes queries
561:     for, you can prevent escaping content by passing an optional th
ird
562:     argument, and setting it to FALSE.</p>
563:
564: <p>
565:         <code>
566:             $this-&gt;db-&gt;having('user_id', 45); <br /> // Produ
ces: HAVING
567:             'user_id' = 45 in some databases such as MySQL <br />
568:             $this-&gt;db-&gt;having('user_id', 45, FALSE); <br /> /
/ Produces:
569:             HAVING user_id = 45
570:         </code>
571:     </p>
572: <h2>$this-&gt;db-&gt;or_having();</h2>
573: <p>Identical to having(), only separates multiple clauses with
574:     &quot;OR&quot;.</p>
575: <h2>$this->db->order_by();</h2>
576: <p>
577:     Lets you set an ORDER BY clause. The first parameter contains t
he
578:     name of the column you would like to order by. The second param
eter
579:     lets you set the direction of the result. Options are
580:     <code>asc</code>
581:     or
582:     <code>desc</code>
583:     , or
584:     <code>random</code>
585:     .
586: </p>
587:
588: <code>
589:     $this->db->order_by("title", "desc"); <br /> <br /> // Produces:
590:     ORDER BY title DESC
591: </code>
592: <p>You can also pass your own string in the first parameter:</p>
593:
594: <code>
595:     $this->db->order_by('title desc, name asc'); <br /> <br /> //
596:     Produces: ORDER BY title DESC, name ASC
597: </code>
598:
599: <p>Or multiple function calls can be made if you need multiple
600:     fields.</p>
```

active_record.html

```
601:
602:         <p>
603:             <code>
604:                 $this->db->order_by("title", "desc");<br />
605:                 $this->db->order_by("name", "asc"); <br /> <br /> // Pr
roduces:
606:                 ORDER BY title DESC, name ASC
607:             </code>
608:         </p>
609:         <p class="important">Note: order_by() was formerly known as
610:             orderby(), which has been removed.</p>
611:         <p class="important">Note: random ordering is not currently
612:             supported in Oracle or MSSQL drivers. These will default to 'AS
C' .</p>
613:         <h2>$this->db->limit();</h2>
614:         <p>Lets you limit the number of rows you would like returned by
615:             the query:</p>
616:
617:         <code>
618:             $this->db->limit(10);<br /> <br /> // Produces: LIMIT 10
619:         </code>
620:
621:
622:         <p>The second parameter lets you set a result offset.</p>
623:
624:         <code>
625:             $this->db->limit(10, 20);<br /> <br /> // Produces: LIMIT 20, 1
0
626:             (in MySQL. Other databases have slightly different syntax)
627:         </code>
628:
629:
630:         <h2>$this->db->count_all_results();</h2>
631:
632:         <p>Permits you to determine the number of rows in a particular
633:             Active Record query. Queries will accept Active Record restrict
ors
634:             such as where(), or_where(), like(), or_like(), etc. Example:</
p>
635:         <code>
636:             echo $this->db->count_all_results('
637:                 <var>my_table</var>
638:             ');<br /> // Produces an integer, like 25<br /> <br />
639:             $this->db->like('title', 'match');<br />
640:             $this->db->from('
641:                 <var>my_table</var>
642:             ');<br /> echo $this->db->count_all_results();<br /> //
643:             Produces an integer, like 17
644:         </code>
645:
646:         <h2>$this->db->count_all();</h2>
647:
648:         <p>Permits you to determine the number of rows in a particular
649:             table. Submit the table name in the first parameter. Example:</
p>
650:
651:         <code>
652:             echo $this->db->count_all('
653:                 <var>my_table</var>
654:             ');<br /> <br /> // Produces an integer, like 25
655:         </code>
656:
657:
```


active_record.html

```
915:
916:     <code>
917:         $this->db->delete('mytable', array('id' => $id)); <br />
918:     <br /> // Produces:<br /> // DELETE FROM mytable <br /> // WHERE
E
919:         id = $id
920:     </code>
921:
922:     <p>
923:         The first parameter is the table name, the second is the where
924:         clause. You can also use the
925:         <dfn>where()</dfn>
926:         or
927:         <dfn>or_where()</dfn>
928:         functions instead of passing the data to the second parameter o
f the
929:         function:
930:     </p>
931:
932:     <p>
933:         <code>
934:             $this->db->where('id', $id);<br /> $this->db->delete('m
ytable'); <br />
935:             <br /> // Produces:<br /> // DELETE FROM mytable <br />
// WHERE
936:             id = $id
937:         </code>
938:     </p>
939:     <p>An array of table names can be passed into delete() if you
940:         would like to delete data from more than 1 table.</p>
941:     <p>
942:         <code>
943:             $tables = array('table1', 'table2', 'table3');<br />
944:             $this-&gt;db-&gt;where('id', '5');<br />
945:             $this-&gt;db-&gt;delete($tables);
946:         </code>
947:     </p>
948:     <p>
949:         If you want to delete all data from a table, you can use the
950:         <dfn>truncate()</dfn>
951:         function, or
952:         <dfn>empty_table()</dfn>
953:         .
954:     </p>
955:     <h2>$this-&gt;db-&gt;empty_table();</h2>
956:     <p>
957:         Generates a delete SQL string and runs the query.
958:         <code>
959:             $this-&gt;db-&gt;empty_table('mytable'); <br /> <br />
// Produces<br />
960:             // DELETE FROM mytable
961:         </code>
962:     </p>
963:     <h2>$this-&gt;db-&gt;truncate();</h2>
964:     <p>Generates a truncate SQL string and runs the query.</p>
965:     <code>
966:         $this-&gt;db-&gt;from('mytable'); <br />
967:         $this-&gt;db-&gt;truncate(); <br /> // or <br />
968:         $this-&gt;db-&gt;truncate('mytable'); <br /> <br /> // Produce:
<br />
969:         // TRUNCATE mytable <br />
970:     </code>
971:     <p class="important">
```

active_record.html

```
972:         <strong>Note:</strong> If the TRUNCATE command isn't available,
973:         truncate() will execute as &quot;DELETE FROM table&quot;;.
974:     </p>
975:
976:     <h1>
977:         <a name="chaining">&nbsp;</a>Method Chaining
978:     </h1>
979:
980:     <p>Method chaining allows you to simplify your syntax by
981:         connecting multiple functions. Consider this example:</p>
982:
983:     <code>
984:         <dfn>$this->db</dfn>
985:         <kbd>-></kbd>
986:         <var>select</var>
987:         ('title')
988:         <kbd>-></kbd>
989:         <var>from</var>
990:         ('mytable')
991:         <kbd>-></kbd>
992:         <var>where</var>
993:         ('id', $id)
994:         <kbd>-></kbd>
995:         <var>limit</var>
996:         (10, 20);<br /> <br /> $query = $this->db->get();
997:     </code>
998:
999:     <p>&nbsp;</p>
1000:
1001:     <h1>
1002:         <a name="caching">&nbsp;</a>Active Record Caching
1003:     </h1>
1004:
1005:     <p>While not &quot;true&quot;, caching, Active Record enables you
1006:         to save (or &quot;cache&quot;;) certain parts of your queries fo
r
1007:         reuse at a later point in your script's execution. Normally, wh
en an
1008:         Active Record call is completed, all stored information is rese
t for
1009:         the next call. With caching, you can prevent this reset, and re
use
1010:         information easily.</p>
1011:
1012:     <p>Cached calls are cumulative. If you make 2 cached select()
1013:         calls, and then 2 uncached select() calls, this will result in
4
1014:         select() calls. There are three Caching functions available:</p>
>
1015:
1016:     <h2>$this-&gt;db-&gt;start_cache()</h2>
1017:
1018:     <p>This function must be called to begin caching. All Active
1019:         Record queries of the correct type (see below for supported que
ries)
1020:         are stored for later use.</p>
1021:
1022:     <h2>$this-&gt;db-&gt;stop_cache()</h2>
1023:
1024:     <p>This function can be called to stop caching.</p>
1025:
1026:     <h2>$this-&gt;db-&gt;flush_cache()</h2>
1027:
```


queries.html

```

1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml
tml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Queries : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype-lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28:     <!-- START NAVIGATION -->
29:     <div id="nav">
30:         <div id="nav_inner">
31:             <script type="text/javascript">create_menu('../');</script>
32:         </div>
33:     </div>
34:     <div id="nav2">
35:         <a name="top"></a><a href="javascript:void(0);"
36:         onclick="myHeight.toggle();"></a>
40:     </div>
41:     <div id="masthead">
42:         <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43:             <tr>
44:                 <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45:                 <td id="breadcrumb_right"><a href="../toc.html">Table of
46:
47:                 Contents Page</a></td>
48:             </tr>
49:         </table>
50:     </div>
51:     <!-- END NAVIGATION -->
52:
53:     <!-- START BREADCRUMB -->
54:     <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
55:         <tr>
56:             <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
57:             er
58:             <a href="#">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="../index.
html">User Guide
59:             <a href="#">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="index.htm
l">Database Library</a>

```


queries.html

```
59:          &nbsp;&#8250;&nbsp;&nbsp;&nbsp; Queries</td>
60:          <td id="searchbox"><form method="get"
61:              action="http://www.google.com/search"
62:              <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
63:                  value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;&nbsp;&nbsp; <input
64:              type="text" class="input" style="width:
200px;" name="q" id="q"
65:              size="31" maxlength="255" value="" />&nbsp;&nbsp;&nbsp;
bsp;<input type="submit"
66:              class="submit" name="sa" value="Go" />
67:          </form></td>
68:      </tr>
69: </table>
70: <!-- END BREADCRUMB -->
71:
72:
73:
74: <br clear="all" />
75:
76:
77: <!-- START CONTENT -->
78: <div id="content">
79:
80:
81:     <h1>Queries</h1>
82:
83:     <h2>${this->db->query()}</h2>
84:
85:     <p>To submit a query, use the following function:</p>
86:
87:     <code>${this->db->query('YOUR QUERY HERE');</code>
88:
89:     <p>
90:         The
91:         <dfn>query()</dfn>
92:         function returns a database result <strong>object</strong> when
93:         "read" type queries are run, which you can use to <a
94:             href="results.html">show your results</a>. When "write"
type queries
95:         are run it simply returns TRUE or FALSE depending on success or
96:         failure. When retrieving data you will typically assign the que
ry to
97:         your own variable, like this:
98:
99:         <p>
100:            <code>
101:                <var>$query</var>
102:                = ${this->db->query('YOUR QUERY HERE');
103:            </code>
104:
105:            <h2>${this->db->simple_query()}</h2>
106:
107:            <p>
108:                This is a simplified version of the
109:                <dfn>${this->db->query()}</dfn>
110:                function. It ONLY returns TRUE/FALSE on success or failure. It
DOES
111:                NOT return a database result set, nor does it set the query tim
er, or
112:                compile bind data, or store your query for debugging. It simply
lets
```

queries.html

```
113:         you submit a query. Most users will rarely use this function.
114:     </p>
115:
116:
117:     <h1>Working with Database prefixes manually</h1>
118:     <p>If you have configured a database prefix and would like to
119:         prepend it to a table name for use in a native SQL query for ex
ample,
120:         then you can use the following:</p>
121:     <p>
122:         <code>
123:             $this-&gt;db-&gt;dbprefix('tablename');<br /> // output
S
124:             prefix_tablename
125:         </code>
126:     </p>
127:
128:     <p>If for any reason you would like to change the prefix
129:         programatically without needing to create a new connection, you
can
130:         use this method:</p>
131:     <p>
132:         <code>
133:             $this-&gt;db-&gt;set_dbprefix('newprefix');<br />
134:             <br /> $this-&gt;db-&gt;dbprefix('tablename');<br /> //
outputs
135:             newprefix_tablename
136:         </code>
137:     </p>
138:
139:
140:     <h1>Protecting identifiers</h1>
141:     <p>
142:         In many databases it is advisable to protect table and field na
mes -
143:         for example with backticks in MySQL. <strong>Active Record
144:         queries are automatically protected</strong>, however i
f you need to manually
145:         protect an identifier you can use:
146:     </p>
147:     <p>
148:         <code>$this-&gt;db-&gt;protect_identifiers('table_name');</code>
>
149:     </p>
150:
151:     <p class="important">Although Active Record will try its best to
152:         properly quote any field and table names that you feed it, note
that
153:         it is NOT designed to work with arbitrary user input. DO NOT fe
ed it
154:         with unsanitized user data.</p>
155:
156:     <p>
157:         This function will also add a table prefix to your table, assum
ing
158:         you have a prefix specified in your database config file. To en
able
159:         the prefixing set
160:         <kbd>TRUE</kbd>
161:         (boolean) via the second parameter:
162:     </p>
163:     <p>
164:         <code>
```

queries.html

```
165:         $this->db->protect_identifiers('table_name',
166:         <kbd>TRUE</kbd>
167:         );
168:     </code>
169: </p>
170:
171:
172: <h1>Escaping Queries</h1>
173: <p>It's a very good security practice to escape your data before
174:     submitting it into your database. CodeIgniter has three methods
that
175:     help you do this:</p>
176:
177: <ol>
178:     <li><strong>$this->db->escape()</strong> This function
179:         determines the data type so that it can escape only str
ing data. It
180:         also automatically adds single quotes around the data s
o you don't
181:         have to: <code>$sql = "INSERT INTO table (title)
182:             VALUES(".$this->db->escape($title).")";</code><
/li>
183:
184:     <li><strong>$this->db->escape_str()</strong> This function
185:         escapes the data passed to it, regardless of type. Most
of the time
186:         you'll use the above function rather than this one. Use
the function
187:         like this: <code>$sql = "INSERT INTO table (title)
188:             VALUES('".$this->db->escape_str($title).")";</
code></li>
189:
190:     <li><strong>$this->db->escape_like_str()</strong> This method
191:         should be used when strings are to be used in LIKE cond
itions so
192:         that LIKE wildcards ('%', '_') in the string are also p
roperly
193:         escaped. <code>
194:             $search = '20% raise';<br /> $sql = "SELECT id
FROM table WHERE
195:             column LIKE '%" . $this->db->escape_like_str($sea
rch). "%'";
196:         </code></li>
197:
198: </ol>
199:
200:
201: <h1>Query Bindings</h1>
202:
203:
204: <p>Bindings enable you to simplify your query syntax by letting
205:     the system put the queries together for you. Consider the follo
wing
206:     example:</p>
207:
208: <code>
209:     $sql = "SELECT * FROM some_table WHERE id =
210:     <var>?</var>
211:     AND status =
212:     <var>?</var>
213:     AND author =
214:     <var>?</var>
215:     "; <br />
```


hooks.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Hooks : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype_lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28: <!-- START NAVIGATION -->
29: <div id="nav">
30: <div id="nav_inner">
31: <script type="text/javascript">create_menu('../');</script>
32: </div>
33: </div>
34: <div id="nav2">
35: <a name="top"></a><a href="javascript:void(0);"
36: onclick="myHeight.toggle();"></a>
40: </div>
41: <div id="masthead">
42: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43: <tr>
44: <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45: <td id="breadcrumb_right"><a href="../toc.html">Table of
46: Contents Page</a></td>
47: </tr>
48: </table>
49: </div>
50: <!-- END NAVIGATION -->
51:
52: <!-- START BREADCRUMB -->
53: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54: <tr>
55: <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
56: er
57: <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="../index.
58: html">User Guide
59: <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;Hooks - Extending
60: the Framework Core</td>
```

hooks.html

```
59:                 <td id="searchbox"><form method="get"
60:                   action="http://www.google.com/search">
61:                     <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                       value="codeigniter.com/user_guide/" />S
earch User Guide    <input
63:                       type="text" class="input" style="width:
200px;" name="q" id="q"
64:                       size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                       class="submit" name="sa" value="Go" />
66:                 </form></td>
67:             </tr>
68:         </table>
69:         <!-- END BREADCRUMB -->
70:
71:         <br clear="all" />
72:
73:
74:         <!-- START CONTENT -->
75:         <div id="content">
76:
77:             <h1>Hooks - Extending the Framework Core</h1>
78:
79:             <p>
80:                 CodeIgniter's Hooks feature provides a means to tap into and mo
dify
81:                 the inner workings of the framework without hacking the core fi
les.
82:                 When CodeIgniter runs it follows a specific execution process,
83:                 diagramed in the <a href="../overview/appflow.html">Application
84:                 Flow</a> page. There may be instances, however, where y
ou'd like to
85:                 cause some action to take place at a particular stage in the
86:                 execution process. For example, you might want to run a script
right
87:                 before your controllers get loaded, or right after, or you migh
t want
88:                 to trigger one of your own scripts in some other location.
89:             </p>
90:
91:             <h2>Enabling Hooks</h2>
92:
93:             <p>
94:                 The hooks feature can be globally enabled/disabled by setting t
he
95:                 following item in the
96:                 <kbd>application/config/config.php</kbd>
97:                 file:
98:             </p>
99:
100:            <code>$config['enable_hooks'] = TRUE;</code>
101:
102:
103:            <h2>Defining a Hook</h2>
104:
105:            <p>
106:                 Hooks are defined in
107:                 <dfn>application/config/hooks.php</dfn>
108:                 file. Each hook is specified as an array with this prototype:
109:            </p>
110:
111:            <code>
```


hooks.html

```
191:
192:         <p>Notice the brackets after each array index:</p>
193:
194:         <code>
195:             $hook['pre_controller']
196:             <kbd>[]</kbd>
197:         </code>
198:
199:         <p>This permits you to have the same hook point with multiple
200:             scripts. The order you define your array will be the execution
order.</p>
201:
202:
203:         <h2>Hook Points</h2>
204:
205:         <p>The following is a list of available hook points.</p>
206:
207:         <ul>
208:             <li><strong>pre_system</strong><br /> Called very early during
209:                 system execution. Only the benchmark and hooks class ha
ve been
210:                 loaded at this point. No routing or other processes hav
e happened.</li>
211:             <li><strong>pre_controller</strong><br /> Called immediately
212:                 prior to any of your controllers being called. All base
classes,
213:                 routing, and security checks have been done.</li>
214:             <li><strong>post_controller_constructor</strong><br /> Called
215:                 immediately after your controller is instantiated, but
prior to any
216:                 method calls happening.</li>
217:             <li><strong>post_controller</strong><br /> Called immediately
218:                 after your controller is fully executed.</li>
219:             <li><strong>display_override</strong><br /> Overrides the <dfn>
_display()</dfn>
220:                 function, used to send the finalized page to the web br
rowser at the
221:                 end of system execution. This permits you to use your o
wn display
222:                 methodology. Note that you will need to reference the C
I superobject
223:                 with <dfn>$this->CI =&amp; get_instance()</dfn> and the
n the
224:                 finalized data will be available by calling <dfn>$this-
>CI->output->get_output()</dfn></li>
225:             <li><strong>cache_override</strong><br /> Enables you to call
226:                 your own function instead of the <dfn>_display_cache()<
/dfn>
227:                 function in the output class. This permits you to use y
our own cache
228:                 display mechanism.</li>
229:             <li><strong>post_system</strong><br /> Called after the final
230:                 rendered page is sent to the browser, at the end of sys
tem execution
231:                 after the finalized data is sent to the browser.</li>
232:         </ul>
233:     </div>
234:     <!-- END CONTENT -->
235:
236:
237:     <div id="footer">
238:         <p>
239:             Previous Topic:&nbsp;&nbsp;&nbsp;<a href="core_classes.html">Creating
```


controllers.html

```
60:             action="http://www.google.com/search">
61:             <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:             value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;  <input
63:             type="text" class="input" style="width:
200px;" name="q" id="q"
64:             size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:             class="submit" name="sa" value="Go" />
66:             </form></td>
67:         </tr>
68:     </table>
69:     <!-- END BREADCRUMB -->
70:
71:     <br clear="all" />
72:
73:
74:     <!-- START CONTENT -->
75:     <div id="content">
76:
77:         <h1>Controllers</h1>
78:
79:         <p>Controllers are the heart of your application, as they
80:           determine how HTTP requests should be handled.</p>
81:
82:
83:         <ul>
84:             <li><a href="#what">What is a Controller?</a></li>
85:             <li><a href="#hello">Hello World</a></li>
86:             <li><a href="#functions">Functions</a></li>
87:             <li><a href="#passinguri">Passing URI Segments to Your
88:               Functions</a></li>
89:             <li><a href="#default">Defining a Default Controller</a></li>
90:             <li><a href="#remapping">Remapping Function Calls</a></li>
91:             <li><a href="#output">Controlling Output Data</a></li>
92:             <li><a href="#private">Private Functions</a></li>
93:             <li><a href="#subfolders">Organizing Controllers into
94:               Sub-folders</a></li>
95:             <li><a href="#constructors">Class Constructors</a></li>
96:             <li><a href="#reserved">Reserved Function Names</a></li>
97:         </ul>
98:
99:
100:        <a name="what"></a>
101:        <h2>What is a Controller?</h2>
102:
103:        <p>
104:            <dfn>A Controller is simply a class file that is named in a way
105:            that can be associated with a URI.</dfn>
106:        </p>
107:
108:        <p>Consider this URI:</p>
109:
110:        <code>
111:            example.com/index.php/
112:            <var>blog</var>
113:            /
114:        </code>
115:
116:        <p>
117:            In the above example, CodeIgniter would attempt to find a contr
oller
```

controllers.html

```
118:         named
119:         <dfn>blog.php</dfn>
120:         and load it.
121:     </p>
122:
123: <p>
124:     <strong>When a controller's name matches the first segment
125:         of a URI, it will be loaded.</strong>
126: </p>
127:
128: <a name="hello"></a>
129: <h2>Let's try it:&nbsp;  Hello World!</h2>
130:
131: <p>
132:     Let's create a simple controller so you can see it in action. U
sing
133:     your text editor, create a file called
134:     <dfn>blog.php</dfn>
135:     , and put the following code in it:
136: </p>
137:
138:
139:     <textarea class="textarea" style="width: 100%" cols="50" rows="10">
140: &lt;?php
141: class Blog extends CI_Controller {
142:
143:     public function index()
144:     {
145:         echo 'Hello World!';
146:     }
147: }
148: ?&gt;
149: </textarea>
150:
151:
152:
153: <p>
154:     Then save the file to your
155:     <dfn>application/controllers/</dfn>
156:     folder.
157: </p>
158:
159: <p>Now visit the your site using a URL similar to this:</p>
160:
161: <code>
162:     example.com/index.php/
163:     <var>blog</var>
164:     /
165: </code>
166:
167: <p>
168:     If you did it right, you should see
169:     <samp>Hello World!</samp>
170:     .
171: </p>
172:
173: <p>Note: Class names must start with an uppercase letter. In other
174:     words, this is valid:</p>
175:
176: <code>
177: &lt;?php<br /> class
178: <var>Blog</var>
179: extends CI_Controller {<br /> <br /> }<br /> ?&gt;
```

controllers.html

```
180:         </code>
181:
182:         <p>
183:             This is <strong>not</strong> valid:
184:         </p>
185:
186:         <code>
187:             &lt;?php<br /> class
188:             <var>blog</var>
189:             extends CI_Controller {<br /> <br /> }<br /> ?&gt;
190:         </code>
191:
192:         <p>
193:             Also, always make sure your controller
194:             <dfn>extends</dfn>
195:             the parent controller class so that it can inherit all its func
tions.
196:         </p>
197:
198:
199:
200:         <a name="functions"></a>
201:         <h2>Functions</h2>
202:
203:         <p>
204:             In the above example the function name is
205:             <dfn>index()</dfn>
206:             . The "index" function is always loaded by default if the <stro
ng>second
207:             segment</strong> of the URI is empty. Another way to sh
ow your "Hello World"
208:             message would be this:
209:         </p>
210:
211:         <code>
212:             example.com/index.php/
213:             <var>blog</var>
214:             /
215:             <samp>index</samp>
216:             /
217:         </code>
218:
219:         <p>
220:             <strong>The second segment of the URI determines which
221:             function in the controller gets called.</strong>
222:         </p>
223:
224:         <p>Let's try it. Add a new function to your controller:</p>
225:
226:
227:         <textarea class="textarea" style="width: 100%" cols="50" rows="15">
228:         &lt;?php
229:         class Blog extends CI_Controller {
230:
231:             public function index()
232:             {
233:                 echo 'Hello World!';
234:             }
235:
236:             public function comments()
237:             {
238:                 echo 'Look at this!';
239:             }

```


controllers.html

he

```
353: URI) will be passed as a parameter to the
354: _remap()
355: function:
356: 
```

```
357: 
```

```
358: 
```

```
359: public function _remap(
360:     $method
361: )  
362: {  
363:     if ($method ==  
364:         'some_method')  
365:     {  
366:         $this->$method(  
367:             ...  
368:         );  
369:     }  
370: }  
371: 
```

```
372: 
```

```
373: Any extra segments after the method name are passed into  
374: _remap()  
375: as an optional second parameter. This array can be used in  
376: combination with PHP's call\_user\_func\_array  
377: to emulate CodeIgniter's default behavior.
```

```
378: 
```

```
379: 
```

```
380: public function _remap($method, $params = array())  
381: {  
382:     $method = 'process_'. $method;  
383:     if (method_exists($this, $method))  
384:     {  
385:         return call_user_func_array(array($this, $method), $params);  
386:     }  
387:     show_404();  
388: }  
389: 
```

```
390: 
```

```
391: 
```

```
392: 
```

```
393: 
```

```
394: CodeIgniter has an output class that takes care of sending your  
395: rendered data to the web browser automatically. More informatio  
396: n on this can be found in the Views and Output class pages.  
397: In some  
398: cases, however, you might want to post-process the finalized da  
399: ta in  
400: some way and send it to the browser yourself. CodeIgniter permi  
401: ts you  
402: to add a function named  
403: _output()  
404: to your controller that will receive the finalized output data.
```


environments.html

```
59:          <td id="searchbox"><form method="get"
60:              action="http://www.google.com/search">
61:          <input type="hidden" name="as_sitesearch" id="as_sitesearch"
62:              value="codeigniter.com/user_guide/" />Search User Guide<br><input
63:          type="text" class="input" style="width: 200px;" name="q" id="q"
64:          size="31" maxlength="255" value="" /><input type="submit"
65:          class="submit" name="sa" value="Go" />
66:          </form></td>
67:      </tr>
68:  </table>
69:  <!-- END BREADCRUMB -->
70:
71:  <br clear="all" />
72:
73:  <!-- START CONTENT -->
74:  <div id="content">
75:
76:      <h1>Handling Multiple Environments</h1>
77:
78:      <p>Developers often desire different system behavior depending on
79:          whether an application is running in a development or production
80:          environment. For example, verbose error output is something that
81:          would be useful while developing an application, but it may also
82:          pose a security issue when "live".</p>
83:
84:      <h2>The ENVIRONMENT Constant</h2>
85:
86:      <p>By default, CodeIgniter comes with the environment constant set
87:          to '
88:          <code><code><code>development</code>'. At the top of index.php, you will see:
89:
90:
91:      </p>
92:
93:      <code>
94:          define(
95:              <code><code><code>ENVIRONMENT</code>
96:              ,
97:              <code><code><code>development</code>
98:              ');
99:      </code>
100:
101:      <p>In addition to affecting some basic framework behavior (see the
102:          next section), you may use this constant in your own development
103:          to differentiate between which environment you are running in.</p>
104:
105:      <h2>Effects On Default Framework Behavior</h2>
106:
107:      <p>There are some places in the CodeIgniter system where the
108:          <code><code><code>ENVIRONMENT</code> constant is used. This section describes how default framework
109:          behavior is affected.
110:
111:
112:      </p>
```


autoloader.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml
tml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Auto-loading Resources : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12:      href='../userguide.css' />
13:
14: <script type="text/javascript" src=" ../nav/nav.js"></script>
15: <script type="text/javascript" src=" ../nav/prototype.lite.js"></script>
16: <script type="text/javascript" src=" ../nav/moo.fx.js"></script>
17: <script type="text/javascript" src=" ../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28:     <!-- START NAVIGATION -->
29:     <div id="nav">
30:         <div id="nav_inner">
31:             <script type="text/javascript">create_menu('../');</script>
32:         </div>
33:     </div>
34:     <div id="nav2">
35:         <a name="top"></a><a href="javascript:void(0);"
36:             onclick="myHeight.toggle();"></a>
40:     </div>
41:     <div id="masthead">
42:         <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43:             <tr>
44:                 <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45:                 <td id="breadcrumb_right"><a href=" ../toc.html">Table o
f
46:                                     Contents Page</a></td>
47:             </tr>
48:         </table>
49:     </div>
50:     <!-- END NAVIGATION -->
51:
52:
53:     <!-- START BREADCRUMB -->
54:     <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
55:         <tr>
56:             <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
er
57:                                     Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href=" ../index.
html">User Guide
58:                                     Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;Auto-loading Resou
rces</td>
```

autoloader.html

```
59:                 <td id="searchbox"><form method="get"
60:                   action="http://www.google.com/search">
61:                   <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                   value="codeigniter.com/user_guide/" />S
earch User Guide<input
63:                   type="text" class="input" style="width:
200px;" name="q" id="q"
64:                   size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                   class="submit" name="sa" value="Go" />
66:                 </form></td>
67:             </tr>
68:         </table>
69:         <!-- END BREADCRUMB -->
70:
71:         <br clear="all" />
72:
73:
74:         <!-- START CONTENT -->
75:         <div id="content">
76:
77:             <h1>Auto-loading Resources</h1>
78:
79:             <p>CodeIgniter comes with an "Auto-load" feature that permits
80:             every
81:             time the system runs. If you need certain resources globally
82:             throughout your application you should consider auto-loading th
em for
83:             convenience.</p>
84:
85:             <p>The following items can be loaded automatically:</p>
86:
87:             <ul>
88:                 <li>Core classes found in the "libraries" folder</li>
89:                 <li>Helper files found in the "helpers" folder</li>
90:                 <li>Custom config files found in the "config" folder</li>
91:                 <li>Language files found in the "system/language" folder</li>
92:                 <li>Models found in the &quot;models&quot; folder</li>
93:             </ul>
94:
95:             <p>
96:             To autoload resources, open the
97:             <var>application/config/autoload.php</var>
98:             file and add the item you want loaded to the
99:             <samp>autoload</samp>
100:             array. You'll find instructions in that file corresponding to e
ach
101:             type of item.
102:             </p>
103:
104:             <p class="important">
105:                 <strong>Note:</strong> Do not include the file extension (.php)
when
106:                 adding items to the autoload array.
107:             </p>
108:
109:
110:
111:
112:
113:
```


routing.html

```
60:                                     action="http://www.google.com/search">
61:                                     <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                                     value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp; <input
63:                                     type="text" class="input" style="width:
200px;" name="q" id="q"
64:                                     size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                                     class="submit" name="sa" value="Go" />
66:                                     </form></td>
67:                                     </tr>
68:                                     </table>
69:                                     <!-- END BREADCRUMB -->
70:
71:                                     <br clear="all" />
72:
73:
74:                                     <!-- START CONTENT -->
75:                                     <div id="content">
76:
77:                                     <h1>URI Routing</h1>
78:
79:                                     <p>Typically there is a one-to-one relationship between a URL
80:                                     string and its corresponding controller class/method. The segme
nts in
81:                                     a URI normally follow this pattern:</p>
82:
83:                                     <code>
84:                                     example.com/
85:                                     <dfn>class</dfn>
86:                                     /
87:                                     <samp>function</samp>
88:                                     /
89:                                     <var>id</var>
90:                                     /
91:                                     </code>
92:
93:                                     <p>In some instances, however, you may want to remap this
94:                                     relationship so that a different class/function can be called i
nstead
95:                                     of the one corresponding to the URL.</p>
96:
97:                                     <p>For example, lets say you want your URLs to have this
98:                                     prototype:</p>
99:
100:                                     <p>
101:                                     example.com/product/1/<br /> example.com/product/2/<br />
102:                                     example.com/product/3/<br /> example.com/product/4/
103:                                     </p>
104:
105:                                     <p>Normally the second segment of the URL is reserved for the
106:                                     function name, but in the example above it instead has a produc
t ID.
107:                                     To overcome this, CodeIgniter allows you to remap the URI handl
er.</p>
108:
109:
110:                                     <h2>Setting your own routing rules</h2>
111:
112:                                     <p>
113:                                     Routing rules are defined in your
114:                                     <var>application/config/routes.php</var>
```

routing.html

```
115:         file. In it you'll see an array called
116:         <dfn>$route</dfn>
117:         that permits you to specify your own routing criteria. Routes c
an
118:         either be specified using
119:         <dfn>wildcards</dfn>
120:         or
121:         <dfn>Regular Expressions</dfn>
122:     </p>
123:
124:
125:     <h2>Wildcards</h2>
126:
127:     <p>A typical wildcard route might look something like this:</p>
128:
129:     <code>$route['product/:num'] = "catalog/product_lookup";</code>
130:
131:     <p>In a route, the array key contains the URI to be matched, while
132:         the array value contains the destination it should be re-routed
to.
133:         In the above example, if the literal word "product" is found in
the
134:         first segment of the URL, and a number is found in the second
135:         segment, the "catalog" class and the "product_lookup" method ar
e
136:         instead used.</p>
137:
138:     <p>You can match literal values or you can use two wildcard types:</p>
139:
140:     <p>
141:         <strong>(:num)</strong> will match a segment containing only nu
mbers.<br />
142:         <strong>(:any)</strong> will match a segment containing any
143:         character.
144:     </p>
145:
146:     <p class="important">
147:         <strong>Note:</strong> Routes will run in the order they are de
fined.
148:         Higher routes will always take precedence over lower ones.
149:     </p>
150:
151:     <h2>Examples</h2>
152:
153:     <p>Here are a few routing examples:</p>
154:
155:     <code>$route['journals'] = "blogs";</code>
156:     <p>A URL containing the word "journals" in the first segment will
157:         be remapped to the "blogs" class.</p>
158:
159:     <code>$route['blog/joe'] = "blogs/users/34";</code>
160:     <p>A URL containing the segments blog/joe will be remapped to the
161:         "blogs" class and the "users" method. The ID will be set to "34
".</p>
162:
163:     <code>$route['product/(:any)'] = "catalog/product_lookup";</code>
164:     <p>A URL with "product" as the first segment, and anything in the
165:         second will be remapped to the "catalog" class and the
166:         "product_lookup" method.</p>
167:
168:     <code>$route['product/(:num)'] =
169:         "catalog/product_lookup_by_id/$1";</code>
170:     <p>A URL with "product" as the first segment, and a number in the
```

routing.html

```
171:         second will be remapped to the "catalog" class and the
172:         "product_lookup_by_id" method passing in the match as a variabl
e to
173:         the function.</p>
174:
175:         <p class="important">
176:             <strong>Important:</strong> Do not use leading/trailing slashes
.
177:         </p>
178:
179:         <h2>Regular Expressions</h2>
180:
181:         <p>If you prefer you can use regular expressions to define your
182:         routing rules. Any valid regular expression is allowed, as are
183:         back-references.</p>
184:
185:         <p class="important">
186:             <strong>Note:</strong>&nbsp; If you use back-references you mus
t use
187:             the dollar syntax rather than the double backslash syntax.
188:         </p>
189:
190:         <p>A typical RegEx route might look something like this:</p>
191:
192:         <code>$route['products/([a-z]+)/(\d+)'] = "$1/id_$2";</code>
193:
194:         <p>
195:             In the above example, a URI similar to
196:             <dfn>products/shirts/123</dfn>
197:             would instead call the
198:             <dfn>shirts</dfn>
199:             controller class and the
200:             <dfn>id_123</dfn>
201:             function.
202:         </p>
203:
204:         <p>You can also mix and match wildcards with regular expressions.</p>
205:
206:         <h2>Reserved Routes</h2>
207:
208:         <p>There are two reserved routes:</p>
209:
210:         <code>$route['default_controller'] = 'welcome';</code>
211:
212:         <p>This route indicates which controller class should be loaded if
213:         the URI contains no data, which will be the case when people lo
ad
214:         your root URL. In the above example, the "welcome" class would
be
215:         loaded. You are encouraged to always have a default route other
wise a
216:         404 page will appear by default.</p>
217:
218:         <code>$route['404_override'] = '';</code>
219:
220:         <p>
221:             This route indicates which controller class should be loaded if
the
222:             requested controller is not found. It will override the default
404
223:             error page. It won't affect to the
224:             <samp>show_404()</samp>
225:             function, which will continue loading the default
```


styleguide.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Style Guide : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <style type="text/css" media="screen">
15: code {
16:     white-space: pre;
17: }
18: </style>
19:
20: <script type="text/javascript" src="../../nav/nav.js"></script>
21: <script type="text/javascript" src="../../nav/prototype.lite.js"></script>
22: <script type="text/javascript" src="../../nav/moo.fx.js"></script>
23: <script type="text/javascript" src="../../nav/user_guide_menu.js"></script>
24:
25: <meta http-equiv='expires' content='-1' />
26: <meta http-equiv='pragma' content='no-cache' />
27: <meta name='robots' content='all' />
28: <meta name='author' content='ExpressionEngine Dev Team' />
29: <meta name='description' content='CodeIgniter User Guide' />
30:
31: </head>
32: <body>
33:
34:     <!-- START NAVIGATION -->
35:     <div id="nav">
36:         <div id="nav_inner">
37:             <script type="text/javascript">create_menu('../');</script>
38:         </div>
39:     </div>
40:     <div id="nav2">
41:         <a name="top"></a><a href="javascript:void(0);"
42:             onclick="myHeight.toggle();"></a>
46:     </div>
47:     <div id="masthead">
48:         <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
49:             <tr>
50:                 <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
51:                 <td id="breadcrumb_right"><a href="../../toc.html">Table o
52:
53:                                     Contents Page</a></td>
54:             </tr>
55:         </table>
56:     </div>
57:     <!-- END NAVIGATION -->
58:
59:     <!-- START BREADCRUMB -->
60:     <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
61:         <tr>
```

f

styleguide.html

```
116:         <li><a href="#private_methods_and_variables">Private Methods
117:             and Variables</a></li>
118:     <li><a href="#php_errors">PHP Errors</a></li>
119:     <li><a href="#short_open_tags">Short Open Tags</a></li>
120:     <li><a href="#one_statement_per_line">One Statement Per Line</a
></li>
121:     <li><a href="#strings">Strings</a></li>
122:     <li><a href="#sql_queries">SQL Queries</a></li>
123:     <li><a href="#default_function_arguments">Default Function
124:         Arguments</a></li>
125: </ul>
126:
127:     <li>
128:
129:         <h2>
130:             <a name="file_format"></a>File Format
131:         </h2>
132:         <div class="guidelineDetails">
133:             <p>
134:                 Files should be saved with Unicode (UTF-8) encod
ding. The <abbr
135:                 <abbr title="Byte Order Mark">BOM</abbr> shou
ld <em>not</em> be used.
136:                 Unlike UTF-16 and UTF-32, there's no byte order
to indicate in a
137:                 UTF-8 encoded file, and the <abbr title="Byte O
rder Mark">BOM</abbr>
138:                 can have a negative side effect in PHP of sendi
ng output,
139:                 preventing the application from being able to s
et its own headers.
140:                 Unix line endings should be used (LF).
141:             </p>
142:             <p>Here is how to apply these settings in some of the m
ore
143:             common text editors. Instructions for your text
editor may vary;
144:             check your text editor's documentation.</p>
145:             <h5>TextMate</h5>
146:             <ol>
147:                 <li>Open the Application Preferences</li>
148:                 <li>Click Advanced, and then the "Saving" tab</
149:                 <li>In "File Encoding", select "UTF-8 (recommen
ded)"</li>
150:                 <li>In "Line Endings", select "LF (recommended)"
151:                 <li><em>Optional:</em> Check "Use for existing
files as well"
152:                 if you wish to modify the line endings
of files you open to your
153:                 new preference.</li>
154:             </ol>
155:             <h5>BBEdit</h5>
156:             <ol>
157:                 <li>Open the Application Preferences</li>
158:                 <li>Select "Text Encodings" on the left.</li>
159:                 <li>In "Default text encoding for new documents
160:
161:
162:
163:
164:
```

styleguide.html

```
" , select
165:         "Unicode (UTF-8, no BOM)"</li>
166:     <li><em>Optional:</em> In "If file's encoding c
an't be
167:         guessed, use", select "Unicode (UTF-8,
no BOM)"</li>
168:     <li>Select "Text Files" on the left.</li>
169:     <li>In "Default line breaks", select "Mac OS X
and Unix (LF)"</li>
170:         </ol>
171:     </div>
172:
173:     <h2>
174:         <a name="php_closing_tag"></a>PHP Closing Tag
175:     </h2>
176:     <div class="guidelineDetails">
177:         <p>
178:             The PHP closing tag on a PHP document <strong>?
&gt;</strong> is
179:             optional to the PHP parser. However, if used, a
ny whitespace
180:             following the closing tag, whether introduced b
y the developer,
181:             user, or an FTP application, can cause unwanted
output, PHP errors,
182:             or if the latter are suppressed, blank pages. F
or this reason, all
183:             PHP files should <strong>OMIT</strong> the clos
ing PHP tag, and
184:             instead use a comment block to mark the end of
file and it's
185:             location relative to the application root. This
allows you to still
186:             identify a file as being complete and not trunc
ated.
187:         </p>
188:         <code>
189:             <strong>INCORRECT</strong>: &lt;?php echo "Here
's my code!"; ?&gt;
190:             <strong>CORRECT</strong>: &lt;?php echo "Here's
my code!"; /* End
191:             of file myfile.php */ /* Location:
192:             ./system/modules/mymodule/myfile.php */
193:         </code>
194:     </div>
195:
196:
197:
198:     <h2>
199:         <a name="class_and_method_naming"></a>Class and Method
Naming
200:     </h2>
201:     <div class="guidelineDetails">
202:         <p>Class names should always start with an uppercase le
tter.
203:             Multiple words should be separated with an unde
rscore, and not
204:             CamelCased. All other class methods should be e
ntirely lowercased
205:             and named to clearly indicate their function, p
referably including
206:             a verb. Try to avoid overly long and verbose na
mes.</p>
```

styleguide.html

```
207:
208:         <code>
209:         <strong>INCORRECT</strong>: class superclass cl
ass SuperClass <strong>CORRECT</strong>:
210:         class Super_class
211:         </code>
212:
213:
214:         <code>class Super_class { function __construct() { } }<
/code>
215:
216:         <p>Examples of improper and proper method naming:</p>
217:
218:         <code>
219:         <strong>INCORRECT</strong>: function fileproper
ties() // not
220:         descriptive and needs underscore separator func
tion
221:         fileProperties() // not descriptive and uses Ca
melCase function
222:         getFileProperties() // Better! But still missin
g underscore
223:         separator function getFileProperties() // uses
CamelCase function
224:         get_the_file_properties_from_the_file() // word
y <strong>CORRECT</strong>:
225:         function get_file_properties() // descriptive,
underscore
226:         separator, and all lowercase letters
227:         </code>
228:
229:     </div>
230:
231:
232:     <h2>
233:     <a name="variable_names"></a>Variable Names
234:     </h2>
235:     <div class="guidelineDetails">
236:     <p>The guidelines for variable naming is very similar t
o that
237:     used for class methods. Namely, variables shoul
d contain only
238:     lowercase letters, use underscore separators, a
nd be reasonably
239:     named to indicate their purpose and contents. V
ery short, non-word
240:     variables should only be used as iterators in f
or() loops.</p>
241:     <code>
242:     <strong>INCORRECT</strong>: $j = &apos;foo&apos;
; // single letter
243:     variables should only be used in for() loops $$
tr // contains
244:     uppercase letters $bufferedText // uses CamelCa
sing, and could be
245:     shortened without losing semantic meaning $grou
pid // multiple
246:     words, needs underscore separator $name_of_last
_city_used // too
247:     long <strong>CORRECT</strong>: for ($j = 0; $j
&lt; 10; $j++) $str
248:     $buffer $group_id $last_city
249:     </code>
```

styleguide.html

```
250:                 </div>
251:
252:
253:                 <h2>
254:                     <a name="commenting"></a>Commenting
255:                 </h2>
256:                 <div class="guidelineDetails">
257:                     <p>In general, code should be commented prolifically. I
t not
258:                         only helps describe the flow and intent of the
code for less
259:                         experienced programmers, but can prove invaluable
le when returning to
260:                         your own code months down the line. There is no
t a required format
261:                         for comments, but the following are recommended
.</p>
262:
263:                     <p>
264:                         <a
265:                             href="http://manual.phpdoc.org/HTMLSmartyConverter/HandS/phpDocumentor/tutorial_phpDocumentor.howto.pkg.html#basics.docblock">DocBlock</a>
266:                         style comments preceding class and method declarations so they can
267:                         be picked up by IDEs:
268:                     </p>
269:
270:                     <code>/** * Super Class * * @package Package Name *
271:                         @subpackage Subpackage * @category Category * @
author Author Name *
272:                         @link http://example.com */ class Super_class {
</code>
273:
274:                     <code>/** * Encodes string for use in XML * * @access public
275:                         * @param string * @return string */ function xml_encode($str)</code>
276:
277:                     <p>Use single line comments within code, leaving a blank line
278:                         between large comment blocks and code.</p>
279:
280:                     <code>// break up the string by newlines $parts =
281:                         explode("\n", $str); // A longer comment that needs to give greater
282:                         detail on what is // occurring and why can use
multiple single-line
283:                         comments. Try to // keep the width reasonable,
around 70 characters
284:                         is the easiest to // read. Don't hesitate to link to permanent
285:                         external resources // that may provide greater
detail: // //
286:                         http://example.com/information_about_something/
in_particular/
287:
288:                         $parts = $this->foo($parts); </code>
289:                 </div>
290:
291:
292:                 <h2>
293:                     <a name="constants"></a>Constants
```

styleguide.html

```
294:                                     </h2>
295:                                     <div class="guidelineDetails">
296:                                         <p>
297:                                             Constants follow the same guidelines as do vari
ables, except
298:                                             constants should always be fully uppercase. <em
>Always use
299:                                             CodeIgniter constants when appropriate,
300:                                             PATH_CACHE, etc.</em>
301:                                         </p>
302:                                         <code>
303:                                             <strong>INCORRECT</strong>: myConstant // missi
ng underscore
304:                                             separator and not fully uppercase N // no singl
e-letter constants
305:                                             S_C_VER // not descriptive $str = str_replace('
{foo}', 'bar',
306:                                             $str); // should use LD and RD constants <stron
g>CORRECT</strong>:
307:                                             MY_CONSTANT NEWLINE SUPER_CLASS_VERSION $str =
308:                                             str_replace(LD.'foo'.RD, 'bar', $str);
309:                                         </code>
310:                                     </div>
311:
312:
313:                                     <h2>
314:                                         <a name="true_false_and_null"></a>TRUE, FALSE, and NULL
315:                                     </h2>
316:                                     <div class="guidelineDetails">
317:                                         <p>
318:                                             <strong>TRUE</strong>, <strong>FALSE</strong>,
and <strong>NULL</strong>
319:                                             keywords should always be fully uppercase.
320:                                         </p>
321:                                         <code>
322:                                             <strong>INCORRECT</strong>: if ($foo == true) $
bar = false;
323:                                             function foo($bar = null) <strong>CORRECT</stro
ng>: if ($foo ==
324:                                             TRUE) $bar = FALSE; function foo($bar = NULL)
325:                                         </code>
326:                                     </div>
327:
328:
329:                                     <h2>
330:                                         <a name="logical_operators"></a>Logical Operators
331:                                     </h2>
332:                                     <div class="guidelineDetails">
333:                                         <p>
334:                                             Use of <strong>||</strong> is discouraged as it
s clarity on some
335:                                             output devices is low (looking like the number
336:                                             11 for instance). <strong>&amp;&amp;</strong>
337:                                             is preferred over <strong>AND</strong> but eith
er are acceptable,
338:                                             and a space should always precede and follow <s
trong>!</strong>.
339:                                         </p>
340:                                         <code>
341:                                             <strong>INCORRECT</strong>: if ($foo || $bar) i
f ($foo AND $bar) //
```

styleguide.html

```
342: okay but not recommended for common syntax high
lighting applications if (! $foo) if (! is_array($foo)) <
343: if ($foo OR $bar) if ($foo && $bar) // recommen
strong>CORRECT</strong>: ( ! is_array($foo))
344: </code>
345: </div>
346:
347:
348:
349:
350:
351: <h2>
352: <a name="comparing_return_values_and_typecasting"></a>C
omparing Return Values and Typecasting
353: </h2>
354: <div class="guidelineDetails">
355: <p>Some PHP functions return FALSE on failure, but may
356: also have a valid return value of "" or 0, which would ev
357: aluate to FALSE in loose comparisons. Be explicit by comparing the
358: variable type when using these return values in conditionals to en
359: sure the return value is indeed what you expect, and not a valu
360: e that has an equivalent loose-type evaluation.</p>
361: <p>
362: Use the same stringency in returning and checki
363: ng your own variables. Use <strong>===</strong> and <strong>
364: >!==</strong> as necessary.
365: <code>
366: <strong>INCORRECT</strong>: // If 'foo'
367: is at the beginning of the string, strpos will return a 0, // resu
368: lting in this conditional evaluating as TRUE if (strpos($str, 'fo
369: o') == FALSE) <strong>CORRECT</strong>:
370: <code>
371: if (strpos($str, 'foo') === FALSE)
372: </code>
373: <code>
374: <strong>INCORRECT</strong>: function bu
375: ild_string($str = "") { if
376: ($str == "") // uh-oh! What if FALSE or
377: the integer 0 is passed as an argument? { } } <strong>CORRECT</str
378: ong>: function build_string($str = "") { if ($str ===
379: "") { } }
380: </code>
381: <p>
382: See also information regarding <a
383: href="http://us3.php.net/manual
/en/language.types.type-juggling.php#language.types.typecasting">typecasting</a>,
which can be quite useful. Typecasting
384: has a slightly different
```


styleguide.html

```
385:                                     effect which may be desirable. When cas
ting a variable as a
386:                                     string, for instance, NULL and boolean
FALSE variables become
387:                                     empty strings, 0 (and other numbers) be
come strings of digits, and
388:                                     boolean TRUE becomes "1":
389:                                     </p>
390:                                     <code>$str = (string) $str; // cast $str as a s
tring</code>
391:                                     </div>
392:                                     <h2>
393:                                     <a name="debugging_code"></a>Debugging Code
394:                                     </h2>
395:                                     <div class="guidelineDetails">
396:                                     <p>No debugging code can be left in place for submitted
add-ons
397:                                     unless it is commented out, i.e. no var_dump(),
400:                                     and exit() calls that were used while creating
401:                                     the add-on, unless
402:                                     they are commented out.</p>
403:                                     <code>// print_r($foo);</code>
404:                                     </div>
405:                                     <h2>
406:                                     <a name="whitespace_in_files"></a>Whitespace in Files
407:                                     </h2>
408:                                     <div class="guidelineDetails">
409:                                     <p>No whitespace can precede the opening PHP tag or fol
410:                                     low the
411:                                     closing PHP tag. Output is buffered, so whitesp
412:                                     ace in your files
413:                                     can cause output to begin before CodeIgniter ou
414:                                     tputs its content,
415:                                     leading to errors and an inability for CodeIgni
416:                                     ter to send proper
417:                                     headers. In the examples below, select the text
418:                                     with your mouse to
419:                                     reveal the incorrect whitespace.</p>
420:                                     <p>
421:                                     <strong>INCORRECT</strong>:
422:                                     </p>
423:                                     <code> &lt;?php // ...there is whitespace and a linebre
ak
424:                                     above the opening PHP tag // as well as whitesp
425:                                     ace after the
426:                                     closing PHP tag ?&gt; </code>
427:                                     <p>
428:                                     <strong>CORRECT</strong>:
429:                                     </p>
430:                                     <code>&lt;?php // this sample has no whitespace before
or
431:                                     after the opening and closing PHP tags ?&gt;</c
ode>
```

styleguide.html

```
432:                                     </div>
433:
434:
435:                                     <h2>
436:                                     <a name="compatibility"></a>Compatibility
437:                                     </h2>
438:                                     <div class="guidelineDetails">
439:                                     <p>Unless specifically mentioned in your add-on's docum
entation,
440:                                     all code must be compatible with PHP version 5.
441:                                     not use PHP functions that require non-default
442:                                     installed unless your code contains an alternat
443:                                     function is not available, or you implicitly do
444:                                     add-on requires said PHP libraries.</p>
445:                                     </div>
446:
447:
448:
449:                                     <h2>
450:                                     <a name="class_and_file_names_using_common_words"></a>C
lass and File
451:                                     Names using Common Words
452:                                     </h2>
453:                                     <div class="guidelineDetails">
454:                                     <p>When your class or filename is a common word, or mig
ht quite
455:                                     likely be identically named in another PHP scri
456:                                     prefix to help prevent collision. Always realiz
457:                                     e that your end
458:                                     users may be running other add-ons or third par
459:                                     ty PHP scripts.
460:                                     Choose a prefix that is unique to your identity
461:                                     company.</p>
462:                                     <code>
463:                                     <strong>INCORRECT</strong>: class Email pi.emai
464:                                     ext.xml.php class Import mod.import.php <strong>
465:                                     class Pre_email pi.pre_email.php class Pre_xml
466:                                     class Pre_import mod.pre_import.php
467:                                     </code>
468:                                     </div>
469:
470:                                     <h2>
471:                                     <a name="database_table_names"></a>Database Table Names
472:                                     </h2>
473:                                     <div class="guidelineDetails">
474:                                     <p>Any tables that your add-on might use must use the '
exp_'
475:                                     prefix, followed by a prefix uniquely identifyi
476:                                     ng you as the
477:                                     developer or company, and then a short descript
478:                                     ive table name. You
479:                                     do not need to be concerned about the database
```

styleguide.html

prefix being used on

478: the user's installation, as CodeIgniter's datab

ase class will

479: automatically convert 'exp_' to what is actuall

y being used.</p>

480:

481: <code>

482: INCORRECT: email_addresses //

missing both

483: prefixes pre_email_addresses // missing exp_ pr

efix

484: exp_email_addresses // missing unique prefix <s

trong>CORRECT:

485: exp_pre_email_addresses

486: </code>

487: <p class="important">

488: NOTE: Be mindful that MySQL ha

489: s a limit of 64

s a limit of 64

490: characters for table names. This should not be

an issue as table

491: names that would exceed this would likely have

unreasonable names.

492: For instance, the following table name exceeds

this limitation by

493: one character. Silly, no? exp_pre_email

_addresses_of_registered_users_in_seattle_washington

494: </div>

495:

496:

497:

498: <h2>

499: One File per Class

500: </h2>

501: <div class="guidelineDetails">

502: <p>

ses, unless the

503: Use separate files for each class your add-on u

ses, unless the

504: classes are closely related. An exampl

e of CodeIgniter

505: files that contains multiple classes is the Dat

abase class file,

506: which contains both the DB class and the DB_Cac

he class, and the

507: Magpie plugin, which contains both the Magpie a

he class, and the

508: </p>

nd Snoopy classes.

509: </div>

510:

511:

512:

513: <h2>

514: Whitespace

515: </h2>

516: <div class="guidelineDetails">

517: <p>Use tabs for whitespace in your code, not spaces. Th

is may

518: seem like a small thing, but using tabs instead

is may

519: of whitespace

of whitespace

520: allows the developer looking at your code to ha

ve indentation at

521: levels that they prefer and customize in whatev

ve indentation at

522: er application they

er application they

523:

er application they

524:

er application they

525:

er application they

526:

er application they

527:

er application they

528:

er application they

529:

er application they

styleguide.html

```
521: use. And as a side benefit, it results in (slig
htly) more compact
522: files, storing one tab character versus, say, f
our space
523: characters.</p>
524: </div>
525:
526:
527:
528: <h2>
529:     <a name="line_breaks"></a>Line Breaks
530: </h2>
531: <div class="guidelineDetails">
532:     <p>Files must be saved with Unix line breaks. This is m
ore of an
533:         issue for developers who work in Windows, but i
n any case ensure
534:         that your text editor is setup to save files wi
th Unix line breaks.</p>
535: </div>
536:
537:
538:
539: <h2>
540:     <a name="code_indenting"></a>Code Indenting
541: </h2>
542: <div class="guidelineDetails">
543:     <p>Use Allman style indenting. With the exception of Cl
ass
544:         declarations, braces are always placed on a lin
e by themselves, and
545:         indented at the same level as the control state
ment that "owns"
546:         them.</p>
547:
548:     <code>
549:         <strong>INCORRECT</strong>: function foo($bar)
{ // ... } foreach
550:         ($arr as $key => $val) { // ... } if ($foo == $
bar) { // ... } else
551:         { $j = 0; $j &lt;
552:         10; $j++) { // ... } } <strong>CORRECT</strong>
: function foo($bar)
553:         { // ... } foreach ($arr as $key => $val) { //
... } if ($foo ==
554:         $bar) { // ... } else { // ... } for ($i = 0; $
i &lt;
555:         10; $i++) {
556:         for ($j = 0; $j &lt;
557:         10; $j++) { // ... } }
558:     </code>
559: </div>
560:
561: <h2>
562:     <a name="bracket_spacing"></a>Bracket and Parenthetic S
pacing
563: </h2>
564: <div class="guidelineDetails">
565:     <p>In general, parenthesis and brackets should not use
any
566:         additional spaces. The exception is that a spac
e should always
follow PHP control structures that accept argum
```

styleguide.html

ents with
567: parenthesis (declare, do-while, elseif, for, fo
reach, if, switch,
568: while), to help distinguish them from functions
and increase
569: readability.</p>
570:
571: <code>INCORRECT: \$arr[\$foo] = 'foo'; CORRECT: \$arr[\$f
oo] =
572: 'foo'; // no spaces around array keys INCORRECT
: function foo (
573: \$bar) { } CORRECT: function foo(\$bar) // no sp
aces around
574: parenthesis in function declarations { } IN CORR
ECT: foreach(
575: \$query->result() as \$row) CORRECT: foreach (\$q
uery->result() as
576: \$row) // single space following PHP control str
uctures, but not in
577: interior parenthesis </code>
578: </div>
579:
580:
581:
582: <h2>
583: Localized Text
584: </h2>
585: <div class="guidelineDetails">
586: <p>Any text that is output in the control panel should
use
587: language variables in your lang file to allow l
ocalization.</p>
588:
589: <code>INCORRECT: return "Invalid Selection"; CORRECT: r
eturn
590: \$this->lang->line('invalid_selection');</code>
591: </div>
592:
593:
594: <h2>
595: Private Met
596: Variables
597: Variables
598: </h2>
599: <div class="guidelineDetails">
600: <p>Methods and variables that are only accessed interna
lly by
601: your class, such as utility and helper function
s that your public
602: methods use for code abstraction, should be pre
fixed with an
603: underscore.</p>
604:
605: <code>convert_text() // public method _convert_text() /
/
606: private method</code>
607: </div>
608:
609:
610: <h2>
611: PHP Errors
612:

styleguide.html

```
613: </h2>
614: <div class="guidelineDetails">
615:     <p>Code must run error free and not rely on warnings and
d notices
616:         to be hidden to meet this requirement. For instance, never access a
617:         variable that you did not set yourself (such as
$_POST array keys)
618:         without first checking to see that it is set().<
/p>
619:     <p>Make sure that while developing your add-on, error reporting
620:         is enabled for ALL users, and that display_errors is enabled in the
621:         PHP environment. You can check this setting with:
622:
623:     <code>if (ini_get('display_errors') == 1) { exit "Enabled"; }</code>
624:     <p>On some servers where display_errors is disabled, and you do
625:         not have the ability to change this in the php.ini, you can often
626:         enable it with:
627:
628:     <code>ini_set('display_errors', 1);</code>
629:
630:     <p class="important">
631:         <strong>NOTE:</strong> Setting the <a href="http://us.php.net/manual/en/ref.errorfunc.php#ini.display-errors">display_errors</a>
632:         setting with ini_set() at runtime is not identical to having it
633:         enabled in the PHP environment. Namely, it will not have any effect
634:         if the script has fatal errors
635:
636:     </p>
637: </div>
638:
639: <h2>
640:     <a name="short_open_tags"></a>Short Open Tags
641: </h2>
642: <div class="guidelineDetails">
643:     <p>Always use full PHP opening tags, in case a server does not
644:         have short_open_tag enabled.</p>
645:
646:     <code>
647:         <strong>INCORRECT</strong>: <? echo $foo; ?&
648:         <strong>CORRECT</strong>: <?php echo $foo; ?
649:     </code>
650: </div>
651:
652: <h2>
```

styleguide.html

```
660:         <a name="one_statement_per_line"></a>One Statement Per
Line
661:     </h2>
662:     <div class="guidelineDetails">
663:         <p>Never combine statements on one line.</p>
664:
665:         <code>
666:             <strong>INCORRECT</strong>: $foo = 'this'; $bar
= 'that'; $bat =
667:             str_replace($foo, $bar, $bag); <strong>CORRECT<
/strong>: $foo =
668:             'this'; $bar = 'that'; $bat = str_replace($foo,
$bar, $bag);
669:         </code>
670:     </div>
671:
672:
673:
674:     <h2>
675:         <a name="strings"></a>Strings
676:     </h2>
677:     <div class="guidelineDetails">
678:         <p>Always use single quoted strings unless you need var
iables
679:         parsed, and in cases where you do need variable
s parsed, use braces
680:         to prevent greedy token parsing. You may also u
se double-quoted
681:         strings if the string contains single quotes, s
o you do not have to
682:         use escape characters.</p>
683:
684:         <code>
685:             <strong>INCORRECT</strong>: "My String" // no v
ariable parsing, so
686:             no use for double quotes "My string $foo" // ne
eds braces 'SELECT
687:             foo FROM bar WHERE baz = \'bag\'' // ugly <stro
ng>CORRECT</strong>:
688:             'My String' "My string {$foo}" "SELECT foo FROM
bar WHERE baz =
689:             'bag'"
690:         </code>
691:     </div>
692:
693:
694:
695:     <h2>
696:         <a name="sql_queries"></a>SQL Queries
697:     </h2>
698:     <div class="guidelineDetails">
699:         <p>MySQL keywords are always capitalized: SELECT, INSER
T,
700:         UPDATE, WHERE, AS, JOIN, ON, IN, etc.</p>
701:
702:         <p>Break up long queries into multiple lines for legibi
lity,
703:         preferably breaking for each clause.</p>
704:
705:         <code>
706:             <strong>INCORRECT</strong>: // keywords are low
ercase and query is
707:             too long for // a single line (... indicates co
```


styleguide.html

755: </html>

cli.html

```
</kbd>
115:         </li>
116:         <li>Make interactive "tasks" that can do things like set
117:             permissions, prune cache folders, run backups, etc.</li>
>
118:         <li>Integrate with other applications in other languages. For
119:             example, a random C++ script could call one command and
run code in
120:             your models!</li>
121:     </ul>
122:
123:     <a name="how"></a>
124:     <h2>Let's try it:&nbsp;   Hello World!</h2>
125:
126:     <p>
127:         Let's create a simple controller so you can see it in action. U
sing
128:         your text editor, create a file called
129:         <dfn>tools.php</dfn>
130:         , and put the following code in it:
131:     </p>
132:
133:     <textarea class="textarea" style="width: 100%" cols="50" rows="10">
134: &lt;?php
135: class Tools extends CI_Controller {
136:
137:     public function message($to = 'World')
138:     {
139:         echo "Hello {$to}!".PHP_EOL;
140:     }
141: }
142: ?&gt;
143: </textarea>
144:
145:     <p>
146:         Then save the file to your
147:         <dfn>application/controllers/</dfn>
148:         folder.
149:     </p>
150:
151:     <p>Now normally you would visit the your site using a URL similar
152:         to this:</p>
153:
154:     <code>
155:         example.com/index.php/
156:         <var>tools</var>
157:         /
158:         <var>message</var>
159:         /
160:         <var>to</var>
161:     </code>
162:
163:     <p>Instead, we are going to open Terminal in Mac/Linux or go to
164:         Run > "cmd" in Windows and navigate to our CodeIgniter project.
</p>
165:
166:     <blockquote>
167:         $ cd /path/to/project;<br /> $ php index.php tools message
168:     </blockquote>
169:
170:     <p>
171:         If you did it right, you should see
172:         <samp>Hello World!</samp>
```


core_classes.html

```
59:             <td id="searchbox"><form method="get"
60:                 action="http://www.google.com/search">
61:                 <input type="hidden" name="as_sitesearch" id="as_
s_sitesearch"
62:                     value="codeigniter.com/user_guide/" />S
earch User Guide<input
63:                 type="text" class="input" style="width:
200px;" name="q" id="q"
64:                 size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                 class="submit" name="sa" value="Go" />
66:             </form></td>
67:         </tr>
68:     </table>
69:     <!-- END BREADCRUMB -->
70:
71:     <br clear="all" />
72:
73:
74:     <!-- START CONTENT -->
75:     <div id="content">
76:
77:         <h1>Creating Core System Classes</h1>
78:
79:         <p>Every time CodeIgniter runs there are several base classes that
80:             are initialized automatically as part of the core framework. It
is
81:             possible, however, to swap any of the core system classes with
your
82:             own versions or even extend the core versions.</p>
83:
84:         <p>
85:             <strong>Most users will never have any need to do this, but
86:                 the option to replace or extend them does exist for tho
se who would
87:                 like to significantly alter the CodeIgniter core.</stro
ng>
88:         </p>
89:
90:         <p class="important">
91:             <strong>Note:</strong>&nbsp; Messing with a core system class h
as a
92:             lot of implications, so make sure you know what you are doing b
efore
93:             attempting it.
94:         </p>
95:
96:
97:         <h2>System Class List</h2>
98:
99:         <p>The following is a list of the core system files that are
100:             invoked every time CodeIgniter runs:</p>
101:
102:         <ul>
103:             <li>Benchmark</li>
104:             <li>Config</li>
105:             <li>Controller</li>
106:             <li>Exceptions</li>
107:             <li>Hooks</li>
108:             <li>Input</li>
109:             <li>Language</li>
110:             <li>Loader</li>
111:             <li>Log</li>
```

core_classes.html

```
112:         <li>Output</li>
113:         <li>Router</li>
114:         <li>URI</li>
115:         <li>Utf8</li>
116:     </ul>
117:
118: <h2>Replacing Core Classes</h2>
119:
120: <p>
121:     To use one of your own system classes instead of a default one
simply
122:     place your version inside your local
123:     <dfn>application/core</dfn>
124:     directory:
125: </p>
126:
127: <code>
128:     application/core/
129:     <dfn>some-class.php</dfn>
130: </code>
131:
132: <p>If this directory does not exist you can create it.</p>
133:
134: <p>Any file named identically to one from the list above will be
135:     used instead of the one normally used.</p>
136:
137: <p>
138:     Please note that your class must use
139:     <kbd>CI</kbd>
140:     as a prefix. For example, if your file is named
141:     <kbd>Input.php</kbd>
142:     the class will be named:
143: </p>
144:
145: <code>
146:     class CI_Input {<br />
147:     <br /> }
148: </code>
149:
150:
151:
152: <h2>Extending Core Class</h2>
153:
154: <p>If all you need to do is add some functionality to an existing
155:     library - perhaps add a function or two - then it's overkill to
156:     replace the entire library with your version. In this case it's
157:     better to simply extend the class. Extending a class is nearly
158:     identical to replacing a class with a couple exceptions:</p>
159:
160: <ul>
161:     <li>The class declaration must extend the parent class.</li>
162:     <li>Your new class name and filename must be prefixed with <kbd>
>MY_</kbd>
163:         (this item is configurable. See below.).
164:     </li>
165: </ul>
166:
167: <p>
168:     For example, to extend the native
169:     <kbd>Input</kbd>
170:     class you'll create a file named
171:     <dfn>application/core/</dfn>
172:     <kbd>MY_Input.php</kbd>
```


creating_drivers.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Creating Drivers : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototypelite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28: <!-- START NAVIGATION -->
29: <div id="nav">
30: <div id="nav_inner">
31: <script type="text/javascript">create_menu('../');</script>
32: </div>
33: </div>
34: <div id="nav2">
35: <a name="top"></a><a href="javascript:void(0);"
36: onclick="myHeight.toggle();"></a>
40: </div>
41: <div id="masthead">
42: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43: <tr>
44: <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45: <td id="breadcrumb_right"><a href="../toc.html">Table of
46: Contents Page</a></td>
47: </tr>
48: </table>
49: </div>
50: <!-- END NAVIGATION -->
51:
52: <!-- START BREADCRUMB -->
53: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54: <tr>
55: <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
56: er
57: <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="../index.
html">User Guide
58: <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;Creating Drivers</
td>
```

creating_drivers.html

```
59:             <td id="searchbox"><form method="get"
60:                 action="http://www.google.com/search">
61:                 <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                     value="codeigniter.com/user_guide/" />S
earch User Guide<input
63:                 type="text" class="input" style="width:
200px;" name="q" id="q"
64:                 size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                 class="submit" name="sa" value="Go" />
66:             </form></td>
67:         </tr>
68:     </table>
69:     <!-- END BREADCRUMB -->
70:
71:     <br clear="all" />
72:
73:
74:     <!-- START CONTENT -->
75:     <div id="content">
76:
77:         <h1>Creating Drivers</h1>
78:
79:         <h2>Driver Directory and File Structure</h2>
80:
81:         <p>Sample driver directory and file structure layout:</p>
82:
83:         <ul>
84:             <li>/application/libraries/Driver_name
85:                 <ul>
86:                     <li>Driver_name.php</li>
87:                     <li>drivers
88:                         <ul>
89:                             <li>Driver_name_subclass_1.php<
/li>
90:                             <li>Driver_name_subclass_2.php<
/li>
91:                             <li>Driver_name_subclass_3.php<
/li>
92:                         </ul>
93:                     </li>
94:                 </ul>
95:             </li>
96:         </ul>
97:
98:         <p class="important">
99:             <strong>NOTE:</strong> In order to maintain compatibility on
100:             case-sensitive file systems, the
101:             <samp>Driver_name</samp>
102:             directory must be
103:             <var>ucfirst()</var>
104:         </p>
105:
106:         <!-- @todo write this! -->
107:
108:
109:     </div>
110:     <!-- END CONTENT -->
111:
112:
113:     <div id="footer">
114:         <p>
```


requirements.html

111: </html>

managing_apps.html

```
59:                 <td id="searchbox"><form method="get"
60:                   action="http://www.google.com/search">
61:                   <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                   value="codeigniter.com/user_guide/" />S
earch User Guide<input
63:                   type="text" class="input" style="width:
200px;" name="q" id="q"
64:                   size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                   class="submit" name="sa" value="Go" />
66:                 </form></td>
67:             </tr>
68:         </table>
69:         <!-- END BREADCRUMB -->
70:
71:         <br clear="all" />
72:
73:         <!-- START CONTENT -->
74:         <div id="content">
75:
76:             <h1>Managing your Applications</h1>
77:
78:             <p>
79:                 By default it is assumed that you only intend to use CodeIgnite
r to
80:                 manage one application, which you will build in your
81:                 <dfn>application</dfn>
82:                 directory. It is possible, however, to have multiple sets of
83:                 applications that share a single CodeIgniter installation, or e
84:                 rename or relocate your
85:                 <dfn>application</dfn>
86:                 folder.
87:             </p>
88:
89:             <h2>Renaming the Application Folder</h2>
90:
91:             <p>
92:                 If you would like to rename your
93:                 <dfn>application</dfn>
94:                 folder you may do so as long as you open your main
95:                 <kbd>index.php</kbd>
96:                 file and set its name using the
97:                 <samp>$application_folder</samp>
98:                 variable:
99:             </p>
100:
101:             <code>$application_folder = "application";</code>
102:
103:             <h2>Relocating your Application Folder</h2>
104:
105:             <p>
106:                 It is possible to move your
107:                 <dfn>application</dfn>
108:                 folder to a different location on your server than your
109:                 <kbd>system</kbd>
110:                 folder. To do so open your main
111:                 <kbd>index.php</kbd>
112:                 and set a <em>full server path</em> in the
113:                 <samp>$application_folder</samp>
114:                 variable.
115:
```


managing_apps.html

```
116:         </p>
117:
118:
119:         <code>$application_folder = "/Path/to/your/application";</code>
120:
121:
122:         <h2>Running Multiple Applications with one CodeIgniter
123:             Installation</h2>
124:
125:         <p>
126:             If you would like to share a common CodeIgniter installation to
127:             manage several different applications simply put all of the
128:             directories located inside your
129:             <code>application</code>
130:             folder into their own sub-folder.
131:         </p>
132:
133:         <p>For example, let's say you want to create two applications,
134:             "foo" and "bar". You could structure your application folders l
like
135:             this:</p>
136:
137:         <code>
138:             applications/
139:             <code>foo</code>
140:             /<code> /> applications/
141:             <code>foo</code>
142:             /config/<code> /> applications/
143:             <code>foo</code>
144:             /controllers/<code> /> applications/
145:             <code>foo</code>
146:             /errors/<code> /> applications/
147:             <code>foo</code>
148:             /libraries/<code> /> applications/
149:             <code>foo</code>
150:             /models/<code> /> applications/
151:             <code>foo</code>
152:             /views/<code> /> applications/
153:             <code>bar</code>
154:             /<code> /> applications/
155:             <code>bar</code>
156:             /config/<code> /> applications/
157:             <code>bar</code>
158:             /controllers/<code> /> applications/
159:             <code>bar</code>
160:             /errors/<code> /> applications/
161:             <code>bar</code>
162:             /libraries/<code> /> applications/
163:             <code>bar</code>
164:             /models/<code> /> applications/
165:             <code>bar</code>
166:             /views/
167:         </code>
168:
169:
170:         <p>
171:             To select a particular application for use requires that you op
en
172:             your main
173:             <code>index.php</code>
174:             file and set the
175:             <code>$application_folder</code>
176:             variable. For example, to select the "foo" application for use
```


urls.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml
tml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>CodeIgniter URLs : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype-lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28: <!-- START NAVIGATION -->
29: <div id="nav">
30: <div id="nav_inner">
31: <script type="text/javascript">create_menu('../');</script>
32: </div>
33: </div>
34: <div id="nav2">
35: <a name="top"></a><a href="javascript:void(0);"
36: onclick="myHeight.toggle();"></a>
40: </div>
41: <div id="masthead">
42: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43: <tr>
44: <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45: <td id="breadcrumb_right"><a href="../toc.html">Table o
46: Contents Page</a></td>
47: </tr>
48: </table>
49: </div>
50: <!-- END NAVIGATION -->
51:
52: <!-- START BREADCRUMB -->
53: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54: <tr>
55: <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
56: er
57: <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="../index.
html">User Guide
58: <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;URLS</td>
59: <td id="searchbox"><form method="get">
```

```

urls.html
60:                                action="http://www.google.com/search">
61:                                <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                                value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp; <input
63:                                type="text" class="input" style="width:
200px;" name="q" id="q"
64:                                size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                                class="submit" name="sa" value="Go" />
66:                                </form></td>
67:                                </tr>
68:                                </table>
69:                                <!-- END BREADCRUMB -->
70:
71:                                <br clear="all" />
72:
73:
74:                                <!-- START CONTENT -->
75:                                <div id="content">
76:
77:
78:                                <h1>CodeIgniter URLs</h1>
79:
80:                                <p>
81:                                By default, URLs in CodeIgniter are designed to be search-engin
e and
82:                                human friendly. Rather than using the standard "query string"
83:                                approach to URLs that is synonymous with dynamic systems, CodeI
gniter
84:                                uses a <strong>segment-based</strong> approach:
85:                                </p>
86:
87:                                <code>
88:                                example.com/
89:                                <var>news</var>
90:                                /
91:                                <dfn>article</dfn>
92:                                /
93:                                <samp>my_article</samp>
94:                                </code>
95:
96:                                <p class="important">
97:                                <strong>Note:</strong> Query string URLs can be optionally enab
led,
98:                                as described below.
99:                                </p>
100:
101:                                <h2>URI Segments</h2>
102:
103:                                <p>The segments in the URL, in following with the
104:                                Model-View-Controller approach, usually represent:</p>
105:
106:                                <code>
107:                                example.com/
108:                                <var>class</var>
109:                                /
110:                                <dfn>function</dfn>
111:                                /
112:                                <samp>ID</samp>
113:                                </code>
114:
115:                                <ol>

```

urls.html

```
116:         <li>The first segment represents the controller <strong>class</strong>
strong>
117:             that should be invoked.
118:         </li>
119:         <li>The second segment represents the class <strong>function</strong>,
strong>,
120:             or method, that should be called.
121:         </li>
122:         <li>The third, and any additional segments, represent the ID and
d
123:             any variables that will be passed to the controller.</li>
i>
124:     </ol>
125:
126:     <p>
127:         The <a href="../../../libraries/uri.html">URI Class</a> and the <a
128:             href="../../../helpers/url_helper.html">URL Helper</a> contain
n functions
129:         that make it easy to work with your URI data. In addition, your
URLs
130:         can be remapped using the <a href="routing.html">URI Routing</a>
>
131:         feature for more flexibility.
132:     </p>
133:
134:
135:
136:     <h2>Removing the index.php file</h2>
137:
138:     <p>
139:         By default, the <strong>index.php</strong> file will be included
d in
140:         your URLs:
141:     </p>
142:
143:     <code>
144:         example.com/
145:         <var>index.php</var>
146:         /news/article/my_article
147:     </code>
148:
149:     <p>You can easily remove this file by using a .htaccess file with
150:         some simple rules. Here is an example of such a file, using the
151:         "negative" method in which everything is redirected except the
152:         specified items:</p>
153:
154:     <code>
155:         RewriteEngine on<br /> RewriteCond $1
156:         !^(index\.php|images|robots\.txt)<br /> RewriteRule ^(.*)$
157:         /index.php/$1 [L]
158:     </code>
159:
160:     <p>In the above example, any HTTP request other than those for
161:         index.php, images, and robots.txt is treated as a request for
our
162:         index.php file.</p>
163:
164:
165:     <h2>Adding a URL Suffix</h2>
166:
167:     <p>
168:         In your
169:         <dfn>config/config.php</dfn>
```


reserved_names.html

```
60:                                     action="http://www.google.com/search">
61:                                     <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                                     value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;   <input
63:                                     type="text" class="input" style="width:
200px;" name="q" id="q"
64:                                     size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                                     class="submit" name="sa" value="Go" />
66:                                     </form></td>
67:                                     </tr>
68:     </table>
69:     <!-- END BREADCRUMB -->
70:
71:     <br clear="all" />
72:
73:     <!-- START CONTENT -->
74:     <div id="content">
75:
76:
77:         <h1>Reserved Names</h1>
78:
79:         <p>In order to help out, CodeIgniter uses a series of functions
80:            and names in its operation. Because of this, some names cannot
be
81:            used by a developer. Following is a list of reserved names that
82:            cannot be used.</p>
83:         <h3>Controller names</h3>
84:         <p>Since your controller classes will extend the main application
85:            controller you must be careful not to name your functions ident
icall
86:            to the ones used by that class, otherwise your local functions
will
87:            override them. The following is a list of reserved names. Do no
t name
88:            your controller any of these:</p>
89:         <ul>
90:             <li>Controller</li>
91:             <li>CI_Base</li>
92:             <li>_ci_initialize</li>
93:             <li>Default</li>
94:             <li>index</li>
95:         </ul>
96:         <h3>Functions</h3>
97:         <ul>
98:             <li>is_really_writable</li>
99:             <li>load_class</li>
100:            <li>get_config</li>
101:            <li>config_item</li>
102:            <li>show_error</li>
103:            <li>show_404</li>
104:            <li>log_message</li>
105:            <li>_exception_handler</li>
106:            <li>get_instance</li>
107:        </ul>
108:        <h3>Variables</h3>
109:        <ul>
110:            <li>$config</li>
111:            <li>$mimes</li>
112:            <li>$lang</li>
113:        </ul>
114:        <h3>Constants</h3>
```


errors.html

```
59:                 <td id="searchbox"><form method="get"
60:                   action="http://www.google.com/search">
61:                   <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                               value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp; <input
63:                               type="text" class="input" style="width:
200px;" name="q" id="q"
64:                               size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                               class="submit" name="sa" value="Go" />
66:                   </form></td>
67:           </tr>
68:   </table>
69:   <!-- END BREADCRUMB -->
70:
71:   <br clear="all" />
72:
73:
74:   <!-- START CONTENT -->
75:   <div id="content">
76:
77:       <h1>Error Handling</h1>
78:
79:       <p>CodeIgniter lets you build error reporting into your
80:         applications using the functions described below. In addition,
it has
81:         an error logging class that permits error and debugging message
s to
82:         be saved as text files.</p>
83:
84:       <p class="important">
85:         <strong>Note:</strong> By default, CodeIgniter displays all PHP
86:         errors. You might wish to change this behavior once your develo
pment
87:         is complete. You'll find the
88:         <dfn>error_reporting()</dfn>
89:         function located at the top of your main index.php file. Disabl
ing
90:         error reporting will NOT prevent log files from being written i
f
91:         there are errors.
92:       </p>
93:
94:       <p>Unlike most systems in CodeIgniter, the error functions are
95:         simple procedural interfaces that are available globally throug
hout
96:         the application. This approach permits error messages to get
97:         triggered without having to worry about class/function scoping.
</p>
98:
99:       <p>The following functions let you generate errors:</p>
100:
101:       <h2>
102:         show_error( '
103:         <var>message</var>
104:         ' [ , int
105:         <var>$status_code</var>
106:         = 500 ] )
107:       </h2>
108:       <p>This function will display the error message supplied to it
109:         using the following error template:</p>
110:
111:       <p>
```


caching.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Web Page Caching : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../../nav/nav.js"></script>
15: <script type="text/javascript" src="../../nav/prototypelite.js"></script>
16: <script type="text/javascript" src="../../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28: <!-- START NAVIGATION -->
29: <div id="nav">
30: <div id="nav_inner">
31: <script type="text/javascript">create_menu('../');</script>
32: </div>
33: </div>
34: <div id="nav2">
35: <a name="top"></a><a href="javascript:void(0);"
36: onclick="myHeight.toggle();"></a>
40: </div>
41: <div id="masthead">
42: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43: <tr>
44: <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45: <td id="breadcrumb_right"><a href="../../toc.html">Table of Contents
46: </td>
47: </tr>
48: </table>
49: </div>
50: <!-- END NAVIGATION -->
51:
52: <!-- START BREADCRUMB -->
53: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54: <tr>
55: <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgniter
56: </td>
57: <td><a href="../../index.html">User Guide
58: </td>
59: <td id="searchbox"><form method="get"
```

caching.html

```
60:                                     action="http://www.google.com/search">
61:                                     <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                                     value="codeigniter.com/user_guide/" />S
earch User Guide<input
63:                                     type="text" class="input" style="width:
200px;" name="q" id="q"
64:                                     size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                                     class="submit" name="sa" value="Go" />
66:                                     </form></td>
67:                                     </tr>
68:                                     </table>
69:                                     <!-- END BREADCRUMB -->
70:
71:                                     <br clear="all" />
72:
73:
74:                                     <!-- START CONTENT -->
75:                                     <div id="content">
76:
77:
78:                                     <h1>Web Page Caching</h1>
79:
80:                                     <p>CodeIgniter lets you cache your pages in order to achieve
81:                                     maximum performance.</p>
82:
83:                                     <p>Although CodeIgniter is quite fast, the amount of dynamic
84:                                     information you display in your pages will correlate directly t
o the
85:                                     server resources, memory, and processing cycles utilized, which
86:                                     affect your page load speeds. By caching your pages, since they
are
87:                                     saved in their fully rendered state, you can achieve performanc
e that
88:                                     nears that of static web pages.</p>
89:
90:
91:                                     <h2>How Does Caching Work?</h2>
92:
93:                                     <p>
94:                                     Caching can be enabled on a per-page basis, and you can set the
95:                                     length of time that a page should remain cached before being
96:                                     refreshed. When a page is loaded for the first time, the cache
file
97:                                     will be written to your
98:                                     <dfn>application/cache</dfn>
99:                                     folder. On subsequent page loads the cache file will be retriev
ed and
100:                                     sent to the requesting user's browser. If it has expired, it wi
ll be
101:                                     deleted and refreshed before being sent to the browser.
102:                                     </p>
103:
104:                                     <p>Note: The Benchmark tag is not cached so you can still view
105:                                     your page load speed when caching is enabled.</p>
106:
107:                                     <h2>Enabling Caching</h2>
108:
109:                                     <p>To enable caching, put the following tag in any of your
110:                                     controller functions:</p>
111:
112:                                     <code>
```


alternative_php.html

```
ax</td>
60:                 <td id="searchbox"><form method="get"
61:                   action="http://www.google.com/search">
62:                     <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
63:                       value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;   <input
64:                       type="text" class="input" style="width:
200px;" name="q" id="q"
65:                       size="31" maxlength="255" value="" />&nbsp;
bsp;<input type="submit"
66:                       class="submit" name="sa" value="Go" />
67:                 </form></td>
68:             </tr>
69:     </table>
70:     <!-- END BREADCRUMB -->
71:
72:     <br clear="all" />
73:
74:
75:     <!-- START CONTENT -->
76:     <div id="content">
77:
78:         <h1>Alternate PHP Syntax for View Files</h1>
79:
80:         <p>
81:             If you do not utilize CodeIgniter's <a
82:                 href=" ../libraries/parser.html">template engine</a>, yo
u'll be using
83:             pure PHP in your View files. To minimize the PHP code in these
files,
84:             and to make it easier to identify the code blocks it is recomme
nded
85:             that you use PHP's alternative syntax for control structures and
short
86:             tag echo statements. If you are not familiar with this syntax,
it
87:             allows you to eliminate the braces from your code, and eliminat
e
88:             "echo" statements.
89:         </p>
90:
91:         <h2>Automatic Short Tag Support</h2>
92:
93:         <p>
94:             <strong>Note:</strong> If you find that the syntax described in
this
95:             page does not work on your server it might be that "short tags"
are
96:             disabled in your PHP ini file. CodeIgniter will optionally rewr
ite
97:             short tags on-the-fly, allowing you to use that syntax even if
your
98:             server doesn't support it. This feature can be enabled in your
99:             <dfn>config/config.php</dfn>
100:            file.
101:         </p>
102:
103:         <p class="important">
104:             Please note that if you do use this feature, if PHP errors are
105:             encountered in your <strong>view files</strong>, the error mess
age
106:             and line number will not be accurately shown. Instead, all erro
```

alternative_php.html

```
rs
107:         will be shown as
108:         <code>eval()</code>
109:         errors.
110:     </p>
111:
112:     <h2>Alternative Echos</h2>
113:
114:     <p>Normally to echo, or print out a variable you would do this:</p>
115:
116:     <code>&lt;?php echo $variable; ?></code>
117:
118:     <p>With the alternative syntax you can instead do it this way:</p>
119:
120:     <code>&lt;?=$variable?></code>
121:
122:
123:
124:
125:     <h2>Alternative Control Structures</h2>
126:
127:     <p>
128:         Controls structures, like
129:         <code>if</code>
130:         ,
131:         <code>for</code>
132:         ,
133:         <code>foreach</code>
134:         , and
135:         <code>while</code>
136:         can be written in a simplified format as well. Here is an exampl
le
137:         using foreach:
138:     </p>
139:
140:     <code>
141:         &lt;ul><br /> <br />
142:         <code>&lt;?php foreach ($todo as $item): ?></code>
143:         <br /> <br /> &lt;li>
144:         <code>&lt;?=$item?></code>
145:         &lt;/li><br /> <br />
146:         <code>&lt;?php endforeach; ?></code>
147:         <br /> <br /> &lt;/ul>
148:     </code>
149:
150:     <p>
151:         Notice that there are no braces. Instead, the end brace is repl
aced
152:         with
153:         <code>endforeach</code>
154:         . Each of the control structures listed above has a similar clo
sing
155:         syntax:
156:         <code>endif</code>
157:         ,
158:         <code>endfor</code>
159:         ,
160:         <code>endforeach</code>
161:         , and
162:         <code>endwhile</code>
163:     </p>
164:
165:     <p>Also notice that instead of using a semicolon after each
```


security.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Security : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype_lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28: <!-- START NAVIGATION -->
29: <div id="nav">
30: <div id="nav_inner">
31: <script type="text/javascript">create_menu('../');</script>
32: </div>
33: </div>
34: <div id="nav2">
35: <a name="top"></a><a href="javascript:void(0);"
36: onclick="myHeight.toggle();"></a>
40: </div>
41: <div id="masthead">
42: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43: <tr>
44: <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45: <td id="breadcrumb_right"><a href="../toc.html">Table of
46: Contents Page</a></td>
47: </tr>
48: </table>
49: </div>
50: <!-- END NAVIGATION -->
51:
52: <!-- START BREADCRUMB -->
53: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54: <tr>
55: <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
56: er
57: <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="../index.
58: html">User Guide
59: <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;Security</td>
60: <td id="searchbox"><form method="get">
```

security.html

```

60:                                action="http://www.google.com/search">
61:                                <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                                value="codeigniter.com/user_guide/" />S
earch User Guide<input
63:                                type="text" class="input" style="width:
200px;" name="q" id="q"
64:                                size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                                class="submit" name="sa" value="Go" />
66:                                </form></td>
67:                                </tr>
68:                                </table>
69:                                <!-- END BREADCRUMB -->
70:
71:                                <br clear="all" />
72:
73:                                <!-- START CONTENT -->
74:                                <div id="content">
75:
76:                                    <h1>Security</h1>
77:
78:                                    <p>This page describes some "best practices" regarding web
79:                                    security, and details CodeIgniter's internal security features.
80:                                    </p>
</p>
81:
82:                                    <h2>URI Security</h2>
83:
84:                                    <p>CodeIgniter is fairly restrictive regarding which characters it
85:                                    allows in your URI strings in order to help minimize the possib
86:                                    ility
87:                                    that malicious data can be passed to your application. URIs may
88:                                    only
89:                                    contain the following:</p>
90:
91:                                    <ul>
92:                                        <li>Alpha-numeric text</li>
93:                                        <li>Tilde: ~</li>
94:                                        <li>Period: .</li>
95:                                        <li>Colon: :</li>
96:                                        <li>Underscore: _</li>
97:                                        <li>Dash: -</li>
98:                                    </ul>
99:
100:                                    <h2>Register_globals</h2>
101:
102:                                    <p>During system initialization all global variables are unset,
103:                                    except those found in the $_GET, $_POST, and $_COOKIE arrays. T
he
104:                                    unsetting routine is effectively the same as register_globals =
off.</p>
105:
106:                                    <a name="error_reporting"></a>
107:                                    <h2>error_reporting</h2>
108:
109:                                    <p>In production environments, it is typically desirable to
110:                                    disable PHP's error reporting by setting the internal error_rep
orting
111:                                    flag to a value of 0. This disables native PHP errors from bein
g
                                    rendered as output, which may potentially contain sensitive

```


security.html

```
112:         information.</p>
113:
114:     <p>
115:         Setting CodeIgniter's
116:         <code>ENVIRONMENT</code>
117:         constant in index.php to a value of '
118:         <code>production</code>
119:         ' will turn off these errors. In development mode, it is recomm
ended
120:         that a value of '
121:         <code>development</code>
122:         ' is used. More information about differentiating between
123:         environments can be found on the <a href="environments.html">Ha
ndling
124:         Environments</a> page.
125:     </p>
126:
127:     <h2>magic_quotes_runtime</h2>
128:
129:     <p>The magic_quotes_runtime directive is turned off during system
130:         initialization so that you don't have to remove slashes when
131:         retrieving data from your database.</p>
132:
133:     <h1>Best Practices</h1>
134:
135:     <p>Before accepting any data into your application, whether it be
136:         POST data from a form submission, COOKIE data, URI data, XML-RP
C
137:         data, or even data from the SERVER array, you are encouraged to
138:         practice this three step approach:</p>
139:
140:     <ol>
141:         <li>Filter the data as if it were tainted.</li>
142:         <li>Validate the data to ensure it conforms to the correct type
,
143:         length, size, etc. (sometimes this step can replace ste
p one)</li>
144:         <li>Escape the data before submitting it into your database.</l
i>
145:     </ol>
146:
147:     <p>CodeIgniter provides the following functions to assist in this
148:         process:</p>
149:
150:     <ul>
151:
152:         <li><h2>XSS Filtering</h2>
153:
154:             <p>
155:                 CodeIgniter comes with a Cross Site Scripting f
ilter. This filter
156:                 looks for commonly used techniques to embed mal
icious Javascript
157:                 into your data, or other types of code that att
empt to hijack
158:                 cookies or do other malicious things. The XSS F
ilter is described <a
159:                 href=" ../libraries/security.html">here<
/a>.
160:             </p></li>
161:
162:         <li><h2>Validate the data</h2>
163:
```


helpers.html

```
59:                 <td id="searchbox"><form method="get"
60:                 action="http://www.google.com/search">
61:                 <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                 value="codeigniter.com/user_guide/" />S
earch User Guide<input
63:                 type="text" class="input" style="width:
200px;" name="q" id="q"
64:                 size="31" maxlength="255" value="" />&nb
bsp;<input type="submit"
65:                 class="submit" name="sa" value="Go" />
66:                 </form></td>
67:             </tr>
68:         </table>
69:         <!-- END BREADCRUMB -->
70:
71:         <br clear="all" />
72:
73:
74:         <!-- START CONTENT -->
75:         <div id="content">
76:
77:             <h1>Helper Functions</h1>
78:
79:             <p>
80:                 Helpers, as the name suggests, help you with tasks. Each helper
file
81:                 is simply a collection of functions in a particular category. T
here
82:                 are
83:                 <dfn>URL Helpers</dfn>
84:                 , that assist in creating links, there are
85:                 <dfn>Form Helpers</dfn>
86:                 that help you create form elements,
87:                 <dfn>Text Helpers</dfn>
88:                 perform various text formatting routines,
89:                 <dfn>Cookie Helpers</dfn>
90:                 set and read cookies,
91:                 <dfn>File Helpers</dfn>
92:                 help you deal with files, etc.
93:             </p>
94:
95:             <p>Unlike most other systems in CodeIgniter, Helpers are not
96:                 written in an Object Oriented format. They are simple, procedur
al
97:                 functions. Each helper function performs one specific task, wit
h no
98:                 dependence on other functions.</p>
99:
100:            <p>
101:                CodeIgniter does not load Helper Files by default, so the first
step
102:                in using a Helper is to load it. Once loaded, it becomes global
ly
103:                available in your <a href="../general/controllers.html">control
ler</a>
104:                and <a href="../general/views.html">views</a>.
105:            </p>
106:
107:            <p>
108:                Helpers are typically stored in your
109:                <dfn>system/helpers</dfn>
110:                , or
```

helpers.html

```
111:         <dfn>application/helpers </dfn>
112:         directory. CodeIgniter will look first in your
113:         <dfn>application/helpers</dfn>
114:         directory. If the directory does not exist or the specified hel
per is
115:         not located there CI will instead look in your global
116:         <dfn>system/helpers</dfn>
117:         folder.
118:     </p>
119:
120:
121:     <h2>Loading a Helper</h2>
122:
123:     <p>Loading a helper file is quite simple using the following
124:         function:</p>
125:
126:     <code>
127:         $this->load->helper('
128:         <var>name</var>
129:         ');
130:     </code>
131:
132:     <p>
133:         Where
134:         <var>name</var>
135:         is the file name of the helper, without the .php file extension
or
136:         the "helper" part.
137:     </p>
138:
139:     <p>
140:         For example, to load the
141:         <dfn>URL Helper</dfn>
142:         file, which is named
143:         <var>url_helper.php</var>
144:         , you would do this:
145:     </p>
146:
147:     <code>
148:         $this->load->helper('
149:         <var>url</var>
150:         ');
151:     </code>
152:
153:     <p>A helper can be loaded anywhere within your controller
154:         functions (or even within your View files, although that's not
a good
155:         practice), as long as you load it before you use it. You can lo
ad
156:         your helpers in your controller constructor so that they become
157:         available automatically in any function, or you can load a help
er in
158:         a specific function that needs it.</p>
159:
160:     <p class="important">Note: The Helper loading function above does
161:         not return a value, so don't try to assign it to a variable. Ju
st use
162:         it as shown.</p>
163:
164:
165:     <h2>Loading Multiple Helpers</h2>
166:
167:     <p>If you need to load more than one helper you can specify them
```

helpers.html

```
168:         in an array, like this:</p>
169:
170:     <code>
171:         $this->load->helper(
172:         <samp>array(</samp>
173:         '
174:         <var>helper1</var>
175:         ',
176:         <var>helper2</var>
177:         ',
178:         <var>helper3</var>
179:         '
180:         <samp>)</samp>
181:         );
182:     </code>
183:
184:     <h2>Auto-loading Helpers</h2>
185:
186:     <p>
187:         If you find that you need a particular helper globally througho
ut
188:         your application, you can tell CodeIgniter to auto-load it duri
ng
189:         system initialization. This is done by opening the
190:         <var>application/config/autoload.php</var>
191:         file and adding the helper to the autoload array.
192:     </p>
193:
194:     <h2>Using a Helper</h2>
195:
196:     <p>Once you've loaded the Helper File containing the function you
197:     intend to use, you'll call it the way you would a standard PHP
198:     function.</p>
199:
200:     <p>
201:         For example, to create a link using the
202:         <dfn>anchor()</dfn>
203:         function in one of your view files you would do this:
204:     </p>
205:
206:     <code>&lt;?php echo anchor('blog/comments', 'Click Here');?&gt;</code>
207:
208:     <p>Where "Click Here" is the name of the link, and "blog/comments"
209:         is the URI to the controller/function you wish to link to.</p>
210:
211:     <h2>"Extending" Helpers</h2>
212:
213:     <p>
214:         To "extend" Helpers, create a file in your
215:         <dfn>application/helpers/</dfn>
216:         folder with an identical name to the existing Helper, but prefi
xed
217:         with
218:         <kbd>MY_</kbd>
219:         (this item is configurable. See below.).
220:     </p>
221:
222:     <p>
223:         If all you need to do is add some functionality to an existing
224:         helper - perhaps add a function or two, or change how a particu
lar
225:         helper function operates - then it's overkill to replace the en
tire
```


credits.html

```
60:                                     action="http://www.google.com/search">
61:                                     <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                                     value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;   <input
63:                                     type="text" class="input" style="width:
200px;" name="q" id="q"
64:                                     size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                                     class="submit" name="sa" value="Go" />
66:                                     </form></td>
67:                                     </tr>
68:     </table>
69:     <!-- END BREADCRUMB -->
70:
71:     <br clear="all" />
72:
73:
74:     <!-- START CONTENT -->
75:     <div id="content">
76:
77:         <h1>Credits</h1>
78:
79:         <p>
80:             CodeIgniter was originally developed by <a
81:                 href="http://www.ellislab.com/">Rick Ellis</a> (CEO of
<a
82:                 href="http://ellislab.com/">EllisLab, Inc.</a>). The fr
amework was
83:             written for performance in the real world, with many of the cla
ss
84:             libraries, helpers, and sub-systems borrowed from the code-base
of <a
85:                 href="http://www.expressionengine.com/">ExpressionEngin
e</a>.
86:         </p>
87:
88:         <p>It was, for years, developed and maintained by EllisLab, the
89:             ExpressionEngine Development Team and a group of community memb
ers
90:             called the Reactor Team.</p>
91:
92:             In 2014, CodeIgniter was acquired by the <a href="http://bcit.ca/">Brit
ish
93:             Columbia Institute of Technology</a> and was then officially an
nounced as
94:             a community-maintaned project.
95:         </p>
96:
97:         <p>Bleeding edge development is spearheaded by the handpicked
98:             contributors of the Reactor Team.</p>
99:
100:        <p>A hat tip goes to Ruby on Rails for inspiring us to create a
101:            PHP framework, and for bringing frameworks into the general
102:            consciousness of the web community.</p>
103:
104:    </div>
105:    <!-- END CONTENT -->
106:
107:
108:    <div id="footer">
109:        <p>
110:            Previous Topic:&nbsp;&nbsp; <a href=" ../changelog.html">Change L
```


libraries.html

```
59:                 <td id="searchbox"><form method="get"
60:                   action="http://www.google.com/search">
61:                   <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                   value="codeigniter.com/user_guide/" />S
earch User Guide<input
63:                   type="text" class="input" style="width:
200px;" name="q" id="q"
64:                   size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                   class="submit" name="sa" value="Go" />
66:                 </form></td>
67:             </tr>
68:         </table>
69:         <!-- END BREADCRUMB -->
70:
71:         <br clear="all" />
72:
73:         <!-- START CONTENT -->
74:         <div id="content">
75:
76:             <h1>Using CodeIgniter Libraries</h1>
77:
78:
79:
80:             <p>
81:                 All of the available libraries are located in your
82:                 <dfn>system/libraries</dfn>
83:                 folder. In most cases, to use one of these classes involves
84:                 initializing it within a <a href="controllers.html">controller<
/a>
85:                 using the following initialization function:
86:             </p>
87:
88:             <code>
89:                 $this->load->library('
90:                 <var>class name</var>
91:                 ');
92:             </code>
93:
94:             <p>
95:                 Where
96:                 <var>class name</var>
97:                 is the name of the class you want to invoke. For example, to lo
ad the
98:                 form validation class you would do this:
99:             </p>
100:
101:             <code>
102:                 $this->load->library('
103:                 <var>form_validation</var>
104:                 ');
105:             </code>
106:
107:             <p>Once initialized you can use it as indicated in the user guide
108:                 page corresponding to that class.</p>
109:
110:             <p>Additionally, multiple libraries can be loaded at the same time
111:                 by passing an array of libraries to the load function.</p>
112:
113:             <code>
114:                 $this->load->library(array('
115:                 <var>email</var>
```


creating_libraries.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml
tml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Creating Libraries : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype-lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28:     <!-- START NAVIGATION -->
29:     <div id="nav">
30:         <div id="nav_inner">
31:             <script type="text/javascript">create_menu('../');</script>
32:         </div>
33:     </div>
34:     <div id="nav2">
35:         <a name="top"></a><a href="javascript:void(0);"
36:         onclick="myHeight.toggle();"></a>
40:     </div>
41:     <div id="masthead">
42:         <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43:             <tr>
44:                 <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45:                 <td id="breadcrumb_right"><a href="../toc.html">Table o
46:
47:                                     Contents Page</a></td>
48:             </tr>
49:         </table>
50:     </div>
51:     <!-- END NAVIGATION -->
52:
53:     <!-- START BREADCRUMB -->
54:     <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
55:         <tr>
56:             <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
57:
58:                                     Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="../index.
er
html">User Guide
58:                                     Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;Creating Libraries
</td>
```

creating_libraries.html

```
59:                 <td id="searchbox"><form method="get"
60:                   action="http://www.google.com/search">
61:                   <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                   value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;   <input
63:                   type="text" class="input" style="width:
200px;" name="q" id="q"
64:                   size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                   class="submit" name="sa" value="Go" />
66:                 </form></td>
67:           </tr>
68:     </table>
69:     <!-- END BREADCRUMB -->
70:
71:     <br clear="all" />
72:
73:
74:     <!-- START CONTENT -->
75:     <div id="content">
76:
77:         <h1>Creating Libraries</h1>
78:
79:         <p>
80:             When we use the term "Libraries" we are normally referring to t
he
81:             classes that are located in the
82:             <kbd>libraries</kbd>
83:             directory and described in the Class Reference of this user gui
de. In
84:             this case, however, we will instead describe how you can create
your
85:             own libraries within your
86:             <dfn>application/libraries</dfn>
87:             directory in order to maintain separation between your local
88:             resources and the global framework resources.
89:         </p>
90:
91:         <p>
92:             As an added bonus, CodeIgniter permits your libraries to
93:             <kbd>extend</kbd>
94:             native classes if you simply need to add some functionality to
an
95:             existing library. Or you can even replace native libraries just
by
96:             placing identically named versions in your
97:             <dfn>application/libraries</dfn>
98:             folder.
99:         </p>
100:
101:         <p>In summary:</p>
102:
103:         <ul>
104:             <li>You can create entirely new libraries.</li>
105:             <li>You can extend native libraries.</li>
106:             <li>You can replace native libraries.</li>
107:         </ul>
108:
109:         <p>The page below explains these three concepts in detail.</p>
110:
111:         <p class="important">
112:             <strong>Note:</strong> The Database classes can not be extended
```


creating_libraries.html

```
221:         <dfn>application/config/</dfn>
222:         folder. Note that if you dynamically pass parameters as describ
ed
223:         above, the config file option will not be available.
224:     </p>
225:
226:
227:
228:
229:
230:
231:
232:     <h2>Utilizing CodeIgniter Resources within Your Library</h2>
233:
234:
235:     <p>
236:         To access CodeIgniter's native resources within your library us
e the
237:         <code>get_instance()</code>
238:         function. This function returns the CodeIgniter super object.
239:     </p>
240:
241:     <p>
242:         Normally from within your controller functions you will call an
y of
243:         the available CodeIgniter functions using the
244:         <code>$this</code>
245:         construct:
246:     </p>
247:
248:     <code>
249:         <strong>$this</strong>->load->helper('url');<br /> <strong>$thi
s</strong>->load->library('session');<br />
250:         <strong>$this</strong>->config->item('base_url');<br /> etc.
251:     </code>
252:
253:     <p>
254:         <code>$this</code>
255:         , however, only works directly within your controllers, your mo
dels,
256:         or your views. If you would like to use CodeIgniter's classes f
rom
257:         within your own custom classes you can do so as follows:
258:     </p>
259:
260:
261:     <p>First, assign the CodeIgniter object to a variable:</p>
262:
263:     <code>$CI =& get_instance();</code>
264:
265:     <p>
266:         Once you've assigned the object to a variable, you'll use that
267:         variable <em>instead</em> of
268:         <code>$this</code>
269:         :
270:     </p>
271:
272:     <code>
273:         $CI =& get_instance();<br /> <br /> $CI->load->helper('url'
);<br />
274:         $CI->load->library('session');<br /> $CI->config->item('base_ur
l');<br />
275:         etc.
```

creating_libraries.html

```
276:         </code>
277:
278:         <p class="important">
279:             <strong>Note:</strong> You'll notice that the above get_instanc
e()
280:             function is being passed by reference: <br />
281:             <br />
282:             <var>$CI =&amp; get_instance();</var>
283:             <br /> <br />
284:             <kbd>This is very important.</kbd>
285:             Assigning by reference allows you to use the original CodeIgnit
er
286:             object rather than creating a copy of it.
287:         </p>
288:
289:         <h2>Replacing Native Libraries with Your Versions</h2>
290:
291:         <p>
292:             Simply by naming your class files identically to a native libra
ry
293:             will cause CodeIgniter to use it instead of the native one. To
use
294:             this feature you must name the file and the class declaration e
xactly
295:             the same as the native library. For example, to replace the nat
ive
296:             <code>
297:             <code>Email</code>
298:             library you'll create a file named
299:             <code>application/libraries/Email.php</code>
300:             , and declare your class with:
301:         </p>
302:
303:         <code>
304:             class CI_Email {<br />
305:             <br /> }
306:         </code>
307:
308:         <p>
309:             Note that most native classes are prefixed with
310:             <code>CI_</code>
311:             .
312:         </p>
313:
314:         <p>To load your library you'll see the standard loading function:</p>
315:
316:         <code>
317:             $this->load->library('
318:             <code>email</code>
319:             ');
320:         </code>
321:
322:         <p class="important">
323:             <strong>Note:</strong> At this time the Database classes can no
t be
324:             replaced with your own versions.
325:         </p>
326:
327:         <h2>Extending Native Libraries</h2>
328:
329:         <p>If all you need to do is add some functionality to an existing
330:             library - perhaps add a function or two - then it's overkill to
331:
```



```
views.html
60:                                     action="http://www.google.com/search">
61:                                     <input type="hidden" name="as_sitesearch" id="as_
s_sitesearch"
62:                                     value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;   <input
63:                                     type="text" class="input" style="width:
200px;" name="q" id="q"
64:                                     size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                                     class="submit" name="sa" value="Go" />
66:                                     </form></td>
67:                                     </tr>
68:                               </table>
69:                               <!-- END BREADCRUMB -->
70:
71:                               <br clear="all" />
72:
73:                               <!-- START CONTENT -->
74:                               <div id="content">
75:
76:                                   <h1>Views</h1>
77:
78:                                   <p>
79:                                       A
80:                                       <dfn>view</dfn>
81:                                       is simply a web page, or a page fragment, like a header, footer
82:                                       sidebar, etc. In fact, views can flexibly be embedded within ot
her
83:                                       views (within other views, etc., etc.) if you need this type of
84:                                       hierarchy.
85:
86:                                   </p>
87:
88:                                   <p>
89:                                       Views are never called directly, they must be loaded by a <a
90:                                       href="controllers.html">controller</a>. Remember that i
n an MVC
91:                                       framework, the Controller acts as the traffic cop, so it is
92:                                       responsible for fetching a particular view. If you have not rea
d the
93:                                       <a href="controllers.html">Controllers</a> page you should do s
o
94:                                       before continuing.
95:
96:                                   </p>
97:
98:                                   <p>
99:                                       Using the example controller you created in the <a
a view to it.
100:                                       href="controllers.html">controller</a> page, let's add
101:
102:                                   <h2>Creating a View</h2>
103:
104:                                   <p>
105:                                       Using your text editor, create a file called
106:                                       <dfn>blogview.php</dfn>
107:                                       , and put this in it:
108:
109:                                   </p>
110:                                   <textarea class="textarea" style="width: 100%" cols="50" rows="10">
111: <!-->html>
112: <!-->head>
```

views.html

```
113: <title>My Blog</title>
114: </head>
115: <body>
116:     <h1>Welcome to my Blog!</h1>
117: </body>
118: </html>
119: </textarea>
120:
121:     <p>
122:         Then save the file in your
123:         <dfn>application/views/</dfn>
124:         folder.
125:     </p>
126:
127:     <h2>Loading a View</h2>
128:
129:     <p>To load a particular view file you will use the following
130:         function:</p>
131:
132:     <code>
133:         $this->load->view('
134:         <var>name</var>
135:         ');
136:     </code>
137:
138:     <p>
139:         Where
140:         <var>name</var>
141:         is the name of your view file. Note: The .php file extension do
142:         es not
143:         need to be specified unless you use something other than
144:         <kbd>.php</kbd>
145:         .
146:     </p>
147:
148:     <p>
149:         Now, open the controller file you made earlier called
150:         <dfn>blog.php</dfn>
151:         , and replace the echo statement with the view loading function
152:         :
153:
154:     </p>
155:     <code class="text" style="width: 100%; height: 100px;">
156: <code>?php
157: class Blog extends CI_Controller {
158:
159:     function index()
160:     {
161:         $this->load->view('blogview');
162:     }
163: }
164: ?&gt;
165: </code>
166:
167:
168:     <p>If you visit your site using the URL you did earlier you should
169:         see your new view. The URL was similar to this:</p>
170:
171:     <code>
172:         example.com/index.php/
173:         <var>blog</var>
```


views.html

```
284:         <h2>Creating Loops</h2>
285:
286:         <p>The data array you pass to your view files is not limited to
287:             simple variables. You can pass multi dimensional arrays, which
can be
288:             looped to generate multiple rows. For example, if you pull data
from
289:             your database it will typically be in the form of a multi-dimen
sional
290:             array.</p>
291:
292:         <p>Here's a simple example. Add this to your controller:</p>
293:
294:         <textarea class="textarea" style="width: 100%" cols="50" rows="17">
295: &lt;?php
296: class Blog extends CI_Controller {
297:
298:     function index()
299:     {
300:         $data['todo_list'] = array('Clean House', 'Call Mom', 'Run Errands');
301:
302:         $data['title'] = "My Real Title";
303:         $data['heading'] = "My Real Heading";
304:
305:         $this->load->view('blogview', $data);
306:     }
307: }
308: ?&gt;
309: </textarea>
310:
311:
312:         <p>Now open your view file and create a loop:</p>
313:
314:
315:         <textarea class="textarea" style="width: 100%" cols="50" rows="24">
316: &lt;?html>
317: &lt;?head>
318: &lt;?title>&lt;?php echo $title;?&gt;&lt;?/title>
319: &lt;?/head>
320: &lt;?body>
321: &lt;?h1>&lt;?php echo $heading;?&gt;&lt;?/h1>
322:
323: &lt;?h3>My Todo List&lt;?/h3>
324:
325: &lt;?ul>
326: &lt;?php foreach ($todo_list as $item):?>
327:
328: &lt;?li>&lt;?php echo $item;?&gt;&lt;?/li>
329:
330: &lt;?php endforeach;?>
331: &lt;?/ul>
332:
333: &lt;?/body>
334: &lt;?/html>
335: </textarea>
336:         <p>
337:             <strong>Note:</strong> You'll notice that in the example above
we are
338:             using PHP's alternative syntax. If you are not familiar with it
you
339:             can read about it <a href="alternative_php.html">here</a>.
340:
341:         </p>
```


drivers.html

```
59:                 <td id="searchbox"><form method="get"
60:                   action="http://www.google.com/search">
61:                   <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                   value="codeigniter.com/user_guide/" />S
earch User Guide<input
63:                   type="text" class="input" style="width:
200px;" name="q" id="q"
64:                   size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                   class="submit" name="sa" value="Go" />
66:                 </form></td>
67:             </tr>
68:         </table>
69:         <!-- END BREADCRUMB -->
70:
71:         <br clear="all" />
72:
73:
74:         <!-- START CONTENT -->
75:         <div id="content">
76:
77:             <h1>Using CodeIgniter Drivers</h1>
78:
79:
80:             <p>
81:                 Drivers are a special type of Library that has a parent class a
nd any
82:                 number of potential child classes. Child classes have access to
the
83:                 parent class, but not their siblings. Drivers provide an elegan
t
84:                 syntax in your <a href="controllers.html">controllers</a> for
85:                 libraries that benefit from or require being broken down into
86:                 discrete classes.
87:             </p>
88:
89:             <p>
90:                 Drivers are found in the
91:                 <dfn>system/libraries</dfn>
92:                 folder, in their own folder which is identically named to the p
arent
93:                 library class. Also inside that folder is a subfolder named
94:                 <kbd>drivers</kbd>
95:                 , which contains all of the possible child class files.
96:             </p>
97:
98:             <p>To use a driver you will initialize it within a controller
99:                 using the following initialization function:</p>
100:
101:             <code>
102:                 $this->load->driver('
103:                 <var>class name</var>
104:                 ');
105:             </code>
106:
107:             <p>
108:                 Where
109:                 <var>class name</var>
110:                 is the name of the driver class you want to invoke. For example
, to
111:                 load a driver named "Some Parent" you would do this:
112:             </p>
```


profiling.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Profiling Your Application : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12:      href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype_lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28:     <!-- START NAVIGATION -->
29:     <div id="nav">
30:         <div id="nav_inner">
31:             <script type="text/javascript">create_menu('../');</script>
32:         </div>
33:     </div>
34:     <div id="nav2">
35:         <a name="top"></a><a href="javascript:void(0);"
36:            onclick="myHeight.toggle();"></a>
40:     </div>
41:     <div id="masthead">
42:         <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43:             <tr>
44:                 <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45:                 <td id="breadcrumb_right"><a href="../toc.html">Table of
46:
47:                     Contents Page</a></td>
48:             </tr>
49:         </table>
50:     </div>
51:     <!-- END NAVIGATION -->
52:
53:     <!-- START BREADCRUMB -->
54:     <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
55:         <tr>
56:             <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgniter
57:
58:             Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="../index.html">User Guide
59:
60:             Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;Profiling Your App
61:             lication</td>
62:         </tr>
63:     </table>
64:
65: </body>
66: </html>
```


profiling.html

```
al
170:         execution time</td>
171:     <td class="td">TRUE</td>
172: </tr>
173: <tr>
174:     <td class="td"><strong>config</strong></td>
175:     <td class="td">CodeIgniter Config variables</td>
176:     <td class="td">TRUE</td>
177: </tr>
178: <tr>
179:     <td class="td"><strong>controller_info</strong></td>
180:     <td class="td">The Controller class and method requeste
d</td>
181:     <td class="td">TRUE</td>
182: </tr>
183: <tr>
184:     <td class="td"><strong>get</strong></td>
185:     <td class="td">Any GET data passed in the request</td>
186:     <td class="td">TRUE</td>
187: </tr>
188: <tr>
189:     <td class="td"><strong>http_headers</strong></td>
190:     <td class="td">The HTTP headers for the current request
</td>
191:     <td class="td">TRUE</td>
192: </tr>
193: <tr>
194:     <td class="td"><strong>memory_usage</strong></td>
195:     <td class="td">Amount of memory consumed by the current
196:         request, in bytes</td>
197:     <td class="td">TRUE</td>
198: </tr>
199: <tr>
200:     <td class="td"><strong>post</strong></td>
201:     <td class="td">Any POST data passed in the request</td>
202:     <td class="td">TRUE</td>
203: </tr>
204: <tr>
205:     <td class="td"><strong>queries</strong></td>
206:     <td class="td">Listing of all database queries executed
,
207:         including execution time</td>
208:     <td class="td">TRUE</td>
209: </tr>
210: <tr>
211:     <td class="td"><strong>uri_string</strong></td>
212:     <td class="td">The URI of the current request</td>
213:     <td class="td">TRUE</td>
214: </tr>
215: <tr>
216:     <td class="td"><strong>query_toggle_count</strong></td>
217:     <td class="td">The number of queries after which the qu
ery
218:         block will default to hidden.</td>
219:     <td class="td">25</td>
220: </tr>
221: </table>
222:
223:
224: </div>
225: <!-- END CONTENT -->
226:
227:
```


ancillary_classes.html

```
59:                 <td id="searchbox"><form method="get"
60:                   action="http://www.google.com/search">
61:                     <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                       value="codeigniter.com/user_guide/" />S
earch User Guide<input
63:                       type="text" class="input" style="width:
200px;" name="q" id="q"
64:                       size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                       class="submit" name="sa" value="Go" />
66:                 </form></td>
67:             </tr>
68:         </table>
69:         <!-- END BREADCRUMB -->
70:
71:         <br clear="all" />
72:
73:
74:         <!-- START CONTENT -->
75:         <div id="content">
76:
77:             <h1>Creating Ancillary Classes</h1>
78:
79:             <p>In some cases you may want to develop classes that exist apart
80:               from your controllers but have the ability to utilize all of
81:               CodeIgniter's resources. This is easily possible as you'll see.
</p>
82:
83:             <h2>get_instance()</h2>
84:
85:
86:             <p>
87:                 <strong>Any class that you instantiate within your
88:                 controller functions can access CodeIgniter's native re
sources</strong>
89:                 simply by using the
90:                 <code>get_instance()</code>
91:                 function. This function returns the main CodeIgniter object.
92:             </p>
93:
94:             <p>
95:                 Normally, to call any of the available CodeIgniter functions re
quires
96:                 you to use the
97:                 <code>$this</code>
98:                 construct:
99:             </p>
100:
101:             <code>
102:                 <strong>$this</strong>->load->helper('url');<br /> <strong>$thi
s</strong>->load->library('session');<br />
103:                 <strong>$this</strong>->config->item('base_url');<br /> etc.
104:             </code>
105:
106:             <p>
107:                 <code>$this</code>
108:                 , however, only works within your controllers, your models, or
your
109:                 views. If you would like to use CodeIgniter's classes from with
in
110:                 your own custom classes you can do so as follows:
111:             </p>
```


ancillary_classes.html

165: </html>

features.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml
html1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>CodeIgniter Features : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12:     href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype-lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28:     <!-- START NAVIGATION -->
29:     <div id="nav">
30:         <div id="nav_inner">
31:             <script type="text/javascript">create_menu('../');</script>
32:         </div>
33:     </div>
34:     <div id="nav2">
35:         <a name="top"></a><a href="javascript:void(0);"
36:             onclick="myHeight.toggle();"></a>
40:     </div>
41:     <div id="masthead">
42:         <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43:             <tr>
44:                 <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45:                 <td id="breadcrumb_right"><a href="../toc.html">Table of
46:
47:                                     Contents Page</a></td>
48:             </tr>
49:         </table>
50:     </div>
51:     <!-- END NAVIGATION -->
52:
53:     <!-- START BREADCRUMB -->
54:     <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
55:         <tr>
56:             <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
er
57:                                     Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="../index.
html">User Guide
58:                                     Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<td id="searchbox"><form method="get"
59:                                     Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;Features</td>

```


at_a_glance.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>CodeIgniter at a Glance : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype.lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28:     <!-- START NAVIGATION -->
29:     <div id="nav">
30:         <div id="nav_inner">
31:             <script type="text/javascript">create_menu('../');</script>
32:         </div>
33:     </div>
34:     <div id="nav2">
35:         <a name="top"></a><a href="javascript:void(0);"
36:         onclick="myHeight.toggle();"></a>
40:     </div>
41:     <div id="masthead">
42:         <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43:             <tr>
44:                 <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45:                 <td id="breadcrumb_right"><a href="../toc.html">Table of
46:
47:                                     Contents Page</a></td>
48:             </tr>
49:         </table>
50:     </div>
51:     <!-- END NAVIGATION -->
52:
53:     <!-- START BREADCRUMB -->
54:     <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
55:         <tr>
56:             <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgniter
57:
58:                                     Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="../index.html">User Guide
59:
60:                                     Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;What is CodeIgniter?</td>
61:         </tr>
62:     </table>
63: </body>
64: </html>
```


at_a_glance.html

```
109: <h2>CodeIgniter Uses M-V-C</h2>
110: <p>CodeIgniter uses the Model-View-Controller approach, which
111:     allows great separation between logic and presentation. This is
112:     particularly good for projects in which designers are working w
ith
113:     your template files, as the code these file contain will be
114:     minimized. We describe MVC in more detail on its own page.</p>
115:
116: <h2>CodeIgniter Generates Clean URLs</h2>
117: <p>The URLs generated by CodeIgniter are clean and search-engine
118:     friendly. Rather than using the standard "query string" approac
h to
119:     URLs that is synonymous with dynamic systems, CodeIgniter uses
a
120:     segment-based approach:</p>
121:
122: <code>
123:     example.com/
124:     <var>news</var>
125:     /
126:     <dfn>article</dfn>
127:     /
128:     <samp>345</samp>
129: </code>
130:
131: <p>Note: By default the index.php file is included in the URL but
132:     it can be removed using a simple .htaccess file.</p>
133:
134: <h2>CodeIgniter Packs a Punch</h2>
135: <p>CodeIgniter comes with full-range of libraries that enable the
136:     most commonly needed web development tasks, like accessing a
137:     database, sending email, validating form data, maintaining sess
ions,
138:     manipulating images, working with XML-RPC data and much more.</
p>
139:
140: <h2>CodeIgniter is Extensible</h2>
141: <p>The system can be easily extended through the use of your own
142:     libraries, helpers, or through class extensions or system hooks
.</p>
143:
144:
145: <h2>CodeIgniter Does Not Require a Template Engine</h2>
146: <p>
147:     Although CodeIgniter <em>does</em> come with a simple template
parser
148:     that can be optionally used, it does not force you to use one.
149:
150:     Template engines simply can not match the performance of native
PHP,
151:     and the syntax that must be learned to use a template engine is
152:     usually only marginally easier than learning the basics of PHP.
153:     Consider this block of PHP code:
154: </p>
155:
156: <code>
157:     &lt;ul><br /> <br /> &lt;?php foreach ($addressbook as $name):?
><br />
158:     <br /> &lt;li>&lt;?=$name?&lt;/li><br /> <br /> &lt;?php
159:     endforeach; ?><br /> <br /> &lt;/ul>
160: </code>
161:
162: <p>Contrast this with the pseudo-code used by a template engine:</p>
```


at_a_glance.html

```
212: Copyright © 2014 - 2015 &nbsp;&middot;&nbsp;&nbsp; <a
213: href="http://bcit.ca/">British Columbia Institute of Te
chnology</a>
214: </p>
215: </div>
216:
217: </body>
218: </html>
```

goals.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Design and Architectural Goals : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28: <!-- START NAVIGATION -->
29: <div id="nav">
30: <div id="nav_inner">
31: <script type="text/javascript">create_menu('../');</script>
32: </div>
33: </div>
34: <div id="nav2">
35: <a name="top"></a><a href="javascript:void(0);"
36: onclick="myHeight.toggle();"></a>
40: </div>
41: <div id="masthead">
42: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43: <tr>
44: <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45: <td id="breadcrumb_right"><a href="../toc.html">Table o
46: Contents Page</a></td>
47: </tr>
48: </table>
49: </div>
50: <!-- END NAVIGATION -->
51:
52: <!-- START BREADCRUMB -->
53: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54: <tr>
55: <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
56: er
57: <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="../index.
58: html">User Guide
59: <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="http://codeigniter.com/">Goals</td>
60: <td id="searchbox"><form method="get"
```

goals.html

```
60:                                     action="http://www.google.com/search">
61:                                     <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                                     value="codeigniter.com/user_guide/" />S
earch User Guide<input
63:                                     type="text" class="input" style="width:
200px;" name="q" id="q"
64:                                     size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                                     class="submit" name="sa" value="Go" />
66:                                     </form></td>
67:                                     </tr>
68:     </table>
69:     <!-- END BREADCRUMB -->
70:
71:     <br clear="all" />
72:
73:
74:     <!-- START CONTENT -->
75:     <div id="content">
76:
77:
78:
79:         <h2>Design and Architectural Goals</h2>
80:
81:         <p>
82:             Our goal for CodeIgniter is
83:             <dfn>maximum performance, capability, and flexibility in the
84:             smallest, lightest possible package</dfn>
85:             .
86:         </p>
87:
88:         <p>To meet this goal we are committed to benchmarking,
89:             re-factoring, and simplifying at every step of the development
90:             process, rejecting anything that doesn't further the stated
91:             objective.</p>
92:
93:         <p>From a technical and architectural standpoint, CodeIgniter was
94:             created with the following objectives:</p>
95:
96:         <ul>
97:             <li><strong>Dynamic Instantiation.</strong> In CodeIgniter,
98:                 components are loaded and routines executed only when r
equested,
99:                 rather than globally. No assumptions are made by the sy
stem
100:                 regarding what may be needed beyond the minimal core re
sources, so
101:                 the system is very light-weight by default. The events,
as triggered
102:                 by the HTTP request, and the controllers and views you
design will
103:                 determine what is invoked.</li>
104:             <li><strong>Loose Coupling.</strong> Coupling is the degree to
105:                 which components of a system rely on each other. The le
ss components
106:                 depend on each other the more reusable and flexible the
system
107:                 becomes. Our goal was a very loosely coupled system.</l
i>
108:             <li><strong>Component Singularity.</strong> Singularity is the
109:                 degree to which components have a narrowly focused purp
ose. In
```


cheatsheets.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>CodeIgniter Cheatsheets : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28: <!-- START NAVIGATION -->
29: <div id="nav">
30: <div id="nav_inner">
31: <script type="text/javascript">create_menu('../');</script>
32: </div>
33: </div>
34: <div id="nav2">
35: <a name="top"></a><a href="javascript:void(0);"
36: onclick="myHeight.toggle();"></a>
40: </div>
41: <div id="masthead">
42: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43: <tr>
44: <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45: <td id="breadcrumb_right"><a href="../toc.html">Table of
46: Contents Page</a></td>
47: </tr>
48: </table>
49: </div>
50: <!-- END NAVIGATION -->
51:
52: <!-- START BREADCRUMB -->
53: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54: <tr>
55: <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
56: er
57: <a href="http://codeigniter.com/">CodeIgniter User Guide
58: <a href="http://codeigniter.com/">CodeIgniter Cheats
59: sheets</td>
```


cheatsheets.html

```
109: Copyright &#169; 2006 - 2014 &nbsp;&#x2014;&nbsp;&nbsp; <a  
110: href="http://ellislab.com/">EllisLab, Inc.</a> &nbsp;&#x2014;&nbsp;&nbsp;  
iddot;&nbsp;&nbsp; &nbsp;&nbsp;  
111: Copyright &#169; 2014 - 2015 &nbsp;&#x2014;&nbsp;&nbsp; <a  
112: href="http://bcit.ca/">British Columbia Institute of Te  
chnology</a>  
113: </p>  
114: </div>  
115:  
116: </body>  
117: </html>
```


mvc.html

```
60:                                     action="http://www.google.com/search">
61:                                     <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                                     value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;  <input
63:                                     type="text" class="input" style="width:
200px;" name="q" id="q"
64:                                     size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                                     class="submit" name="sa" value="Go" />
66:                                     </form></td>
67:                                     </tr>
68:     </table>
69:     <!-- END BREADCRUMB -->
70:
71:     <br clear="all" />
72:
73:
74:     <!-- START CONTENT -->
75:     <div id="content">
76:
77:
78:         <h1>Model-View-Controller</h1>
79:
80:         <p>CodeIgniter is based on the Model-View-Controller development
81:             pattern. MVC is a software approach that separates application
logic
82:             from presentation. In practice, it permits your web pages to co
ntrain
83:             minimal scripting since the presentation is separate from the P
HP
84:             scripting.</p>
85:
86:         <ul>
87:             <li>The <strong>Model</strong> represents your data structures.
88:                 Typically your model classes will contain functions tha
t help you
89:                 retrieve, insert, and update information in your databa
se.
90:             </li>
91:             <li>The <strong>View</strong> is the information that is being
92:                 presented to a user. A View will normally be a web page
, but in
93:                 CodeIgniter, a view can also be a page fragment like a
header or
94:                 footer. It can also be an RSS page, or any other type o
f "page".
95:             </li>
96:             <li>The <strong>Controller</strong> serves as an <em>intermedia
ry</em>
97:                 between the Model, the View, and any other resources ne
eded to
98:                 process the HTTP request and generate a web page.
99:             </li>
100:
101:         </ul>
102:
103:         <p>CodeIgniter has a fairly loose approach to MVC since Models are
104:             not required. If you don't need the added separation, or find t
hat
105:             maintaining models requires more complexity than you want, you
can
106:             ignore them and build your application minimally using Controll
```


appflow.html

```
60:                                                     action="http://www.google.com/search">
61:                                                     <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
62:                                                     value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp;   <input
63:                                                     type="text" class="input" style="width:
200px;" name="q" id="q"
64:                                                     size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
65:                                                     class="submit" name="sa" value="Go" />
66:                                                     </form></td>
67:                                                     </tr>
68: </table>
69: <!-- END BREADCRUMB -->
70:
71: <br clear="all" />
72:
73:
74: <!-- START CONTENT -->
75: <div id="content">
76:
77:
78:     <h1>Application Flow Chart</h1>
79:
80:     <p>The following graphic illustrates how data flows throughout the
81:         system:</p>
82:
83:     <div>
84:         
86:     </div>
87:
88:
89:     <ol>
90:         <li>The index.php serves as the front controller, initializing
91:             the base resources needed to run CodeIgniter.</li>
92:         <li>The Router examines the HTTP request to determine what
93:             should be done with it.</li>
94:         <li>If a cache file exists, it is sent directly to the browser,
95:             bypassing the normal system execution.</li>
96:         <li>Security. Before the application controller is loaded, the
97:             HTTP request and any user submitted data is filtered fo
r security.</li>
98:         <li>The Controller loads the model, core libraries, helpers, an
d
99:             any other resources needed to process the specific requ
est.</li>
100:        <li>The finalized View is rendered then sent to the web browser
101:            to be seen. If caching is enabled, the view is cached f
irst so that
102:            on subsequent requests it can be served.</li>
103:        </ol>
104:
105:
106:
107:
108:     </div>
109:     <!-- END CONTENT -->
110:
111:
112:     <div id="footer">
113:         <p>
114:             Previous Topic:&nbsp;  &nbsp; <a href="features.html">CodeIgniter
```


getting_started.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xh
tml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>Getting Started With CodeIgniter : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('../userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all'
12: href='../userguide.css' />
13:
14: <script type="text/javascript" src="../nav/nav.js"></script>
15: <script type="text/javascript" src="../nav/prototype_lite.js"></script>
16: <script type="text/javascript" src="../nav/moo.fx.js"></script>
17: <script type="text/javascript" src="../nav/user_guide_menu.js"></script>
18:
19: <meta http-equiv='expires' content='-1' />
20: <meta http-equiv='pragma' content='no-cache' />
21: <meta name='robots' content='all' />
22: <meta name='author' content='ExpressionEngine Dev Team' />
23: <meta name='description' content='CodeIgniter User Guide' />
24:
25: </head>
26: <body>
27:
28: <!-- START NAVIGATION -->
29: <div id="nav">
30: <div id="nav_inner">
31: <script type="text/javascript">create_menu('../');</script>
32: </div>
33: </div>
34: <div id="nav2">
35: <a name="top"></a><a href="javascript:void(0);"
36: onclick="myHeight.toggle();"></a>
40: </div>
41: <div id="masthead">
42: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
43: <tr>
44: <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
45: <td id="breadcrumb_right"><a href="../toc.html">Table o
46: Contents Page</a></td>
47: </tr>
48: </table>
49: </div>
50: <!-- END NAVIGATION -->
51:
52: <!-- START BREADCRUMB -->
53: <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54: <tr>
55: <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
56: er
57: <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="../index.
html">User Guide
58: <a href="http://codeigniter.com/">Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="#">Getting Started</t
d>
```

getting_started.html

```
59:                 <td id="searchbox"><form method="get"
60:                   action="http://www.google.com/search">
61:                   <input type="hidden" name="as_sitesearch" id="as_
s_sitesearch"
62:                   value="codeigniter.com/user_guide/" /><input
earch User Guide    <input
63:                   type="text" class="input" style="width:
200px;" name="q" id="q"
64:                   size="31" maxlength="255" value="" />&nbsp;&nbsp;
bsp;<input type="submit"
65:                   class="submit" name="sa" value="Go" />
66:                 </form></td>
67:             </tr>
68:         </table>
69:         <!-- END BREADCRUMB -->
70:
71:         <br clear="all" />
72:
73:         <!-- START CONTENT -->
74:         <div id="content">
75:
76:             <h1>Getting Started With CodeIgniter</h1>
77:
78:             <p>Any software application requires some effort to learn. We've
79:             done our best to minimize the learning curve while making the p
80:             rocess as enjoyable as possible.</p>
81:
82:             <p>
83:             The first step is to <a href="../../../installation/index.html">insta
84:             ll</a>
85:             CodeIgniter, then read all the topics in the <strong>Introducti
on</strong>
86:             section of the Table of Contents.
87:
88:             <p>
89:             Next, read each of the <strong>General Topics</strong> pages in
90:             order. Each topic builds on the previous one, and includes code
91:             examples that you are encouraged to try.
92:
93:             <p>
94:             <p>
95:             Once you understand the basics you'll be ready to explore the <
strong>Class
96:             Reference</strong> and <strong>Helper Reference</strong>
97:             > pages to learn to
98:             utilize the native libraries and helper files.
99:
100:            <p>
101:            Feel free to take advantage of our <a
102:                href="http://forum.codeigniter.com/">Community Forums</
103:            a> if you
104:            have questions or problems, and our <a
105:                href="https://github.com/bcit-ci/CodeIgniter/wiki">Wiki
106:            </a> to see
107:            code examples posted by other users.
108:
109:
110:        </div>
```


changelog.html

```
113:                                     </ul>
114:                                     </li>
115: </ul>
116:
117: <h3>Bug fixes:</h3>
118: <ul>
119:     <li>Fixed a bug (#3094) - <samp>CI_Input::_clean_input_data()</
<samp>
120:         breaks encrypted session cookies.
121:     </li>
122:     <li>Fixed a bug (#2268) - <samp>CI_Security::xss_clean()</samp>
123:         didn't properly match JavaScript events.
124:     </li>
125:     <li>Fixed a bug (#3309) - <samp>CI_Security::xss_clean()</samp>
126:         used an overly-invasive pattern to strip JS event handl
ers.
127:     </li>
128:     <li>Fixed a bug (#2771) - <samp>CI_Security::xss_clean()</samp>
129:         didn't take into account HTML5 entities.
130:     </li>
131:     <li>Fixed a bug (#73) - <samp>CI_Security::sanitize_filename()<
/samp>
132:         could be tricked by an XSS attack.
133:     </li>
134:     <li>Fixed a bug (#2681) - <samp>CI_Security::entity_decode()</s
<samp>
135:         used the PREG_REPLACE_EVAL flag, which is deprecated si
nce PHP 5.5.
136:     </li>
137:     <li>Fixed a bug (#3302) - Internal function <samp>get_config()<
/samp>
138:         triggered an E_NOTICE message on PHP 5.6.
139:     </li>
140:     <li>Fixed a bug (#2508) - <a href="libraries/config.html">Confi
g
141:         Library</a> didn't properly detect if the curre
nt request is via HTTPS.
142:     </li>
143:     <li>Fixed a bug (#3314) - SQLSRV <a href="database/index.html">
Database
144:         driver</a>'s method <samp>count_all()</samp> di
dn't escape the supplied
145:         table name.
146:     </li>
147:     <li>Fixed a bug (#3404) - MySQLi <a href="database/index.html">
Database
148:         driver</a>'s method <samp>escape_str()</samp> h
ad a wrong fallback to <samp>mysql_escape_string()</samp>
149:         when there was no active connection.
150:     </li>
151:     <li>Fixed a bug in the <a href="libraries/sessions.html">Sessio
n
152:         Library</a> where session ID regeneration occur
red during AJAX
153:         requests.
154:     </li>
155: </ul>
156:
157: <h2>Version 2.2.0</h2>
158: <p>Release Date: June 5, 2014</p>
159:
160: <ul>
161:     <li>General Changes
```

changelog.html

```
162:                                     <ul>
163:                                     <li><b>Security:</b> The <samp>xor_encode()</sa
mp> method in
164:                                     the Encrypt Class has been removed. The
Encrypt Class now requires
165:                                     the Mcrypt extension to be installed.</
li>
166:                                     <li><b>Security:</b> The <a href="libraries/ses
sions.html">Session
167:                                     Library</a> now uses HMAC authe
ntication instead of a simple MD5
168:                                     checksum.</li>
169:                                     </ul>
170:                                     </li>
171:                               </ul>
172:
173:                               <h3>Bug fixes for 2.2.0:</h3>
174:                               <ul>
175:                                   <li>Fixed an edge case (#2583) in the <a
176:                                   href="libraries/email.html">Email Library</a> where <a
177:                                   href="http://www.hardened-php.net/suhosin/">Suhosin</a>
blocked
178:                                   messages sent via <samp>mail()</samp> due to trailing n
ewspaces in
179:                                   headers.
180:                                   </li>
181:                                   <li>Fixed a bug (#696) - make <samp>oci_execute()</samp> calls
182:                                   inside <samp>num_rows()</samp> non-committing, since th
ey are only
183:                                   there to reset which row is next in line for oci_fetch
calls and
184:                                   thus don't need to be committed.
185:                                   </li>
186:                                   <li>Fixed a bug (#2689) - <a href="database/forge.html">Databas
e
187:                                   Forge Class</a> methods <samp>create_table()</s
amp>, <samp>drop_table()</samp>
188:                                   and <samp>rename_table()</samp> produced broken SQL for
189:                                   driver.
190:                                   </li>
191:                                   <li>Fixed a bug (#2427) - PDO <a href="database/index.html">Dat
abase
192:                                   driver</a> didn't properly check for query fail
ures.
193:                                   </li>
194:                                   <li>Fixed a bug in the <a href="libraries/sessions.html">Sessio
n
195:                                   Library</a> where authentication was not perfor
med for encrypted
196:                                   cookies.
197:                                   </li>
198:                               </ul>
199:
200:                               <h2>Version 2.1.4</h2>
201:                               <p>Release Date: July 8, 2013</p>
202:
203:                               <ul>
204:                                   <li>General Changes
205:                                       <ul>
206:                                           <li>Improved security in <kbd>xss_clean()</kbd>
207:                                           </li>
```

changelog.html

```
208:                                     </ul>
209:                                     </li>
210: </ul>
211:
212: <h3>Bug fixes for 2.1.4:</h3>
213: <ul>
214:     <li>Fixed a bug (#1936) - <a href="libraries/migration.html">Mi
grations
215:         Library</a> method <samp>latest()</samp> had a
typo when retrieving
216:         language values.
217:     </li>
218:     <li>Fixed a bug (#2021) - <a href="libraries/migration.html">Mi
grations
219:         Library</a> configuration file was mistakenly u
sing Windows style line
220:         feeds.
221:     </li>
222:     <li>Fixed a bug (#1273) - <a href="database/active_record.html"
>Active
223:         Record</a> method <samp>set_update_batch()</sam
p> was using the
224:         incorrect variables and would cause an error.
225:     </li>
226:     <li>Fixed a bug (#2337) - <a href="libraries/email.html">Email
Library</a> method <samp>print_debugger()</samp>
227:         > was not using <samp>htmlspecialchars()</samp>
when being shown in the browser.
228:         when being shown in the browser.
229:     </li>
230: </ul>
231:
232:
233: <h2>Version 2.1.3</h2>
234: <p>Release Date: October 8, 2012</p>
235:
236: <h3>Bug fixes for 2.1.3:</h3>
237: <ul>
238:     <li>Fixed a bug (#1543) - <a href="libraries/caching.html#file"
>File-based
239:         Caching</a> method <samp>get_metadata()</samp>
used a non-existent
240:         array key to look for the TTL value.
241:     </li>
242:     <li>Fixed a bug (#1314) - <a href="libraries/session.html">Sess
ion
243:         Library</a> method <samp>sess_destroy()</samp>
didn't destroy the
244:         userdata array.
245:     </li>
246:     <li>Fixed a bug where the <a href="libraries/profiler.html">Pro
filer
247:         Library</a> issued an E_WARNING error if <a
href="libraries/session.html">Session</a> userdata cont
248:         ains objects.
249:     </li>
250:     <li>Fixed a bug (#1699) - <a href="libraries/migration.html">Mi
gration
251:         Library</a> ignored the <samp>$config['migratio
n_path']</samp> setting.
252:     </li>
253:     <li>Fixed a bug (#227) - <a href="libraries/input.html">Input
Library</a> allowed unconditional spoofing of H
254:         TTP clients' IP
```

changelog.html

```
255: addresses through the HTTP_CLIENT_IP header.
256: </li>
257: <li>Fixed a bug (#907) - <a href="libraries/input.html">Input
258: Library</a> ignored HTTP_X_CLUSTER_CLIENT_IP an
d HTTP_X_CLIENT_IP
259: headers when checking for proxies.
260: </li>
261: <li>Fixed a bug (#940) - <samp>csrf_verify()</samp> used to set
262: the CSRF cookie while processing a POST request with no
actual POST
263: data, which resulted in validating a request that shoul
d be
264: considered invalid.
265: </li>
266: <li>Fixed a bug in the <a href="libraries/security.html">Securi
ty
267: Library</a> where a CSRF cookie was created eve
n if <samp>$config['csrf_protection']</samp>
268: is set tot FALSE.
269: </li>
270: <li>Fixed a bug (#1715) - <a href="libraries/input.html">Input
271: Library</a> triggered <samp>csrf_verify()</samp>
> on CLI requests.
272: </li>
273: </ul>
274:
275: <h2>Version 2.1.2</h2>
276: <p>Release Date: June 29, 2012</p>
277:
278: <ul>
279: <li>General Changes
280: <ul>
281: <li>Improved security in <kbd>xss_clean()</kbd>
282: </li>
283: </ul>
284: </li>
285: </ul>
286:
287: <h2>Version 2.1.1</h2>
288: <p>Release Date: June 12, 2012</p>
289:
290: <ul>
291: <li>General Changes
292: <ul>
293: <li>Fixed support for docx, xlsx files in mimes
.php.</li>
294: </ul>
295: </li>
296: <li>Libraries
297: <ul>
298: <li>Further improved MIME type detection in the
<a
299: href="libraries/file_uploading.html">File
Uploading Library</a>.
300: </li>
301: <li>Added support for IPv6 to the <a
302: href="libraries/input.html">Input Libra
ry</a>.
303: </li>
304: <li>Added support for the IP format parameter t
o the <a
305: href="libraries/form_validation.html">F
```

changelog.html

```
orm Validation Library</a>.
306:                                     </li>
307:                                     </ul>
308:                                     </li>
309:                                     <li>Helpers
310:                                     <ul>
311:                                     <li><samp>url_title()</samp> performance and ou
tput improved.
312:                                     You can now use any string as the word
delimiter. Backwards
313:                                     compatible with 'dash' or 'underscore'
as words delimiters.</li>
314:                                     </ul>
315:                                     </li>
316:                                     </ul>
317:
318:                                     <h3>Bug fixes for 2.1.1</h3>
319:                                     <ul>
320:                                     <li>Fixed a bug (#697) - A wrong array key was used in the
321:                                     Upload library to check for mime-types.</li>
322:                                     <li>Fixed a bug - form_open() compared $action against
323:                                     site_url() instead of base_url()</li>
324:                                     <li>Fixed a bug - CI_Upload::_file_mime_type() could've failed
325:                                     if mime_content_type() is used for the detection and re
turns FALSE.</li>
326:                                     <li>Fixed a bug (#538) - Windows paths were ignored when using
327:                                     the <a href="libraries/image_lib.html">Image Manipulati
on Class</a>
328:                                     to create a new file.
329:                                     </li>
330:                                     <li>Fixed a bug - When database caching was enabled,
331:                                     $this->db->query() checked the cache before binding var
iables which
332:                                     resulted in cached queries never being found.</li>
333:                                     <li>Fixed a bug - CSRF cookie value was allowed to be any
334:                                     (non-empty) string before being written to the output,
making code
335:                                     injection a risk.</li>
336:                                     <li>Fixed a bug (#726) - PDO put a 'dbname' argument in it's
337:                                     connection string regardless of the database platform i
n use, which
338:                                     made it impossible to use SQLite.</li>
339:                                     <li>Fixed a bug - CI_DB_pdo_result::num_rows() was not returnin
g
340:                                     properly value with SELECT queries, cause it was relyin
g on
341:                                     PDOStatement::rowCount().</li>
342:                                     <li>Fixed a bug (#1059) - CI_Image_lib::clear() was not
343:                                     correctly clearing all necessary object properties, nam
ely width and
344:                                     height.</li>
345:                                     </ul>
346:
347:
348:                                     <h2>Version 2.1.0</h2>
349:                                     <p>Release Date: November 14, 2011</p>
350:
351:                                     <ul>
352:                                     <li>General Changes
353:                                     <ul>
354:                                     <li>Fixed a potential parameter injection flaw
in the <a
355:                                     href="libraries/security.html">Security
```


changelog.html

```
Library</a> and
356: strengthened the XSS filter for HTML5 v
ulnerabilites.
357: </li>
358: <li>Callback validation rules can now accept pa
rameters like
359: any other validation rule.</li>
360: <li>Added html_escape() to the <a
361: href="general/common_functions.html">Co
mmon functions</a> to
362: escape HTML output for preventing XSS e
asliy.
363: </li>
364: </ul>
365: </li>
366: <li>Helpers
367: <ul>
368: <li>Added <samp>increment_string()</samp> to <a
369: href="helpers/string_helper.html">Strin
g Helper</a> to turn "foo"
370: into "foo-1" or "foo-1" into "foo-2".
371: </li>
372: <li>Altered form helper - made action on form_o
pen_multipart
373: helper function call optional. Fixes (#
65)</li>
374: <li><samp>url_title()</samp> will now trim extr
a dashes from
375: beginning and end.</li>
376: <li>Improved speed of <a href="helpers/string_h
elper.html">String
377: Helper</a>'s <b>random_string()
</b> method
378: </li>
379: </ul>
380: </li>
381: <li>Database
382: <ul>
383: <li>Added a <a href="http://www.cubrid.org/" ta
rget="_blank">CUBRID</a>
384: driver to the <a href="libraries/databa
se.html">Database
385: driver</a>. Thanks to the CUBRI
D team for supplying this patch.
386: </li>
387: <li>Added a <a
388: href="http://www.php.net/manual/en/intr
o.pdo.php" target="_blank">PDO</a>
389: driver to the <a href="libraries/databa
se.html">Database
390: driver</a>.
391: </li>
392: <li>Typecast limit and offset in the <a
393: href="database/queries.html">Database d
river</a> to integers to
394: avoid possible injection.
395: </li>
396: <li>Added additional option 'none' for the opti
onal third
397: argument for <code>$this->db->like()</code>
398: <a href="database/active_record.html">Data
base driver</a>.
```

changelog.html

```
399:
400: support to the
401:
402:
403:
404:
405:
406:
407: the <a
408: </a> to return the Row ID
409: lly.
410:
411: in your
412: <a
413: orm Validation library</a>.
414:
415: Migration
416: ying incremental updates to your
417:
418:
419: ge path.</li>
420:
421: orm Validation library</a>.
422:
423: > to the <a
424: tion library</a>, which
425:
426:
427:
428:
429:
430:
431:
432: ries/cart.html">URI
433: b>MY_URI</kdb> can override them.
434:
435: nt from
436: fferent Reactor and Core
437:
438:
439:
440:
441:
442:
443: <h3>Bug fixes for 2.1.0</h3>
444:
```

```
</li>
<li>Added <kbd>$this->db->insert_batch()</kbd>
    OCI8 (Oracle) driver.
</li>
<li>Changed <kbd>$this->cart->insert()</kbd> in
    href="libraries/cart.html">Cart library
    if a single item was inserted successfu
</li>
<li>Added support to set an optional parameter
    callback rules of validation using the
    href="libraries/form_validation.html">F
</li>
<li>Added a <a href="libraries/migration.html">
    library</a> to assist with appl
    database schema.
</li>
<li>Driver children can be located in any packa
</li>
<li>Added <samp>is_unique</samp> to the <a
    href="libraries/form_validation.html">F
</li>
<li>Added <kbd>$config['use_page_numbers']</kbd>
    href="libraries/pagination.html">Pagina
    enables real page numbers in the URI.
</li>
<li>Added TLS and SSL Encryption for SMTP.</li>
<li>Changed private functions in <a href="libra
    library</a> to protected so <kd
</li>
<li>Removed <samp>CI_CORE</samp> boolean consta
    CodeIgniter.php (there are no longer di
    versions).
</li>
</ul>
</li>
<h3>Bug fixes for 2.1.0</h3>
<ul>
```

changelog.html

```
445: <li>Fixed #378 Robots identified as regular browsers by the <a
446: href="libraries/user_agent.html">User Agent class</a>.
447: </li>
448: <li>If a config class was loaded first then a library with the
449: same name is loaded, the config would be ignored.</li>
450: <li>Fixed a bug (Reactor #19) where 1) the 404_override route
451: was being ignored in some cases, and 2) auto-loaded lib
raries were
452: not available to the 404_override controller when a con
troller
453: existed but the requested method did not.</li>
454: <li>Fixed a bug (Reactor #89) where MySQL export would fail if
455: the table had hyphens or other non alphanumeric/undersc
ore
456: characters.</li>
457: <li>Fixed a bug (#200) where MySQL queries would be malformed
458: after calling <kbd>$this->db->count_all()</kbd> then <k
db>$this->db->get()</kdb>
459: </li>
460: <li>Fixed bug #105 that stopped query errors from being logged
461: unless database debugging was enabled</li>
462: <li>Fixed a bug (#160) - Removed unneeded array copy in the fil
e
463: cache driver.</li>
464: <li>Fixed a bug (#150) - <samp>field_data()</samp> now correctl
y
465: returns column length.
466: </li>
467: <li>Fixed a bug (#8) - <samp>load_class()</samp> now looks for
468: core classes in <samp>APPPATH</samp> first, allowing th
em to be
469: replaced.
470: </li>
471: <li>Fixed a bug (#24) - ODBC database driver called incorrect
472: parent in __construct().</li>
473: <li>Fixed a bug (#85) - OCI8 (Oracle) database <kbd>escape_str(
)</kbd>
474: function did not escape correct.
475: </li>
476: <li>Fixed a bug (#344) - Using schema found in <a
477: href="libraries/sessions.html">Saving Session Data to a
Database</a>,
478: system would throw error "user_data does not have a def
ault value"
479: when deleting then creating a session.
480: </li>
481: <li>Fixed a bug (#112) - OCI8 (Oracle) driver didn't pass the
482: configured database character set when connecting.</li>
483: <li>Fixed a bug (#182) - OCI8 (Oracle) driver used to re-execut
e
484: the statement whenever num_rows() is called.</li>
485: <li>Fixed a bug (#82) - WHERE clause field names in the DB <kbd>
>update_string()</kbd>
486: method were not escaped, resulting in failed queries in
some cases.
487: </li>
488: <li>Fixed a bug (#89) - Fix a variable type mismatch in DB <kbd>
>display_error()</kbd>
489: where an array is expected, but a string could be set i
nstead.
490: </li>
491: <li>Fixed a bug (#467) - Suppress warnings generated from <kbd>
get_magic_quotes_gpc()</kbd>
```

changelog.html

```
492:                                     (deprecated in PHP 5.4)
493: </li>
494: <li>Fixed a bug (#484) - First time <code>_csrf_set_hash()</code>
495:     is called, hash is never set to the cookie (in Security
.php).
496: </li>
497: <li>Fixed a bug (#60) - Added <code>_file_mime_type()</code>
498:     method to the <a href="libraries/file_uploading.html">File
file
499:         Uploading Library</a> in order to fix a possible
e MIME-type injection
500:         (also fixes bug #394).
501: </li>
502: <li>Fixed a bug (#537) - Support for all wav type in browser.</
li>
503: <li>Fixed a bug (#576) - Using <code>ini_get()</code> function to
504:     detect if apc is enabled or not.
505: </li>
506: <li>Fixed invalid date time format in <a
507:     href="helpers/date_helper.html">Date helper</a> and <a
508:     href="libraries/xmlrpc.html">XMLRPC library</a>.
509: </li>
510: </ul>
511:
512: <h2>Version 2.0.3</h2>
513: <p>Release Date: August 20, 2011</p>
514:
515: <ul>
516:     <li>Security
517:         <ul>
518:             <li>An improvement was made to the MySQL and My
SQLi drivers to
519:                 prevent exposing a potential vector for
520:                 using multi-byte character sets in the
database client connection.
521:
522:                 <p>
523:                     An incompatibility in PHP versi
ons <code>5.2.3</code> and MySQL <code>5.
t()</code> creates a situation where
524:                 on these environments may
525:                 potentially expose a SQL inject
ion attack vector. Latin-1, UTF-8,
526:                 and other "low ASCII" character
sets are unaffected on all
527:                 environments.
528:                 <p class="critical">If you are running
529:                 a multi-byte character set for
or considering running
530:                 pay close attention to the serv
your database connection, please
531:                 on to ensure you are not vulner
er environment you are deploying
532:                 able.</p>
533:             </li>
534:         </ul>
535:     </li>
536:     <li>General Changes
537:         <ul>
538:             <li>Fixed a bug where there was a misspelling w
```

changelog.html

```
ithin a code
 539:
 540:
profiler.
 541:
on HTTP Headers, Session
 542:
 543:
<code> constant.
 544:
 545:
file and
 546:
 547:
anijelb</code> for the pull
 548:
 549:
 550:
>
 551:
<code>Email
 552:
"." in the name would using invalid
 553:
 554:
 555:
parameters like
 556:
 557:
 558:
 559:
 560:
 561:
heading()</code>
 562:
the rendered heading tag.
 563:
 564:
en (Cross-site
 565:
n the form's action is
 566:
(Reactor #165)</code>
 567:
ingular()</code>
 568:
<code>ctor_helper.html</code> Inflector
 569:
bly more words.
 570:
 571:
 572:
 573:
 574:
 575:
st the
 576:
using database sessions.</code>
 577:
 578:
orm Validation library</code>.
 579:
 580:
```

```
comment in the index.php file.</li>
<li>Added Session Class userdata to the output
    Additionally, added a show/hide toggle
    Data and Config Variables.</li>
<li>Removed internal usage of the <code>EXT</code>
</li>
<li>Visual updates to the welcome_message view
    default error templates. Thanks to <a
    href="https://bitbucket.org/danijelb">d
    request.
</li>
<li>Added "application/x-csv" to mimes.php.</li>
<li>Fixed a bug where <a href="libraries/email.
    library</a> attachments with a
    MIME-types.
</li>
<li>Callback validation rules can now accept pa
    any other validation rule.</li>
<li>Added an optional third parameter to <code>
    which allows adding html attributes to
</li>
<li><code>form_open()</code> now only adds a hidd
    Reference Forgery) protection field whe
    internal and is set to the post method.
<li>Re-worked <code>plural()</code> and <code>s
    functions in the <a href="helpers/infle
    helper</a> to support considera
</li>
<li>Altered Session to use a longer match again
    user_agent string. See upgrade notes if
<li>Added <code>is_unique</code> to the <a
    href="libraries/form_validation.html">F
</li>
<li>Added <code>$this->db->set_dbprefix()</code>
```

changelog.html

```
to the <a
581:                                     href="database/queries.html">Database D
river</a>.
582:                                     </li>
583:                                     <li>Changed <kbd>$this->cart->insert()</kbd> in
the <a
584:                                     href="libraries/cart.html">Cart Library
</a> to return the Row ID
585:                                     if a single item was inserted successfu
lly.
586:                                     </li>
587:                                     <li>Added <kbd>$this->load->get_var()</kbd> to
the <a
588:                                     href="libraries/loader.html">Loader lib
rary</a> to retrieve global
589:                                     vars set with <kbd>$this->load->view()<
/kbd> and <kbd>$this->load->vars()</kbd>.
590:                                     </li>
591:                                     <li>Changed <kbd>$this->db->having()</kbd> to i
nsert quotes
592:                                     using escape() rather than escape_str()
.
593:                                     </li>
594:                                     </ul>
595:                                     </li>
596:                                     </ul>
597:
598:                                     <h3>Bug fixes for 2.0.3</h3>
599:                                     <ul>
600:                                     <li>Added ENVIRONMENT to reserved constants. (Reactor #196)</li>
>
601:                                     <li>Changed server check to ensure SCRIPT_NAME is defined.
602:                                     (Reactor #57)</li>
603:                                     <li>Removed <samp>APPPATH.'third_party'</samp> from the package
s
604:                                     autoloader to negate needless file stats if no packages
exist or if
605:                                     the developer does not load any other packages by defau
lt.
606:                                     </li>
607:                                     <li>Fixed a bug (Reactor #231) where Sessions Library database
608:                                     table example SQL did not contain an index on last_acti
vity. See <a
609:                                     href="installation/upgrade_203.html">Upgrade Notes</a>.
610:                                     </li>
611:                                     <li>Fixed a bug (Reactor #229) where the Sessions Library
612:                                     example SQL in the documentation contained incorrect SQ
L.</li>
613:                                     <li>Fixed a bug (Core #340) where when passing in the second
614:                                     parameter to $this->db->select(), column names in subse
quent queries
615:                                     would not be properly escaped.</li>
616:                                     <li>Fixed issue #199 - Attributes passed as string does not
617:                                     include a space between it and the opening tag.</li>
618:                                     <li>Fixed a bug where the method <kbd>$this->cart->total_items(
)</kbd>
619:                                     from <a href="libraries/cart.html">Cart Library</a> now
returns the
620:                                     sum of the quantity of all items in the cart instead of
your total
621:                                     count.
622:                                     </li>
623:                                     <li>Fixed a bug where not setting 'null' when adding fields in
```

changelog.html

```
624: db_forge for mysql and mysqli drivers would default to
NULL instead
625: of NOT NULL as the docs suggest.</li>
626: <li>Fixed a bug where using <code>
627: $this->db->select_max()
628: </code>
629: '
630: <code>
631: $this->db->select_min()
632: </code>
633: , etc could throw notices. Thanks to w4
3l for the patch.</li>
634: <li>Replace checks for STDIN with php_sapi_name() == 'cli' which
h
635: on the whole is more reliable. This should get parameters
rs in crontab
636: working.</li>
637: </ul>
638:
639: <h2>Version 2.0.2</h2>
640: <p>
641: Release Date: April 7, 2011<br /> Hg Tag: v2.0.2
642: </p>
643:
644: <ul>
645: <li>General changes
646: <ul>
647: <li>The <a href="./libraries/security.html">Security
urity
648: library</a> was moved to the core and is now loaded automatically.
649: Please remove your loading calls.
650: </li>
651: <li>The CI_SHA class is now deprecated. All supported versions
ported versions
652: of PHP provide a <code>sha1()</code> function.
653: </li>
654: <li><code>constants.php</code> will now be loaded
from the
655: environment folder if available.</li>
656: <li>Added language key error logging</li>
657: <li>Made Environment Support optional. Comment
out or delete
658: the constant to stop environment checks
.</li>
659: <li>Added Environment Support for Hooks.</li>
660: <li>Added CI_Prefix to the <a href="libraries/
caching.html">Cache
661: driver</a>.
662: </li>
663: <li>Added <a href="./general/cli.html">CLI usage
e</a>
664: documentation.
665: </li>
666: </ul>
667: </li>
668: <li>Helpers
669: <ul>
670: <li>Removed the previously deprecated <code>doha
sh()</code> from
671: the <a href="./helpers/security_helper.
html">Security helper</a>;
```

changelog.html

```
672:                                     use do_hash() instead.
673:                                     </li>
674:                                     <li>Changed the 'plural' function so that it do
esn't ruin the                                     capitalization of your string. It also t
675:                                     ake into consideration acronyms which are all caps.</li>
676:                                     </ul>
677:                                     </li>
678:                                     <li>Database
679:                                     <ul>
680:                                     <li><code>$this->db->count_all_results()</code> w
ill now return                                     an integer instead of a string.</li>
681:                                     </ul>
682:                                     </li>
683:                                     </ul>
684:                                     </li>
685:                                     </ul>
686:                                     <h3>Bug fixes for 2.0.2</h3>
687:                                     <ul>
688:                                     <li>Fixed a bug (Reactor #145) where the Output Library had
689:                                     parse_exec_vars set to protected.</li>
690:                                     <li>Fixed a bug (Reactor #80) where is_really_writable would
691:                                     create an empty file when on Windows or with safe_mode
692:                                     enabled.</li>
693:                                     <li>Fixed various bugs with User Guide.</li>
694:                                     <li>Added is_cli_request() method to documentation for <a
695:                                     href="libraries/input.html">Input class</a>.
696:                                     </li>
697:                                     <li>Added form_validation_lang entries for <code>decimal</code>,
<code>less_than</code>
698:                                     and <code>greater_than</code>.
699:                                     </li>
700:                                     <li><a
701:                                     href="https://bitbucket.org/ellislab/codeigniter-reacto
r/issue/153/escape-str-bug-in-mssql-driver">Fixed
702:                                     issue #153</a> Escape Str Bug in MSSQL driver.<
/li>
703:                                     <li><a
704:                                     href="https://bitbucket.org/ellislab/codeigniter-reacto
r/issue/172/bug-in-chrome-and-form_open-in-201">Fixed
705:                                     issue #172</a> Google Chrome 11 posts incorrect
ly when action is empty.</li>
706:                                     </ul>
707:                                     </li>
708:                                     <h2>Version 2.0.1</h2>
709:                                     <p>
710:                                     <p>
711:                                     Release Date: March 15, 2011<br /> Hg Tag: v2.0.1
712:                                     </p>
713:                                     </p>
714:                                     <ul>
715:                                     <li>General changes
716:                                     <ul>
717:                                     <li>Added <code>$config['cookie_secure']</code> t
o the config
718:                                     file to allow requiring a secure (HTTPS
) in order to set cookies.
719:                                     </li>
720:                                     <li>Added the constant <code>CI_CORE</code> to he
lp
721:                                     differentiate between Core: TRUE and Re
actor: FALSE.
```


changelog.html

```
722:
723: index.php,
724: ngs, and optionally, which
725: ow). Read more on the <a
726: ng Environments</a> page.
727:
728:
729: s">environment-specific</a>
730:
731:
732:                                     </ul>
733:                                     </li>
734:                                     <li>Libraries
735:                                     <ul>
736: <li>Added <kbd>decimal</kbd>, <kbd>less_than</k
737: rules to the <a href="libraries/form_va
738: validation.html">Form
739:
740: validation.html">Form
741: /a> methods <kbd>post()</kbd>
742: full array if the first
743:
744: d>set_cookie()</kbd>
745: l">Input Class</a> method.
746:
747: ss</a> to set the output <kbd>Content-Type</kbd>
748: onfig/mimes.php array key.
749:
750: s</a> will now
751:
752:                                     </ul>
753:                                     </li>
754:                                     <li>Helpers
755:                                     <ul>
756: > in <a
757: lper</a>. If no value is
758: L.
759:
760:                                     </ul>
761:                                     </li>
762:                                     </ul>
763:
764: <h3>Bug fixes for 2.0.1</h3>
765: <ul>
766: <li>CLI requests can now be run from any folder, not just when
767: CD'ed next to index.php.</li>
```

changelog.html

```

768:         <li>Fixed issue #41: Added audio/mp3 mime type to mp3.</li>
769:         <li>Fixed a bug (Core #329) where the file caching driver
770:             referenced the incorrect cache directory.</li>
771:         <li>Fixed a bug (Reactor #69) where the SHA1 library was named
772:             incorrectly.</li>
773:     </ul>
774:
775:     <h2>Version 2.0.0</h2>
776:     <p>
777:         Release Date: January 28, 2011<br /> Hg Tag: v2.0.0
778:     </p>
779:
780:     <ul>
781:         <li>General changes
782:             <ul>
783:                 <li>PHP 4 support is removed. CodeIgniter now r
equires PHP
784:                     5.1.6.</li>
785:                 <li>Scaffolding, having been deprecated for a n
umber of
786:                     versions, has been removed.</li>
787:                 <li>Plugins have been removed, in favor of Help
ers. The
788:                     CAPTCHA plugin has been converted to a
Helper and <a
789:                     href="./helpers/captcha_helper.html">do
cumented</a>. The
790:                     JavaScript calendar plugin was removed
due to the ready
791:                     availability of great JavaScript calend
ars, particularly with
792:                     jQuery.
793:                 </li>
794:                 <li>Added new special Library type: <a
795:                     href="./general/drivers.html">Drivers</
a>.
796:                 </li>
797:                 <li>Added full query-string support. See the co
nfig file for
798:                     details.</li>
799:                 <li>Moved the application folder outside of the
system folder.</li>
800:                 <li>Moved system/cache and system/logs director
ies to the
801:                     application directory.</li>
802:                 <li>Added routing overrides to the main index.p
hp file,
803:                     enabling the normal routing to be overr
idden on a per "index" file
804:                     basis.</li>
805:                 <li>Added the ability to set config values (or
override config
806:                     values) directly from data set in the m
ain index.php file. This
807:                     allows a single application to be used
with multiple front
808:                     controllers, each having its own config
values.</li>
809:                 <li>Added <code>$config['directory_trigger']</code>
d> to the
810:                     config file so that a controller sub-di
rectory can be specified
811:                     when running _GET strings instead of UR

```

changelog.html

I segments.

812:

813:

ific paths

814:

uld try to look first for a

815:

n of sub-applications with

816:

es, etc. in a single

817:

ibraries/loader.html">Loader

818:

e details.

819:

820:

821:

eigniter-reactor/">BitBucket.

822:

823:

i>

824:

825:

826:

827:

oad library

828:

ension, the extension will

829:

of the given name.

830:

orge the

831:

->modify_column()'s 2nd

832:

833:

834:

by default and

835:

836:

bility with <kbd>config['uri_protocol']

837:

838:

839:

840:

841:

842:

html">Cache driver

843:

ased support.

844:

845:

fix</var> and <var>\$first_url</var>

846:

ies/pagination.html">Pagination

847:

848:

849:

, previous, next,

850:

their values to FALSE in the <a

851:

ml">Pagination library.

Added ability to set "Package" paths - spec

where the Loader and Config classes sho

requested file. This allows distributio

their own libraries, models, config fil

"package" directory. See the <a href="1

class documentation for mor

In-development code is now hosted at <a href="http://bitbucket.org/ellislab/cod

Removed the deprecated Validation Class.</l

Added CI_Prefix to all core classes.

Package paths can now be set in application/config/autoload.php.

Upl

file_name can now be set without an ext

be taken from the uploaded file instead

In Database F

name can be omitted from \$this->dbforge

param if you aren't changing the name.

<kbd>\$config['base_url']</kbd> is now empty

will guess what it should be.

Enabled full Command Line Interface compati

= 'CLI';</kbd>.

Libraries

Added a <a href="libraries/caching.

with APC, memcached, and file-b

Added <var>\$prefix</var>, <var>\$suf

properties to <a href="./librar

library.

Added the ability to suppress first

last, and page links by setting

href="./libraries/pagination.ht

changelog.html

```
852:
853:
.html">Security
854:
contains the <dfn>xss_clean</dfn> function, <dfn>filename_security</dfn>
855:
ated functions.
856:
857:
rgery) protection to
858:
ty.html">Security library</a>.
859:
860:
roperty to Output
861:
862:
863:
ndividual sections of
864:
html">Profiler</a>
865:
866:
g['allowed_types']
867:
ref="./libraries/file_uploading.html">File
868:
869:
870:
o the XML-RPC Server
871:
the object to look for requested
872:
is in the $CI superobject.</li>
873:
unit tests capable of
874:
875:
y cell with a blank
876:
877:
for individual cells
878:
879:
ethod to the <a
880:
rser Class</a>.
881:
882:
mation to the <a
883:
rofiler</a> output.
884:
885:
of detectable browsers
886:
<a
887:
">User Agent Class</a>.
888:
889:
.html">Unit Test
890:
```

```
</li>
<li>Added <a href="./libraries/security
library</a>, which now
function and other security rel
</li>
<li>Added CSRF (Cross-site Reference Fo
the <a href="./libraries/securi
</li>
<li>Added <var>$parse_exec_vars</var> p
library.
</li>
<li>Added ability to enable / disable i
the <a href="general/profiling.
</li>
<li>Added a wildcard option <kbd>$confi
= '*'</kbd> to the <a h
Uploading Class</a>.
</li>
<li>Added an 'object' config variable t
library so that one can specify
methods, instead of assuming it
<li>Added "is_object" into the list of
being run.</li>
<li>Table library will generate an empt
string, or NULL value.</li>
<li>Added ability to set tag attributes
in the Table library</li>
<li>Added a <kbd>parse_string()</kbd> m
href="libraries/parser.html">Pa
</li>
<li>Added HTTP headers and Config infor
href="general/profiling.html">P
</li>
<li>Added Chrome and Flock to the list
by <kbd>browser()</kbd> in the
href="libraries/user_agent.html
</li>
<li>The <a href="libraries/unit_testing
Class</a> now has an op
```

changelog.html

tional "notes" field available to it, and
891: allows for discrete display of
test result items using `<kbd>$this->unit->set_test_items()`.

892: ``
893: `Added a <kbd>$xss_clean class`
variable to the
894: XMLRPC library, enabling contro
l over the use of the Security
895: library's `<kbd>xss_clean()`

`> method.`
896: ``
897: `Added a <kbd>download() metho`
d to the `<a`
898: `href="libraries/ftp.html">FTP l`
ibrary``
899: `Changed <kbd>do_xss_clean() t`
o return FALSE if the
900: uploaded file fails XSS checks.
901: ``
902: `Added stripslashes() and trim()ing`
of double quotes from
903: `$_FILES` type value to standardi
ze input in Upload library.``
904: `Added a second parameter (boolean)`
to `<kbd>$this->zip->read_dir('/path/to/directory'`,
905: `FALSE)` to remove
the preceding trail of empty folders when
906: creating a Zip archive. This ex
ample would contain a zip with
907: "directory" and all of its cont
ents.
908: ``
909: `Added ability in the Image Library`
to handle PNG
910: transparency for resize operati
ons when using the GD lib.``
911: `Modified the Session class to preve`
nt use if no
912: encryption key is set in the co
nfig file.``
913: `Added a new config item to the Sess`
ion class `<kbd>sess_expire_on_close`
914: to allow sessions to auto-expir
e when the browser window is
915: closed.
916: ``
917: `Improved performance of the Encrypt`
ion library on servers
918: where Mcrypt is available.``
919: `Changed the default encryption mode`
in the Encryption
920: library to CBC.``
921: `Added an <kbd>encode_from_legacy()<`
`` method to
922: provide a way to transition enc
rypted data from CodeIgniter 1.x
923: to CodeIgniter 2.x. Please see
the `<a`
924: `href="./installation/upgrade_20`
`0.html">upgrade instructions`
925: for details.
926: ``
927: `Altered Form_Validation library to`

changelog.html

```
allow for method
928: chaining on set_rules() and
set_message() and
929: set_error_delimiters()
d functions.
930:
931:
method chaining.
932: get_request_header()
933: > to the input class.
934:
935:
nt.html>User
936: at is_browser(), is_mobile()
937: optionally check for a specific
938:
939:
940:
ml>Input library</a>
941: get() will return all
942: y) if there are no parameters
943:
944:
945: </ul>
946: </li>
947: <li>Database
948: <ul>
949:
tml>database
950:
951:
<a
952:
html>database configuration</a>.
953:
954:
<a
955:
html>database configuration</a>.
956:
957:
to the <a
958: Database Utilities Class</a>.
959:
960:
nction to allow a list
961:
h functions to return version
962:
med SQL queries. Currently this
963:
QLite.
964:
able identifier
965:
med queries in the field_data()
```

changelog.html

966:
967:
968:
s variable was
969:

970:
w the filename and
971:
query.
972:
nctions: orwhere,
973:
rby, getwhere.
974:
and
975:
om the db utility drivers.
976:
ction for the Postgres
977:
978:
979:
980:
981:
982:
ers()</kbd> function
983:
er.html">text helper.
984:
985:
inserted attributes
986:
e <a
987:
>Form Helper.
988:
989:
function in favour of
990:
consistency.
991:
992:
e to <kbd>get_dir_file_info()</kbd>
993:
lper.html">File Helper. No
994:
s to encourage responsible use
995:
performance issues when used
996:
997:
998:
<bd>directory_map()</kbd>
999:
ry_helper.html">Directory
1000:
integer to specify recursion depth.
1001:
1002:
in the <a
1003:
>File Helper to return FALSE

functions.

Fixed a bug where an undefined clas
referenced in database drivers.
Modified the database errors to sho
line number of the problematic
Removed the following deprecated fu
orlike, groupby, orhaving, orde
Removed deprecated _drop_database()
_create_database() functions fr
Improved dbforge create_table() fun
driver.

Helpers

Added <kbd>convert_accented_charact
to <a href="./helpers/text_help

Added accept-charset to the list of
of <kbd>form_open()</kbd> in th
href="helpers/form_helper.html"

Deprecated the <kbd>dohash()</kbd>
<kbd>do_hash()</kbd> for naming

Non-backwards compatible change mad
in the <a href="helpers/file_he
longer recurses by default so a
(this function can cause server
without caution).

Modified the second parameter of <k
in the <a href="helpers/directo
Helper to accept an

Modified <kbd>delete_files()</kbd>
href="helpers/file_helper.html"

changelog.html

1004:
1005:
1006:
to `<kbd>byte_format()</kbd>`
1007:
`helper.html`>Number Helper to
1008:
1009:
1010:
to `<kbd>random_string()</kbd>`
1011:
`helper.html`>String Helper.
1012:
1013:
s to not prepend
1014:
already has a scheme.
1015:
1016:
n the file helper,
1017:
me() for dates.
1018:
1019:
add optional third
1020:
vascript with no script tags.
1021:
1022:
e `<a`
1023:
`l">HTML helper` will now
1024:
alt attribute if one is not
1025:
1026:
1027:
on config file, `<kbd>form_open()</kbd>`
1028:
a hidden field.
1029:
1030:
> into the `<a`
1031:
`.html">Security helper`.
1032:
1033:
`<a`
1034:
`l">Text Helper`
1035:
`<a`
1036:
`ml">Array Helper`
1037:
1038:
1039:
1040:
1041:
to `<kbd>show_404()</kbd>`
1042:
1043:
1044:

on failure.

Added an optional second parameter
in the `<a href="helpers/number_`
allow for decimal precision.

Added alpha, and sha1 string types
in the `<a href="helpers/string_`

Modified `<kbd>prep_url()</kbd>` so a
http:// if the supplied string

Modified `<kbd>get_file_info</kbd>` i
changing filectime() to filemti

Modified `<kbd>smiley_js()</kbd>` to
parameter to return only the ja

The `<kbd>img()</kbd>` function of th
`href="./helpers/html_helper.htm`
generate an empty string as an
provided.

If CSRF is enabled in the applicati
will automatically insert it as

Added `<kbd>sanitize_filename()</kbd>`
`href="./helpers/security_helper`

Added `<kbd>ellipsize()</kbd>` to the
`href="./helpers/text_helper.htm`
Added `<kbd>elements()</kbd>` to the
`href="./helpers/array_helper.ht`

Other Changes

Added an optional second parameter
to disable logging.

Updated loader to automatically app

changelog.html

ly the sub-class

1045: g classes. Class names can be

1046: " or the same prefix as the

1047: t all.

1048: eally_writable()</kbd>

1049: undreds or thousands of requests

1050:

1051:

1052: nt uses to

1053: not directories are being

1054:

1055:

is now case

1056:

1057:

1058:

ml">Reserved Names.

1059:

1060: r .exe files and

1061: ' for .tgz files to

1062:

1063:

nger compress output

1064:

if the server runs with

1065:

.

1066:

able() on each request

1067:

put caching)

1068:

d> in the <a

1069:

tput Class.

1070:

1071:

<kbd>decode()</kbd>

1072:

ries/encryption.html">Encryption

1073:

1074:

1075:

1076:

ault route with any

1077:

1078:

) function from the <a

1079:

Security Library to <a

1080:

html">common functions.

1081:

ype for MP3.

prefix as an option when loadin

prefixed with the standard "CI_

subclass prefix, or no prefix a

Increased randomness with <kbd>is_r

to avoid file collisions when h

occur at once.

Switched some DIR_WRITE_MODE consta

FILE_WRITE_MODE where files and

operated on.

<kbd>get_mime_by_extension()</kbd>

insensitive.

Added "default" to the list <a href="general/reserved_names.ht

Added 'application/x-msdownload' fo

'application/x-gzip-compressed

config/mimes.php.

Updated the output library to no lo

or send content-length headers

zlib.output_compression enabled

Eliminated a call to is_really_writ

unless it is really needed (Out

Documented <kbd>append_output()</kb

href="libraries/output.html">Ou

Documented a second argument in the

function for the <a href="libra

Class.

Documented db->close().

Updated the router to support a def

number of segments.

Moved _remove_invisible_characters(

href="libraries/security.html">

href="general/common_functions.

Added audio/mpeg3 as a valid mime t

changelog.html

```
1082:                                     </ul>
1083:                                     </li>
1084: </ul>
1085:
1086: <h3>Bug fixes for 2.0.0</h3>
1087: <ul>
1088:     <li>Fixed a bug where you could not change the User-Agent when
1089:         sending email.</li>
1090:     <li>Fixed a bug where the Output class would send incorrect
1091:         cached output for controllers implementing their own <b>
fn>_output()</b>
1092:         method.
1093:     </li>
1094:     <li>Fixed a bug where a failed query would not have a saved
1095:         query execution time causing errors in the Profiler</li>
>
1096:     <li>Fixed a bug that was writing log entries when multiple
1097:         identical helpers and plugins were loaded.</li>
1098:     <li>Fixed assorted user guide typos or examples (#10693, #8951,
1099:         #7825, #8660, #7883, #6771, #10656).</li>
1100:     <li>Fixed a language key in the profiler:
1101:         "profiler_no_memory_usage" to "profiler_no_memory".</li>
>
1102:     <li>Fixed an error in the Zip library that didn't allow
1103:         downloading on PHP 4 servers.</li>
1104:     <li>Fixed a bug in the Form Validation library where fields
1105:         passed as rule parameters were not being translated (#9
132)</li>
1106:     <li>Modified inflector helper to properly pluralize words that
1107:         end in 'ch' or 'sh'</li>
1108:     <li>Fixed a bug in xss_clean() that was not allowing hyphens in
1109:         query strings of submitted URLs.</li>
1110:     <li>Fixed bugs in get_dir_file_info() and get_file_info() in th
e
1111:         File Helper with recursion, and file paths on Windows.<
/li>
1112:     <li>Fixed a bug where Active Record override parameter would no
t
1113:         let you disable Active Record if it was enabled in your
database
1114:         config file.</li>
1115:     <li>Fixed a bug in reduce_double_slashes() in the String Helper
1116:         to properly remove duplicate leading slashes (#7585)</l
i>
1117:     <li>Fixed a bug in values_parsing() of the XML-RPC library whic
h
1118:         prevented NULL variables typed as 'string' from being h
andled
1119:         properly.</li>
1120:     <li>Fixed a bug where form_open_multipart() didn't accept string
1121:         attribute arguments (#10930).</li>
1122:     <li>Fixed a bug (#10470) where <b>get_mime_by_extension()</b>
d>
1123:         was case sensitive.</li>
1124:     <li>Fixed a bug where some error messages for the SQLite and
1125:         Oracle drivers would not display.</li>
1126:     <li>Fixed a bug where files created with the Zip Library would
1127:         result in file creation dates of 1980.</li>
1128:     <li>Fixed a bug in the Session library that would result in PHP
1129:         error when attempting to store values with objects.</li>
>
1130:     <li>Fixed a bug where extending the Controller class would
1131:         result in a fatal PHP error.</li>
```

changelog.html

```
1132: <li>Fixed a PHP Strict Standards Error in the index.php file.</li>
1133: <li>Fixed a bug where getimagesize() was being needlessly
1134:         checked on non-image files in is_allowed_type().</li>
1135: <li>Fixed a bug in the Encryption library where an empty key was
s
1136:         not triggering an error.</li>
1137: <li>Fixed a bug in the Email library where CC and BCC recipient
s
1138:         were not reset when using the clear() method (#109).</li>
1139: <li>Fixed a bug in the URL Helper where prep_url() could cause
a
1140:         PHP error on PHP versions &lt; 5.1.2.</li>
1141: <li>Added a log message in core/output if the cache directory
1142:         config value was not found.</li>
1143: <li>Fixed a bug where multiple libraries could not be loaded by
1144:         passing an array to load-&gt;library().</li>
1145: <li>Fixed a bug in the html helper where too much white space
1146:         was rendered between the src and alt tags in the img()
function.</li>
1147: <li>Fixed a bug in the profilers _compile_queries() function.</li>
1148: <li>Fixed a bug in the date helper where the DATE_ISO8601
1149:         variable was returning an incorrectly formatted date string.</li>
1150: </ul>
1151:
1152: <h2>Version 1.7.2</h2>
1153: <p>
1154:     Release Date: September 11, 2009<br /> Hg Tag: v1.7.2
1155: </p>
1156:
1157: <ul>
1158:     <li>Libraries
1159:         <ul>
1160:             <li>Added a new <a href="libraries/cart.html">Cart Class</a>.
1161:             </li>
1162:             <li>Added the ability to pass $config['file_name'] for the <a
1163:                 href="libraries/file_uploading.html">File Uploading Class</a> and
1164:                 rename the uploaded file.
1165:             </li>
1166:             <li>Changed order of listed user-agents so Safari would more
1167:                 accurately report itself. (#6844)</li>
1168:         </ul>
1169:     </li>
1170:     <li>Database
1171:         <ul>
1172:             <li>Switched from using gettype() in escape() to
1173:                 since future PHP versions might change
1174:             </li>
1175:             <li>Updated all database drivers to handle arrays in
1176:                 escape_str().</li>
1177:             <li>Added escape_like_str() method for escaping
1178:                 strings to be used in LIKE conditions</li>
1179:             <li>Updated Active Record to utilize the new LI
```

changelog.html

KE escaping

1179:

1180:

ry to keep

1181:

a long idle.

1182:

_message() for

1183:

1184:

1185:

1186:

1187:

1188:

<a

1189:

lper.

1190:

1191:

1192:

lper to accept

1193:

1194:

1195:

1196:

lper to keep track of

1197:

utation from subsequent

1198:

alidation and form helper

1199:

1200:

1201:

<a

1202:

rectory helper to allow

1203:

eturn FALSE on failure to

1204:

1205:

1206:

.html">Smiley

1207:

e fields and insert the smiley at the

1208:

1209:

1210:

1211:

1212:

1213:

1214:

1215:

_error() to

1216:

odes.

1217:

1218:

_404() to

1219:

compatible header()

1220:

1221:

1222:

mechanism.

Added reconnect() method to DB drivers to t

alive / reestablish a connection after

Modified MSSQL driver to use mssql_get_last

error messages.

Helpers

Added <code>form_multiselect()</code> to the

[Form he](helpers/form_helper.html)

Modified <code>form_hidden()</code> in the <a

[Form he](helpers/form_helper.html)

multi-dimensional arrays.

Modified <code>form_prep()</code> in the <a

[Form he](helpers/form_helper.html)

prepped fields to avoid multiple prep/m

calls which can occur when using Form V

functions to output form fields.

Modified <code>directory_map()</code> in the

[Di](helpers/directory_helper.html)

the inclusion of hidden files, and to r

read directory.

Modified the <a href="helpers/smiley_helper

helper to work with multipl

last known cursor position.

Compatible with PHP 5.3.0

Modified show

allow sending of HTTP server response c

Modified show

send 404 status code, removing non-CGI

statement from error_404.php template.

Added set_status_header() to the <a

changelog.html

1223: [Common functions](general/common_functions.html) to allow use when the output class is unavailable.

1224: Added `is_php()` to [Common functions](general/common_functions.html) to facilitate PHP version comparisons.

1225: Added 2 CodeIgniter "cheatsheets" (thanks to DesignFellow.com for this contribution).

1226: Bug fixes for 1.7.2

- 1227: Fixed assorted user guide typos or examples (#6743, #7214, #7516, #7287, #7852, #8224, #8324, #8349).
- 1228: Fixed a bug in the Form Validation library where multiple callbacks weren't working (#6110)
- 1229: doctype helper default value was missing a "1".
- 1230: Fixed a bug in the language class when outputting an error for an unfound file.
- 1231: Fixed a bug in the Calendar library where the shortname was output for "May".
- 1232: Fixed a bug with `ORIG_PATH_INFO` that was allowing URIs of just a slash through.
- 1233: Fixed a fatal error in the Oracle and ODBC drivers (#6752)
- 1234: Fixed a bug where `xml_from_result()` was checking for a nonexistent method.
- 1235: Fixed a bug where Database Forge's `add_column` and `modify_column` were not looping through when sent multiple fields.
- 1236: Fixed a bug where the File Helper was using `'/'` instead of the `DIRECTORY_SEPARATOR` constant.
- 1237: Fixed a bug to prevent PHP errors when attempting to use `sendmail` on servers that have manually disabled the `PHP popen()` function.
- 1238: Fixed a bug that would cause PHP errors in XML-RPC data if the PHP data type did not match the specified XML-RPC type.
- 1239: Fixed a bug in the XML-RPC class with parsing `dateTime.iso8601` data types.
- 1240: Fixed a case sensitive string replacement in `xss_clean()`
- 1241: Fixed a bug in `form_textarea()` where form data was not prepped correctly.
- 1242: Fixed a bug in `form_prep()` causing it to not preserve entities in the user's original input when called back into a form element
- 1243: Fixed a bug in `_protect_identifiers()` where the swap prefix (`$swap_pre`) was not being observed.
- 1244: Fixed a bug where the 400 status header sent with the 'disallowed URI characters' was not compatible with CGI environments.
- 1245: Fixed a bug in the typography class where heading tags could

changelog.html

```
1273:                                     have paragraph tags inserted when using auto_typography
( )</li>
1274:                                     </ul>
1275:
1276:                                     <h2>Version 1.7.1</h2>
1277:                                     <p>
1278:                                     Release Date: February 10, 2009<br /> Hg Tag: 1.7.1
1279:                                     </p>
1280:
1281:                                     <ul>
1282:                                     <li>Libraries
1283:                                     <ul>
1284:                                     <li>Fixed an arbitrary script execution security
y flaw (#6068)
1285:                                     in the Form Validation library (thanks
to hkk)</li>
1286:                                     <li>Changed default current page indicator in t
he Pagination
1287:                                     library to use <b>strong</b> instead o
f <b>strong</b></li>
1288:                                     <li>A "HTTP/1.1 400 Bad Request" header is now
sent when
1289:                                     disallowed characters are encountered.<
/li>
1290:                                     <li>Added <b>big</b>, <small>, <q>,
, and
1291:                                     <tt> to the Typography parser's i
nline elements.</li>
1292:                                     <li>Added more accurate error reporting for the
Email library
1293:                                     when using sendmail.</li>
1294:                                     <li>Removed a strict type check from the <code>r
otate()</code>
1295:                                     function of the <a href="libraries/imag
e_lib.html">Image
1296:                                     Manipulation Class</a>.
1297:                                     </li>
1298:                                     <li>Added enhanced error checking in file savin
g in the Image
1299:                                     library when using the GD lib.</li>
1300:                                     <li>Added an additional newline between multipa
rt email
1301:                                     headers and the MIME message text for b
etter compatibility with a
1302:                                     variety of MUAs.</li>
1303:                                     <li>Made modest improvements to efficiency and
accuracy of
1304:                                     explode_name() in the Image lib.</li>
1305:                                     </ul>
1306:                                     </li>
1307:                                     <li>Database
1308:                                     <ul>
1309:                                     <li>Added where_in to the list of expected argu
ments received
1310:                                     by <code>delete()</code>.
1311:                                     </li>
1312:                                     </ul>
1313:                                     </li>
1314:                                     <li>Helpers
1315:                                     <ul>
1316:                                     <li>Added the ability to have optgroups in <code>
>form_dropdown()</code>
1317:                                     within the <a href="helpers/form_helper
```

changelog.html

```
.html">form helper</a>.
1318:
1319:
1320:
lper</a>.
1321:
1322:
rl_title()</kbd>
1323:
">URL helper</a>.
1324:
1325:
ton()</kbd> to
1326:
helpers/form_helper.html">form
1327:
1328:
1329:
ow redirections
1330:
1331:
okie using the
1332:
ookie name doesn't exist.</li>
1333:
1334:
1335:
1336:
1337:
to help
1338:
orer.
1339:
1340:
.php for .xls
1341:
1342:
reverse proxy
1343:
FORWARDED_FOR header to to
1344:
>
1345:
etype() for
1346:
1347:
1348:
1349:
1350:
1351:
1352:
1353:
1354:
1355:
() (#5706).</li>
1356:
ugh Forge could
1357:
1358:
databases
1359:
1360:
ed a "write"
```

```
</li>
<li>Added a doctype() function to the <a
href="helpers/html_helper.html">HTML he
</li>
<li>Added ability to force lowercase for <kbd>u
in the <a href="helpers/url_helper.html
</li>
<li>Changed the default "type" of <kbd>form_but
"button" from "submit" in the <a href="
helper</a>.
</li>
<li>Changed redirect() in the URL helper to all
to URLs outside of the CI site.</li>
<li>Updated get_cookie() to try to fetch the co
global cookie prefix if the requested c
</ul>
</li>
<li>Other Changes
<ul>
<li>Improved security in <kbd>xss_clean()</kbd>
prevent attacks targeting Internet Expl
</li>
<li>Added 'application/msexcel' to config/mimes
files.</li>
<li>Added 'proxy_ips' config item to whitelist
servers from which to trust the HTTP_X_
determine the visitor's IP address.</li>
<li>Improved accuracy of Upload::is_allowed_fil
images (#6715)</li>
</ul>
</li>
</ul>
</li>
<h3>Bug fixes for 1.7.1</h3>
<ul>
<li>Database
<ul>
<li>Fixed a bug when doing 'random' on order_by
<li>Fixed a bug where adding a primary key thro
fail (#5731).</li>
<li>Fixed a bug when using DB cache on multiple
(#5737).</li>
<li>Fixed a bug where TRUNCATE was not consider
```

changelog.html

```
1361:                                     query (#6619).</li>
1362:                                     <li>Fixed a bug where csv_from_result() was che
cking for a
1363:                                     nonexistent method.</li>
1364:                                     <li>Fixed a bug _protect_identifiers() where it
was improperly
1365:                                     removing all pipe symbols from items</l
i>
1366:                                     </ul>
1367:                                     </li>
1368:                                     <li>Fixed assorted user guide typos or examples (#5998, #6093,
1369:                                     #6259, #6339, #6432, #6521).</li>
1370:                                     <li>Fixed a bug in the MySQLi driver when no port is specified<
/li>
1371:                                     <li>Fixed a bug (#5702), in which the field label was not being
1372:                                     fetched properly, when "matching" one field to another.
</li>
1373:                                     <li>Fixed a bug in which identifers were not being escaped
1374:                                     properly when reserved characters were used.</li>
1375:                                     <li>Fixed a bug with the regular expression used to protect
1376:                                     submitted paragraph tags in auto typography.</li>
1377:                                     <li>Fixed a bug where double dashes within tag attributes were
1378:                                     being converted to em dash entities.</li>
1379:                                     <li>Fixed a bug where double spaces within tag attributes were
1380:                                     being converted to non-breaking space entities.</li>
1381:                                     <li>Fixed some accuracy issues with curly quotes in
1382:                                     Typography::format_characters()</li>
1383:                                     <li>Changed a few docblock comments to reflect actual return
1384:                                     values.</li>
1385:                                     <li>Fixed a bug with high ascii characters in subject and from
1386:                                     email headers.</li>
1387:                                     <li>Fixed a bug in xss_clean() where whitespace following a
1388:                                     validated character entity would not be preserved.</li>
1389:                                     <li>Fixed a bug where HTML comments and &lt;pre> tags were
1390:                                     being parsed in Typography::auto_typography().</li>
1391:                                     <li>Fixed a bug with non-breaking space cleanup in
1392:                                     Typography::auto_typography().</li>
1393:                                     <li>Fixed a bug in database escaping where a compound statement
1394:                                     (ie: SUM()) wasn't handled correctly with database pref
ixes.</li>
1395:                                     <li>Fixed a bug when an opening quote is preceded by a paragra
ph
1396:                                     tag and immediately followed by another tag.</li>
1397:                                     <li>Fixed a bug in the Text Helper affecting some locales where
1398:                                     word_censor() would not work on words beginning or endi
ng with an
1399:                                     accented character.</li>
1400:                                     <li>Fixed a bug in the Text Helper character limiter where the
1401:                                     provided limit intersects the last word of the string.<
/li>
1402:                                     <li>Fixed a bug (#6342) with plural() in the Inflection helper
1403:                                     with words ending in "y".</li>
1404:                                     <li>Fixed bug (#6517) where Routed URI segments returned by
1405:                                     URI::rsegment() method were incorrect for the default c
ontroller.</li>
1406:                                     <li>Fixed a bug (#6706) in the Security Helper where xss_clean(
)
1407:                                     was using a deprecated second argument.</li>
1408:                                     <li>Fixed a bug in the URL helper url_title() function where
1409:                                     trailing periods were allowed at the end of a URL.</li>
1410:                                     <li>Fixed a bug (#6669) in the Email class when CRLF's are used
1411:                                     for the newline character with headers when used with t
he "mail"
```


changelog.html

```
1412:                                     protocol.</li>
1413: <li>Fixed a bug (#6500) where URI::A_filter_uri() was exit()ing
1414: an error instead of using show_error().</li>
1415: <li>Fixed a bug (#6592) in the File Helper where
1416: get_dir_file_info() where recursion was not occurring p
roperly.</li>
1417: <li>Tweaked Typography::auto_typography() for some edge-cases.<
/li>
1418: </ul>
1419:
1420:
1421: <h2>Version 1.7</h2>
1422: <p>
1423: Release Date: October 23, 2008<br /> Hg Tag: 1.7.0
1424: </p>
1425:
1426: <ul>
1427: <li>Libraries
1428: <ul>
1429: <li>Added a new <a href="libraries/form_validat
ion.html">Form
1430: Validation Class</a>. It simpli
fies setting rules and field names,
1431: supports arrays as field names, allows
groups of validation rules
1432: to be saved in a config file, and adds
some helper functions for
1433: use in view files. <strong>Please note
that the old
1434: Validation class is now depreca
ted</strong>. We will leave it in the
1435: library folder for some time so that ex
isting applications that
1436: use it will not break, but you are enco
uraged to migrate to the
1437: new version.
1438: </li>
1439: <li>Updated the <a href="libraries/sessions.htm
l">Sessions
1440: class</a> so that any custom da
ta being saved gets stored to a
1441: database rather than the session cookie
(assuming you are using a
1442: database to store session data), permit
ting much more data to be
1443: saved.
1444: </li>
1445: <li>Added the ability to store libraries in sub
directories
1446: within either the main "libraries" or t
he local application
1447: "libraries" folder. Please see the <a h
ref="libraries/loader.html">Loader
1448: class</a> for more info.
1449: </li>
1450: <li>Added the ability to assign library objects
to your own
1451: variable names when you use <kbd>$this-
>load->library()</kbd>.
1452: Please see the <a href="libraries/loade
r.html">Loader class</a>
1453: for more info.
1454: </li>
```

changelog.html

1455:
1456:
class and support for
1457:
1458:
1459:
moved it out of
1460:
ies/typography.html">Typography
1461:
1462:
1463:
ss_clean()</kbd>,
1464:
on image/file tests.
1465:
1466:
Parser
1467:
ls to the parse() function. The output
1468:
1469:
1470:
a file name
1471:
file_uploading.html">File
1472:
1473:
1474:
on to <a
1475:
ss.
1476:
1477:
">Pagination
1478:
hen the link for page one
1479:
1480:
1481:
1482:
orm Validation class so
1483:
e string lengths properly.
1484:
1485:
1486:
1487: Database
1488:
1489:
path column and
1490:
lumn. Also improved the
1491:
1492:
aped and
1493:
when adding prefixes and
1494:
1495:
ate" and
1496:
1497:

Added controller class/method info to <a
href="general/profiling.html">Profiler

multiple database connections.

Improved the "auto typography" feature and

the helper into its own <a href="librar

Class.

Improved performance and accuracy of <kbd>x

including reduction of false positives

Improved

class to allow multiple cal

of each is appended in the output.

Added <dfn>max_filename</dfn> option to set

length limit in the <a href="libraries/

Upload Class.

Added <dfn>set_status_header()</dfn> functi

href="libraries/output.html">Output cla

Modified <a href="libraries/pagination.html

class to only output the "First" link w

would not be shown.

Added support for mb_strlen in the <a
href="libraries/form_validation.html">F

that multi-byte languages will calculat

Improved Active Record class to allow full

table names: hostname.database.table.co

alias handling.
Improved how table and column names are esc

prefixed. It now honors full path names

escaping.
Added Active Record caching feature to "upd

"delete" functions.
Added removal of non-printing control chara

changelog.html

```
acters in escape_str()
1498:
HP escaping mechanisms
1499:
SQL errors, and possible
1500:
1501:
1502:
SQL database
1503:
1504:
r, based on bug
1505:
1506:
1507:
1508:
1509:
1510:
e 
1511:
lper that allow POST
1512:
. These are intended to be
1513:
1514:
orm Validation Class.
1515:
1516:
ents() to 
1517:
er.
1518:
1519:
1520:
er so that email
1521:
ked.
1522:
1523:
1524:
lper.
1525:
1526:
1527:
r helper.
1528:
1529:
m and html
1530:
1531:
1532:
lper function human_to_unix(),
1533:
1534:
1535:
Date helper
1536:
and offsets.
1537:
1538:
1539:
er to remove characters
1540:
of DB drivers that do not have native P
(mssql, oci8, odbc), to avoid potential
sources of SQL injection.

```

``

``Added port support to MySQL, MySQLi, and MS
drivers.``

``Added driver name variable in each DB drive
report #4436.``

``

``

``Helpers
``

``Added several new "setting" functions to th
`href="helpers/form_helper.html">`Form he
data to be retrieved and set into forms
used on their own, or with the new `<a`
`href="libraries/form_validation.html">`F
``

``Added `current_url()` and `uri_segme`
`href="helpers/url_helper.html">`URL help
``

``Altered `auto_link()` in the `<a`
`href="helpers/url_helper.html">`URL help
addresses with "+" included will be lin
``

``Added `meta()` function to `<a`
`href="helpers/html_helper.html">`HTML he
``

``Improved accuracy of calculations in `<a`
`href="helpers/number_helper.html">`Numbe
``

``Removed added newlines ("`\n`") from most for
helper functions.``

``Tightened up validation in the `<a`
`href="helpers/date_helper.html">`Date he
and eliminated the POSIX regex.
``

``Updated ``
to match the world's current time zones
``

``Modified `url_title()` in the `<a`
`href="helpers/url_helper.html">`URL help
and digits that are part of character e

changelog.html

```
ntities, to allow dashes,
1541:                                     underscores, and periods regardless of
the $separator, and to
1542:                                     allow uppercase characters.
1543:                                     </li>
1544:                                     <li>Added support for arbitrary attributes in a
nchor_popup()
1545:                                     of the <a href="helpers/url_helper.html"
">URL helper</a>.
1546:                                     </li>
1547:                                     </ul>
1548:                                     </li>
1549:                                     <li>Other Changes
1550:                                     <ul>
1551:                                     <li>Added <a href="./general/styleguide.html">P
HP Style
1552:                                     Guide</a> to docs.
1553:                                     </li>
1554:                                     <li>Added sanitization in <code>xss_clean()</code>
> for a
1555:                                     deprecated HTML tag that could be abuse
d in user input in Internet
1556:                                     Explorer.
1557:                                     </li>
1558:                                     <li>Added a few openxml document mime types, an
d an additional
1559:                                     mobile agent to mimes.php and user_agen
ts.php respectively.</li>
1560:                                     <li>Added a file lock check during caching, bef
ore trying to
1561:                                     write to the file.</li>
1562:                                     <li>Modified Cookie key cleaning to unset a few
troublesome
1563:                                     key names that can be present in certai
n environments, preventing
1564:                                     CI from halting execution.</li>
1565:                                     <li>Changed the output of the profiler to use s
tyle attribute
1566:                                     rather than clear, and added the id "co
deigniter_profiler" to the
1567:                                     container div.</li>
1568:                                     </ul>
1569:                                     </li>
1570:                                     </ul>
1571:
1572:                                     <h3>Bug fixes for 1.7.0</h3>
1573:                                     <ul>
1574:                                     <li>Fixed bug in <code>xss_clean()</code> that could remove some
1575:                                     desirable tag attributes.
1576:                                     </li>
1577:                                     <li>Fixed assorted user guide typos or examples (#4807, #4812,
1578:                                     #4840, #4862, #4864, #4899, #4930, #5006, #5071, #5158,
#5229,
1579:                                     #5254, #5351).</li>
1580:                                     <li>Fixed an edit from 1.6.3 that made the $robots array in
1581:                                     user_agents.php go poof.</li>
1582:                                     <li>Fixed a bug in the <a href="libraries/email.html">Email
1583:                                     library</a> with quoted-printable encoding impr
operly encoding space
1584:                                     and tab characters.
1585:                                     </li>
1586:                                     <li>Modified XSS sanitization to no longer add semicolons after
1587:                                     &[single letter], such as in M&M's, B&B, et
```

changelog.html

c.
1588: Modified XSS sanitization to no longer strip XHTML image
1589: tags of closing slashes.
1590: Fixed a bug in the Session class when database sessions are
1591: used where upon session update all userdata would be er
rantly
1592: written to the session cookie.
1593: Fixed a bug (#4536) in backups with the MySQL driver where
1594: some legacy code was causing certain characters to be d
ouble
1595: escaped.
1596: Fixed a routing bug (#4661) that occurred when the default
1597: route pointed to a subfolder.
1598: Fixed the spelling of "Dhaka" in the <code>timezone_menu()</
code>
1599: function of the Date
1600: helper.
1601: Fixed the spelling of "raspberry" in config/smileys.php.</l
i>
1602: Fixed incorrect parenthesis in <code>form_open()</code>
1603: function (#5135).
1604:
1605: Fixed a bug that was ignoring case when comparing controlle
r
1606: methods (#4560).
1607: Fixed a bug (#4615) that was not setting SMTP authorization
1608: settings when using the initialize function.
1609: Fixed a bug in <code>highlight_code()</code> in the <a
1610: href="helpers/text_helper.html">Text helper that wo
uld leave a
1611: stray <code></code> in certain cases.
1612:
1613: Fixed Oracle bug (#3306) that was preventing multiple
1614: queries in one action.
1615: Fixed ODBC bug that was ignoring connection params due to
1616: its use of a constructor.
1617: Fixed a DB driver bug with num_rows() that would cause an
1618: error with the Oracle driver.
1619: Fixed MS SQL bug (#4915). Added brackets around database
1620: name in MS SQL driver when selecting the database, in t
he event that
1621: reserved characters are used in the name.
1622: Fixed a DB caching bug (4718) in which the path was
1623: incorrect when no URI segments were present.
1624: Fixed Image_lib class bug #4562. A path was not defined for
1625: NetPBM.
1626: Fixed Image_lib class bug #4532. When cropping an image wit
h
1627: identical height/width settings on output, a copy is ma
de.
1628: Fixed DB_driver bug (4900), in which a database error was
1629: not being logged correctly.
1630: Fixed DB backup bug in which field names were not being
1631: escaped.
1632: Fixed a DB Active Record caching bug in which multiple call
s
1633: to cached data were not being honored.
1634: Fixed a bug in the Session class that was disallowing
1635: slashes in the serialized array.
1636: Fixed a Form Validation bug in which the "isset" error
1637: message was being triggered by the "required" rule.
1638: Fixed a spelling error in a Loader error message.
Fixed a bug (5050) with IP validation with empty segments.<

changelog.html

```
/li>
1639:             <li>Fixed a bug in which the parser was being greedy if multipl
e
1640:                 identical sets of tags were encountered.</li>
1641:         </ul>
1642:
1643:         <h2>Version 1.6.3</h2>
1644:         <p>
1645:             Release Date: June 26, 2008<br /> Hg Tag: v1.6.3
1646:         </p>
1647:
1648:         <p>Version 1.6.3 is a security and maintenance release and is
1649:             recommended for all users.</p>
1650:         <ul>
1651:             <li>Database
1652:                 <ul>
1653:                     <li>Modified MySQL/MySQLi Forge class to give e
xplicit names
1654:                         to keys</li>
1655:                     <li>Added ability to set multiple column non-pr
imary keys to
1656:                         the <a href="database/forge.html">Forge
class</a>
1657:                     </li>
1658:                     <li>Added ability to set additional database co
nfig values in
1659:                         <a href="database/connecting.html">DSN
connections</a> via the
1660:                         query string.
1661:                     </li>
1662:                 </ul>
1663:             </li>
1664:             <li>Libraries
1665:                 <ul>
1666:                     <li>Set the mime type check in the <a
1667:                         href="libraries/file_uploading.html">Up
load class</a> to reference
1668:                         the global mimes variable.
1669:                     </li>
1670:                     <li>Added support for query strings to the <a
1671:                         href="libraries/pagination.html">Pagina
tion class</a>,
1672:                         automatically detected or explicitly de
clared.
1673:                     </li>
1674:                     <li>Added <code>get_post()</code> to the <a
1675:                         href="libraries/input.html">Input class
</a>.
1676:                     </li>
1677:                     <li>Documented <code>get()</code> in the <a
1678:                         href="libraries/input.html">Input class
</a>.
1679:                     </li>
1680:                     <li>Added the ability to automatically output 1
language items
1681:                         as form labels in the <a href="librarie
s/language.html">Language
class</a>.
1682:                     </li>
1683:                 </ul>
1684:             </li>
1685:             <li>Helpers
1686:                 <ul>
1687:                     <li>
```

changelog.html

```
1688:
ml">Language
1689:
1690:
1691:
">Number
1692:
1693:
1694:
per</a>
1695:
bd> and <kbd>form_fieldset()</kbd>
1696:
s.</li>
1697:
1698:
1699:
1700:
1701:
.
1702:
1703:
_display_cache()</kbd>.
1704:
1705:
tml">use
1706:
for XSS, useful for upload security.
1707:
1708:
gents in
1709:
1710:
en moved above
1711:
(#4614).</li>
1712:
request
1713:
"libraries/xmlrpc.html">XMLRPC
1714:
1715:
1716:
added
1717:
1718:
lper</a>.
1719:
1720:
te_format()</kbd>
1721:
tml">number helper</a>.
1722:
1723:
1724:
1725:
1726:
1727:
1728:
1729:
1730:
1731:
1732:
```

```
<li>Added a <a href="helpers/language_helper.ht
        helper</a>.
</li>
<li>Added a <a href="helpers/number_helper.html
        helper</a>.
</li>
<li><a href="helpers/form_helper.html">Form hel
        refactored to allow <kbd>form_open()</k
        to accept arrays or strings as argument
</ul>
</li>
<li>Other changes
    <ul>
<li>Improved security in <kbd>xss_clean()</kbd>
</li>
<li>Removed an unused Router reference in <kbd>
</li>
<li>Added ability to <a href="libraries/input.h
        xss_clean() to test images</a>
</li>
<li>Considerably expanded list of mobile user-a
        config/user_agents.php.</li>
<li>Charset information in the userguide has be
        title for internationalization purposes
<li>Added &quot;Using Associative Arrays In a R
        Parameter&quot; example to the <a href=
        userguide page</a>.
</li>
<li>Removed maxlength and size as automatically
        attributes of form_input() in the <a
        href="helpers/form_helper.html">form he
</li>
<li>Documented the language file use of <kbd>by
        in the <a href="helpers/number_helper.h
</li>
</ul>
</li>
</ul>
</li>
<h3>Bug fixes for 1.6.3</h3>
<ul>
<li>Added a language key for valid_emails in
        validation_lang.php.</li>
```

changelog.html

```
1733: <li>Amended fixes for bug (#3419) with parsing DSN database
1734: connections.</li>
1735: <li>Moved the <code>_has_operators()</code> function (#4535) into
1736: DB_driver from DB_active_rec.
1737: </li>
1738: <li>Fixed a syntax error in upload_lang.php.</li>
1739: <li>Fixed a bug (#4542) with a regular expression in the Image
1740: library.</li>
1741: <li>Fixed a bug (#4561) where <code>orhaving()</code> wasn't
1742: properly passing values.
1743: </li>
1744: <li>Removed some unused variables from the code (#4563).</li>
1745: <li>Fixed a bug where <code>having()</code> was not adding an =
1746: into the statement (#4568).
1747: </li>
1748: <li>Fixed assorted user guide typos or examples (#4574, #4706).
</li>
1749: <li>Added quoted-printable headers to Email class when the
1750: multi-part override is used.</li>
1751: <li>Fixed a double opening <code>&lt;p>&gt;</code> tag in the index pages of
1752: each system directory.</li>
1753: </ul>
1754:
1755: <h2>Version 1.6.2</h2>
1756: <p>
1757: Release Date: May 13, 2008<br /> Hg Tag: 1.6.2
1758: </p>
1759: <ul>
1760: <li>Active Record
1761: <ul>
1762: <li>Added the ability to prevent escaping in <code>having()</code>
1763: clauses.
1764: </li>
1765: <li>Added <code>rename_table()</code> into <a
1766: href="./database/forge.html">DBForge</a>
>.
1767: </li>
1768: <li>Fixed a bug that wasn't allowing escaping t
o be turned off
if the value of a query was NULL.</li>
1769: <li>DB Forge is now assigned to any models that
exist after
loading (#3457).</li>
1770: </ul>
1771: </ul>
1772: </li>
1773: <li>Database
1774: <ul>
1775: <li>Added <a href="./database/transactions.html
">Strict
1776: Mode</a> to database transactio
ns.
1777: </li>
1778: <li>Escape behaviour in where() clauses has cha
nged; values in
those with the <code>'FALSE'</code> argume
nt are no longer escaped
(ie: quoted).</li>
1779: </ul>
1780: </li>
1781: <li>Config
1782: <ul>
1783: <li>Added 'application/vnd.ms-powerpoint' to li
```


changelog.html

```
st of mime
 1787:
 1788:
i>
 1789:
p containing
 1790:
 1791:
config in the <a
 1792:
lass.
 1793:
 1794:
 1795:
 1796:
 1797:
 1798:
ing in the
 1799:
 1800:
lient-side data
 1801:
 1802:
MySQL forge.</li>
 1803:
config in the <a
 1804:
lass.
 1805:
 1806:
Testing</a>
 1807:
nge was made to the default
 1808:
 1809:
class.</li>
 1810:
/a> now exits
 1811:
 1812:
 1813:
/a> has
 1814:
eed and clarity (thanks
 1815:
ntribution in bug report
 1816:
 1817:
 1818:
 1819:
 1820:
 1821:
 1822:
er.html">Compatibility
 1823:
n PHP 5 functions safely in
 1824:
rvers (thanks Seppo for the
 1825:
 1826:
 1827:
 1828:
lper</a>.
      types.</li>
<li>Added 'audio/mpg' to list of mime types.</li>
<li>Added new user-modifiable file constants.php
      file mode and fopen constants.</li>
<li>Added the ability to set CRLF settings via
      href="libraries/email.html">Email</a> c
</li>
</li>
<li>Libraries
      <ul>
<li>Added increased security for filename handl
      Upload library.</li>
<li>Added increased security for sessions for c
      tampering.</li>
<li>The MySQLi forge class is now in sync with
<li>Added the ability to set CRLF settings via
      href="libraries/email.html">Email</a> c
</li>
<li><a href="libraries/unit_testing.html">Unit
      results are now colour coded, and a cha
      template of results.</li>
<li>Added a valid_emails rule to the Validation
<li>The <a href="libraries/zip.html">Zip class<
      within <kbd>download()</kbd>.
</li>
<li>The <a href="libraries/zip.html">Zip class<
      undergone a substantial re-write for sp
      stanleyxu for the hard work and code co
      #3425!)
</li>
</li>
<li>Helpers
      <ul>
<li>Added a <a href="helpers/compatibility_help
      Helper</a> for using some commo
      applications that might run on PHP 4 se
      hard work and code contribution!)
</li>
<li>Added <kbd>form_button()</kbd> in the <a
      href="helpers/form_helper.html">Form he
```

changelog.html

```
1829:
1830:
checkbox()</kbd>
1831:
default.
1832:
1833:
TP Response
1834:
ion of the <a
1835:
er</a>.
1836:
1837:
1838:
lper</a> to remove an
1839:
1840:
1841:
1842:
er</a> to no longer add a
1843:
1844:
1845:
Download
1846:
d>force_download()</kbd>.
1847:
1848:
get_file_info()</kbd>,
1849:
to the <a
1850:
lper</a>.
1851:
1852:
<kbd>octal_permissions()</kbd>
1853:
l'>File helper</a>.
1854:
1855:
1856:
1857:
1858:
1859:
or the function
1860:
magecreate if it isn't
1861:
1862:
1863:
1864:
1865:
1866:
html">xss_clean()</a>
1867:
1868:
1869:
to avoid
1870:
modify headers' errors.</li>
1871:
oad search

</li>
<li>Changed the <kbd>radio()</kbd> and <kbd>che
functions to default to not checked by
</li>
<li>Added the ability to include an optional HT
Code in the <kbd>redirect()</kbd> funct
href="helpers/url_helper.html">URL Help
</li>
<li>Modified <kbd>img()</kbd> in the <a
href="helpers/html_helper.html">HTML He
unneeded space (#4208).
</li>
<li>Modified <kbd>anchor()</kbd> in the <a
href="helpers/url_helper.html">URL help
default title= attribute (#4209).
</li>
<li>The <a href="helpers/download_helper.html">
helper</a> now exits within <kb
</li>
<li>Added <kbd>get_dir_file_info()</kbd>, <kbd>
and <kbd>get_mime_by_extension()</kbd>
href="helpers/file_helper.html">File He
</li>
<li>Added <kbd>symbolic_permissions()</kbd> and
to the <a href='helpers/file_helper.htm
</li>
</ul>
</li>
<li>Plugins
<ul>
<li>Modified captcha generation to first look f
imagecreatetruecolor, and fallback to i
available (#4226).</li>
<li>Added ability for <a href="libraries/input.
to accept arrays.
</li>
<li>Removed closing PHP tags from all PHP files
accidental output and potential 'cannot
<li>Removed &quot;scripts&quot; from the auto-l
```

changelog.html

1872: path. Scripts were deprecated in Version 1.4.1 (September 21, 2006). If you still need to use them for legacy reasons, they must now be manually loaded in each Controller.

- 1873: Added a [Reserved Names](general/reserved_names.html) page to the userguide, and migrated reserved controller names into it.
- 1874: Added a [Common Functions](general/common_functions.html) page to the userguide for globally available functions.
- 1875: Improved security and performance of `xss_clean()`.

Bugfixes for 1.6.2

Fixed a bug where SET queries were not being handled as "write" queries.

Fixed a bug (#3191) with ORIG_PATH_INFO URI parsing.

Fixed a bug in DB Forge, when inserting an id field (#3456)

Fixed a bug in the table library that could cause identically constructed rows to be dropped (#3459).

Fixed DB Driver and MySQLi result driver checking for resources instead of objects (#3461).

Fixed an AR_caching error where it wasn't tracking table aliases (#3463).

Fixed a bug in AR compiling, where select statements with arguments got incorrectly escaped (#3478).

Fixed an incorrect documentation of `$this->load->language` (#3520).

Fixed bugs (#3523, #4350) in `get_filenames()` with recursion and problems with Windows when `$include_path` is used.

Fixed a bug (#4153) in the XML-RPC class preventing `dateTime.iso8601` from being used.

Fixed an AR bug with `or_where_not_in()` (#4171).

Fixed a bug with [xss_clean\(\)](libraries/input.html) that would add semicolons to GET URI variable strings.

Fixed a bug (#4206) in the Directory Helper where the directory resource was not being closed, and minor improvements.

Fixed a bug in the FTP library where `delete_dir()` was not working recursively (#4215).

Fixed a Validation bug when `set_rules()` is used with a non-array field name and rule (#4220).

Fixed a bug (#4223) where DB caching would not work for returned DB objects or multiple DB connections.

Fixed a bug in the Upload library that might output the same error twice (#4390).

changelog.html

```
1921: <li>Fixed an AR bug when joining with a table alias and table
1922: prefix (#4400).</li>
1923: <li>Fixed a bug in the DB class testing the $params argument.</li>
1924: <li>Fixed a bug in the Table library where the integer 0 in cell
1925: data would be displayed as a blank cell.</li>
1926: <li>Fixed a bug in <code>link_tag()</code> of the <a
1927: href="helpers/url_helper.html">URL helper</a> where a key
ey was passed
1928: instead of a value.
1929: </li>
1930: <li>Fixed a bug in DB_result::row() that prevented it from
1931: returning individual fields with MySQL NULL values.</li>
1932: <li>Fixed a bug where SMTP emails were not having dot
1933: transformation performed on lines that begin with a dot
1934: </li>
1935: <li>Fixed a bug in display_error() in the DB driver that was
1936: instantiating new Language and Exception objects, and not
1937: using the error heading.</li>
1938: <li>Fixed a bug (#4413) where a URI containing slashes only e.g
1939: 'http://example.com/index.php?/' would result in PHP errors</li>
1940: <li>Fixed an array to string conversion error in the Validation
1941: library (#4425)</li>
1942: <li>Fixed bug (#4451, #4299, #4339) where failed transactions
1943: will not rollback when debug mode is enabled.</li>
1944: <li>Fixed a bug (#4506) with overlay_watermark() in the Image
1945: library preventing support for PNG-24s with alpha transparency</li>
1946: <li>Fixed assorted user guide typos (#3453, #4364, #4379, #4399
1947: #4408, #4412, #4448, #4488).</li>
1948: </ul>
1949: <h2>Version 1.6.1</h2>
1950: <p>
1951: Release Date: February 12, 2008<br /> Hg Tag: 1.6.1
1952: </p>
1953: <ul>
1954: <li>Active Record
1955: <ul>
1956: <li>Added <a href="./database/active_record.html#<b>#caching</b>">Active
1957: Record Caching</a>.
1958: </li>
1959: <li>Made Active Record fully database-prefix aware.</li>
1960: </ul>
1961: </li>
1962: <li>Database drivers
1963: <ul>
1964: <li>Added support for setting client character
1965: set and collation for MySQLi.</li>
1966: </ul>
1967: </li>
1968: <li>Core Changes
1969: <ul>
1970: <li>Modified <code>xss_clean()</code> to be more
```

changelog.html

intelligent
1971: with its handling of URL encoded string
s.
1972:
1973: Added \$_SERVER, \$_FILES, \$_ENV, and \$_SESSI
ON to sanitization of globals.
1974: Added a <a href="./helpers/path_helper.html
>Path Helper.
1975:
1976: Simplified <code>_reindex_segments()</code> i
n the URI class.
1977:
1978: Escaped the '-' in the default 'permitted_u
ri_chars' config item, to prevent errors if devel
1979: additional characters to the end of the
1980: Modified method calling to controllers to s
1981: a private or protected method is access
1982: Modified framework initiated 404s to log th
1983: and method for invalid requests.
1984:
1985:
1986: Helpers
1987:
1988: Modified <code>get_filenames()</code> in the
1989: return FALSE if the \$source_dir is not
1990: readable.
1991:
1992:
1993:
1994:
1995:
1996:
1997:
1998:
1999:
2000: <h3>Bugfixes for 1.6.1</h3>
2001:
2002: Deprecated is_numeric as a validation rule. Use of numeric
2003: and integer are preferred.
2004: Fixed bug (#3379) in DBForge with SQLite for table creation
2005: Made Active Record fully database prefix aware (#3384).
Fixed a bug where DBForge was outputting invalid SQL in
2006: Postgres by adding brackets around the tables in FROM.<

2007: Changed the behaviour of Active Record's update() to make
2008: the WHERE clause optional (#3395).
2009: Fixed a bug (#3396) where certain POST variables would caus
2010: e
2011: a PHP warning.
2012: Fixed a bug in query binding (#3402).
2013: Changed order of SQL keywords in the Profiler \$highlight
2014: array so OR would not be highlighted before ORDER BY.</

2015: Fixed a bug (#3404) where the MySQLi driver was testing if

changelog.html

2016: `$this->conn_id` was a resource instead of an object.

- 2017: Fixed a bug (#3419) connecting to a database via a DSN string.
- 2018: Fixed a bug (#3445) where the routed segment array was not re-indexed to begin with 1 when the default controller is used.
- 2020: Fixed assorted user guide typos.

2022:

2023:

2024:

2025:

2026: **Version 1.6.0**

2027: **Release Date: January 30, 2008**

- 2028:
- 2029: **DBForge**
 - 2030: **Added [DBForge](./database/forge.html) to the database tools.**
 - 2031: **Moved `create_database()` and `drop_database()` into [DBForge](./database/forge.html).**
 - 2032: **Added `add_field()`, `add_key()`, `create_table()`, `drop_table()`, `add_column()`, `drop_column()`, `modify_column()` into [DBForge](./database/forge.html).**
- 2040:
- 2041: **Active Record**
 - 2042: **Added `protect_identifiers()` in [Active Record](./database/active_record.html).**
 - 2043:
 - 2044: **All AR queries are backticked if appropriate to the database.**
 - 2045: **Added `where_in()`, `or_where_in()`, `where_not_in()`, `or_where_not_in()`, `not_like()` and `or_not_like()` to [Active Record](./database/active_record.html).**
 - 2046: **Added support for `limit()` into `update()` and `delete()` statements in [Active Record](./database/active_record.html).**
 - 2047:
 - 2048: **Added `empty_table()` and `truncate_table()` to [Active Record](./database/active_record.html).**

2050:

2051:

2052:

2053:

2054:

2055:

2056:

2057:

2058:

2059:

2060:

changelog.html

ml">Active Record.
2061:
2062: s to the `delete()`
2063: [_record.html](#)>Active
2064: Record.
2065: `count_all_results()` functi
2066: on to [Active Record](#)
2067: `select_max()`, `select_
2068: min()`, `select_avg()`
2069: and `select_sum()` to [Ac
2070: tive Record](#)
2071: `select_max()`, `select_
2072: and`
2073: `select_sum()` to [Ac
2074: tive Record](#)
2075: `select_max()`, `select_
2076: and`
2077: `select_sum()` to [Ac
2078: tive Record](#)
2079: `select_max()`, `select_
2080: and`
2081: `select_sum()` to [Ac
2082: tive Record](#)
2083: `select_max()`, `select_
2084: and`
2085: `select_sum()` to [Ac
2086: tive Record](#)
2087: `select_max()`, `select_
2088: and`
2089: `select_sum()` to [Ac
2090: tive Record](#)
2091: `select_max()`, `select_
2092: and`
2093: `select_sum()` to [Ac
2094: tive Record](#)
2095: `select_max()`, `select_
2096: and`
2097: `select_sum()` to [Ac
2098: tive Record](#)
2099: `select_max()`, `select_
2100: and`
2101: `select_sum()` to [Ac
2102: tive Record](#)

Added the ability to pass an array of table
statement in [Active Record](#).

Added `count_all_results()` functi
[Active Record](#)

Added `select_max()`, `select_
and select_sum() to Ac
Active Record

Added the ability to use aliases with joins
Active Record

Added a third parameter to Active Record's
clause to control where the wildcard go

Added a third parameter to set\(\)
Active Record
withholds escaping data.

Changed the behaviour of variables submitte
where\(\) clause with no values to auto s

Other Database Related

MySQL driver now requires MySQL 4.1+
Added \$this->DB->save_queries variable to D
enabling queries to get saved or not. P
saved.
Added \$this->db->dbprefix\(\) to manually add
prefixes.
Added 'random' as an order_by\(\)
removed rand\(\) as a listed
only.

Added a check for NULL fields in the MySQL
utility.
Added constrain_by_prefix param`

changelog.html

eter to
2103: db->list_table() function. If set to
TRUE it will limit the result to only table names with the cur
2104: rent prefix.
2105: Deprecated from Active Record; <code>getwher
e()</code> for <code>get_where()</code>
2106: <code>groupby()</code> for <code>group_by(
</code><code>havingor()</code>
<code>having_or()</code>; <code>orderb
<code>orwhere()</code> for <code>or_where(
<code>or_like()</code> for <code>or_like()</code>.

2107: y()</code> for <code>order_by</code>;
2108: <code>orlike()</code>
<code>orlike()</code>
2109: <code>csv_from_result()</code> to ou
2110: tput CSV data
2111: more in the spirit of basic rules of RF
2112: C 4180.
2113:
2114: Added 'char_set' and 'dbcollat' database co
nfiguration settings, to explicitly set the client
2115: communication properly.
2116: Removed 'active_r' configuration setting an
d replaced with a global \$active_record setting, which
2117: is more in harmony with the global nature of the behavior (#1834).<
2118: /li>
2119:
2120:
2121: Core changes
2122:
2123: Added ability to load multiple views, whose
2124: content will be appended to the output in the order
2125: loaded.
2126: Added the ability to <a href="./general/aut
Models<
2127: /a>.
2128:
2129: Reorganized the URI and Routes classes for
2130: better clarity.
2131: Added Compat.php to allow function override
s for older versions of PHP or PHP environments mis
2132: sing certain extensions / libraries
2133: Added memory usage, GET, URI string data, a
2134: nd individual query execution time to Profiler output

2135: Deprecated Scaffolding.
2136: Added is_really_writable() to Common.php to
2137: provide a cross-platform reliable method of testi
ng file/folder writability.
2138:
2139:
2140:

changelog.html

```
2141:                                     <li>Libraries
2142:                                         <ul>
2143:                                             <li>Changed the load protocol of Models to allo
w for
2144:                                                 extension.</li>
2145:                                             <li>Strengthened the Encryption library to help
protect
2146:                                                 against man in the middle attacks when
MCRYPT_MODE_CBC mode is
2147:                                                 used.</li>
2148:                                             <li>Added Flashdata variables, session_id regen
eration and
2149:                                                 configurable session update times to th
e <a
2150:                                                 href="./libraries/sessions.html">Sessio
n class.</a>
2151:                                             </li>
2152:                                             <li>Removed 'last_visit' from the Session class
.</li>
2153:                                             <li>Added a language entry for valid_ip validat
ion error.</li>
2154:                                             <li>Modified prep_for_form() in the Validation
class to accept
2155:                                                 arrays, adding support for POST array v
alidation (via callbacks
2156:                                                 only)</li>
2157:                                             <li>Added an &quot;integer&quot; rule into the
Validation
2158:                                                 library.</li>
2159:                                             <li>Added valid_base64() to the Validation libr
ary.</li>
2160:                                             <li>Documented clear() in the <a
2161:                                                 href="./libraries/image_lib.html">Imag
e Processing</a> library.
2162:                                             </li>
2163:                                             <li>Changed the behaviour of custom callbacks s
o that they no
2164:                                                 longer trigger the &quot;required&quot;
rule.</li>
2165:                                             <li>Modified Upload class $_FILES error message
s to be more
2166:                                                 precise.</li>
2167:                                             <li>Moved the safe mode and auth checks for the
Email library
2168:                                                 into the constructor.</li>
2169:                                             <li>Modified variable names in _ci_load() metho
d of Loader
2170:                                                 class to avoid conflicts with view vari
ables.</li>
2171:                                             <li>Added a few additional mime type variations
for CSV.</li>
2172:                                             <li>Enabled the 'system' methods for the XML-RP
C Server
2173:                                                 library, except for 'system.multicall'
which is still disabled.</li>
2174:                                         </ul>
2175:                                     </li>
2176:                                     <li>Helpers & Plugins
2177:                                         <ul>
2178:                                             <li>Added link_tag() to the <a
2179:                                                 href="./helpers/html_helper.html">HTML
helper.</a></li>
```

changelog.html

```
2181:
helper.html">HTML
2182:
2183: .html">"extend"
2184:
2185:
2186: ml">email
2187:
2188:
2189: <a
2190: ing helper</a>.
2191:
2192: n to <a
2193: ing helper</a>.
2194:
2195: ion to <a
2196: ing helper</a>.
2197:
2198: _fieldset_close()</kbd>,
2199: reset()</kbd> function to <a
2200: helper</a>.
2201:
2202: m_open()</kbd>.
2203:
2204: to
2205:
2206: ths and days
2207: d of hard-coded values,
2208:
2209: </ul>
2210: </li>
2211:
2212:
2213: <li>Documentation Changes
2214: <ul>
2215: <li>Added <a href="./doc_style/index.html">Writing
2216: Documentation</a> section for the
2217: community to use in writing their
2218: own documentation.
2219: </li>
2220: <li>Added titles to all user manual pages.</li>
2221: <li>Added attributes into <code><html></code> of userguide
2222: for valid
2223: <code>html.</code>
2224: <li>Added <a
2225: href="http://codeigniter.com/user_guide
2226: /libraries/zip.html">Zip
```

changelog.html

```
2224:                                     Encoding Class</a> to the table
of contents of the userguide.
2225:                                     </li>
2226:                                     <li>Moved part of the userguide menu javascript
to an external
2227:                                     file.</li>
2228:                                     <li>Documented <kbd>distinct()</kbd> in <a
2229:                                     href="./database/active_record.html">Ac
tive Record</a>.
2230:                                     </li>
2231:                                     <li>Documented the <kbd>timezones()</kbd> funct
ion in the <a
2232:                                     href="./helpers/date_helper.html">Date
Helper</a>.
2233:                                     </li>
2234:                                     <li>Documented unset_userdata in the <a
2235:                                     href="./libraries/sessions.html">Sessio
n class</a>.
2236:                                     </li>
2237:                                     <li>Documented 2 config options to the <a
2238:                                     href="./database/configuration.html">Da
tabase configuration</a>
2239:                                     page.
2240:                                     </li>
2241:                                     </ul>
2242:                                     </li>
2243:                                     </ul>
2244:                                     <h3>Bug fixes for Version 1.6.0</h3>
2245:                                     <ul>
2246:                                     <li>Fixed a bug (#1813) preventing using $CI->db in the same
2247:                                     application with returned database objects.</li>
2248:                                     <li>Fixed a bug (#1842) where the $this->uri->rsegments array
2249:                                     would not include the 'index' method if routed to the c
ontroller
2250:                                     without an implicit method.</li>
2251:                                     <li>Fixed a bug (#1872) where word_limiter() was not retaining
2252:                                     whitespace.</li>
2253:                                     <li>Fixed a bug (#1890) in csv_from_result() where content that
2254:                                     included the delimiter would break the file.</li>
2255:                                     <li>Fixed a bug (#2542) in the clean_email() method of the Email
2256:                                     class to allow for non-numeric / non-sequential array k
eys.</li>
2257:                                     <li>Fixed a bug (#2545) in <kbd>_html_entity_decode_callback()<
2258:                                     /kbd>
2259:                                     when 'global_xss_filtering' is enabled.
2260:                                     </li>
2261:                                     <li>Fixed a bug (#2668) in the <a href="./libraries/parser.html
2262:                                     ">parser
class</a> where numeric data was ignored.
2263:                                     </li>
2264:                                     <li>Fixed a bug (#2679) where the &quot;previous&quot;
2265:                                     pagination link would get drawn on the first page.</li>
2266:                                     <li>Fixed a bug (#2702) in _object_to_array that broke some
2267:                                     types of inserts and updates.</li>
2268:                                     <li>Fixed a bug (#2732) in the SQLite driver for PHP 4.</li>
2269:                                     <li>Fixed a bug (#2754) in Pagination to scan for non-positive
2270:                                     num_links.</li>
2271:                                     <li>Fixed a bug (#2762) in the <a
2272:                                     href="./libraries/sessions.html">Session library</a> wh
ere user
2273:                                     agent matching would fail on user agents ending with a
2274:                                     agent matching would fail on user agents ending with a
```

changelog.html

```
space.
2275: </li>
2276: <li>Fixed a bug (#2784) $field_names[] vs $Ffield_names[] in
2277:     postgres and sqlite drivers.</li>
2278: <li>Fixed a bug (#2810) in the typography helper causing
2279:     extraneous paragraph tags when string contains tags.</li>
i>
2280: <li>Fixed a bug (#2849) where arguments passed to a subfolder
2281:     controller method would be incorrectly shifted, droppin
g the 3rd
2282:     segment value.</li>
2283: <li>Fixed a bug (#2858) which referenced a wrong variable in th
e
2284:     Image class.</li>
2285: <li>Fixed a bug (#2875)when loading plugin files as _plugin. an
d
2286:     not _pi.</li>
2287: <li>Fixed a bug (#2912) in <code>get_filenames()</code> in the <a
2288:     href="./helpers/file_helper.html">File Helper </a>where
the array
2289:     wasn't cleared after each call.
2290: </li>
2291: <li>Fixed a bug (#2974) in <code>highlight_phrase()</code> that
2292:     caused an error with slashes.
2293: </li>
2294: <li>Fixed a bug (#3003) in the Encryption Library to support
2295:     modes other than MCRYPT_MODE_ECB</li>
2296: <li>Fixed a bug (#3015) in the <a
2297:     href="./libraries/user_agent.html">User Agent library</
a> where more
2298:     then 2 languages where not reported with languages().
2299: </li>
2300: <li>Fixed a bug (#3017) in the <a href="./libraries/email.html"
>Email</a>
2301:     library where some timezones were calculated incorrectl
y.
2302: </li>
2303: <li>Fixed a bug (#3024) in which master_dim wasn't getting rese
t
2304:     by clear() in the Image library.</li>
2305: <li>Fixed a bug (#3156) in Text Helper highlight_code() causing
2306:     PHP tags to be handled incorrectly.</li>
2307: <li>Fixed a bug (#3166) that prevented num_rows from working in
2308:     Oracle.</li>
2309: <li>Fixed a bug (#3175) preventing certain libraries from
2310:     working properly when autoloaded in PHP 4.</li>
2311: <li>Fixed a bug (#3267) in the Typography Helper where unordere
d
2312:     list was listed &quot;un.</li>
2313: <li>Fixed a bug (#3268) where the Router could leave '/' as the
2314:     path.</li>
2315: <li>Fixed a bug (#3279) where the Email class was sending the
2316:     wrong Content-Transfer-Encoding for some character sets
.</li>
2317: <li>Fixed a bug (#3284) where the rsegment array would not be
2318:     set properly if the requested URI contained more segmen
ts than the
2319:     routed URI.</li>
2320: <li>Removed extraneous load of $CFG in _display_cache() of the
2321:     Output class (#3285).</li>
2322: <li>Removed an extraneous call to loading models (#3286).</li>
2323: <li>Fixed a bug (#3310) with sanitization of globals in the
2324:     Input class that could unset CI's global variables.</li>
```

changelog.html

```
>
2325: <li>Fixed a bug (#3314) which would cause the top level path to
2326:         be deleted in delete_files() of the File helper.</li>
2327: <li>Fixed a bug (#3328) where the smiley helper might return an
2328:         undefined variable.</li>
2329: <li>Fixed a bug (#3330) in the FTP class where a comparison
2330:         wasn't getting made.</li>
2331: <li>Removed an unused parameter from Profiler (#3332).</li>
2332: <li>Fixed a bug in database driver where num_rows property
2333:         wasn't getting updated.</li>
2334: <li>Fixed a bug in the <a href="./libraries/file_uploading.html
">upload
2335:         library</a> when allowed_files wasn't defined.
2336: </li>
2337: <li>Fixed a bug in <code>word_wrap()</code> of the Text Helper
2338:         that incorrectly referenced an object.
2339: </li>
2340: <li>Fixed a bug in Validation where <code>valid_ip()</code> wasn'
t
2341:         called properly.
2342: </li>
2343: <li>Fixed a bug in Validation where individual error messages
2344:         for checkboxes wasn't supported.</li>
2345: <li>Fixed a bug in captcha calling an invalid PHP function.</li>
>
2346: <li>Fixed a bug in the cookie helper "set_cookie" function. It
2347:         was not honoring the config settings.</li>
2348: <li>Fixed a bug that was making validation callbacks required
2349:         even when not set as such.</li>
2350: <li>Fixed a bug in the XML-RPC library so if a type is
2351:         specified, a more intelligent decision is made as to th
e default
2352:         type.</li>
2353: <li>Fixed an example of comma-separated emails in the email
2354:         library documentation.</li>
2355: <li>Fixed an example in the Calendar library for Showing
2356:         Next/Previous Month Links.</li>
2357: <li>Fixed a typo in the database language file.</li>
2358: <li>Fixed a typo in the image language file &quot;support&quot;
2359:         to &quot;support&quot;.</li>
2360: <li>Fixed an example for XML RPC.</li>
2361: <li>Fixed an example of <code>accept_charset()</code> in the <a
2362:         href="./libraries/user_agent.html">User Agent Library</
a>.
2363: </li>
2364: <li>Fixed a typo in the docblock comments that had CodeIgniter
2365:         spelled CodeIgnitor.</li>
2366: <li>Fixed a typo in the <a href="./helpers/string_helper.html">
String
2367:         Helper</a> (uniquid changed to unqid).
2368: </li>
2369: <li>Fixed typos in the email Language class
2370:         (email_attachment_unredable, email_filed_smtp_login), a
nd FTP Class
2371:         (ftp_unable_to_rename).</li>
2372: <li>Added a stripslashes() into the Upload Library.</li>
2373: <li>Fixed a series of grammatical and spelling errors in the
2374:         language files.</li>
2375: <li>Fixed assorted user guide typos.</li>
2376: </ul>
2377: <h2>Version 1.5.4</h2>
2378: <p>Release Date: July 12, 2007</p>
2379: <ul>
```

changelog.html

2380: Added custom
2381: Language files to the <a href="./general/au
toloader.html">autoload
2382: options.
2383:
2384: Added stripslashes() to the _clean_input_data() function in
2385: the Input class wh
en magic
2386: quotes is on so that data will always be un-slashed wit
hin the
2387: framework.
2388:
2389: Added array to string into the <a
2390: href="general/profiling.html">profiler.
2391:
2392: Added some additional mime types in
2393: application/config/mimes.php.
2394: Added filename_security() method to <a
2395: href="./libraries/input.html">Input library.
2396:
2397: Added some additional arguments to the <a
2398: href="./helpers/inflector_helper.html">Inflection helpe
r
2399: singular() to compensate for words ending in "s". Also
added a force
2400: parameter to pluralize().
2401:
2402: Added \$config['charset'] to the config file. Default value
2403: is 'UTF-8', used in some string handling functions.
>
2404: Fixed MSSQL insert_id().
2405: Fixed a logic error in the DB trans_status() function. It
2406: was incorrectly returning TRUE on failure and FALSE on
success.
2407: Fixed a bug that was allowing multiple load attempts on
2408: extended classes.
2409: Fixed a bug in the bootstrap file that was incorrectly
2410: attempting to discern the full server path even when it
was
2411: explicitly set by the user.
2412: Fixed a bug in the escape_str() function in the MySQL
2413: driver.
2414: Fixed a typo in the Cal
endar
2415: library
2416:
2417: Fixed a typo in rpcs.php library
2418: Fixed a bug in the Zip
2419: library, providing PC Zip file compatibilit
y with Mac OS X
2420:
2421: Fixed a bug in router that was ignoring the scaffolding
2422: route for optimization
2423: Fixed an IP validation bug.
2424: Fixed a bug in display of POST keys in the <a
2425: href="./general/profiling.html">Profiler output
2426:
2427: Fixed a bug in display of queries with characters that woul
d
2428: be interpreted as HTML in the <a href="./general/profil
ing.html">Profiler
2429: output
2430:

changelog.html

2431: Fixed a bug in display of Email class print debugger with
2432: characters that would be interpreted as HTML in the deb
ugging output
2433: Fixed a bug in the Content-Transfer-Encoding of HTML emails
2434: with the quoted-printable MIME type
2435: Fixed a bug where one could unset certain PHP superglobals
2436: by setting them via GET or POST data
2437: Fixed an undefined function error in the insert_id()
2438: function of the PostgreSQL driver
2439: Fixed various doc typos.
2440: Documented two functions from the <a
2441: href="./helpers/string_helper.html">String helper
that were
2442: missing from the user guide: trim_slashes() and
2443: reduce_double_slashes().
2444:
2445: Docs now validate to XHTML 1 transitional
2446: Updated the XSS Filtering to take into account the IE
2447: expression() ability and improved certain deletions to
prevent
2448: possible exploits
2449: Modified the Router so that when Query Strings are Enabled,
2450: the controller trigger and function trigger values are
sanitized for
2451: filename include security.
2452: Modified the is_image() method in the Upload library to tak
e
2453: into account Windows IE 6/7 eccentricities when dealing
with MIMEs
2454: Modified XSS Cleaning routine to be more performance
2455: friendly and compatible with PHP 5.2's new PCRE backtra
ck and
2456: recursion limits.
2457: Modified the URL
2458: Helper to type cast the \$title as a string
in case a numeric value
2459: is supplied
2460:
2461: Modified Form Helper form_dropdown() to type cast the keys
2462: and values of the options array as strings, allowing nu
meric values
2463: to be properly set as 'selected'
2464: Deprecated the use if <code>is_numeric()</code> in various
2465: places since it allows periods. Due to compatibility pr
oblems with <code>ctype_digit()</code>,
2466: making it unreliable in some installations, the followi
ng regular
2467: expression was used instead: <code>preg_match("/[^\0-9]/"
, \$n)</code>
2468: Deprecated: APPVER has been deprecated and replaced with
2469: CI_VERSION for clarity.
2470:
2471: <h2>Version 1.5.3</h2>
2472: <p>Release Date: April 15, 2007</p>
2473:
2474: Added array to string into the profiler
2475: Code Igniter references updated to CodeIgniter
2476: pMachine references updated to EllisLab
2477: Fixed a bug in the repeater function of <a
2478: href="./helpers/string_helper.html">string helper.
2479:
2480: Fixed a bug in ODBC driver
2481: Fixed a bug in result_array() that was returning an empty

changelog.html

```
2482:         array when no result is produced.</li>
2483: <li>Fixed a bug in the redirect function of the <a
2484:         href="./helpers/url_helper.html">url helper</a>.
2485: </li>
2486: <li>Fixed an undefined variable in Loader</li>
2487: <li>Fixed a version bug in the Postgres driver</li>
2488: <li>Fixed a bug in the textarea function of the form helper for
2489:         use with strings</li>
2490: <li>Fixed doc typos.</li>
2491: </ul>
2492: <h2>Version 1.5.2</h2>
2493: <p>Release Date: February 13, 2007</p>
2494: <ul>
2495:         <li>Added <a href="./installation/downloads.html#svn">subversio
n
2496:         information</a> to the <a href="./installation/
downloads.html">downloads</a>
2497:         page.
2498: </li>
2499: <li>Added support for captions in the <a
2500:         href="./libraries/table.html">Table Library</a>
2501: </li>
2502: <li>Fixed a bug in the <a href="./helpers/download_helper.html"
>download_helper</a>
2503:         that was causing Internet Explorer to load rather than
download
2504: </li>
2505: <li>Fixed a bug in the Active Record Join function that was not
2506:         taking table prefixes into consideration.</li>
2507: <li>Removed unescaped variables in error messages of Input and
2508:         Router classes</li>
2509: <li>Fixed a bug in the Loader that was causing errors on
2510:         Libraries loaded twice. A debug message is now silently
made in the
2511:         log.</li>
2512: <li>Fixed a bug in the <a href="./helpers/form_helper.html">form
m
2513:         helper</a> that gave textarea a value attribute
2514: </li>
2515: <li>Fixed a bug in the <a href="./libraries/image_lib.php">Image
e
2516:         Library</a> that was ignoring resizing the same
size image
2517: </li>
2518: <li>Fixed some doc typos.</li>
2519: </ul>
2520:
2521:
2522: <h2>Version 1.5.1</h2>
2523: <p>Release Date: November 23, 2006</p>
2524: <ul>
2525:         <li>Added support for submitting arrays of libraries in the
2526:         $this-&gt;load-&gt;library function.</li>
2527: <li>Added support for naming custom library files in lower or
2528:         uppercase.</li>
2529: <li>Fixed a bug related to output buffering.</li>
2530: <li>Fixed a bug in the active record class that was not
2531:         resetting query data after a completed query.</li>
2532: <li>Fixed a bug that was suppressing errors in controllers.</li>
>
2533: <li>Fixed a problem that can cause a loop to occur when the
2534:         config file is missing.</li>
2535: <li>Fixed a bug that occurred when multiple models were loaded
```


changelog.html

```
2536:         with the third parameter set to TRUE.</li>
2537:     <li>Fixed an oversight that was not unsetting globals properly
2538:         in the input sanitize function.</li>
2539:     <li>Fixed some bugs in the Oracle DB driver.</li>
2540:     <li>Fixed an incorrectly named variable in the MySQLi result
2541:         driver.</li>
2542:     <li>Fixed some doc typos.</li>
2543: </ul>
2544: <h2>Version 1.5.0.1</h2>
2545: <p>Release Date: October 31, 2006</p>
2546: <ul>
2547:     <li>Fixed a problem in which duplicate attempts to load helpers
2548:         and classes were not being stopped.</li>
2549:     <li>Fixed a bug in the word_wrap() helper function.</li>
2550:     <li>Fixed an invalid color Hex number in the Profiler class.</li>
2551:     <li>Fixed a corrupted image in the user guide.</li>
2552: </ul>
2553:
2554:
2555:
2556: <h2>Version 1.5.0</h2>
2557: <p>Release Date: October 30, 2006</p>
2558:
2559: <ul>
2560:     <li>Added <a href="./database/utilities.html">DB utility
2561:         class</a>, permitting DB backups, CVS or XML fi
2562:         les from DB results, and
2563:         various other functions.
2564:     <li>Added <a href="./database/caching.html">Database Caching
2565:         Class</a>.
2566:     <li>Added <a href="./database/transactions.html">transaction
2567:         support</a> to the database classes.
2568:     <li>Added <a href="./general/profiling.html">Profiler Class</a>
2569:         which generates a report of Benchmark execution times,
2570:         queries, and
2571:         POST data at the bottom of your pages.
2572:     <li>Added <a href="./libraries/user_agent.html">User Agent
2573:         Library</a> which allows browsers, robots, and
2574:         mobile devises to be
2575:         identified.
2576:     <li>Added <a href="./libraries/table.html">HTML Table Class</a>
2577:         , enabling tables to be generated from arrays or databa
2578:         se results.
2579:     <li>Added <a href="./libraries/zip.html">Zip Encoding
2580:         Library</a>.
2581:     <li>Added <a href="./libraries/ftp.html">FTP Library</a>.
2582:     <li>Added the ability to <a
2583:         href="./general/creating_libraries.html">extend librari
2584:         es</a> and <a
2585:         href="./general/core_classes.html">extend core classes<
2586:         /a>, in
2587:         addition to being able to replace them.
2588:     <li>Added support for storing <a href="./general/models.html">m
```

changelog.html

```
odels
2592:         within sub-folders</a>.
2593:     </li>
2594:     <li>Added <a href="./helpers/download_helper.html">Download
2595:         Helper</a>.
2596:     </li>
2597:     <li>Added <a href="./database/queries.html">simple_query()</a>
2598:         function to the database classes
2599:     </li>
2600:     <li>Added <a href="./helpers/date_helper.html">standard_date()<
/a>
2601:         function to the Date Helper.
2602:     </li>
2603:     <li>Added <a href="./database/results.html">$query->free_result
( )</a>
2604:         to database class.
2605:     </li>
2606:     <li>Added <a href="./database/fields.html">$query->list_fields(
)/a>
2607:         function to database class
2608:     </li>
2609:     <li>Added <a href="./database/helpers.html">$this->db->platform
( )</a>
2610:         function
2611:     </li>
2612:     <li>Added new <a href="./helpers/file_helper.html">File
2613:         Helper</a>: get_filenames()
2614:     </li>
2615:     <li>Added new helper: <a href="./helpers/smiley_helper.html">Sm
iley
2616:         Helper</a></li>
2617:     <li>Added support for &lt;ul> and &lt;ol> lists in the <a
2618:         href="./helpers/html_helper.html">HTML Helper</a></li>
2619:     <li>Added the ability to rewrite <a
2620:         href="./general/alternative_php.html">short tags</a> on
-the-fly,
2621:         converting them to standard PHP statements, for those s
ervers that
2622:         do not support short tags. This allows the cleaner synt
ax to be used
2623:         regardless of whether it's supported by the server.
2624:     </li>
2625:     <li>Added the ability to <a href="./general/managing_apps.html"
>rename
2626:         or relocate the "application" folder</a>.
2627:     </li>
2628:     <li>Added more thorough initialization in the upload class so
2629:         that all class variables are reset.</li>
2630:     <li>Added "is_numeric" to validation, which uses the native PHP
2631:         is_numeric function.</li>
2632:     <li>Improved the URI handler to make it more reliable when the
2633:         $config['uri_protocol'] item is set to AUTO.</li>
2634:     <li>Moved most of the functions in the Controller class into th
e
2635:         Loader class, allowing fewer reserved function names fo
r controllers
2636:         when running under PHP 5.</li>
2637:     <li>Updated the DB Result class to return an empty array when
2638:         $query->result() doesn't produce a result.</li>
2639:     <li>Updated the <dfn>input->cookie()</dfn> and <dfn>input->post
( )</dfn>
2640:         functions in <a href="./libraries/input.html">Input Cla
ss</a> to
```

changelog.html

2641: permit arrays contained cookies that are arrays to be r
un through
2642: the XSS filter.
2643:
2644: Documented three functions from the <a
2645: href="./libraries/validation.html">Validation class
that were
2646: missing from the user guide: set_select(), set_radio(),
and
2647: set_checkbox().
2648:
2649: Fixed a bug in the Email class related to SMTP Helo data.</
li>
2650: Fixed a bug in the word wrapping helper and function in the
2651: email class.
2652: Fixed a bug in the validation class.
2653: Fixed a bug in the typography helper that was incorrectly
2654: wrapping block level elements in paragraph tags.
2655: Fixed a problem in the form_prep() function that was double
2656: encoding entities.
2657: Fixed a bug that affects some versions of PHP when output
2658: buffering is nested.
2659: Fixed a bug that caused CI to stop working when the PHP
2660: magic __get() or __set() functions were used within mod
els or
2661: controllers.
2662: Fixed a pagination bug that was permitting negative values
2663: in the URL.
2664: Fixed an oversight in which the Loader class was not allowe
d
2665: to be extended.
2666: Changed <dfn>_get_config()</dfn> to <dfn>get_config()</dfn>
2667: since the function is not a private one.
2668:
2669: Deprecated "init" folder. Initialization
2670: happens automatically now. <a
2671: href="./general/creating_libraries.html">Please see
2672: documentation.
2673: Deprecated \$this->db->field_names() USE
2674: \$this->db->list_fields()
2675: Deprecated the <dfn>\$config['log_errors']<
/dfn>
2676: item from the config.php file. Instead, <dfn>\$config['l
og_threshold']</dfn>
2677: can be set to "0" to turn it off.
2678:
2679:
2680:
2681:
2682:
2683: <h2>Version 1.4.1</h2>
2684: <p>Release Date: September 21, 2006</p>
2685:
2686:
2687: Added a new feature that passes URI segments directly to
2688: your function calls as parameters. See the <a
2689: href="./general/controllers.html">Controllers page
for more
2690: info.
2691:
2692: Added support for a function named <dfn>_output()</dfn>,
2693: which when used in your controllers will received the f
inal rendered

changelog.html

2694: output from the output class. More info in the [Controllers](#) page.

2695: [Controllers](#)

2696: [Controllers](#)

2697: [Controllers](#)

2698: [URI Class](#) to let you re

trieve and

2699: manipulate URI segments that have been re-routed using

the [URI](#)

2700: [URI Routing](#) feature. Previously

, the URI

2701: class did not permit you to access any re-routed URI se

gments, but

2702: now it does.

2703: [URI Routing](#)

2704: [URI Routing](#)

header() [URI Routing](#)

2705: function, which allows you to set server headers.

2706: [URI Routing](#)

2707: Updated plugins, helpers, and language classes to allow you

r

2708: [application](#) folder to contain its own plugin

s, helpers,

2709: and language folders. Previously they were always treat

ed as global

2710: for your entire installation. If your application folde

r contains

2711: any of these resources they will be used **instead**

the global

2712: ones.

2713: [Inflector](#)

2714: [Inflector](#)

2715: [Inflector](#)

2716: [Inflector](#)

2717: [element\(\)](#) function in the [array helper](#).

2718: [array helper](#)

2719: [array helper](#)

2720: [RAND\(\)](#) to active record [orderby\(\)](#)

n

2721: function.

2722: [delete_cookie\(\)](#) and [get_cookie\(\)](#)

>

2724: to [Cookie helper](#)

[Cookie helper](#), even

2725: though the input class has a cookie fetching function.

2726: [Cookie helper](#)

2727: [Oracle database driver](#) (still undergoing testing so i

t

2728: might have some bugs).

2729: [combine pseudo-variables and php](#)

2730: [combine pseudo-variables and php](#)

2731: [output compression option to the config file.](#)

2732: [Removed the is_numeric test from the db->escape\(\) function.](#)

2733: [Fixed a MySQLi bug that was causing error messages not to](#)

2734: [contain proper error data.](#)

2735: [Fixed a bug in the email class which was causing it to](#)

2736: [ignore explicitly set alternative headers.](#)

2737: [Fixed a bug that was causing a PHP error when the Exception](#)

s

2738: class was called within the get_config() function since

it was

2739: causing problems.

changelog.html

```
2740: <li>Fixed an oversight in the cookie helper in which the config
2741:         file cookie settings were not being honored.</li>
2742: <li>Fixed an oversight in the upload class. An item mentioned in
n
2743:         the 1.4 changelog was missing.</li>
2744: <li>Added some code to allow email attachments to be reset when
2745:         sending batches of email.</li>
2746: <li>Deprecated the <dfn>application/scripts</dfn> folder. It
2747:         will continue to work for legacy users, but it is recom
mended that
2748:         you create your own <a href="./general/libraries.html">
libraries</a>
2749:         or <a href="./general/models.html">models</a> instead.
It was
2750:         originally added before CI had user libraries or models
, but it's
2751:         not needed anymore.
2752: </li>
2753: <li>Deprecated the <dfn>$autoload['core']</dfn> item from the <
kbd>autoload.php</kbd>
2754:         file. Instead, please now use: <kbd>$autoload['librarie
s']</kbd></li>
2755: <li>Deprecated the following database functions:
2756:         $this->db->smart_escape_str() and $this->db->fields().<
/li>
</ul>
2757:
2758:
2759:
2760:
2761: <h2>Version 1.4.0</h2>
2762: <p>Release Date: September 17, 2006</p>
2763:
2764: <ul>
2765: <li>Added <a href="./general/hooks.html">Hooks</a> feature,
2766:         enabling you to tap into and modify the inner workings
of the
2767:         framework without hacking the core files.
2768: </li>
2769: <li>Added the ability to organize controller files <a
2770:         href="./general/controllers.html">into sub-folders</a>.
Kudos to
2771:         Marco for suggesting this (and the next two) feature.
2772: </li>
2773: <li>Added regular expressions support for <a
2774:         href="./general/routing.html">routing rules</a>.
2775: </li>
2776: <li>Added the ability to <a href="./general/controllers.html">r
emap
2777:         function calls</a> within your controllers.
2778: </li>
2779: <li>Added the ability to <a href="./general/core_classes.html">
replace
2780:         core system classes</a> with your own classes.
2781: </li>
2782: <li>Added support for % character in URL.</li>
2783: <li>Added the ability to supply full URLs using the <a
2784:         href="./helpers/url_helper.html">anchor()</a> helper fu
nction.
2785: </li>
2786: <li>Added mode parameter to <a href="./helpers/file_helper.html
">file_write()</a>
2787:         helper.
2788: </li>
```

changelog.html

```
2789: <li>Added support for changing the port number in the <a
2790:     href="./database/configuration.html">Postgres driver</a
>.
2791: </li>
2792: <li>Moved the list of "allowed URI characters" out of the Router
2793:     class and into the config file.</li>
2794: <li>Moved the MIME type array out of the Upload class and into
2795:     its own file in the applications/config/ folder.</li>
2796: <li>Updated the Upload class to allow the upload field name to
2797:     be set when calling <a href="./libraries/file_uploading
.html">do_upload()</a>.
2798: </li>
2799: <li>Updated the <a href="./libraries/config.html">Config
2800:     Library</a> to be able to load config files silently, and to be able to
2801:     assign config files to their own index (to avoid collisions if you
2802:     use multiple config files).
2803: </li>
2804: <li>Updated the URI Protocol code to allow more options so that
2805:     URLs will work more reliably in different environments.
</li>
2806: <li>Updated the <dfn>form_open()</dfn> helper to allow the GET
2807:     method to be used.
</li>
2808: </li>
2809: <li>Updated the MySQLi <dfn>execute()</dfn> function with some
2810:     code to help prevent lost connection errors.
2811: </li>
2812: <li>Updated the SQLite Driver to check for object support before
2813:     attempting to return results as objects. If unsupported
2814:     it returns an array.</li>
2815: <li>Updated the Models loader function to allow multiple loads
2816:     of the same model.</li>
2817: <li>Updated the MS SQL driver so that single quotes are escaped.
</li>
2818: <li>Updated the Postgres and ODBC drivers for better
2819:     compatibility.</li>
2820: <li>Removed a strtolower() call that was changing URL segments
2821:     to lower case.</li>
2822: <li>Removed some references that were interfering with PHP 4.4.
1
2823:     compatibility.</li>
2824: <li>Removed backticks from Postgres class since these are not
2825:     needed.</li>
2826: <li>Renamed <dfn>display()</dfn> to <dfn>_display()</dfn> in the
e
2827:     Output class to make it clear that it's a private function.
2828: </li>
2829: <li>Deprecated the hash() function due to a naming conflict with
h
2830:     a native PHP function with the same name. Please use <code>dohash()</code>
bd>dohash()</kbd>
2831:     instead.
2832: </li>
2833: <li>Fixed an bug that was preventing the input class from
2834:     unsetting GET variables.</li>
2835: <li>Fixed a router bug that was making it too greedy when
2836:     matching end segments.</li>
2837: <li>Fixed a bug that was preventing multiple discrete database
```

changelog.html

```
2838:         calls.</li>
2839: <li>Fixed a bug in which loading a language file was producing
a
2840:         "file contains no data" message.</li>
2841: <li>Fixed a session bug caused by the XSS Filtering feature
2842:         inadvertently changing the case of certain words.</li>
2843: <li>Fixed some missing prefixes when using the database prefix
2844:         feature.</li>
2845: <li>Fixed a typo in the Calendar class (cal_november).</li>
2846: <li>Fixed a bug in the <dfn>form_checkbox()</dfn> helper.
2847: </li>
2848: <li>Fixed a bug that was allowing the second segment of the URI
2849:         to be identical to the class name.</li>
2850: <li>Fixed an evaluation bug in the database initialization
2851:         function.</li>
2852: <li>Fixed a minor bug in one of the error messages in the
2853:         language class.</li>
2854: <li>Fixed a bug in the date helper <dfn>timespan</dfn> function
.
2855: </li>
2856: <li>Fixed an undefined variable in the DB Driver class.</li>
2857: <li>Fixed a bug in which dollar signs used as binding
2858:         replacement values in the DB class would be treated as
RegEx
2859:         back-references.</li>
2860: <li>Fixed a bug in the <dfn>set_hash()</dfn> function which was
2861:         preventing MD5 from being used.
2862: </li>
2863: <li>Fixed a couple bugs in the Unit Testing class.</li>
2864: <li>Fixed an incorrectly named variable in the Validation class
.</li>
2865: <li>Fixed an incorrectly named variable in the URI class.</li>
2866: <li>Fixed a bug in the config class that was preventing the bas
e
2867:         URL from being called properly.</li>
2868: <li>Fixed a bug in the validation class that was not permitting
2869:         callbacks if the form field was empty.</li>
2870: <li>Fixed a problem that was preventing scaffolding from workin
g
2871:         properly with MySQLi.</li>
2872: <li>Fixed some MS SQL bugs.</li>
2873: <li>Fixed some doc typos.</li>
2874: </ul>
2875:
2876:
2877:
2878: <h2>Version 1.3.3</h2>
2879: <p>Release Date: June 1, 2006</p>
2880:
2881: <ul>
2882:
2883: <li>Models do <strong>not</strong> connect automatically to the
2884:         database as of this version. <a href="./general/models.
html">More
2885:         info here</a>.
2886: </li>
2887: <li>Updated the Sessions class to utilize the active record
2888:         class when running session related queries. Previously
the queries
2889:         assumed MySQL syntax.</li>
2890: <li>Updated alternator() function to re-initialize when called
2891:         with no arguments, allowing multiple calls.</li>
2892: <li>Fixed a bug in the active record "having" function.</li>
```

changelog.html

2893: Fixed a problem in the validation class which was making
2894: checkboxes be ignored when required.
2895: Fixed a bug in the word_limiter() helper function. It was
2896: cutting off the first word.
2897: Fixed a bug in the xss_clean function due to a PHP bug that
2898: affects some versions of html_entity_decode.
2899: Fixed a validation bug that was preventing rules from being
2900: set twice in one controller.
2901: Fixed a calendar bug that was not letting it use dynamicall
y
2902: loaded languages.
2903: Fixed a bug in the active record class when using WHERE
2904: clauses with LIKE
2905: Fixed a bug in the hash() security helper.
2906: Fixed some typos.
2907:
2908:
2909:
2910:
2911:
2912: <h2>Version 1.3.2</h2>
2913: <p>Release Date: April 17, 2006</p>
2914:
2915:
2916: Changed the behavior of the validation class such that if a
2917: "required" rule is NOT explicitly stated for a field th
en all other
2918: tests get ignored.
2919: Fixed a bug in the Controller class that was causing it to
2920: look in the local "init" folder instead of the main sys
tem one.
2921: Fixed a bug in the init_pagination file. The \$config item
2922: was not being set correctly.
2923: Fixed a bug in the auto typography helper that was causing
2924: inconsistent behavior.
2925: Fixed a couple bugs in the Model class.
2926: Fixed some documentation typos and errata.
2927:
2928:
2929:
2930:
2931: <h2>Version 1.3.1</h2>
2932: <p>Release Date: April 11, 2006</p>
2933:
2934:
2935: Added a Unit
2936: Testing Library.
2937:
2938: Added the ability to pass objects to the insert()</
strong>
2939: and update() database functions. This
feature
2940: enables you to (among other things) use your <a
2941: href="./general/models.html">Model class variables
to run
2942: queries with. See the Models page for details.
2943:
2944: Added the ability to pass objects to the <a
2945: href="./general/views.html">view loading function:
2946: \$this->load->view('my_view', <var>\$object</var>);
2947:
2948: Added <kbd>getwhere</kbd> function to <a
2949: href="./database/active_record.html">Active Record clas

changelog.html

```
s</a>.
2950:          </li>
2951:          <li>Added <code>count_all</code> function to <a
2952:              href="./database/active_record.html">Active Record clas
s</a>.
2953:          </li>
2954:          <li>Added language file for scaffolding and fixed a scaffolding
2955:              bug that occurs when there are no rows in the specified
table.</li>
2956:          <li>Added <a href="./database/queries.html">$this->db->last_que
ry()</a>,
2957:              which allows you to view your last query that was run.
2958:          </li>
2959:          <li>Added a new mime type to the upload class for better
2960:              compatibility.</li>
2961:          <li>Changed how cache files are read to prevent PHP errors if
2962:              the cache file contains an XML tag, which PHP wants to
interpret as
2963:              a short tag.</li>
2964:          <li>Fixed a bug in a couple of the active record functions
2965:              (where and orderby).</li>
2966:          <li>Fixed a bug in the image library when realpath() returns
2967:              false.</li>
2968:          <li>Fixed a bug in the Models that was preventing libraries fro
m
2969:              being used within them.</li>
2970:          <li>Fixed a bug in the "exact_length" function of the validatio
n
2971:              class.</li>
2972:          <li>Fixed some typos in the user guide</li>
2973:      </ul>
2974:
2975:
2976:      <h2>Version 1.3</h2>
2977:      <p>Release Date: April 3, 2006</p>
2978:
2979:      <ul>
2980:          <li>Added support for <a href="models.html">Models</a>.
2981:          </li>
2982:          <li>Redesigned the database libraries to support additional
2983:              RDBMs (Postgres, MySQLi, etc.).</li>
2984:          <li>Redesigned the <a href="./database/active_record.html">Acti
ve
2985:              Record class</a> to enable more varied types of
queries with simpler
2986:              syntax, and advanced features like JOINS.
2987:          </li>
2988:          <li>Added a feature to the database class that lets you run <a
2989:              href="./database/call_function.html">custom function ca
lls</a>.
2990:          </li>
2991:          <li>Added support for <a href="controllers.html">private
2992:              functions</a> in your controllers. Any controll
er function name that
2993:              starts with an underscore will not be served by a URI r
equest.
2994:          </li>
2995:          <li>Added the ability to pass your own initialization parameter
s
2996:              to your <a href="creating_libraries.html">custom core l
ibraries</a>
2997:              when using $this->load->library()
2998:          </li>
```

changelog.html

2999: Added support for running standard query string URLs. These can be optionally enabled in your config file.

3000:

3001: Added the ability to specify a "suffix", which will be appended to your URLs. For example, you could add .html to your URLs, making them appear static. This feature is enabled in your config file.

3002:

3003: Added a new error template for use with native PHP errors.

3004: Added "alternator" function in the string helpers.

3005:

3006: Removed slashing from the input class. After much debate we decided to kill this feature.

3007: Change the commenting style in the scripts to the PEAR standard so that IDEs and tools like phpDocumenter can harvest the comments.

3008: Added better class and function name-spacing to avoid collisions with user developed classes. All CodeIgniter classes are now prefixed with <dfn>CI_</dfn> and all controller methods are prefixed with <dfn>_ci</dfn> to avoid controller collisions. A list of reserved function names can be found here.

3009:

3010: Redesigned how the "CI" super object is referenced, depending on whether PHP 4 or 5 is being run, since PHP 5 allows a more graceful way to manage objects that utilizes a bit less resources.

3011: Deprecated: <var>\$this->db->use_table()</var> has been deprecated. Please read the Active Record page for information.

3012:

3013: Deprecated: <var>\$this->db->smart_escape_str()</var> has been deprecated. Please use this instead: <var>\$this->db->escape()</var>

3014: Fixed a bug in the exception handler which was preventing some PHP errors from showing up.

3015: Fixed a typo in the URI class. \$this->total_segment() should be plural: \$this->total_segments()

3016: Fixed some typos in the default calendar template

3017: Fixed some typos in the user guide

3018:

3019:

3020:

3021:

3022:

3023:

3024:

3025:

3026:

3027:

3028:

3029:

3030:

3031:

3032:

3033:

3034:

3035:

3036:

3037:

3038:

3039:

3040:

3041:

3042:

3043:

3044:

3045:

3046:

changelog.html

```
3047:
3048: <h2>Version 1.2</h2>
3049: <p>Release Date: March 21, 2006</p>
3050:
3051: <ul>
3052:     <li>Redesigned some internal aspects of the framework to resolve
e
3053:         scoping problems that surfaced during the beta tests. The problem
3054:         was most notable when instantiating classes in your constructors,
3055:         particularly if those classes in turn did work in their
3056:         constructors.</li>
3057:     <li>Added a global function named <a
3058:         href="ancillary_classes.html">get_instance()</a> allowing the main
3059:         CodeIgniter object to be accessible throughout your own
3060:         classes.</li>
3061:     <li>Added new <a href="./helpers/file_helper.html">File
3062:         Helper</a>: delete_files()
3063:     </li>
3064:     <li>Added new <a href="./helpers/url_helper.html">URL
3065:         Helpers</a>: base_url(), index_page()
3066:     </li>
3067:     <li>Added the ability to create your own <a
3068:         href="creating_libraries.html">core libraries</a> and store them in
3069:         your local application directory.
3070:     </li>
3071:     <li>Added an <kbd>overwrite</kbd> option to the <a
3072:         href="./libraries/file_uploading.html">Upload class</a>
3073:         , enabling files to be overwritten rather than having the file name
3074:         e appended.</li>
3075:     <li>Added Javascript Calendar plugin.</li>
3076:     <li>Added search feature to user guide. Note: This is done using
3077:         Google, which at the time of this writing has not crawled all the
3078:         pages of the docs.</li>
3079:     <li>Updated the parser class so that it allows tag pairs within
3080:         other tag pairs.</li>
3081:     <li>Fixed a bug in the DB "where" function.</li>
3082:     <li>Fixed a bug that was preventing custom config files to be
3083:         auto-loaded.</li>
3084:     <li>Fixed a bug in the mysql class bind feature that prevented
3085:         question marks in the replacement data.</li>
3086:     <li>Fixed some bugs in the xss_clean function</li>
3087: </ul>
3088:
3089:
3090:
3091:
3092:
3093: <h2>Version Beta 1.1</h2>
3094: <p>Release Date: March 10, 2006</p>
3095:
3096: <ul>
3097:     <li>Added a <a href="./libraries/calendar.html">Calendar class</a>.
3098:
3099:     </li>
```


changelog.html

```
3154:          </div>
3155:
3156: </body>
3157: </html>
```

license.html

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml
tml1/DTD/xhtml1-transitional.dtd">
2: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3: <head>
4:
5: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6: <title>CodeIgniter License Agreement : CodeIgniter User Guide</title>
7:
8: <style type='text/css' media='all'>
9: @import url('userguide.css');
10: </style>
11: <link rel='stylesheet' type='text/css' media='all' href='userguide.css' />
12:
13: <script type="text/javascript" src="nav/nav.js"></script>
14: <script type="text/javascript" src="nav/prototype.lite.js"></script>
15: <script type="text/javascript" src="nav/moo.fx.js"></script>
16: <script type="text/javascript" src="nav/user_guide_menu.js"></script>
17:
18: <meta http-equiv='expires' content='-1' />
19: <meta http-equiv='pragma' content='no-cache' />
20: <meta name='robots' content='all' />
21: <meta name='author' content='ExpressionEngine Dev Team' />
22: <meta name='description' content='CodeIgniter User Guide' />
23:
24: </head>
25: <body>
26:
27:     <!-- START NAVIGATION -->
28:     <div id="nav">
29:         <div id="nav_inner">
30:             <script type="text/javascript">create_menu('null');</script>
31:         </div>
32:     </div>
33:     <div id="nav2">
34:         <a name="top"></a><a href="javascript:void(0);"
35:         onclick="myHeight.toggle();"></a>
38:     </div>
39:     <div id="masthead">
40:         <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
41:             <tr>
42:                 <td><h1>CodeIgniter User Guide Version 2.2.2</h1></td>
43:                 <td id="breadcrumb_right"><a href="toc.html">Table of
44:                 Contents Page</a></td>
45:             </tr>
46:         </table>
47:     </div>
48:     <!-- END NAVIGATION -->
49:
50:
51:     <!-- START BREADCRUMB -->
52:
53:     <table cellpadding="0" cellspacing="0" border="0" style="width: 100%">
54:         <tr>
55:             <td id="breadcrumb"><a href="http://codeigniter.com/">CodeIgnit
er
56:             Home</a> &nbsp;&#8250;&nbsp;&nbsp;&nbsp;<a href="index.htm
l">User Guide Home</a>
57:             &nbsp;&#8250;&nbsp;&nbsp;&nbsp;License Agreement</td>
58:             <td id="searchbox"><form method="get"
```

license.html

```
59:                                     action="http://www.google.com/search">
60:                                     <input type="hidden" name="as_sitesearch" id="a
s_sitesearch"
61:                                     value="codeigniter.com/user_guide/" />S
earch User Guide&nbsp; <input
62:                                     type="text" class="input" style="width:
200px;" name="q" id="q"
63:                                     size="31" maxlength="255" value="" />&n
bsp;<input type="submit"
64:                                     class="submit" name="sa" value="Go" />
65:                                     </form></td>
66:                                     </tr>
67:                                     </table>
68:                                     <!-- END BREADCRUMB -->
69:
70:
71:
72:                                     <br clear="all" />
73:
74:
75:                                     <!-- START CONTENT -->
76:                                     <div id="content">
77:
78:                                         <h1>CodeIgniter License Agreement</h1>
79:
80:                                         <p>
81:                                             Copyright (c) 2008 - 2014, EllisLab, Inc.<br /> Copyright (c) 2
014 -
82:                                             2015, British Columbia Institute of Technology<br /> All rights
83:                                             reserved.
84:                                         </p>
85:
86:                                         <p>This license is a legal agreement between you and the British
87:                                             Columbia Institute of Technology for the use of CodeIgniter Sof
tware
88:                                             (the "Software"). By obtaining the Software you agree to comply
with
89:                                             the terms and conditions of this license.</p>
90:
91:                                         <h2>Permitted Use</h2>
92:                                         <p>You are permitted to use, copy, modify, and distribute the
93:                                             Software and its documentation, with or without modification, f
or any
94:                                             purpose, provided that the following conditions are met:</p>
95:
96:                                         <ol>
97:                                             <li>A copy of this license agreement must be included with the
98:                                                 distribution.</li>
99:                                             <li>Redistributions of source code must retain the above
100:                                                 copyright notice in all source code files.</li>
101:                                             <li>Redistributions in binary form must reproduce the above
102:                                                 copyright notice in the documentation and/or other mate
rials
103:                                                 provided with the distribution.</li>
104:                                             <li>Any files that have been modified must carry notices statin
g
105:                                                 the nature of the change and the names of those who cha
nged them.</li>
106:                                             <li>Products derived from the Software must include an
107:                                                 acknowledgment that they are derived from CodeIgniter i
n their
108:                                                 documentation and/or other materials provided with the
distribution.</li>
```


XI. Acknowledgement

This Special Project will not materialize without the collective effort of me, my family, friends, the normal ones and the special. To name a few of them,

Elizabeth Macaraeg, my mother, my loving, caring mother who supported me every step of the way and had given me the hope and courage that no matter what happens there is a brighter day ahead and that she and my little brother will aide me through it.

Peter Jericho Macaraeg, my little brother. The one true love of my life that served as my inspiration for I would do anything for him and my mother no matter what happens to me. When shit happens, I think of them. The people who serve as my inspiration not to stop and always move forward and progress.

Ma. Sheila Magboo, my thesis adviser who helped me every step of the day, who helped me achieve my dream of doing the graduation march. My adviser with whom I shared not only academic stuffs but also my life stories and aspirations and advised me not only to complete my special project but advised me on life things and all. O my gosh. Im crying. Really as I write this acknowledgement. My mother told me not to write anything in Filipino but here it goes because you know life:

Salamat ng madami sa inyong lahat. Kay mama, ma alam mo yun haha! Sobrang sakit ng mga nangyari sa buhay natin pero huwag kang magalala magsisikap ako ng super para sa atin. Darating yung panahon na ipababasa ko sa iyo itong acknowledgement na ito tapos tatawanaan na lang natin yung mga iyon. Ma super. Love you. As in. Kayo ni Pedro. Super. Love you. Wala akong hindi kayang tapusin, gawin para sa inyo.

Ma'am Sheila! Ma'am naiiyak ako kasi you were not only my adviser but served as a friend as well. Naiiyak ako hahaha shzzzz. Kasi hindi mo lang ako tinulungan sa paper ko pero each time we talk you give me advices on life as well. For that I thank you big time. All in one ka ma'am di ka lang adviser parang second mama na din. Sa mga tito at tita ko na lagi akong kinocongrats sa FB dahil sa mga pictures ko at napepressure tuloy ako na magtapos lalo. Love you. Guys hindi ako magaling sa salita pero pinaparamdam ko sa inyo. Love you Super. Super. Super.