

University of the Philippines Manila
College of Arts and Sciences
Department of Physical Sciences and Mathematics

Filipino Language Learning System (FILLES)
for Desktop Computers / Tablets

A Special Problem in partial fulfilment
of the requirements for the degree of
Bachelor of Science in Computer Science

Submitted by:

Bona Rae P. Villarta
2007-03380

April 2013

ACCEPTANCE SHEET

The Special Problem entitled “Filipino Language Learning System (FILLES) for Desktop Computers/Tablets” prepared and submitted by Bona Rae P. Villarta in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

Geoffrey A. Solano, Ph.D. (candidate)

Adviser

EXAMINERS:

	Approved	Disapproved
a) Gregorio B. Baes, Ph.D. (candidate)	_____	_____
b) Avegail D. Carpio, M.S.	_____	_____
c) Richard Bryann L. Chua, Ph.D. (candidate)	_____	_____
d) Aldrich Colin K. Co, M.S. (candidate)	_____	_____
e) Perlita E. Gasmen M.S. (candidate)	_____	_____
f) Ma. Sheila A. Magboo, M.S.	_____	_____
g) Vincent Peter C. Magboo, M.D., M.S.	_____	_____
h) Bernie B. Terrado, M.S. (candidate)	_____	_____

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

Avegail D. Carpio, M.S.

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.

Dean

College of Arts and Sciences

ABSTRACT

The Filipino Language Learning System (FILLES) is a fully-customizable system oriented towards self-instructed and self-directed language learning. The main objective of this system is to make the Filipino language more accessible to people around the world, particularly those with little to no background in the Filipino language and would like to learn the language for different purposes. A variety of both traditional and modern language learning methods are exhibited in this interactive system which utilizes the use of modern web technologies (HTML5). The system allows the integration of these two kinds of methods. While the system is simple in design, it is comprehensive enough to let the users gain proficiency in the language.

KEYWORDS: Computer-assisted language learning, Self-instructed language learning, Filipino language, Language instruction, Computer-assisted instruction

TABLE OF CONTENTS

Acceptance Sheet	i
Abstract	ii
I. Introduction	2
A. Background	2
B. Statement of the Problem	3
C. Objectives of the Study	5
D. Significance of the Problem	6
E. Scope and Limitations	7
F. Assumptions	8
II. Review of Related Literature	9
III. Theoretical Framework	12
IV. Design and Implementation	16
V. Results	34
VI. Discussion	45
VII. Conclusion	46
VIII. Recommendation	47
IX. Bibliography	48
X. Appendix	52
XI. Acknowledgements	252

CHAPTER I

INTRODUCTION TO THE STUDY

A. Background

In today's increasingly globalized society, there is pressure among people around the world to become proficient in as many languages as possible, especially languages of more populous countries such as Mandarin Chinese and Hindi. This is in addition to the usual need for language proficiency in French, Spanish, German and English. Technology plays a significant role in facilitating learning languages that are not native to the speaker, but its role is dependent on what type of technology is involved and how that type is used towards foreign language learning. [1]

The author has observed for several years that technology-driven learning tools for Filipino language (Tagalog-based) are somewhat, if not locally produced and/or locally-oriented, lacking. (Majority of the resources of the Filipino language that are available on the Internet are produced by expatriates or foreign companies.) The market sector for this type of tool is overlooked by companies, especially Filipino companies involved in education, literature and tourism. The target markets for these language learning tools include:

1. *Tourists.* The Philippine government has been promoting tourism as an important program for years; however, only recently they have been aggressively promoting it through the Internet.
2. *Expatriate foreigners working in the Philippines.* The country is a hub for global companies employing foreign workers.

3. *Expatriate Filipinos and their families* who desire to rekindle their connection with their cultural heritage. From the author's own experience, many children of her Filipino relatives and friends who were born and/or raised abroad, even those with full Filipino blood, do not know or have no inclination to learn the Filipino culture and language.

The author believes that with this system, persons willing to learn Filipino, regardless of ethnic origin, will be provided access for learning language proficiency in the Filipino language. It is also envisioned that through this system, the Filipino language will be more appreciated by more people, particularly tourists visiting or foreigners intending to settle in the Philippines.

The Sentro ng Wikang Filipino (SWF) is an extension office of the UP System that focuses on utilizing and maximizing the use of the Filipino language in the university system with offices in every UP campus, however, the two most active SWF offices are those in UP Diliman (UPD) and in UP Manila (UPM). SWF-UPD publishes books about Filipino grammar, style and usage regularly [2], while SWF-UPM hasn't published any book on its own since 1998, according to its current chairperson, Prof. Odessa N. Joson, but carries out Filipino translations of books published by UPM community members from time to time.

B. Statement of the Problem

Mobile computer devices like smartphones and tablets are becoming increasingly popular. Internet searches of applications in the iTunes and Google Play mobile application stores, along with some online software stores such as Herman Street and Amazon.com, conducted by the author in September 2012, revealed that there are no applications for either platforms that offer Filipino as a language offered for learning within an application. The

common applications that claim that users can learn many languages only offers European and some Asian languages, such as Chinese, Japanese, Hindi, Thai and Vietnamese. These applications for Android systems of Google supports up to 50 languages, and for iOS systems of Apple supports up to 40 languages [3], none of which include Filipino (Tagalog). On the commercial language learning marketplace, among the major language software developers, Rosetta Stone [4] offers solutions for learning the Filipino language on desktops or online. Innovative Languages offers a solution for learning Filipino via podcasts and the web [5]. The other major language software developers, Tell Me More, Mango Languages and Transparent Language do not currently offer solutions for learning the Filipino language. If the learning system involves books in addition to multimedia, only Tuttle Publishing [6] provides such learning systems for the Filipino language.

On the other hand, locally-published commercial Filipino learning systems, which are either paper-based or e-learning-based, focus on students currently in Philippine schools, not adults, foreigners or children of expatriates, who all tend to learn languages independently and selectively. Among these are the Wikahon [7], a paper-based Filipino learning system primarily for elementary students, by children's book publisher Adarna House and the Genyo e-learning system [8], a fully-integrated online learning system by educational materials publisher DIWA Learning Systems, which includes a Filipino component within that system for elementary and secondary students.

C. Objectives of the Study

This special project envisions a fully-interactive system of learning the Filipino language. It mainly targets older children of Filipino migrants / expatriates or Filipino migrants who had forgotten their grasp of the language. The special project will provide the following functions:

- 1) Allow learners, or registered users, to:
 - a) Explore the language through different self-instructed lessons.
 - b) Take mastery tests after each lesson is completed.
 - c) View a customized dictionary and phrasebook, complete with audio pronunciations and syllabic pronunciations of terms.
 - d) View a customized grammar guide to the Filipino language.
 - e) Check their proficiency and mastery test scores.
 - f) Leave suggestions/comments to the content administrator for improvement of the system or to the staff for improvement of the guides through change requests.
- 2) Allow the guests, or unregistered users, to:
 - a) View the dictionary, phrasebook and grammar guide for their own use;
 - b) Enroll themselves as registered users to fully participate in the system or to contribute links containing content to the system.
- 3) Allow contributors to:
 - a) Contribute links that contain content for the learners' enjoyment, and can edit or delete their own contributed links as necessary;
 - b) Suggest guide changes to SWF-UPM staff.
- 4) Allow SWF-UPM staff to:
 - a) Add, edit or delete lessons and links;

- b) Maintain the dictionary, phrasebook and grammar guide of the system by adding, editing or deleting the contents of each of these;
 - c) Maintain the question bank by adding, editing, or deleting mastery test questions.
 - d) Maintain the instruction list for the question bank by adding, editing or deleting instructions.
 - e) Review change requests to the guides made by the registered users and contributors.
- 5) Allow the content administrator to:
- a) Approve links to other sites created by the contributors and SWF-UPM staff;
 - b) Respond to suggestions/comments made by other users and guests.
- 6) Allow the system administrator to:
- a) Maintain the list of users and user types in the system by adding, editing or deleting such entities;
 - b) Add additional system administrators or content administrators;
 - c) Add additional staff members as necessary.

D. Significance of the Problem

The Filipino language is an under-appreciated language outside of the Filipino community, especially in the academe [9]. What Manuel L. Quezon envisioned in 1935, for Filipino to become the single language of our people, is still far from being fulfilled, since the regional languages, such as Cebuano and Ilocano, are still being informally taught to older children and adults in those areas where the language is prevalent. Among older generations, there are still fluent speakers of the regional languages, and for a time period in the mid-to-late 20th century, Cebuano speakers were hesitant to accept Tagalog as the national language and the

provincial government of Cebu even fought against the national government's decision to use Filipino/Tagalog because of the pride they took in their language; which is one possible reason why Cebuano is still used by many Filipinos today, especially in the cities and provinces of central Visayas and the Christian regions of Mindanao [10].

However, the author has observed that most Filipinos are hesitant to use the English language or in the provinces, even the national language when talking with anyone regardless of origin, preferring to use the national or regional language most of the time, hence we often have problems communicating with foreigners or out-of-region people; also most *balikbayans* who return home after years of living abroad often have forgotten the language, often resorting to English when talking with their relatives, and that is also an alarming problem as well, and the author has turned her focus towards them.

What Filipino software developers can do to attain Quezon's vision is to develop solutions that bridge the various Philippine languages together and with English, the lingua franca of the world. However, as an initial step, the author proposes a Filipino interactive learning system for mobile tablets and desktops that caters to tourists, Filipino expatriates / migrants and their children.

E. Scope and Limitations

- a) The application's coverage of Filipino grammar and vocabulary is only up to the intermediate level because the target users (primarily preteens, adolescents and adult learners) only need conversational Filipino.

- b) This Special Problem is assisted and collaborated with the Sentro ng Wikang Filipino – UP Manila (SWF-UPM), in which the latter will provide the lessons, exams, dictionary, grammar guide and phrasebook for this SP.
- c) Links are screened for appropriateness and need to be approved by the content administrator before these can become publicly accessible.
- d) It is assumed that the browser the user is using is of a modern standardized one (as noted in the Technical Architecture) so that the HTML5 standard is fully recognizable by the browser. Otherwise, the user’s performance may not be at a full potential.
- e) In the question bank, only true or false and multiple choice types of quiz items are supported.
- f) In the audio files that accompany guide entries, only wave format (.wav) files are supported to comply with browsers’ universal audio standards.
- g) In the picture files, only .jpg, .gif and .png files are supported so that the system is compliant with browsers’ universal image display standards.
- h) Lessons are only supported in .pdf format so that the viewing standards for these are consistent.

F. Assumptions

1. The learner should have at least intermediate proficiency of the English language, since the instructions and website portal will be written in English.
2. The learner should have at least some motivation to learn the Filipino language.
3. The learner should have a PDF viewer installed to view the lessons.
4. The learner should have a sound card installed to hear audio.

CHAPTER II

REVIEW OF RELATED LITERATURE

Foreign-language learning, its various sub-types and its effects on the learners have been well-studied in researches concerning various systems, methods and languages, however, majority of these studied languages are commonly-learned languages; and the Filipino language has not been well-studied in foreign research literature.

Computer-Assisted Language Learning (CALL), the main category of this system, has been in use since the late 1960's [13] and with the increased use of computers in the decades that followed, continues to become a more important category in learning.

There have been a few studies comprehensively reviewing commercially-available integrated language learning systems in third-party research literature, which are found in [14, 15]. However, the subcategory of CALL that both the system and similar systems belong to – self-instructed language learning – has not been well-researched as evidenced in [16].

Although Nedeva & Dimova [17] devised a similar system based online, they used two different but complementary technologies in implementing their system; they adapted a freeware instrumental work environment, Hot Potatoes [18] to create exercises for a foreign language learning system, in their case, in English; while their grading and assignment system was based on a popular open source grading software, Moodle [19]. However, this system is entirely based online and utilizes open source software.

Cultural activities, which are covered in this system, have been covered in relation to CALL in [20]. While integration of cultural activities in foreign language programs is not new, some implementers have had difficulty in integrating such activities in a global context such as

that of a CALL system, in which the target users live around the globe and are of different heritages, faiths and even races. Since this project is targeted towards Filipino expatriates / migrants, the content will reflect the ever-changing landscape of Filipino culture as it spreads around the world, both online and offline.

Challenges to Implementing the Solution

There are challenges in implementing a fully-integrated solution of a foreign language learning system. Aside from lack of interaction, the lack of comprehensible input and output opportunities hinder the potential benefits such a solution might offer [21, 22] in addition to the additional characteristics individual learners should possess [22]. The system is a self-regulated distance learning model that intends to offer learners the basic fundamentals of the Filipino language.

In distance education, Moore's theory of transactional distance includes three key variables: dialogue, structure and learner autonomy. Dialogue refers to student-teacher interaction, structure is the degree the program accommodates to learners' preferences and needs, and learner autonomy allows the learner to explore further resources on his own. Autonomy is represented in distance education as a way of the learner having a choice of what to expect in learning a course through that method [21].

Autonomy is achieved by most language learners learning the language independently but its success is dependent on many behaviors, among these include grasp of the different strategies used in language learning, willingness to learn such languages, and self-commitment to the desired area of learning [21].

Success of distance language learners is also dependent on how the course was designed – regardless of the complexity of the language whose system to be designed is based on, so that sufficient learner support within the course and flexibility in learners' learning styles are both achieved. However, when the student is learning on his own, he does not really learn well nor think critically in that language. [21]

CHAPTER III

THEORETICAL FRAMEWORK

A. Filipino Language

The Filipino language, a member of the Malayo-Polynesian language family and closely related to the Bahasa Malaysia and Bahasa Indonesia languages [23], is the national language of the Philippines. It is based on the Tagalog language, the language spoken by the people of Manila at the time of the Spanish conquest, and throughout the recorded and inferred history of the Philippines, has had many words inherently borrowed from Spanish, English, Japanese, Sanskrit, Chinese (particularly Cantonese), Arabic and of course, ancient Malay, which forms the core of the language.

Tagalog, in its pure form, is closely related to the other Philippine regional languages and the Bahasa languages in terms of grammatical structure and word formation, since they come from the same branch of the said language family. [23] However, some contrasting studies over the first half of the twentieth century, explained in both [12] and [23], had analyzed the similarities of Tagalog with other Philippine and regional languages and those in other language families, hence, the language, like the Filipino people themselves and also Filipino culture, are also a ‘melting pot’; their interdependent evolution continues to this day.

B. Language Learning System

A language learning system is a system that uses a computer or a similar device, such as a mobile phone or a tablet, for a student to learn a foreign language, usually English. Common

features in most integrated language learning systems for desktops or tablets include games, tests, exercises, word processing and speech recognition. [24]

Components of a language learning system vary depending on the subtype and function of the system [25]. Common components include lessons in any virtual learning medium (e.g. on a computer screen), speech recognition, training of grammatical structure using natural language processing [26], and grammatical practice exercises [17]. Son's article [26] states three fundamental components of such a system: the learner, the computer, and the teacher.

A language learner (hereinafter referred to as learner) is a person who learns the language, either through the traditional approach or using a CALL-based system. The learner's success in CALL-based systems is dependent on five variables: age, background, ability, cognitive style and effect [27]. The learner's familiarity with the computer should also be considered so that meaningful activities should be given to the learner. It is for this reason that now, as computers have a significant role in people's lives today, the teachers will have to be familiarized with CALL environments.

C. Language Learning

Computer-assisted or computer-aided language learning (CALL) is a category of computer-aided learning (CAL) that exclusively deals with learning languages [24]. Over the years, CALL has shifted from a specific approach (grammar and translation-related exercises) in the 1970's [13] to a more general, web-based integrated approach in the mid-2000's [24], in which the latter is still the trend in CALL today.

CALL's roots are in language learning and applied linguistics. Although it has a lot in common with computer linguistics, both disciplines are independent of each other because of misunderstanding about each other's disciplines, culture and language learning ideologies. [28]

D. Information System

An information system is an organized set of components for collecting, transmitting, storing and processing data in order to deliver information for action. In the twenty-first century, information technology emerges as the fundamental technology of business. There are several types of information systems, the one type most relevant to this problem is that of the office information system – which supports and helps coordinate knowledge work in an office environment by handling documents and messages in a variety of forms, in which the content management is handled by this type of information system. Knowledge work, in which abstract data instead of tangible materials are handled, is also handled by the professional information system – which supports performance of tasks specific to a given profession, in this case, that of the language learning profession. [29]

E. HTML5

HTML5 is the World Wide Web Consortium (W3C)'s solution to standardize and put together the various Web technologies that all appeared separately throughout the mid and late 1990's and the 2000's. Additionally, it is also the open source community's answer to many proprietary technologies, primarily Adobe's Flash. It is composed of HTML version 4, JavaScript, Cascading Style Sheets (CSS) version 3, and the most significant addition of all, native support for multimedia and dynamic programmable content akin to what can be done in

Flash. [30] Support for Scalable Vector Graphics (SVG) supplements the visual multimedia part of HTML5. Although the standard is still under active development, however, it has already been supported by modern browsers during the early stages of its development that even without an official final version of the standard, developers can already use its elements in their code.

F. CodeIgniter PHP Framework

CodeIgniter [31] is an open-source PHP framework developed and supported by EllisLab, based on the Model-View-Controller (MVC) paradigm. It aims to provide developers with a simple and elegant toolkit to create full-featured web applications. It also claims to have a small server footprint. One advantage of this framework is that it is well-documented and its setup is easier than similar frameworks such as Zend and CakePHP [32].

CHAPTER IV

DESIGN AND IMPLEMENTATION

A. System Design

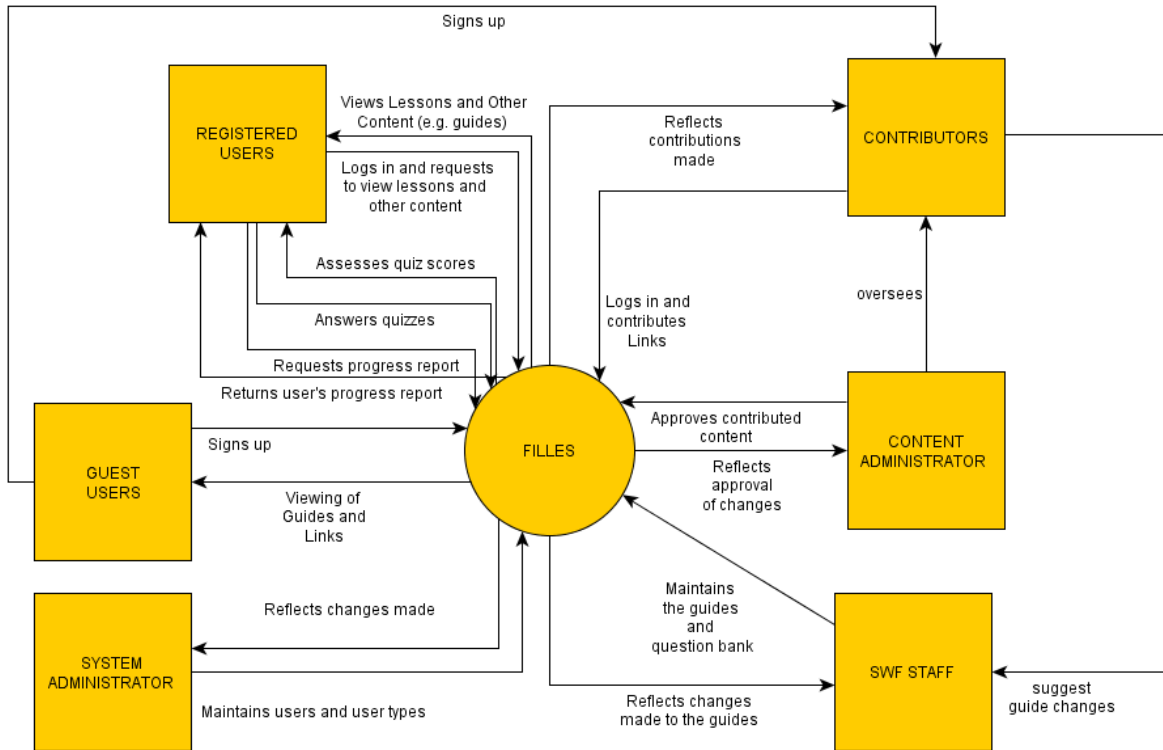


Figure 1: FILLES System Design

The overall users and functions of this system are shown in Figure 1. The image defines the types of users and how different users have roles in the system. The different user types are as follows:

- The guest is a site visitor. He/she is any unlogged user of the website. He/she can view the three guides and approved links and can also sign up to either become a registered user, in which he/she can use the site upon validation of his/her email or become a contributor, in which the guest is allowed to contribute links to the system.

- A registered user is a regular user. He/she has full access to view and participate in the lessons, and can also view approved links and the three guides. He/she can also email the SWF staff for any concern about the guides or the system.
- A contributor is a special user whose roles in the system are to contribute links for approval of the content administrator and to suggest guide changes or voice out concerns to SWF-UPM staff.
- A content administrator is usually the head of SWF-UPM. He/she checks the contributed links the contributors have submitted for approval; if approved, only then will the contributed link publicly appear on the site.
- Members of the SWF-UPM staff will be responsible for maintaining the lessons, question bank and the guides of the system.
- The system administrator installs the system and also manages it through maintenance of users and user types. He/she can work with the content administrator in decision-making for the system, especially on the technical aspect.

Data Flow Diagrams

Level 0.0: Context-Level DFD

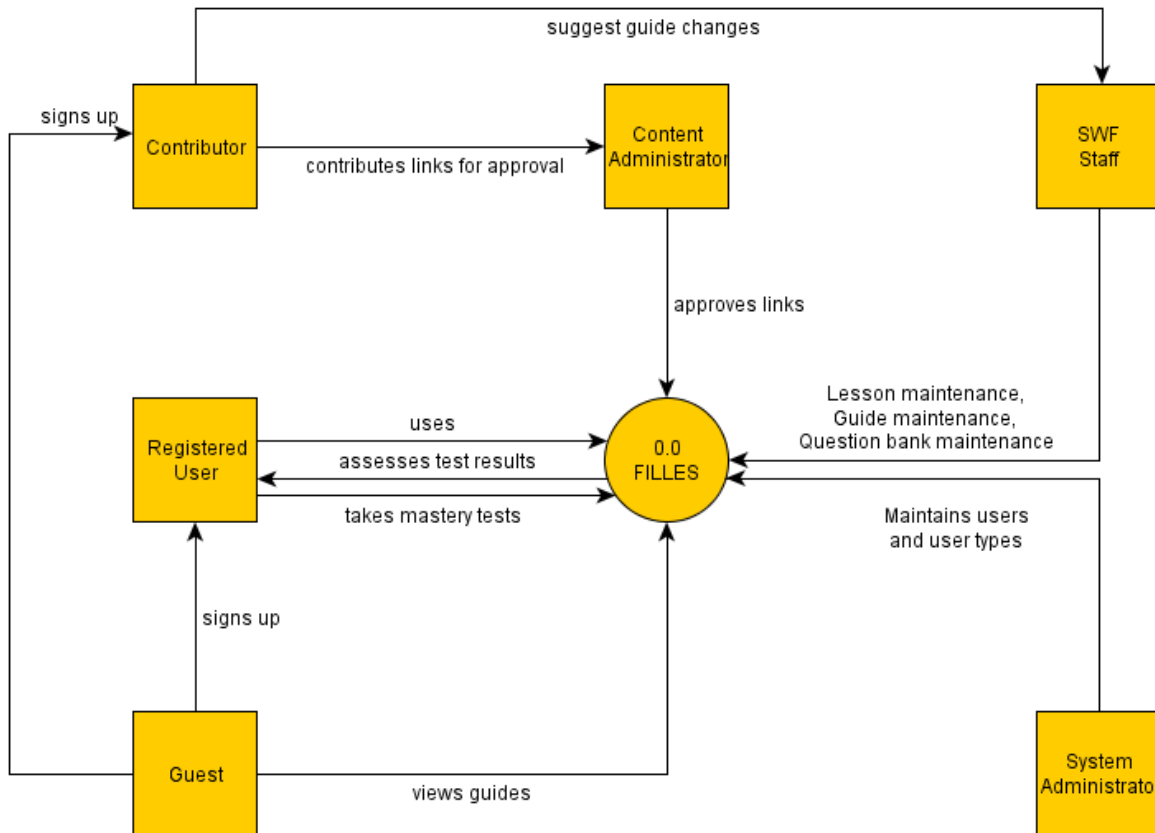


Figure 2: Context-Level DFD for FILLES

The context-level DFD reflects the system's non-reciprocal role except for its role with the registered users. Majority of the roles have interaction with the system, while the contributors do not have a direct role in the system, since their works require screening for language appropriateness/difficulty and approval by the content administrator.

Level 0: Top Level DFD

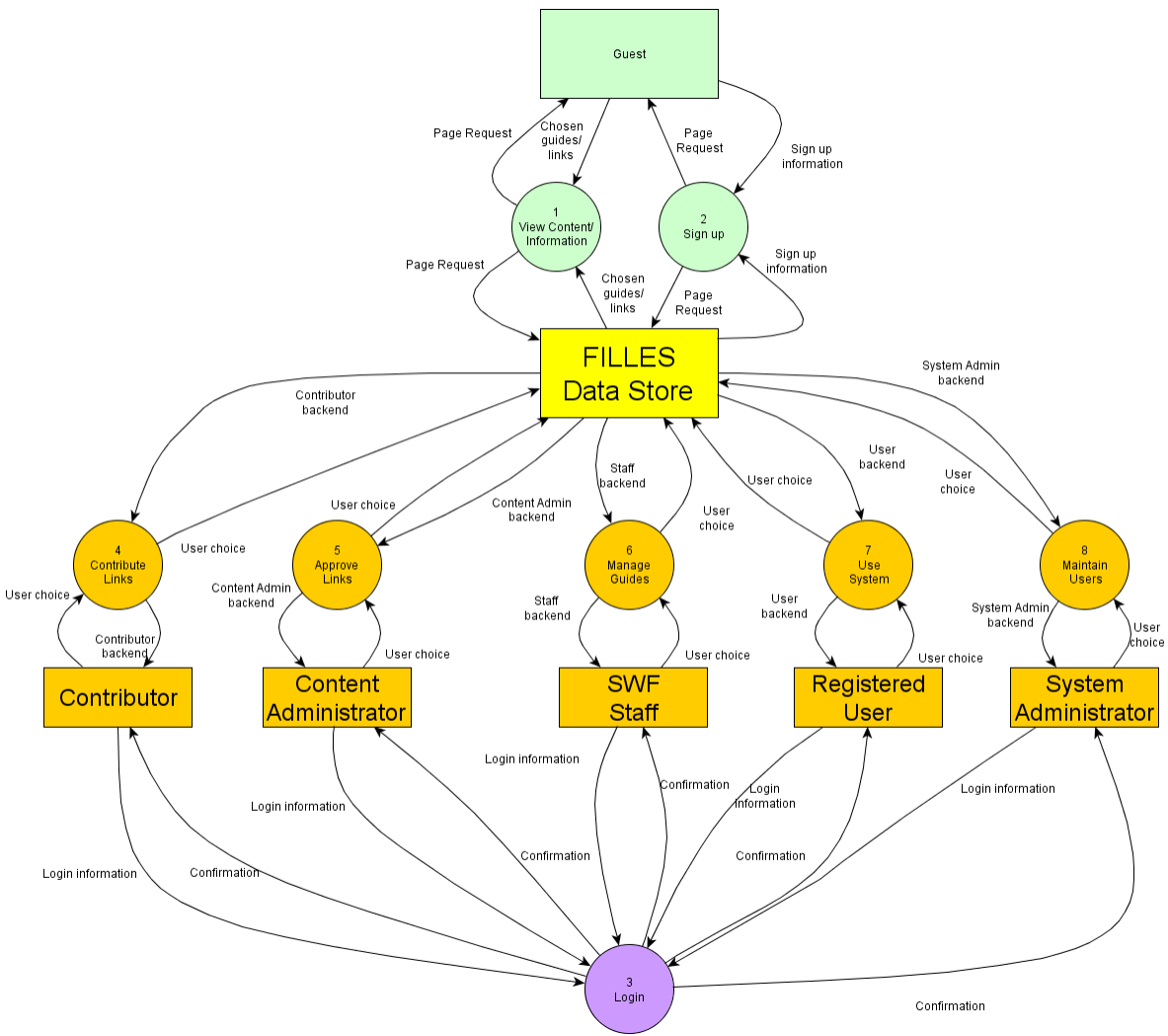


Figure 3: Top Level DFD

FILLES has five (6) user types: Guest, Contributor, Content Administrator, SWF Staff, System Admin and Registered User; the latter five user types have login privileges. The central Data Store stores nearly all of the required information that the system requires. There are seven (7) sub-explussions described herein; each of the five user types with login privileges has functions that are specific for each type of user.

DFD Sub-Explosions

1. 1.0: View Content/Information

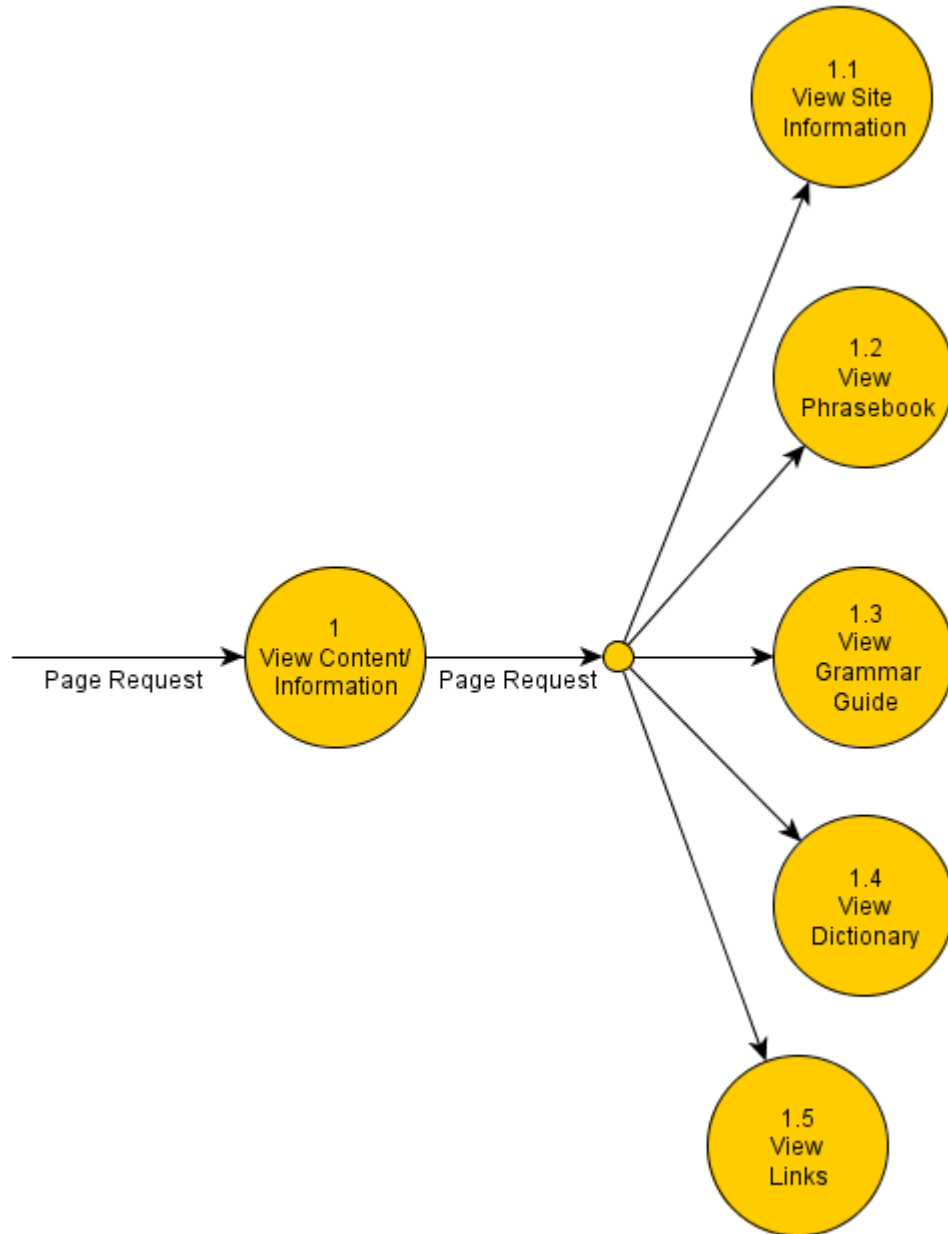


Figure 4: Sub-Explosion 1.0

All unlogged users have access to this section.

‘View Site Information’ allows the user to read about the system in general.

‘View Phrasebook’ allows the user to view a phrasebook of basic Filipino terms/expressions and their English equivalents.

‘View Grammar Guide’ allows the user to view basic grammar rules in the Filipino language.

‘View Links’ allows the user to view approved links either posted by contributors, which are then approved by the content administrator or links posted by the SWF-UPM staff.

‘View Dictionary’ allows the user to view a customized dictionary of basic Filipino terms.

2. 2.0: Sign up

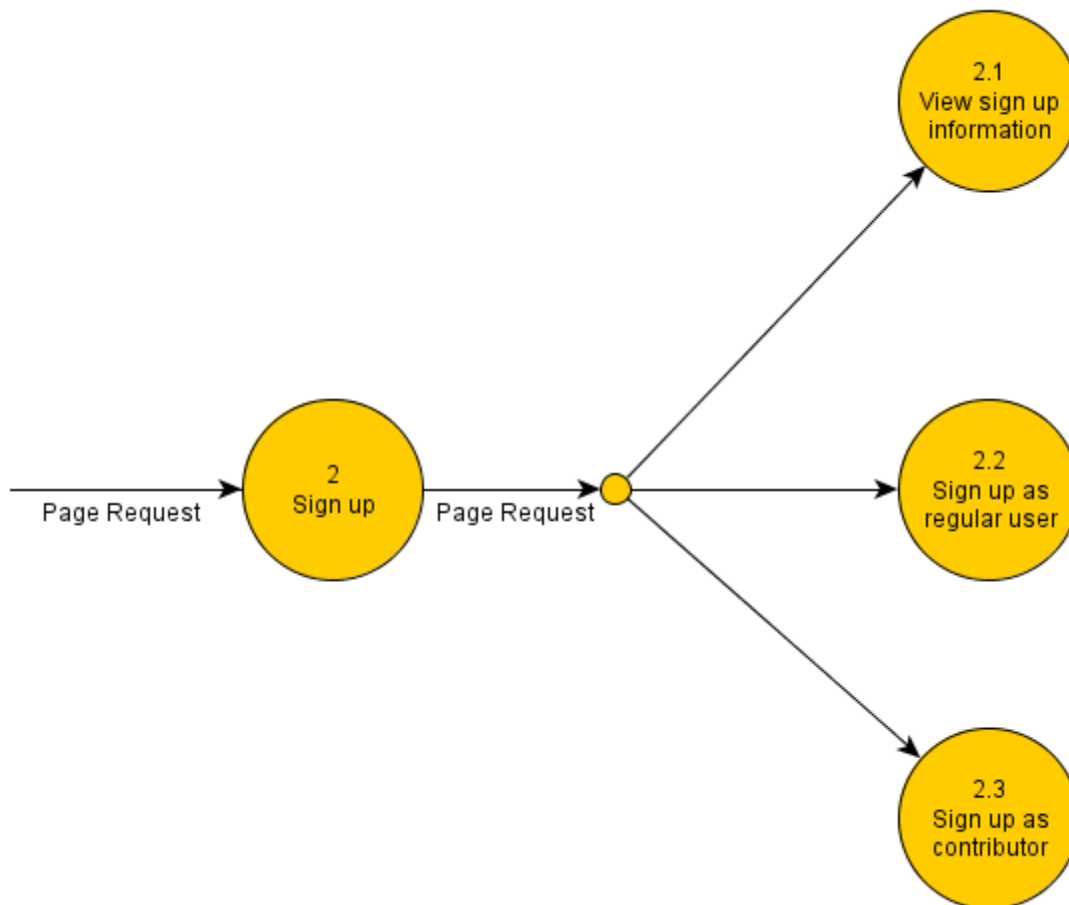


Figure 5: Sub-Explosion 2.0

All unlogged users have access to this section.

‘View sign-up information’ orients the user to how different user accounts work in this system.

‘Sign up as regular user’, when completed, allows the user to log in as a regular user and enables him/her to use the system.

‘Sign up as contributor’, when completed, allows the user to log in as a contributor and enables him/her to use the contributor backend, explained in a later topic.

3. 4.0 Contribute Links

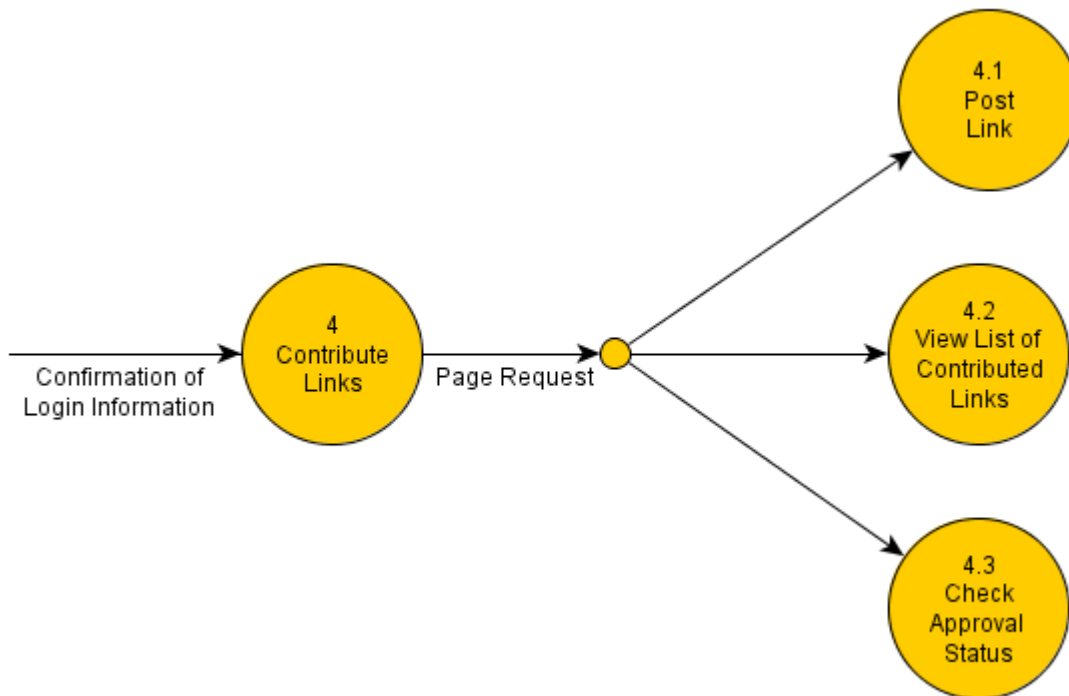


Figure 6: Sub-Explosion 4.0

Only contributors and SWF-UPM staff have access to this section.

‘Post Link’ allows the contributor to post a link for approval.

‘View List of Contributed Links’ allows the contributor to access a list of his/her own approved links, and he/she can edit any of his/her approved links from that screen.

‘Check Approval Status’ informs the contributor about the status of his/her uploaded links that have been approved or rejected by the content administrator.

4. 5.0 Content Admin Functions

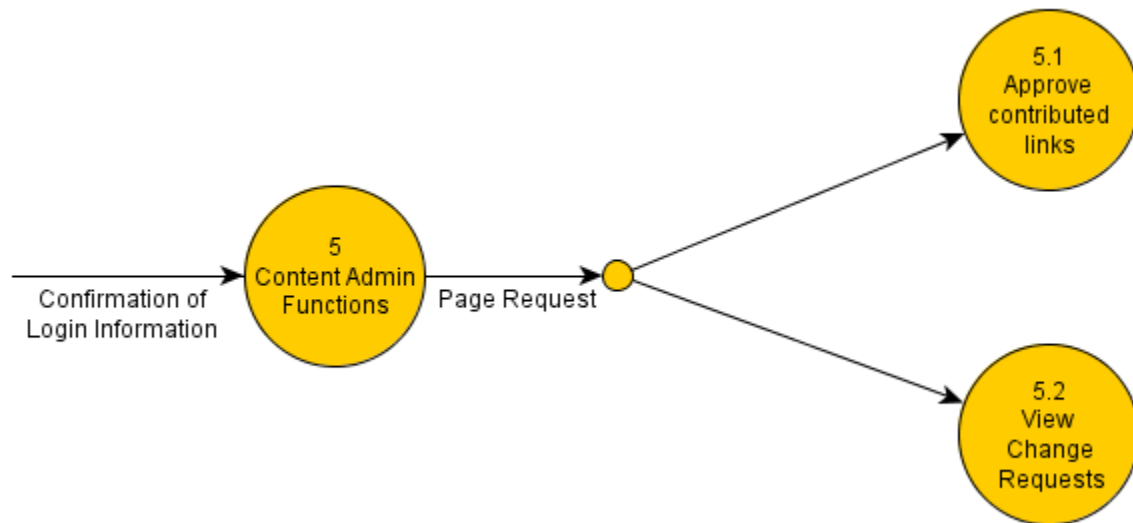


Figure 7: Sub-Explosion 5.0

Only content administrators have access to this section.

‘Approve contributed links’ allows him/her to access a list of links which have been submitted for approval. He/she can accept or reject each link submitted for approval.

‘View change requests’ allows him/her to review general suggestions/comments made by registered users and contributors and allow him/her to coordinate with the staff and the system administrator.

5. 6.0 SWF Staff Functions

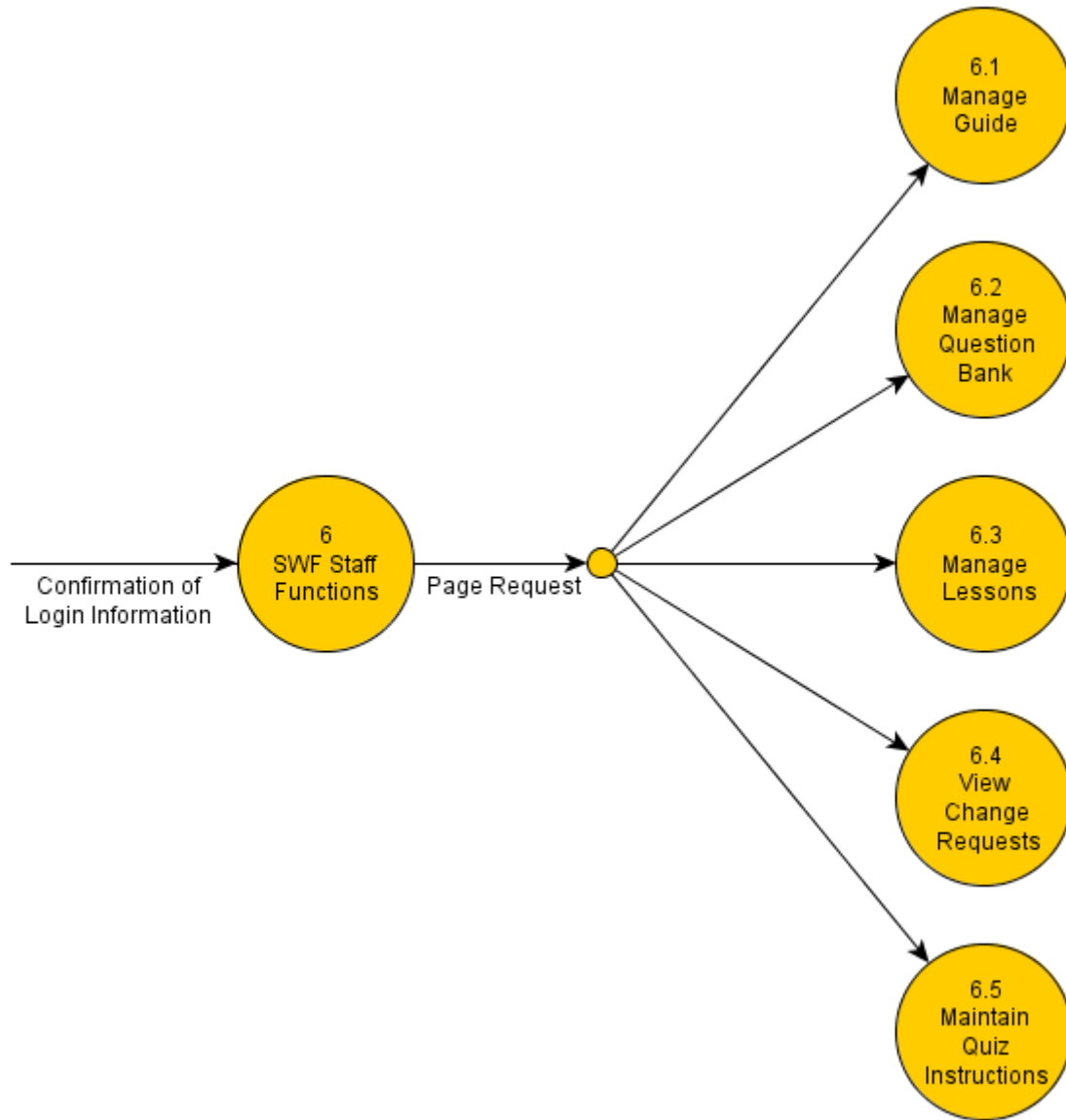


Figure 8: Sub Explosion 6.0

Only SWF-UPM staff has access to this section.

‘Manage Guide’ allows the staff to add, edit, or delete contents of the three guides.

‘Manage Question Bank’ allows the staff to add, edit, or delete questions for the mastery tests.

‘Manage Lessons’ allows the staff to add or edit contents of lessons in the system.

‘View change requests’ allows the staff to review guide changes made by registered users and contributors and allow the staff to coordinate with the content administrator and the system administrator.

‘Maintain Quiz Instructions’ allows the staff to add, edit or delete instructions to be displayed in the end-of-lesson quizzes.

6. 7.0 Use System

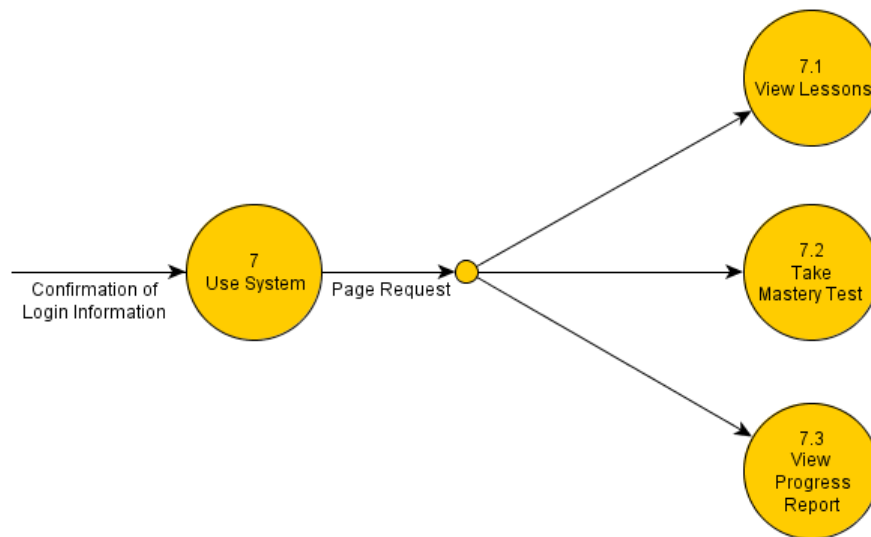


Figure 9: Sub-Explosion 7.0

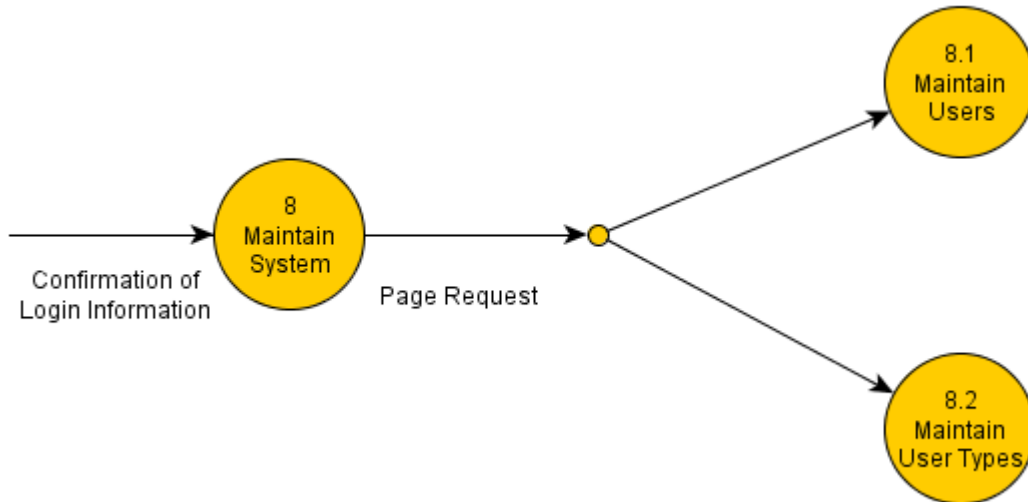
Only registered users have access to this section.

‘View Lessons’ allows the user to take lessons, which are the heart of this system.

‘Take Mastery Test’ allows the user to take a mastery test covering one lesson (accessible from the lesson’s page itself); questions are randomly selected from the question bank.

‘View Progress Report’ allows the user to view his/her progress report for each of the mastery tests he/she has already taken.

7. 8.0 Maintain Users



Only system administrators have access to this section.

‘Maintain Users’ allows the admin to maintain the list of users ever registered in the system. He/she can add users, edit users’ details or delete inactive users from the system.

‘Maintain User Types’ allows the admin to add new, edit existing or remove irrelevant/obsolete user types from the system.

Entity Relationship Diagram: Guides and Lessons

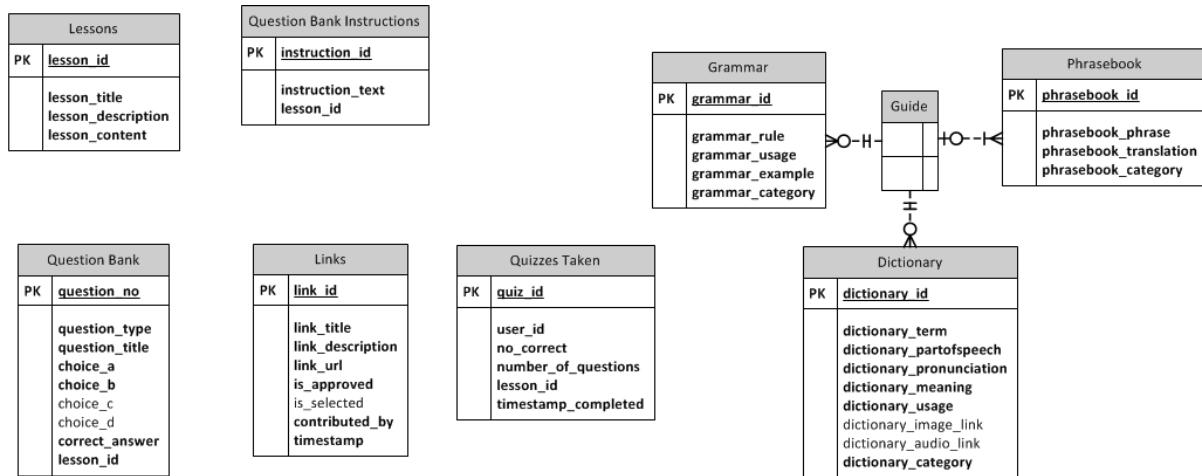


Figure 10: ERD for Guides and Lessons

The Entity Relationship Diagram of the Guides and Lessons is presented here. Except for the Guides, which are linked by a unifying interface, all other non-user tables are designed and operate independently of each other, with the contributed_by field in the Links table hardcoded from retrieving the username of the user who contributed the link and the user_id field in the Quizzes Taken table also hardcoded from retrieving the user_id of the one who took the quiz.

B. Data Dictionary

The data dictionary of the databases is as follows: (Note: The author has used a third-party authentication system, Flexi Auth for CodeIgniter, for functionality of the user-related tables.)

Data-Related Tables Pertaining to the System

The following tables hold the data of the entire system. The lesson contents are stored as database values.

lessons – holds the details of lessons		
FIELD NAME	FIELD TYPE	DESCRIPTION
lesson_id	int (2)	ID of the Lesson

lesson_title	varchar (200)	Title of the Lesson
lesson_description	varchar (1024)	Description of the Lesson
lesson_content	varchar (1024)	Link to the Content of the Lesson (in PDF format)

question_bank – holds the mastery test questions for the system		
FIELD NAME	FIELD TYPE	DESCRIPTION
question_no	int (4)	ID of Question
question_type	enum (‘true_false’, ‘multiple_choice’)	Type of Question (True/False or Multiple Choice are the only allowed types)
question_title	varchar (512)	Question Name
choice_a	varchar (50)	Choice A of Question
choice_b	varchar (50)	Choice B of Question
choice_c	varchar (50)	Choice C of Question (if true/false, not required)
choice_d	varchar (50)	Choice D of Question (if true/false, not required)
correct_answer	varchar (50)	Correct answer of question
lesson_id	int (2)	Lesson where the Question is connected to

links – holds the link exchange and its contents		
FIELD NAME	FIELD TYPE	DESCRIPTION
link_id	int (5)	ID of the Link
link_title	varchar (128)	Title of the Link
link_description	varchar (1024)	Description of the Link

link_url	varchar (1024)	URL of the Link
is_approved	int (1)	Is the Link Approved by the Admin?
contributed_by	varchar (50)	Username of the Contributor
timestamp_created	timestamp	Timestamp when the link was created

quizzes_taken – lists all quizzes ever recorded in the system		
FIELD NAME	FIELD TYPE	DESCRIPTION
quiz_id	int (6)	Quiz ID
user_id	mediumint (8)	ID of User who took the quiz
no_correct	int (2)	Number of Correct Answers in the Quiz
no_of_questions	int (2)	Number of Questions for that Quiz
lesson_id	int (2)	Lesson ID which the quiz's contents was based from
timestamp_completed	timestamp	Timestamp when the quiz was submitted

changes – holds all change requests made by users		
FIELD NAME	FIELD TYPE	DESCRIPTION
change_id	int (5)	Change ID
sender_id	int (8)	ID of the Sender who suggested the change
type	enum ('dictionary', 'phrasebook', 'grammar', 'general')	Type of the Change Request

content	varchar (4096)	Content of the Change Request
timestamp	timestamp	Timestamp when the Change Request was made

bank_instructions – stores the per-lesson instructions for the end-of-lesson quiz		
FIELD NAME	FIELD TYPE	DESCRIPTION
instruction_id	int (2)	Instruction ID
instruction_text	varchar (1024)	Instruction Text
lesson_id	int (2)	Lesson ID where the instruction will be displayed

Guide-Related Tables

The three Guides each have their own table and their respective contents are self-contained in the database.

grammar – table for grammar and style entries		
FIELD NAME	FIELD TYPE	DESCRIPTION
grammar_id	int (4)	ID of Grammar Rule
grammar_rule	varchar (250)	Name of Grammar Rule
grammar_usage	varchar (512)	Usage of Grammar Rule
grammar_example	varchar (512)	Example of the Grammar Rule in use

grammar_category	varchar (50)	Category of the Grammar Rule
------------------	--------------	------------------------------

phrasebook – table for phrasebook entries		
FIELD NAME	FIELD TYPE	DESCRIPTION
phrasebook_id	int (4)	ID of Phrasebook Entry
phrasebook_phrase	varchar (500)	Description of Phrasebook Entry
phrasebook_pronunciation	varchar (256)	Syllabic pronunciation of the Phrasebook Entry
phrasebook_translation	varchar (500)	English translation of Phrasebook Entry
phrasebook_category	varchar (256)	Category of the Phrasebook Entry
phrasebook_audio_link	varchar (1024)	Link to Audio Representation of the Phrasebook Entry

dictionary – table for dictionary entries		
FIELD NAME	FIELD TYPE	DESCRIPTION
dictionary_id	int (5)	ID of Dictionary Entry
dictionary_term	varchar (100)	Name of Dictionary Entry
dictionary_partofspeech	varchar (200)	Part of Speech where this Entry belongs to
dictionary_pronunciation	varchar (256)	Syllabic Pronunciation of the Term
dictionary_meaning	varchar (1000)	Meaning of Dictionary Entry
dictionary_usage	varchar (1000)	Usage Example of the Dictionary Entry
dictionary_image_link	varchar (1024)	Link to Term's Image Representation
dictionary_audio_link	varchar (1024)	Link to Term's Audio Pronunciation

dictionary_category	varchar (200)	Category where this Entry belongs to
---------------------	---------------	--------------------------------------

User-Related Custom Tables

Although Flexi Auth provides a set of pre-defined user tables required for its functionality, often, programmers need to implement more fields than what is provided by the library. One table was defined to extend the user_accounts table of Flexi Auth and its structure is enumerated below.

user_account_info – User Account Information, extends Flexi Auth’s user_accounts		
FIELD NAME	FIELD TYPE	DESCRIPTION
user_id	int (8)	User ID
userinfo_uacc_fk	int (8)	Foreign key to user_accounts’ uacc_id
userinfo_first_name	varchar (256)	User’s First Name
userinfo_last_name	varchar (128)	User’s Last Name

C. Technical Architecture

The system client is supported with the following minimum requirements:

- Any of the following browsers for desktops: Internet Explorer 9+, Mozilla Firefox 4+, Google Chrome 5+, Opera 10.5+, Safari 3+, Maxthon 3+; any browser for tablets (since the tablet OSes' standard browsers are web standards-compliant)
- At least 1 GB RAM for desktops;
- Any operating system supporting the browsers stated above for desktops (Windows XP or above, Mac OS X 10.5 or above, or any Linux distribution released after 2009)
- Minimum screen resolution size: 640x480 (both tablets and desktops)

For the server housing the system, the minimum server hardware requirements are as follows:

- Any server OS, preferably Linux type
- Apache 2.2+
- PHP 5.3+ / MySQL 5.0+
- At least 10 MB of server space for the system's use (more if audio pronunciations and pictures are planned to be available)

The system runs on the CodeIgniter framework (version used was 2.1.3, the latest at the time of writing), and the system also uses three third-party libraries: two CodeIgniter libraries, one for the authentication system (flexi_auth) and another for the CAPTCHA system, used for anti-spam purposes (math_captcha) and one JavaScript library that also doubles as a scripting framework, jQuery 1.9.1. The test system was developed on WampServer 2.2E (64-bit edition) with Apache 2.4.2, MySQL 5.5.24, phpMyAdmin 3.5.1 and PHP 5.4.3.

CHAPTER V

RESULTS

The FILLES main screen, when launched in the Opera browser, is shown in Figure 11.

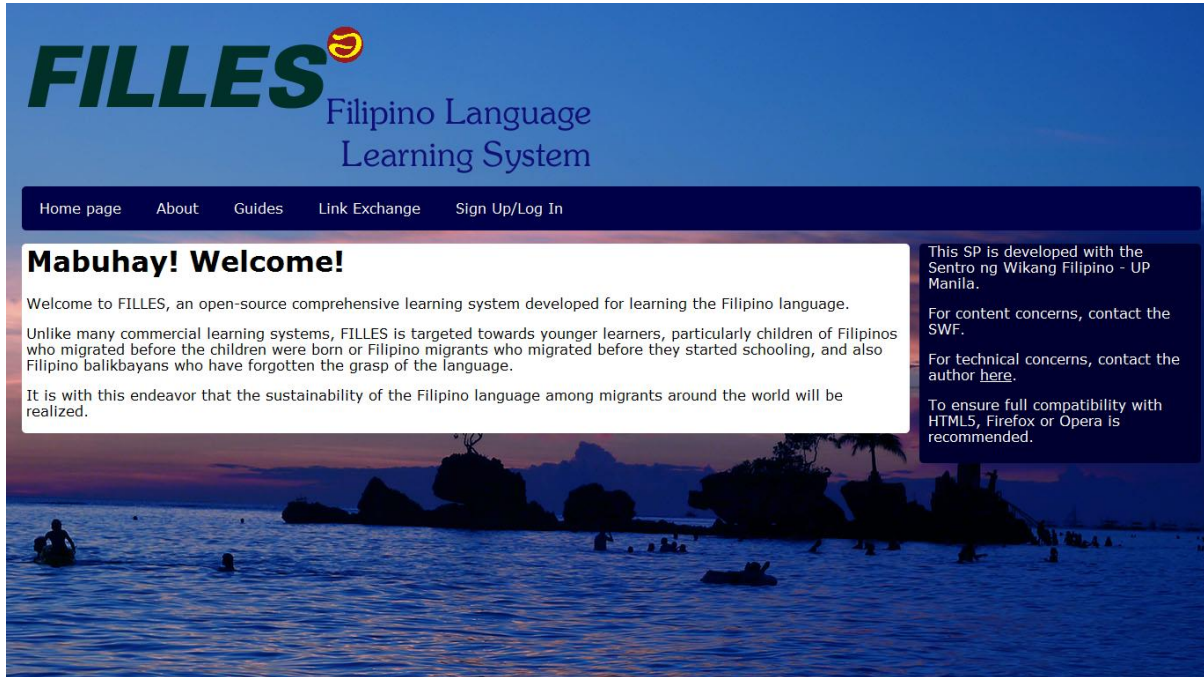


Figure 11: FILLES Main Screen

The navigation bar is kept simple for ease of navigation, and the choice of buttons varies by user role, given the different roles for various types of users in this system. The content area is the area where the main content is shown. The sidebar displays any side notes or other relevant information that can be of help to specific users.

When a guest user or registered user clicks on Guides, he/she is greeted with Figure 12:

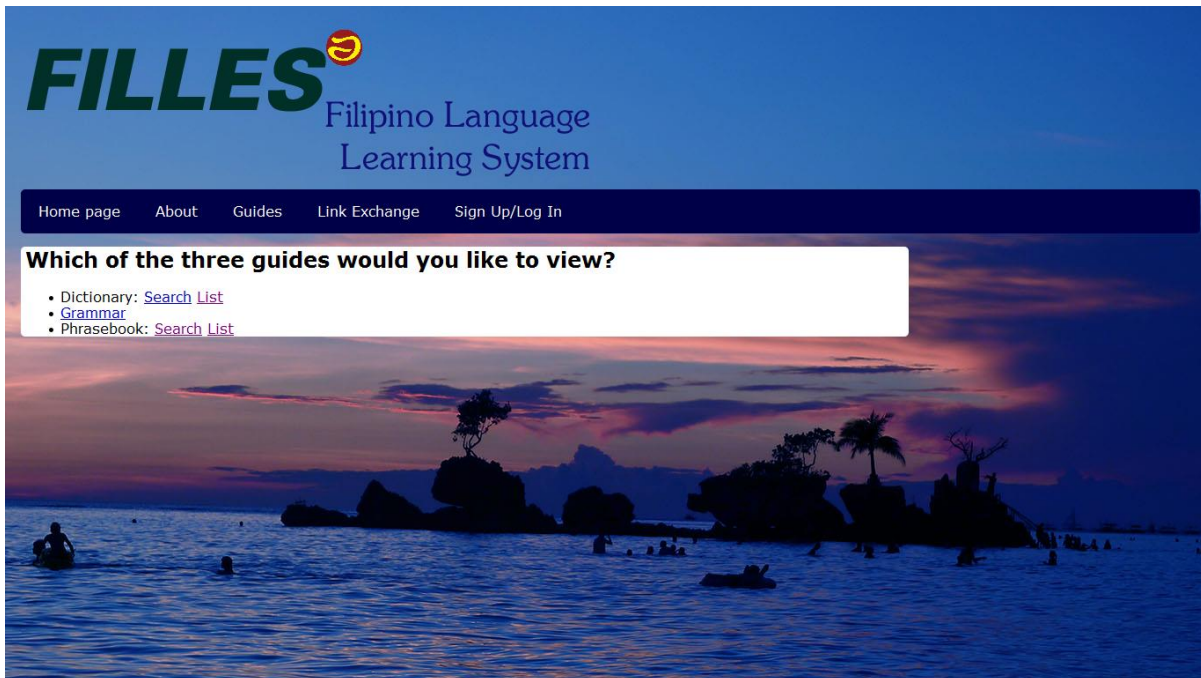


Figure 12: FILLES Guide Portal

As the user clicks on one of the Guides to view it, he/she is greeted by a list of terms reflecting the type of guide, with the Dictionary interface, accessible by a letter, shown in Figure 13.

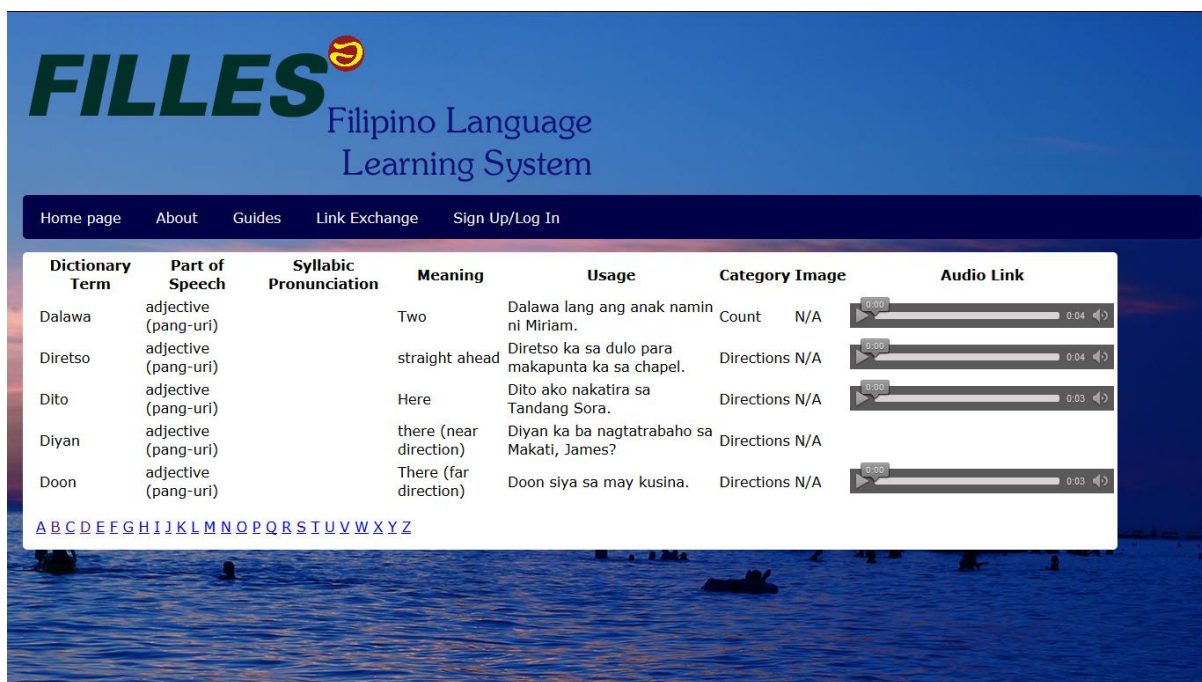


Figure 13: Dictionary (sorted by default order, which is by term)

When a user desires to log in, he/she is greeted by the screen in Figure 14.

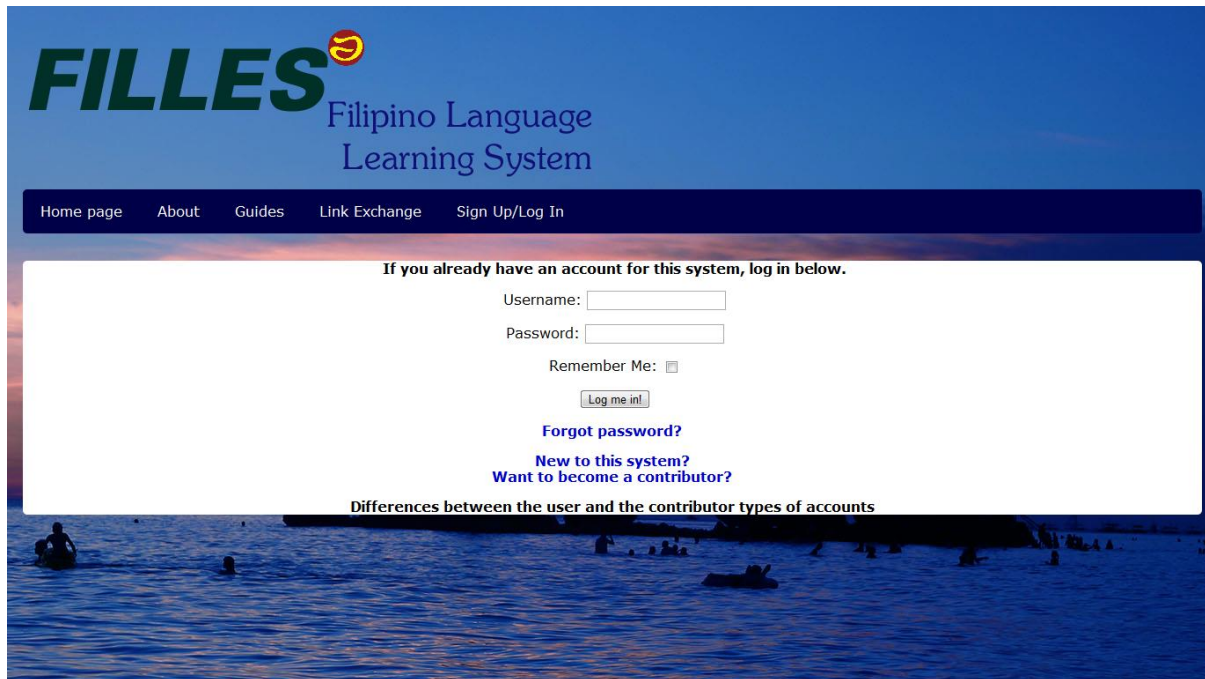


Figure 14: Login Screen

After a user logs in to their respective role's index page (shown in Figure 15 is the Staff Home Page), the list of available options in the navigation bar will differ from the one in Figure 12.

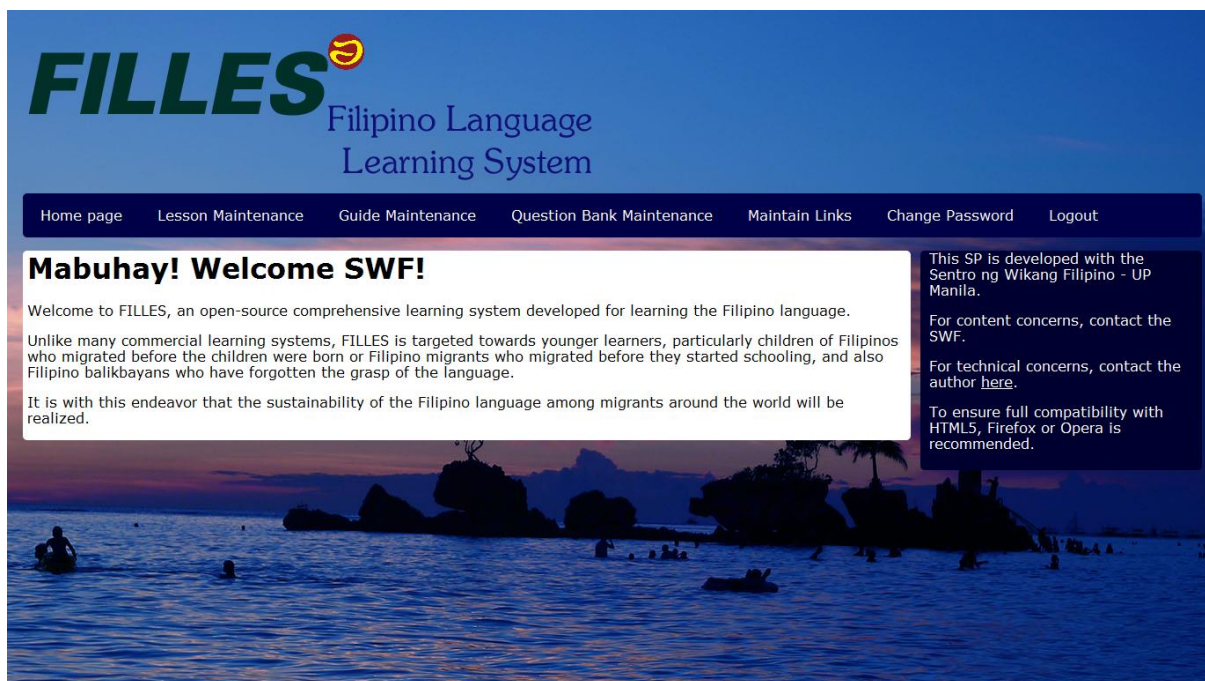


Figure 15: Staff Home Page

For the staff functions, there are five available functions all related to maintenance of the system. These are **Lesson Maintenance**, **Guide Maintenance**, **Question Bank Maintenance**, **View Change Requests**, and **Link Maintenance (Maintain Links)**.

Although the system comes preinstalled with several entries each of both lessons and guides, the staff member can customize the system to their needs: in **Lesson Maintenance** (Figure 17), the staff member can add additional, edit existing or delete unwanted or duplicate lessons. In **Guide Maintenance** (Figure 16), the staff member can add additional entries, edit existing entries, or delete any unwanted or duplicate entries to any of the three Guides. In **Question Bank Maintenance** (Figure 18), the staff member can add additional, edit existing or delete irrelevant questions from the Question Bank, and also add, edit or delete instructions to be displayed per lesson in the quiz function. In **View Change Requests** (Figure 19), the staff member can view suggested changes for the system made by the contributors or registered users. In **Maintain Links**, the staff member can add additional, edit unapproved or approved, or delete yet-to-be-approved links created by him/her.

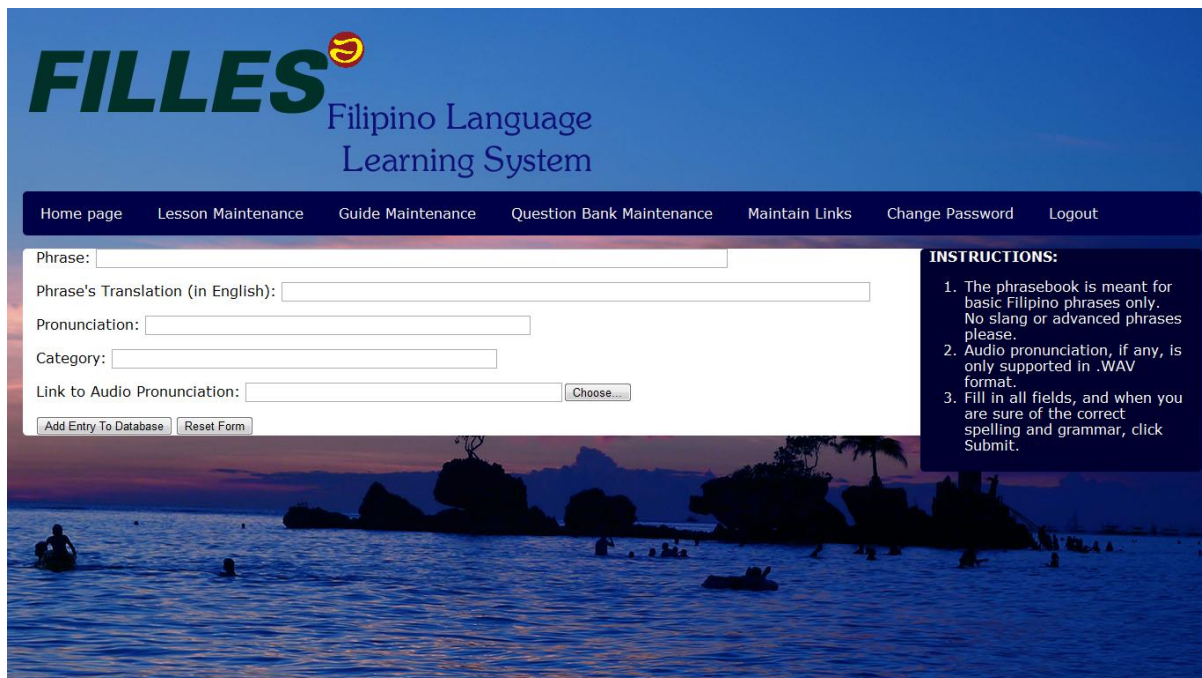


Figure 16: Add Phrase (Part of Guide Maintenance)



Figure 17: Confirmation of Lesson Deletion (Part of Lesson Maintenance)

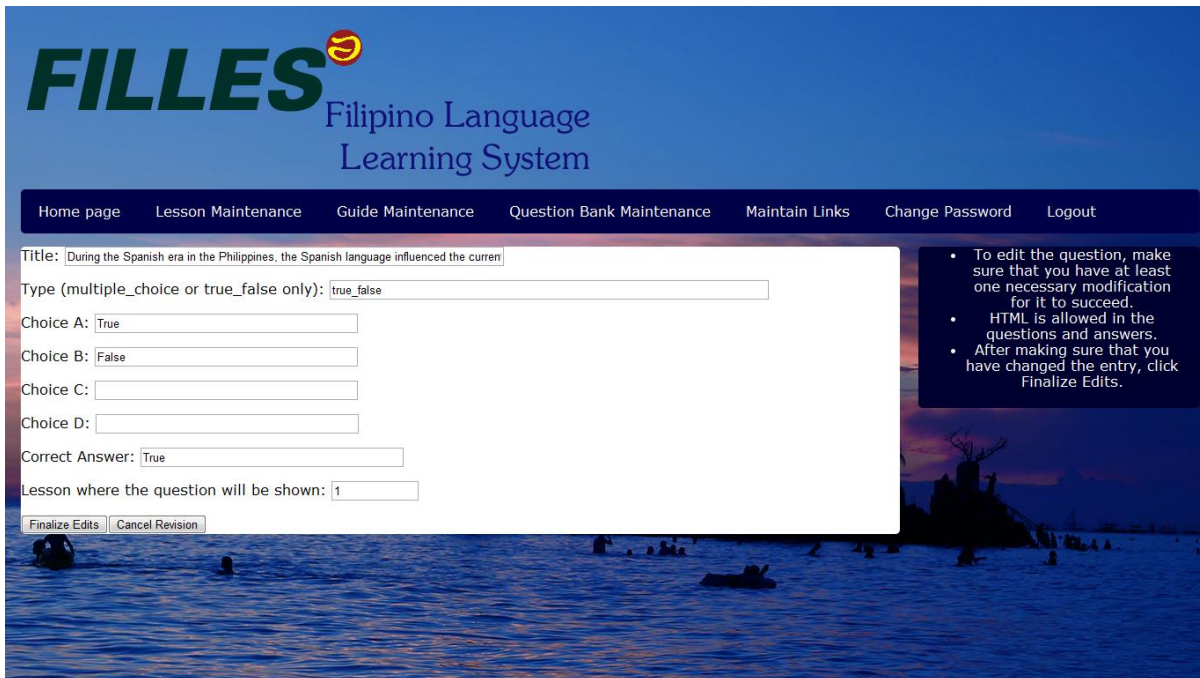


Figure 18: Edit Question (Part of Question Bank Maintenance)

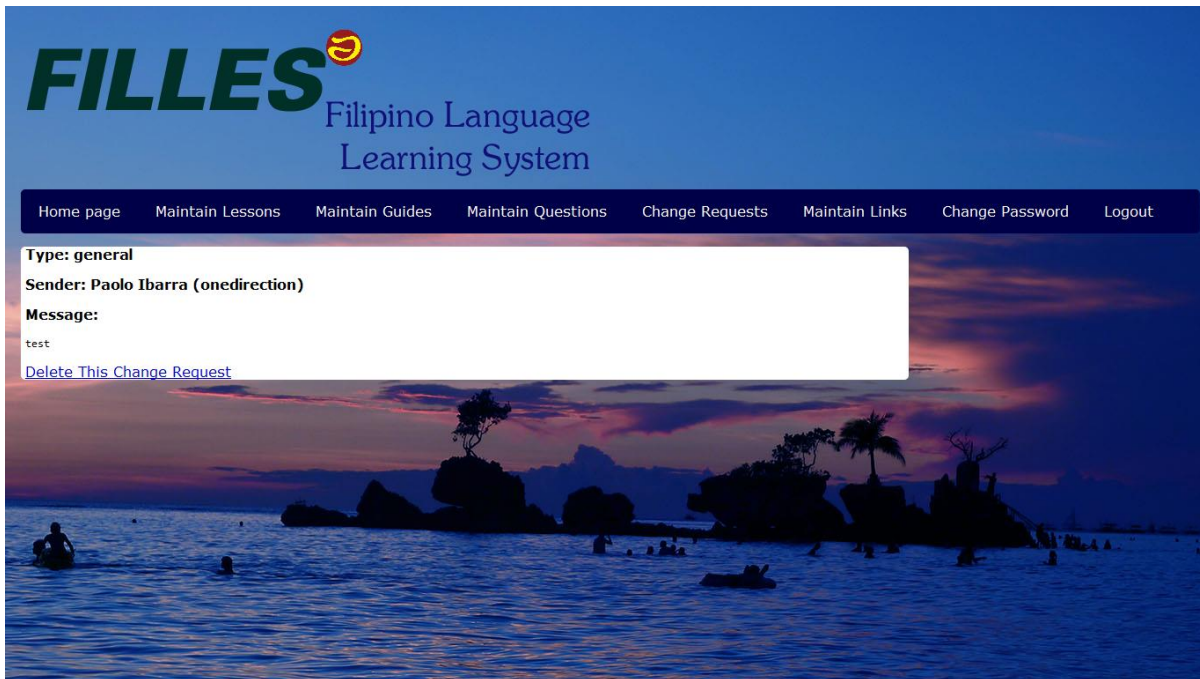


Figure 19: View Change Request

For the content administrator functions, there are two available functions: **Approve Links** and **View Change Requests** (already mentioned in the Staff functions).

In **Approve Links** (Figure 20), the administrator can accept a link for public display, reject (prevent the link from ever displaying in public) or leave the decision status of a link for a later time.



Figure 20: Approve Links Screen

For the registered user functions, these are the following available functions: in the **Lessons** subcategory, the user can view lessons (Figure 21), take an end-of-lesson quiz, whose questions are randomly selected from the question bank specific to that lesson, or view his/her progress through past quizzes he/she has ever taken (Figure 22). In the **Guides** category, the user can view the contents of the three guides. In the **Link Exchange**, he/she can view links approved by the content administrator. He/she can also **Suggest Guide Changes** (Figure 23) by sending change requests to the SWF for checking of the guides or for any suggestions on how to improve FILLES.

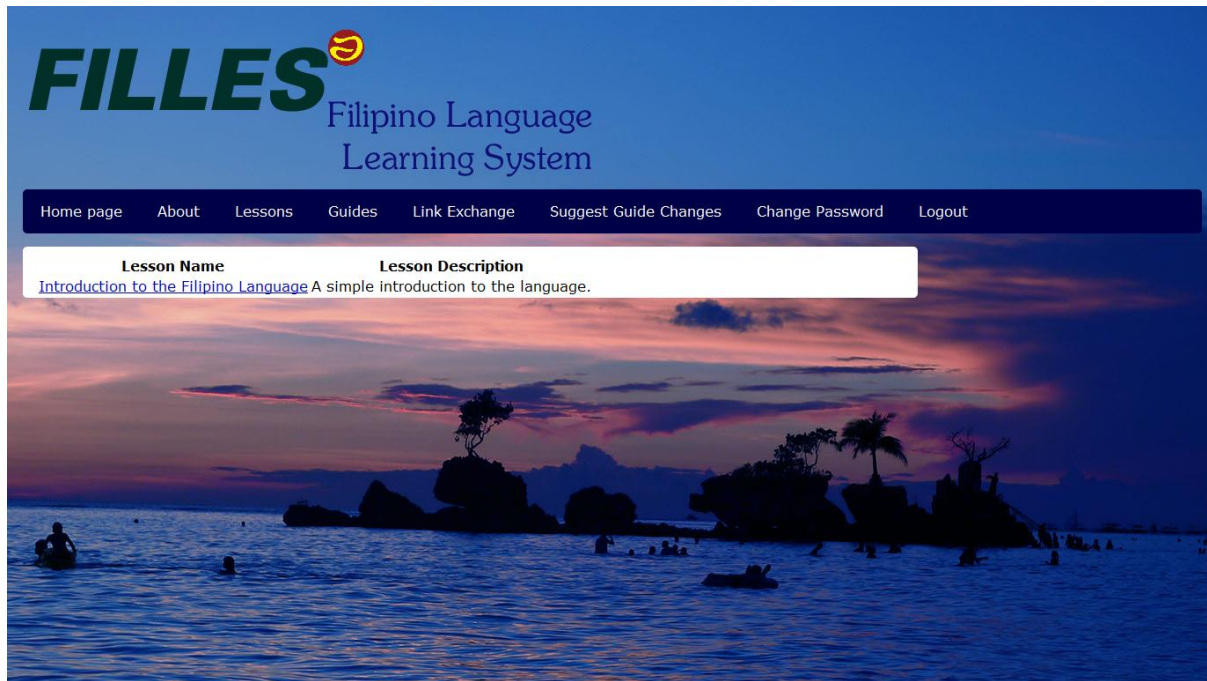


Figure 21: Lessons Navigation interface

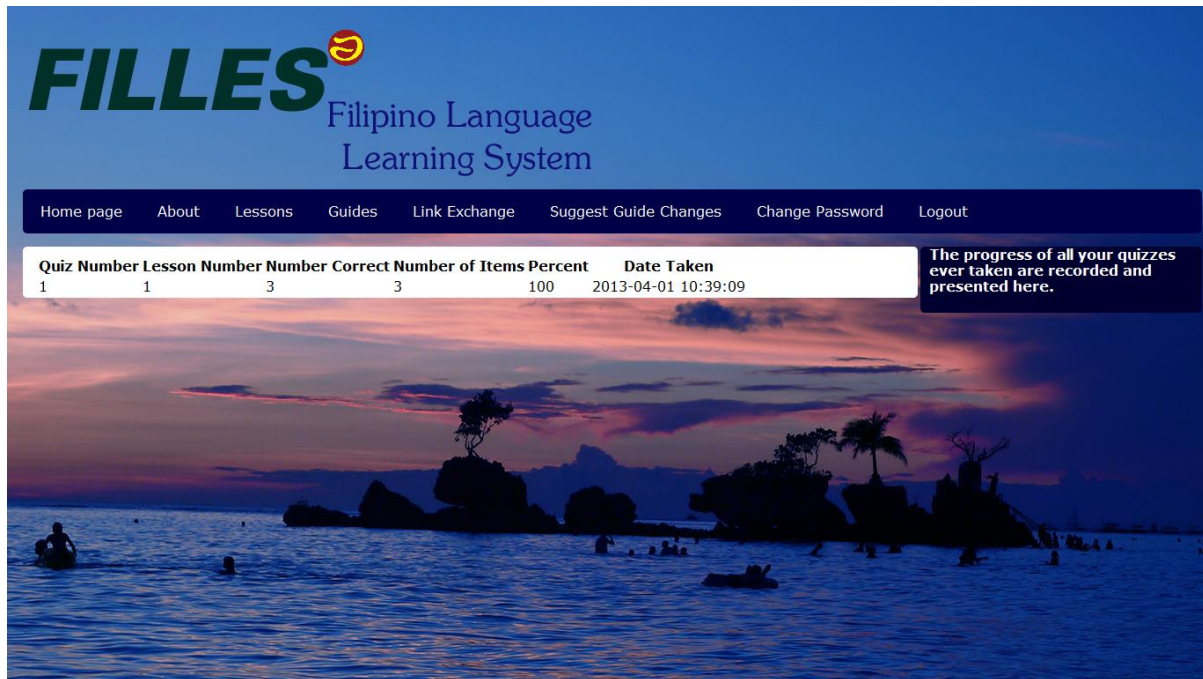


Figure 22: View Progress

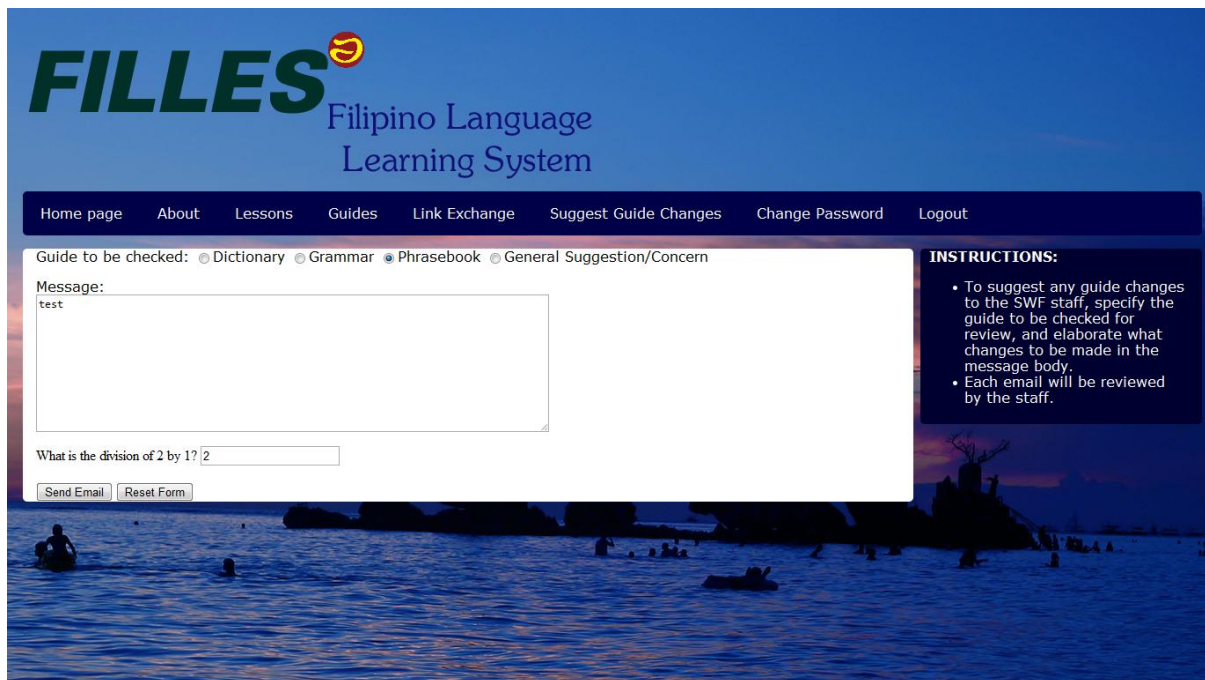


Figure 23: Suggest Guide Changes

For the contributor functions, the only two functions that are allowed for them – **Suggest Guide Changes** and **Maintain Links** (Figure 24) – have already been mentioned earlier, in the Registered Users and the Staff roles respectively, and the respective roles for the contributor are the same as for the earlier roles.

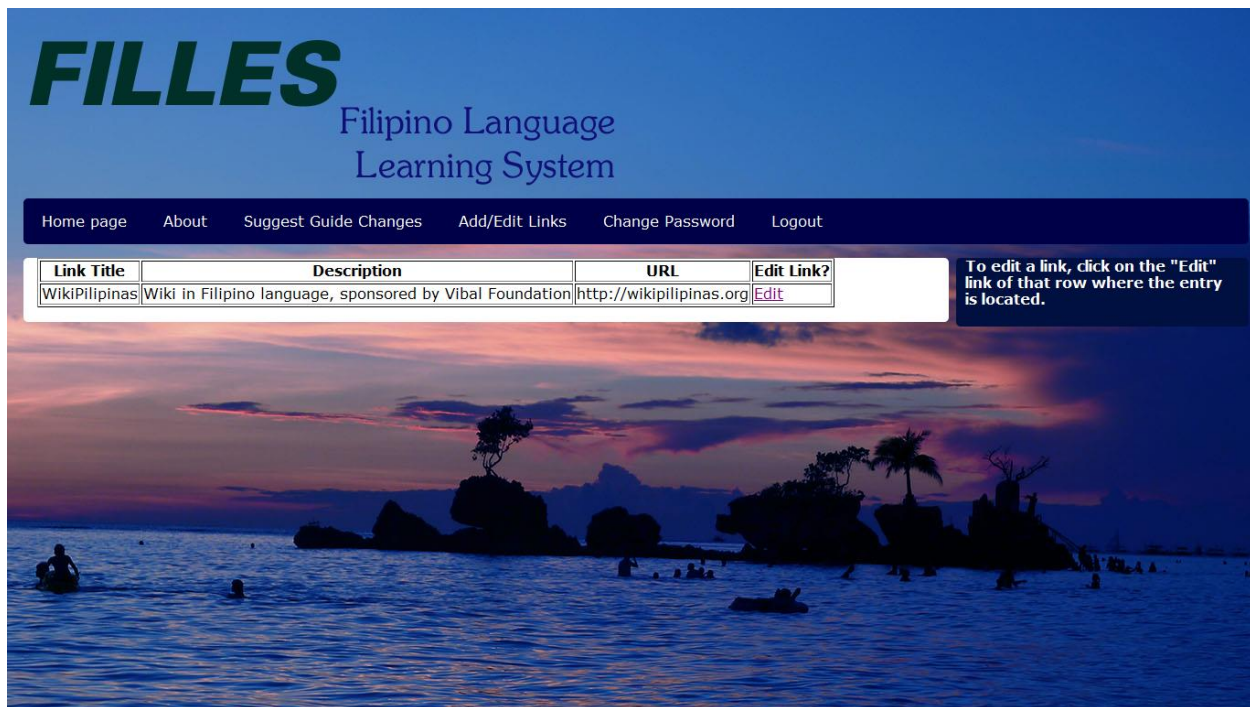


Figure 24: Edit Links (contributor navigation, part of Link Maintenance)

The system administrator has five functions, all unique to its role: **Add System Admin**, **Add Content Admin**, **Add Staff**, **Maintain Users** and **Maintain User Groups**.

In **Add System Admin**, the existing system administrator can add other system administrators, e.g. for when the existing system administrator would resign. In **Add Content Admin**, the system administrator can add another content administrator, e.g. for when the existing content administrator has changed their email address and has forgotten their password. In **Add Staff**, the administrator can add additional staff as necessary. In **Maintain Users** (Figure 25), the system administrator can update the information of its own user account or of other users, and can also delete other users as well, and in **Maintain User Groups**, the system administrator can add, update or delete user groups.

FILLES

Filipino Language Learning System

[Home page](#) [Add System Admin](#) [Add Content Admin](#) [Add Staff Member](#) [Maintain Users](#) [Maintain User Groups](#) [Change Password](#)

Username	User Email	User's Name	Member Since	Update User?	Delete User?
admin	bonarae@gmail.com	System Administrator	2013-02-11 22:40:03	Update	Delete
contentadmin	boeinghelmsman@yahoo.com	Content Administrator	2013-02-13 11:59:07	Update	Delete
swfstaff	kenilworth_maroon_asiam@yahoo.com	SWF Staff	2013-02-13 13:38:48	Update	Delete
contributor	mormongator@hotmail.com	Outside Contributor	2013-02-14 04:08:38	Update	Delete
reguser	200703380@post.upm.edu.ph	Registered User	2013-02-14 04:14:38	Update	Delete



Figure 25: Maintain Users

CHAPTER VI

DISCUSSION

FILLES is a system designed for learning the Filipino language. It brings a few new features not found in other language learning systems of its kind: the most important of these features is its flexibility towards different learning styles of its potential clients through constant maintenance of the guides and that content can also be adjusted to the clients' tastes.

The main advantage of FILLES over commercial counterparts is that content can be evaluated on and expanded over time by the community (through guide change requests to the staff); something that cannot be easily done in commercial systems and the system is also fully customizable to the clients' needs. Another advantage of FILLES is that its sustainability over time can be realized by the addition of additional staff members, content administrators and system administrators should any existing member of these roles become no longer involved in the system. Also, it is entirely developed using open-source technologies: CodeIgniter, HTML5 and PHP, to name a few.

However, FILLES also has disadvantages that can, however, be improved on or be patched upon in the future, should the need arise. The lessons supplied with the system are very basic and do not really help a student master the language through the basic examples alone.

As of this writing, a commercial web-based system similar to FILLES, FilipinoPod101.com by Innovative Languages [5], has done many features as of late that FILLES only does in the most basic ways (e.g. learning words entirely through pictures). But being open-source, FILLES has a lot of potential that when some more important features are added and when it is fully developed, it can even compete with commercial systems in the future.

CHAPTER VII

CONCLUSION

While FILLES is not a groundbreaking system for its kind nor it is a system with as many new features as its commercially-produced counterparts, it finally brings to the language learning market a freely-available, locally-developed and customizable, if not comprehensive, individual-oriented self-instructed language learning system for the Filipino language, that is at par with similar open-source learning systems for other languages, particularly English. The system is comprehensive enough for a learner to even master the Filipino language in a traditional way, but provisions for additional and/or enhanced interactivity can be done in the future.

CHAPTER VIII

RECOMMENDATIONS

Self-instructed language-learning is not a well-tested category when the availability of open-source solutions for that category is concerned; hence, the lack of competition in this category creates much potential and also pressure for systems such as FILLES to deliver the same quality of product for free as commercially-developed solutions currently do for a cost.

As far as the guides and links are concerned, commenting for each guide entry and link entry can also be put into place using commenting frameworks. However, SWF staff and administrators need to monitor the comments made in the entries' detail pages from time to time should commenting be implemented.

Lessons can also be made more interactive by using HTML5 Canvas. The performance of students in quizzes over time can also be implemented in a graphical way, using graphing frameworks, in addition to the textual way that is already implemented in this SP.

CHAPTER IX

BIBLIOGRAPHY

1. Warschauer, M. & Meskill, C. Technology and Second Language Teaching, 2000. Retrieved October 20, 2012 from Warschauer's UC Irvine page: http://www.gse.uci.edu/person/warschauer_m/tslt.html
2. Sentro ng Wikang Filipino – UP Diliman: Bahay (Home Page). Retrieved October 20, 2012 from Organization Site: <http://sentrofilipino.upd.edu.ph/bahay/index.php>.
3. 50languages | book2 Audio Trainer: Home Page. Retrieved October 20, 2012 from Company Site: <http://www.50languages.com/> (Note: 50languages and book2 are actually sister products developed for competing mobile platforms by Goethe.)
4. Rosetta Stone, Ltd.: Learn Tagalog overview page. Retrieved October 20, 2012 from Company Site: <http://www.rosettastone.com/learn-tagalog>.
5. Innovative Languages. FilipinoPod101.com homepage. Retrieved January 21, 2013 from product website: <http://www.filipinopod101.com/>.
6. Tuttle Publishing: Home Page. Retrieved October 20, 2012 from Company Site: <http://www.tuttlepublishing.com>. (Note: Tuttle Publishing doesn't provide a specific portal for its integrated language learning offerings grouped by language through its website; but the company provides different solutions for learning the Tagalog language the traditional way.)
7. Adarna House, Inc.: Ang Wikahon. Retrieved October 19, 2012 from Adarna House: <http://www.adarna.com.ph/wikahon/>

8. Genyo E-Learning System: About. Retrieved October 19, 2012 from Company Site:
http://www.genyo.com.ph/section_content.asp?id=1
9. Jose, V.R. Ang Wika ng Pagpapalaya at ang Papel ng Akademya. *Malay*, 21(2), 15-19.
(Article in Filipino, interpreted in English by the author of this SP)
10. Espiritu, C.C. Tugon ng mga Cebuano sa Kontrobersyang Pangwika: Implikasyon para sa
Debelopment ng Filipino. *Malay*, 22(1), 31-35. (Article in Filipino, interpreted in English
by the author of this SP)
11. Capitol University, Cagayan de Oro City, College of Education: Mother Tongue-Based
Multilingual Education. Retrieved October 11, 2012 from Capitol University:
[http://www.cu.edu.ph/academics/colleges/college-of-education/mother-tongue-based-
multilingual-education-mtb-mle/](http://www.cu.edu.ph/academics/colleges/college-of-education/mother-tongue-based-multilingual-education-mtb-mle/).
12. Constantino, E. *Tagalog and Other Major Languages of the Philippines*. Manuscript
submitted for publication (held at UP Manila CAS Library), 1969.
13. Li, J. An Attempted Evaluation of Computer Assisted Language Learning in China.
Canadian Social Science, 3(3), 109-113.
14. Bidlake, E. "THIS IS THE BIGGEST BLUE CAKE": A TECHNOLOGY REVIEW OF
ROSETTA STONE® VERSION 3. *Annual Review of Education, Communication and
Language Sciences*, 6, 156-164.
15. Kartal, E. An Evaluation of Teacher Candidates' Analyses of Language Learning
Software. *International Online Journal of Educational Sciences*, 2 (3), 905-929. (Article
in Turkish, but an English extended abstract is on the first pages of the full text.)
16. Bidlake, E. LEARNER EXPERIENCE USING SELF-INSTRUCTED CALL:
METHODOLOGICAL AND LEARNER INSIGHTS. *Novitas-ROYAL*, 3(2), 93-109.

17. Nedeva, V. & Dimova, E. SOME ADVANTAGES OF E-LEARNING IN ENGLISH LANGUAGE TRAINING. *Trakia Journal of Sciences*, 8 (Suppl. 3), 21-28.
18. Hot Potatoes: Home Page. Retrieved October 23, 2012 from Product Site: <http://hotpot.uvic.ca/>
19. Moodle: Community Home Page. Retrieved October 23, 2012 from Product Site: <https://moodle.org/>
20. Liaw, M-L. E-LEARNING AND THE DEVELOPMENT OF INTERCULTURAL COMPETENCE. *Language Learning & Technology*, 10(3), 49-64.
21. Andrade, M. & Bunker, E. A model for self-regulated distance learning. *Distance Education*, 30 (1), 47-61. doi:10.1080/01587910902845956
22. Ros i Sole, C. & Hopkins, J. Contrasting Two Approaches to Distance Language Learning. *Distance Education*, 28(3), 351-370. doi:10.1080/01587910701611369
23. Aspillera, P. *A Common Vocabulary for Malay – Pilipino – Bahasa Indonesia*. Self-published, Manila, 1964.
24. Laghos, A. & Zaphiris, P. Computer-Aided Language Learning. In Howard, C., Boettcher, J., & Justice, L. eds. *Encyclopedia of Distance Learning*. Idea Group Reference, Hershey, PA, 2005, Vol. 1, p. 337-340.
25. Tsubota, Y., Kawahara, T. & Dantsuji, M. (n.d.) RECOGNITION AND VERIFICATION OF ENGLISH BY JAPANESE STUDENTS FOR COMPUTER-ASSISTED LANGUAGE LEARNING SYSTEM. Retrieved October 23, 2012 from Tsubota's academic website: <http://www.ar.media.kyoto-u.ac.jp/EN/bib/intl/TSU-ICSLP02.pdf>

26. Hurskainen, A. Intelligent Computer-Assisted Language Learning: Implementation to Swahili. *Technical Reports in Language Technology, Report No. 3*, 1-29.
27. Son, J.-B. Computers, learners and teachers: Teamwork in the CALL classroom. *English Language Teaching*, 14 (2), 239-252.
28. Bergström, H. Evaluation of a Computer Assisted Language Learning System for Swedish Language Learners. Master's thesis, Royal Institute of Technology (KTH), Sweden. 2007.
29. Zwass, V. *Fundamentals of Information Systems*. Irwin/McGraw-Hill, New York, NY, 1998.
30. Freeman, A. *The Definitive Guide to HTML5*. Apress, Berkeley, CA, 2011.
31. EllisLab, Inc.: CodeIgniter product page. Retrieved February 1, 2013 from company website: <http://ellislab.com/codeigniter>.
32. Anonymous. (2012) Comparing Laravel, CodeIgniter, & CakePHP. Retrieved February 1, 2013 from: <http://nerdmom.wordpress.com/2012/08/02/comparing-laravel-codeigniter-cakephp/>

APPENDIX

SOURCE CODE

CONFIG FILES

/application/config/autoload.php

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
/*
| -----
| AUTO-LOADER
| -----
| This file specifies which systems should be loaded by default.
|
| In order to keep the framework as light-weight as possible only the
| absolute minimal resources are loaded by default. For example,
| the database is not connected to automatically since no assumption
| is made regarding whether you intend to use it. This file lets
| you globally define which systems you would like loaded with every
| request.
|
| -----
| Instructions
| -----
|
| These are the things you can load automatically:
|
| 1. Packages
| 2. Libraries
| 3. Helper files
| 4. Custom config files
| 5. Language files
| 6. Models
|
*/

/*
| -----
| Auto-load Packages
| -----
| Prototype:
|
| $autoload['packages'] = array(APPPATH.'third_party', '/usr/local/shared');
*/

$autoload['packages'] = array();

/*
| -----
| Auto-load Libraries
| -----
| These are the classes located in the system/libraries folder
| or in your application/libraries folder.
|
| Prototype:
|
| $autoload['libraries'] = array('database', 'session', 'xmlrpc');
*/

$autoload['libraries'] = array('database','session','auth','flexi_auth');

/*
| -----
| Auto-load Helper Files
```

```

| -----
| Prototype:
|
|     $autoload['helper'] = array('url', 'file');
*/

$autoload['helper'] = array('form','url');

/*
| -----
| Auto-load Config files
| -----
| Prototype:
|
|     $autoload['config'] = array('config1', 'config2');
|
| NOTE: This item is intended for use ONLY if you have created custom
| config files. Otherwise, leave it blank.
|
*/

$autoload['config'] = array();

/*
| -----
| Auto-load Language files
| -----
| Prototype:
|
|     $autoload['language'] = array('lang1', 'lang2');
|
| NOTE: Do not include the "_lang" part of your file. For example
| "codeigniter_lang.php" would be referenced as array('codeigniter');
|
*/

$autoload['language'] = array();

/*
| -----
| Auto-load Models
| -----
| Prototype:
|
|     $autoload['model'] = array('model1', 'model2');
|
*/

$autoload['model'] = array();

/* End of file autoload.php */
/* Location: ./application/config/autoload.php */

/application/config/config.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

/*
| -----
| Base Site URL
| -----
|
| URL to your CodeIgniter root. Typically this will be your base URL,
| WITH a trailing slash:
|
|     http://example.com/
|
| If this is not set then CodeIgniter will guess the protocol, domain and

```



```

| path to your installation.
|
*/
$config['base_url'] = '';

/*
|-----|
| Index File
|-----|
|
| Typically this will be your index.php file, unless you've renamed it to
| something else. If you are using mod_rewrite to remove the page set this
| variable so that it is blank.
|
*/
$config['index_page'] = 'index.php';

/*
|-----|
| URI PROTOCOL
|-----|
|
| This item determines which server global should be used to retrieve the
| URI string. The default setting of 'AUTO' works for most servers.
| If your links do not seem to work, try one of the other delicious flavors:
|
| 'AUTO'                Default - auto detects
| 'PATH_INFO'           Uses the PATH_INFO
| 'QUERY_STRING'        Uses the QUERY_STRING
| 'REQUEST_URI'         Uses the REQUEST_URI
| 'ORIG_PATH_INFO'      Uses the ORIG_PATH_INFO
|
*/
$config['uri_protocol'] = 'AUTO';

/*
|-----|
| URL suffix
|-----|
|
| This option allows you to add a suffix to all URLs generated by CodeIgniter.
| For more information please see the user guide:
|
| http://codeigniter.com/user\_guide/general/urls.html
|
*/

$config['url_suffix'] = '';

/*
|-----|
| Default Language
|-----|
|
| This determines which set of language files should be used. Make sure
| there is an available translation if you intend to use something other
| than english.
|
*/
$config['language'] = 'english';

/*
|-----|
| Default Character Set
|-----|
|
| This determines which character set is used by default in various methods
| that require a character set to be provided.
|
*/
$config['charset'] = 'UTF-8';

```

```

/*
|-----|
| Enable/Disable System Hooks
|-----|
|
| If you would like to use the 'hooks' feature you must enable it by
| setting this variable to TRUE (boolean). See the user guide for details.
|
*/
$config['enable_hooks'] = FALSE;

/*
|-----|
| Class Extension Prefix
|-----|
|
| This item allows you to set the filename/classname prefix when extending
| native libraries. For more information please see the user guide:
|
| http://codeigniter.com/user\_guide/general/core\_classes.html
| http://codeigniter.com/user\_guide/general/creating\_libraries.html
|
*/
$config['subclass_prefix'] = 'MY_';

/*
|-----|
| Allowed URL Characters
|-----|
|
| This lets you specify with a regular expression which characters are permitted
| within your URLs. When someone tries to submit a URL with disallowed
| characters they will get a warning message.
|
| As a security measure you are STRONGLY encouraged to restrict URLs to
| as few characters as possible. By default only these are allowed: a-z 0-9~%.:_-
|
| Leave blank to allow all characters -- but only if you are insane.
|
| DO NOT CHANGE THIS UNLESS YOU FULLY UNDERSTAND THE REPERCUSSIONS!!
|
*/
$config['permitted_uri_chars'] = 'a-z 0-9~%.:_-';

/*
|-----|
| Enable Query Strings
|-----|
|
| By default CodeIgniter uses search-engine friendly segment based URLs:
| example.com/who/what/where/
|
| By default CodeIgniter enables access to the $_GET array. If for some
| reason you would like to disable it, set 'allow_get_array' to FALSE.
|
| You can optionally enable standard query string based URLs:
| example.com?who=me&what=something&where=here
|
| Options are: TRUE or FALSE (boolean)
|
| The other items let you set the query string 'words' that will
| invoke your controllers and its functions:
| example.com/index.php?c=controller&m=function
|
| Please note that some of the helpers won't work as expected when
| this feature is enabled, since CodeIgniter is designed primarily to
| use segment based URLs.
|

```

```

*/
$config['allow_get_array']          = TRUE;
$config['enable_query_strings']    = FALSE;
$config['controller_trigger']     = 'c';
$config['function_trigger']       = 'm';
$config['directory_trigger']      = 'd'; // experimental not currently in use

/*
|-----|
| Error Logging Threshold          |
|-----|
|
| If you have enabled error logging, you can set an error threshold to
| determine what gets logged. Threshold options are:
| You can enable error logging by setting a threshold over zero. The
| threshold determines what gets logged. Threshold options are:
|
|     0 = Disables logging, Error logging TURNED OFF
|     1 = Error Messages (including PHP errors)
|     2 = Debug Messages
|     3 = Informational Messages
|     4 = All Messages
|
| For a live site you'll usually only enable Errors (1) to be logged otherwise
| your log files will fill up very fast.
|
*/
$config['log_threshold'] = 0;

/*
|-----|
| Error Logging Directory Path    |
|-----|
|
| Leave this BLANK unless you would like to set something other than the default
| application/logs/ folder. Use a full server path with trailing slash.
|
*/
$config['log_path'] = '';

/*
|-----|
| Date Format for Logs            |
|-----|
|
| Each item that is logged has an associated date. You can use PHP date
| codes to set your own date formatting
|
*/
$config['log_date_format'] = 'Y-m-d H:i:s';

/*
|-----|
| Cache Directory Path          |
|-----|
|
| Leave this BLANK unless you would like to set something other than the default
| system/cache/ folder. Use a full server path with trailing slash.
|
*/
$config['cache_path'] = '';

/*
|-----|
| Encryption Key                |
|-----|
|
| If you use the Encryption class or the Session class you
| MUST set an encryption key. See the user guide for info.
|
*/

```

```

$config['encryption_key'] = 'StJames1990';

/*
|-----|
| Session Variables
|-----|
|
| 'sess_cookie_name'           = the name you want for the cookie
| 'sess_expiration'           = the number of SECONDS you want the session to last.
|   by default sessions last 7200 seconds (two hours). Set to zero for no expiration.
| 'sess_expire_on_close'      = Whether to cause the session to expire automatically
|   when the browser window is closed
| 'sess_encrypt_cookie'       = Whether to encrypt the cookie
| 'sess_use_database'         = Whether to save the session data to a database
| 'sess_table_name'           = The name of the session database table
| 'sess_match_ip'             = Whether to match the user's IP address when reading the
session data
| 'sess_match_useragent'      = Whether to match the User Agent when reading the session data
| 'sess_time_to_update'       = how many seconds between CI refreshing Session
Information
|
*/
$config['sess_cookie_name']      = 'ci_session';
$config['sess_expiration']       = 7200;
$config['sess_expire_on_close']  = FALSE;
$config['sess_encrypt_cookie']   = FALSE;
$config['sess_use_database']     = TRUE;
$config['sess_table_name']       = 'ci_sessions';
$config['sess_match_ip']         = FALSE;
$config['sess_match_useragent']  = TRUE;
$config['sess_time_to_update']   = 300;

/*
|-----|
| Cookie Related Variables
|-----|
|
| 'cookie_prefix' = Set a prefix if you need to avoid collisions
| 'cookie_domain' = Set to .your-domain.com for site-wide cookies
| 'cookie_path'   = Typically will be a forward slash
| 'cookie_secure' = Cookies will only be set if a secure HTTPS connection exists.
|
*/
$config['cookie_prefix']    = "";
$config['cookie_domain']    = "";
$config['cookie_path']      = "/";
$config['cookie_secure']    = FALSE;

/*
|-----|
| Global XSS Filtering
|-----|
|
| Determines whether the XSS filter is always active when GET, POST or
| COOKIE data is encountered
|
*/
$config['global_xss_filtering'] = TRUE;

/*
|-----|
| Cross Site Request Forgery
|-----|
|
| Enables a CSRF cookie token to be set. When set to TRUE, token will be
| checked on a submitted form. If you are accepting user data, it is strongly
| recommended CSRF protection be enabled.
|
| 'csrf_token_name' = The token name
| 'csrf_cookie_name' = The cookie name
| 'csrf_expire'     = The number in seconds the token should expire.
*/

```

```

$config['csrf_protection'] = FALSE;
$config['csrf_token_name'] = 'csrf_test_name';
$config['csrf_cookie_name'] = 'csrf_cookie_name';
$config['csrf_expire'] = 7200;

/*
|-----|
| Output Compression
|-----|
|
| Enables Gzip output compression for faster page loads. When enabled,
| the output class will test whether your server supports Gzip.
| Even if it does, however, not all browsers support compression
| so enable only if you are reasonably sure your visitors can handle it.
|
| VERY IMPORTANT: If you are getting a blank page when compression is enabled it
| means you are prematurely outputting something to your browser. It could
| even be a line of whitespace at the end of one of your scripts. For
| compression to work, nothing can be sent before the output buffer is called
| by the output class. Do not 'echo' any values with compression enabled.
|
*/
$config['compress_output'] = FALSE;

/*
|-----|
| Master Time Reference
|-----|
|
| Options are 'local' or 'gmt'. This pref tells the system whether to use
| your server's local time as the master 'now' reference, or convert it to
| GMT. See the 'date helper' page of the user guide for information
| regarding date handling.
|
*/
$config['time_reference'] = 'local';

/*
|-----|
| Rewrite PHP Short Tags
|-----|
|
| If your PHP installation does not have short tag support enabled CI
| can rewrite the tags on-the-fly, enabling you to utilize that syntax
| in your view files. Options are TRUE or FALSE (boolean)
|
*/
$config['rewrite_short_tags'] = FALSE;

/*
|-----|
| Reverse Proxy IPs
|-----|
|
| If your server is behind a reverse proxy, you must whitelist the proxy IP
| addresses from which CodeIgniter should trust the HTTP_X_FORWARDED_FOR
| header in order to properly identify the visitor's IP address.
| Comma-delimited, e.g. '10.0.1.200,10.0.1.201'
|
*/
$config['proxy_ips'] = '';

/* End of file config.php */
/* Location: ./application/config/config.php */

```

/application/config/database.php

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
/*
| -----
| DATABASE CONNECTIVITY SETTINGS
| -----
| This file will contain the settings needed to access your database.
|
| For complete instructions please consult the 'Database Connection'
| page of the User Guide.
|
| -----
| EXPLANATION OF VARIABLES
| -----
|
| ['hostname'] The hostname of your database server.
| ['username'] The username used to connect to the database
| ['password'] The password used to connect to the database
| ['database'] The name of the database you want to connect to
| ['dbdriver'] The database type. ie: mysql. Currently supported:
|               mysql, mysqli, postgre, odbc, mssql, sqlite, oci8
| ['dbprefix'] You can add an optional prefix, which will be added
|               to the table name when using the Active Record class
| ['pconnect'] TRUE/FALSE - Whether to use a persistent connection
| ['db_debug'] TRUE/FALSE - Whether database errors should be displayed.
| ['cache_on'] TRUE/FALSE - Enables/disables query caching
| ['cachedir'] The path to the folder where cache files should be stored
| ['char_set'] The character set used in communicating with the database
| ['dbcollat'] The character collation used in communicating with the database
|
|               NOTE: For MySQL and MySQLi databases, this setting is only used
|               as a backup if your server is running PHP < 5.2.3 or MySQL < 5.0.7
|               (and in table creation queries made with DB Forge).
|               There is an incompatibility in PHP with mysql_real_escape_string()
which
|               can make your site vulnerable to SQL injection if you are using a
|               multi-byte character set and are running versions lower than
these.
|               Sites using Latin-1 or UTF-8 database character set and collation
are unaffected.
|       ['swap_pre'] A default table prefix that should be swapped with the dbprefix
|       ['autoinit'] Whether or not to automatically initialize the database.
|       ['stricton'] TRUE/FALSE - forces 'Strict Mode' connections
|
|                               - good for ensuring strict SQL while
developing
|
| The $active_group variable lets you choose which connection group to
| make active. By default there is only one group (the 'default' group).
|
| The $active_record variables lets you determine whether or not to load
| the active record class
*/

$active_group = 'default';
$active_record = TRUE;

$db['default']['hostname'] = 'localhost';
$db['default']['username'] = 'root';
$db['default']['password'] = '';
$db['default']['database'] = 'filles';
$db['default']['dbdriver'] = 'mysql';
$db['default']['dbprefix'] = '';
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = '';
$db['default']['char_set'] = 'utf8';
$db['default']['dbcollat'] = 'utf8_general_ci';
$db['default']['swap_pre'] = '';
$db['default']['autoinit'] = TRUE;
$db['default']['stricton'] = FALSE;
```

```

/* End of file database.php */
/* Location: ./application/config/database.php */

/application/config/flexi_auth.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
/*
* Name: flexi auth Config
*
* Author:
* Rob Hussey
* flexiauth@haseydesign.com
* haseydesign.com/flexi-auth
*
* Copyright 2012 Rob Hussey
*
* Previous Authors / Contributors:
* Ben Edmunds, benedmunds.com
* Phil Sturgeon, philsturgeon.co.uk
* Mathew Davies
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*
* Description: A full login authorisation and user management library for CodeIgniter based on
Ion Auth (By Ben Edmunds) which itself was based on Redux Auth 2 (Mathew Davies)
* Released: 13/09/2012
* Requirements: PHP5 or above and Codeigniter 2.0+
*/

#####
+++++++###
// DATABASE NAMES / ALIASES
#####
+++++++###

/**
 * If required, it is possible to set your own name for each database table and column.
 * Note: Only change the name in the apostrophes (after the '=' sign), and not the
$config array names.
 * Example: Change $config['database']['user_acc']['columns']['id'] = 'uacc_id' to
$config['database']['user_acc']['columns']['id'] = 'new_column_name'
 *
 * Quick Reference Guide on array structuring
 * ['table'] = table name, ['primary_key'] = primary key of table used in joins,
['join'] = column used to join table, ['columns']['xxx'] = specific column name
*/

#####

/**
 * Primary User Account Table
 * The primary user account table contains all of the columns required for different
functions within the flexi auth library.
 *
 * All columns are required.
*/
$config['database']['user_acc']['table'] = 'user_accounts';
$config['database']['user_acc']['join'] = 'user_accounts.uacc_id';
$config['database']['user_acc']['columns']['id'] = 'uacc_id';
$config['database']['user_acc']['columns']['group_id'] = 'uacc_group_fk';

```

```

$config['database']['user_acc']['columns']['email'] = 'uacc_email';
$config['database']['user_acc']['columns']['username'] = 'uacc_username';
$config['database']['user_acc']['columns']['password'] = 'uacc_password';
$config['database']['user_acc']['columns']['ip_address'] = 'uacc_ip_address';
$config['database']['user_acc']['columns']['salt'] = 'uacc_salt';
$config['database']['user_acc']['columns']['activation_token'] = 'uacc_activation_token';
$config['database']['user_acc']['columns']['forgot_password_token'] =
'uacc_forgotten_password_token';
$config['database']['user_acc']['columns']['forgot_password_expire'] =
'uacc_forgotten_password_expire';
$config['database']['user_acc']['columns']['update_email_token'] =
'uacc_update_email_token';
$config['database']['user_acc']['columns']['update_email'] = 'uacc_update_email';
$config['database']['user_acc']['columns']['active'] = 'uacc_active';
$config['database']['user_acc']['columns']['suspend'] = 'uacc_suspend';
$config['database']['user_acc']['columns']['failed_logins'] = 'uacc_fail_login_attempts';
$config['database']['user_acc']['columns']['failed_login_ip'] =
'uacc_fail_login_ip_address';
$config['database']['user_acc']['columns']['failed_login_ban_date'] =
'uacc_date_fail_login_ban';
$config['database']['user_acc']['columns']['last_login_date'] = 'uacc_date_last_login';
$config['database']['user_acc']['columns']['date_added'] = 'uacc_date_added';

// Custom columns can be added to the main user account table to enable library functions
to handle additional custom data stored within the table.
$config['database']['user_acc']['custom_columns'] = array(
    ### Example : 'date_modified', 'modified_user_id' etc.
);

#####

/**
 * User Group Table
 * The user group table is used to allocate a group classification to users, typically
this is used to group users as either admins or public users.
 * The grouped users can then be delivered content specific to their group, or restricted
access to set areas - i.e. an admin only area.
 *
 * All columns are required.
 */
$config['database']['user_group']['table'] = 'user_groups';
$config['database']['user_group']['join'] = 'user_groups.ugrp_id';
$config['database']['user_group']['columns']['id'] = 'ugrp_id';
$config['database']['user_group']['columns']['name'] = 'ugrp_name';
$config['database']['user_group']['columns']['description'] = 'ugrp_desc';
$config['database']['user_group']['columns']['admin'] = 'ugrp_admin';

#####

/**
 * User Privilege Table
 * The user privilege table is used to allocate role privileges to users.
 * Whilst very similar to user groups, multiple privileges can be assigned to a user, the
users privilege (and group if desired) can then be
 * looked up to verify if a user has permission to perform an action or access specific
data.
 * For example, 2 users could be in an 'Moderator' group, 1 of the users could be allowed
to update data, whilst the other could only view the data.
 *
 * All columns are required.
 */
$config['database']['user_privileges']['table'] = 'user_privileges';
$config['database']['user_privileges']['columns']['id'] = 'upriv_id';
$config['database']['user_privileges']['columns']['name'] = 'upriv_name';
$config['database']['user_privileges']['columns']['description'] = 'upriv_desc';

/**
 * User Privilege Users Table
 * The user privilege user table is used to assign privileges to users. Multiple
privileges can be assigned to a user.
 *

```



```

    * All columns are required.
*/
$config['database']['user_privilege_users']['table'] = 'user_privilege_users';
$config['database']['user_privilege_users']['columns']['id'] = 'upriv_users_id';
$config['database']['user_privilege_users']['columns']['user_id'] =
'upriv_users_uacc_fk';
$config['database']['user_privilege_users']['columns']['privilege_id'] =
'upriv_users_upriv_fk';

/**
 * User Privilege Groups Table
 * The user privilege group table is used to assign privileges to user groups. Multiple
privileges can be assigned to a user group.
 *
 * All columns are required.
*/
$config['database']['user_privilege_groups']['table'] = 'user_privilege_groups';
$config['database']['user_privilege_groups']['columns']['id'] = 'upriv_groups_id';
$config['database']['user_privilege_groups']['columns']['group_id'] =
'upriv_groups_ugrp_fk';
$config['database']['user_privilege_groups']['columns']['privilege_id'] =
'upriv_groups_upriv_fk';

#####

/**
 * User Login Session Table
 * The user login session table is used to validate user login credentials. For security
purposes, if a users credentitals do not match those
 * stored within the table, the user is automatically logged out.
 *
 * All columns are required.
*/
$config['database']['user_sess']['table'] = 'user_login_sessions';
$config['database']['user_sess']['join'] = 'user_login_sessions.usess_uacc_fk';
$config['database']['user_sess']['columns']['user_id'] = 'usess_uacc_fk';
$config['database']['user_sess']['columns']['series'] = 'usess_series';
$config['database']['user_sess']['columns']['token'] = 'usess_token';
$config['database']['user_sess']['columns']['date'] = 'usess_login_date';

#####
// Custom User Related Tables
#####

/**
 * Additional custom tables that are directly related to the user account table can be
included in flexi auth CRUD functions by
 * setting their database structure via the the $config['database'] array.
 *
 * Typically, such examples of a custom table you may wish to link to the user account
table would be a user profile table listing the
 * users name and contact details etc.
 *
 * You are not limited to the number of different custom tables you can define.
 *
 * ### Example Custom Table Template ###
 * $config['database']['custom']['#Array Alias#']['table'] = '#Actual table name#';
 * $config['database']['custom']['#Array Alias#']['primary_key'] = '#Table primary key#';
 * $config['database']['custom']['#Array Alias#']['foreign_key'] = '#Table foreign key
(Usually the table join column)#';
 * $config['database']['custom']['#Array Alias#']['join'] = '#Actual table name#.#Foreign
key column to main user table "user_acc"#';
 * $config['database']['custom']['#Array Alias#']['custom_columns'] =
array('#Column1#','#Column2#');
 *
 * Note: No custom tables are required to use flexi auth, see the demo config file for
examples of including custom tables.
*/
$config['database']['custom']['user_info']['table'] = 'user_account_info';
$config['database']['custom']['user_info']['primary_key'] = 'userinfo_id';
$config['database']['custom']['user_info']['foreign_key'] = 'userinfo_uacc_fk';

```

```

        $config['database']['custom']['user_info']['join'] =
'user_account_info.userinfo_uacc_fk';
        $config['database']['custom']['user_info']['custom_columns'] =
array('userinfo_first_name','userinfo_last_name');

        ###+++++#####
+++++#####
        // DATABASE SETTINGS
        ###+++++#####
+++++#####

        /**
         * flexi auth Database Settings
         * If required, it is possible to set your own column names and data types for some
database settings.
         *
         * Note: Only change the value after the '=' sign, and not the $config array names.
         * Example: Change $config['database']['settings']['example'] = 'example_value_1' to
$config['database']['settings']['example'] = 'example_value_2'
        */

        ###+++++#####

        /**
         * Primary User Identity Column
         * Set the column to be used to primarily identify users.
         *
         * Note: The column MUST be either the ['email'] or ['username'] columns from the main
user account table, and must contain a unique column name.
        */
        $config['database']['settings']['primary_identity_col'] = 'uacc_username';

        /**
         * User Identity Columns
         * Set whether the users email address, username or both are to be used to identify users
from data submitted via a login form.
         * This MUST include the ['primary_identity_col'] column set above (Default
'uacc_email').
         * If both the email address and username are used, then users will be able to login by
submitting either value.
         *
         * Note: The only valid columns are the users email address (Default column name
'uacc_email') or username (Default column name'uacc_username').
        */
        $config['database']['settings']['identity_cols'] = array('uacc_email', 'uacc_username');

        /**
         * User Search Query Columns
         * Set the table columns that are looked-up by the libraries search_users() function to
match users against submitted search query terms.
         *
         * Note: Any column within the user main account, custom or group tables can be added to
array
        */
        $config['database']['settings']['search_user_cols'] = array('uacc_email');

        /**
         * Database Date / Time Format
         * Set a native PHP function to format the date and time correctly to be stored within
the user tables.
         * Typically this will either be either DATETIME or TIMESTAMP.
         *
         * MySQL DATETIME = date('Y-m-d H:i:s');
         * Unix TIMESTAMP = time();
         *
         * Note: Ensure you consistently use the same data type in all defined flexi auth tables
for date and time data.
        */
        $config['database']['settings']['date_time'] = date('Y-m-d H:i:s');

```

```

#####
+++++++###
// SESSION NAMES / ALIASES
#####
+++++++###

/**
 * flexi auth Session
 * flexi auth uses CI sessions to store and serve authentication data between pages
loads.
 * All flexi auth session data is stored together within one session array, this helps
maintain a tidy session structure.
 *
 * If required, it is possible to set your own name for each session variable.
 * Note: Only change the name in the apostrophes (after the '=' sign), and not the
$config array names.
 * Example: Change $config['sessions']['user_id'] = 'user_id' to
$config['sessions']['user_id'] = 'new_session_name'
 */

#####

/**
 * Auth Session Name
 * Set the root auth session name saved as an array in the CI session, all other flexi
auth session data is then stored within this array.
 */
$config['sessions']['name'] = 'flexi_auth';

/**
 * Primary User Identifier Session
 * Contains the $config['database']['settings']['primary_identity_col'] column value
(Defined above).
 * This value is then used to internally identify the user when performing CRUD
functions.
 */
$config['sessions']['user_identifier'] = 'user_identifier';

/**
 * User Account Data Sessions
 * Used for performing various CRUD functions.
 */
$config['sessions']['user_id'] = 'user_id';
$config['sessions']['is_admin'] = 'admin';
$config['sessions']['group'] = 'group';
$config['sessions']['privileges'] = 'privileges';

/**
 * Login Via Password
 * Indicate whether the user logged in via entering a password or was logged in
automatically via the 'Remember me' function.
 */
$config['sessions']['logged_in_via_password'] = 'logged_in_via_password';

/**
 * Login Session Token
 * The login session token is used to help validate a users login credentials against a
stored database token.
 *
 * Note: Only used when $config['security']['validate_login_onload'] = TRUE (Defined
Below)
 */
$config['sessions']['login_session_token'] = 'login_session_token';

/**
 * Math Captcha Flash Session
 * Used to store the answer of a math captcha question, this data is stored only in a CI
flash session and so will only be available on the next page and is then deleted.
 */
$config['sessions']['math_captcha'] = 'math_captcha';

```

```

#####
+++++++
// COOKIE NAMES / ALIASES
#####
+++++++

/**
 * flexi auth Cookies
 * flexi auth uses cookies to store and serve authentication data for the next time a
user visits the website.
 *
 * If required, it is possible to set your own name for each cookie variable.
 * Note: Only change the name in the apostrophes (after the '=' sign), and not the
$config array names.
 * Example: Change $config['cookies']['user_id'] = 'user_id' to
$config['cookies']['user_id'] = 'new_session_name'
 */

#####

/**
 * 'Remember me' Cookies
 * Used to store 'Remember me' data to automatically log a user in next time they visit
the website.
 */
$config['cookies']['user_id'] = 'user_id';
$config['cookies']['remember_series'] = 'remember_series';
$config['cookies']['remember_token'] = 'remember_token';

/**
 * Login Session Cookie
 * The cookie login session token is used to invalidate a users login session when they
close their browser by deleting itself.
 *
 * Note: Only used when $config['security']['validate_login_onload'] = TRUE and
$config['security']['logout_user_onclose'] = TRUE (Defined Below)
 */
$config['cookies']['login_session_token'] = 'login_session_token';

/**
 * Login Via Password Cookie
 * The login via password cookie token is used to invalidate a users 'logged in via
password' status when they close their browser by deleting itself.
 *
 * Note: Only used when $config['security']['logout_user_onclose'] = FALSE (Defined
Below)
 */
$config['cookies']['login_via_password_token'] = 'login_via_password_token';

#####
+++++++
// SECURITY CONFIGURATIONS
#####
+++++++

/**
 * flexi auth Security Configurations
 * Many of flexi auths security features are customisable and can even be turned on and
off to suit different websites.
 *
 * Note: Only change the value after the '=' sign, and not the $config array names.
 * Example: Change $config['security']['example'] = TRUE to
$config['security']['example'] = FALSE
 */

#####
// LOGIN COOKIE AND SESSION SETTINGS
#####

```

```

/**
 * Set whether login details are validated on every page load.
 * @param bool
 *
 * TRUE = Login credentials are validated against the database everytime a page is
loaded, invalid users are logged out automatically.
 * FALSE = Login credentials are validated only once at time of login and will not expire
until CI sessions expire (Defined via CI config file).
 */
$config['security']['validate_login_onload'] = TRUE;

/**
 * Set the lifetime of a user login session in seconds.
 * @param int
 *
 * Example: 60*30 = 30 minutes, 60*60*24 = 1 day, 86400 = 1 day, 0 = Unlimited
 * Setting the value as '0' would mean the session would not expire until CIs own session
value (config['sess_expiration'] in CI config file) expired.
 *
 * Note: Only used when $config['security']['validate_login_onload'] = TRUE
 * !IMPORTANT:
 * If the CI config setting '$config['sess_expiration']' is lower, it will cause the
session to expire prior to the 'login_session_expire' value.
 * If 'Remember me' cookies are used, and a users login session expires, they will
remain logged in via the 'Remember me' cookie.
 * There are then functions within the library to check whether a user is logged in via
entering a password, or via a cookie - typically sensitive data should
 * only be available to users logged in via a password, and less sensitive data to
users logged in via 'Remember me' cookies.
 */
$config['security']['login_session_expire'] = 60*60*3;

/**
 * Set whether a users login time is extended when their session token is validated (On
every page load).
 * @param bool
 *
 * Note: Only used when $config['security']['validate_login_onload'] = TRUE
 */
$config['security']['extend_login_session'] = TRUE;

/**
 * Set whether a user is logged out as soon as the browser is closed.
 * Creates a cookie with a 0 lifetime that is deleted when the browser is closed.
 * This invalidates the users session the next time they visit the website as there is no
longer a matching cookie.
 * @param bool
 *
 * Note: Only used when $config['security']['validate_login_onload'] = TRUE
 * !IMPORTANT: 'logout_user_onclose' will also void any 'Remember me' cookies and so both
features should not be used together.
 */
$config['security']['logout_user_onclose'] = TRUE;

/**
 * Set whether a user has their 'logged in via password' status removed as soon as the
browser is closed.
 * If the user enabled the 'Remember me' feature on login, and their session is still
valid, they will have a 'logged in via "Remember me"' status on their next visit.
 * If the user did not enable the 'Remember me' feature on login, they will be logged out
on their next visit.
 *
 * If this setting is not enabled, a user who has logged in via password will have the
same login status if they close the browser and revisit the
 * site before the login session expires ('login_session_expire').
 *
 * The feature works by creating a cookie with a 0 lifetime that is deleted when the
browser is closed.
 * This invalidates the users session the next time they visit the website as there is no
longer a matching cookie.
 * @param bool

```

```

*
* Note: Only used when $config['security']['logout_user_onclose'] = FALSE
*/
$config['security']['unset_password_status_onclose'] = TRUE;

/**
 * Set the lifetime of a users login cookies in seconds, this includes the 'Remember me'
cookies.
 * @param int
 *
 * Example: 60*60*24 = 24 hours, 60*60*24*14 = 14 days, 86400 = 1 day
*/
$config['security']['user_cookie_expire'] = 60*60*24*14;

/**
 * Set whether a users 'Remember me' login cookies have their lifetime extended when
their session token is validated.
 * @param bool
*/
$config['security']['extend_cookies_on_login'] = TRUE;

#####
// PASSWORD SETTINGS
#####

/**
 * Set the minimum required characters for the users password.
 * @param int
*/
$config['security']['min_password_length'] = 8;

/**
 * Set which characters are valid for user passwords.
 * Default allows alpha-numeric, dashes, underscores, periods and commas ('\.\,\_\- a-z0-
9').
 * Note this is a regular expression.
*/
$config['security']['valid_password_chars'] = '\.\,\_\- a-z0-9';

/**
 * Set the static (non-database stored) salt used for password and hash token generation.
 * @param string
 *
 * !IMPORTANT:
 * Do NOT change this salt once users have started registering accounts as their
passwords will not work without the original salt.
 * CHANGE THE DEFAULT STATIC SALT SET BELOW TO YOUR OWN RANDOM SET OF CHARACTERS.
*/
$config['security']['static_salt'] = 'St.James1990';

/**
 * Set whether a salt is stored in the database and then used for password and hash token
generation.
 * @param bool
*/
$config['security']['store_database_salt'] = TRUE;

/**
 * Set the length of a stored database salt (See above).
 * @param int
 *
 * Note: Only used if $config['security']['store_database_salt'] = TRUE
*/
$config['security']['database_salt_length'] = 10;

/**
 * Set the expiry time of unused 'Forgotten Password' tokens.
 * Users will be required to request a new 'Forgotten Password' token once expired.
 * @param int
 *
 * Example: Time set in minutes, 0 = unlimited, 60*24 = 24 hours, 1440 = 24 hours.

```

```

*/
$config['security']['expire_forgotten_password'] = 15;

#####
// FAILED LOGIN ATTEMPT SETTINGS
#####

/**
 * Set a limit to the number of failed login attempts.
 * Once limit is passed, user is blocked from another attempt until time ban passes
(Defined by $config['security']['login_attempt_time_ban'] below).
 * Additionally/alternatively, a captcha can be set to show once this limit is reached by
using the 'ip_login_attempts_exceeded()' library function.
 * @param int
 *
 * Note: If a user exceeds 3 times the limit set, the resulting time ban is doubled to
further slow down attempts.
 * Example: 0 = unlimited attempts, 3 = 3 attempts.
*/
$config['security']['login_attempt_limit'] = 3;

/**
 * If a user has exceeded the failed login attempt limit, set the length of time they
must wait before they can attempt to login again.
 * @param int
 *
 * Note: The time ban is doubled if the failed attempts are 3 times higher than the limit
defined via $config['security']['login_attempt_limit'].
 * Example: If 'login_attempt_limit' = 3 and 'login_attempt_time_ban' = 10, after 3
failed attempts, the user must wait 10 seconds between each next attempt,
 * after 9 consecutive failed attempts, the user must wait 20 seconds between each next
attempt. Attempts within the time ban are ignored and not even checked as being valid.
 * !IMPORTANT: It is NOT recommended that this time ban is set for a long period of time
(> 5 mins).
 * Long time bans could be abused by attackers to deny legitimate users access, it is
designed to SLOW DOWN brute force attackers, not outright ban them.
 * Example: Time in seconds, 0 = no time ban, 10 = 10 seconds, 60*3 = 3 minutes.
*/
$config['security']['login_attempt_time_ban'] = 10;

#####
// Google reCAPTCHA SETTINGS
#####

/**
 * flexi auth Google reCAPTCHA Settings.
 * Google reCAPTCHA can be used to help slow down brute force login attempts, requiring
the user to complete the CAPTCHA before their login details will be submitted.
 *
 * Note: Only change the value after the '=' sign, and not the $config array names.
 * Example: Change $config['security']['example'] = 'example_value_1' to
$config['security']['example'] = 'example_value_2'
*/

#####

/**
 * reCAPTCHA Keys
 * Set your unique reCAPTCHA api keys.
 * !IMPORTANT: Obtain YOUR OWN reCAPTCHA keys from http://www.google.com/recaptcha.
*/
$config['security']['recaptcha_public_key'] = 'ENTER_RECAPTCHA_PUBLIC_KEY_HERE';
$config['security']['recaptcha_private_key'] = 'ENTER_RECAPTCHA_PRIVATE_KEY_HERE';

/**
 * Set the theme of the reCAPTCHA. For custom theming, see
https://developers.google.com/recaptcha/docs/customization
 * Predefined themes: 'red', 'white', 'blackglass', 'clean'. Set 'custom' for custom
themes.

```

```

*/
$config['security']['recaptcha_theme'] = 'white';

/**
 * Set the language of the reCAPTCHA.
 * Supported languages: English 'en', Dutch 'nl', French 'fr', German 'de', Portuguese
'pt', Russian 'ru', Spanish 'es', Turkish 'tr'.
*/
$config['security']['recaptcha_language'] = 'en';

#####
// GENERAL CONFIGURATION SETTINGS
#####

/**
 * General flexi auth Settings
 * Many of flexi auths automatic functions are customisable and can even be turned on and
off to suit different websites.
 *
 * Note: Only change the value after the '=' sign, and not the $config array names.
 * Example: Change $config['settings']['example'] = TRUE to
$config['settings']['example'] = FALSE
*/

#####

/**
 * Set whether an incremented number is added to the end of an unavailable username.
 * Example: If username 'flexi' is already in use, the next user to use 'flexi' as their
username will be automatically updated to 'flexi1'.
 * @param: bool
 *
 * Note: This only applies if the username is not set as the primary identity column
($config['database']['settings']['primary_identity_col'])
*/
$config['settings']['auto_increment_username'] = FALSE;

/**
 * Set whether accounts are suspended by default on registration / inserting user.
 * This option allows admins to verify account details before enabling users.
 * @param: bool
*/
$config['settings']['suspend_new_accounts'] = FALSE;

/**
 * Set a time limit to grant users instant login access, once expired, they are locked
out until they activate their account via an activation email sent to them.
 * @param: int
 *
 * Example: Time in minutes, 0 = unlimited, 60*24 = 24 hours, 1440 = 24 hours
*/
$config['settings']['account_activation_time_limit'] = 0;

/**
 * Set the id of the default group that new users will be added to unless otherwise
specified.
 * @param: int
*/
$config['settings']['default_group_id'] = 1;

/**
 * Set whether user privileges should be determined by individual privileges assigned per
user, or via privileges assigned to a users user group.
 * @param array
 *
 * Options: array('user','group'), array('user'), array('group')
 * Default: individual user privileges only.
*/

```



```

$config['settings']['privilege_sources'] = array('user');

#####
// EMAIL CONFIGURATION SETTINGS
#####

/**
 * flexi auth Email Settings
 * Some of the functions in flexi auth need to send emails to the user (i.e. 'Account
Activation', 'Forgot Password' etc).
 * If required, the title, reply address, email type and the content of these emails can
be configured to suit different website needs.
 *
 * Note: Only change the value after the '=' sign, and not the $config array names.
 * Example: Change $config['email']['example'] = 'example_value_1' to
$config['email']['example'] = 'example_value_2'
 */

#####

// Site title shown as 'from' header on emails.
$config['email']['site_title'] = "FILLES";

// Reply email shown as 'from' header on emails.
$config['email']['reply_email'] = "bonarae@gmail.com";

/**
 * Type of email to send, options: 'html', 'text'.
 * Note: If using 'text', the default code within the flexi auth templates use HTML which
will be emailed as plain text.
 */
$config['email']['email_type'] = 'html';

/**
 * Directory where email templates are stored.
 * Default: 'includes/email/'
 */
$config['email']['email_template_directory'] = 'includes/email/';

/**
 * 'Activate Account' email template.
 * Default: 'activate_account.tpl.php'
 */
$config['email']['email_template_activate'] = 'activate_account.tpl.php';

/**
 * 'Forgot Password' email template.
 * Default: 'forgot_password.tpl.php'
 */
$config['email']['email_template_forgot_password'] = 'forgot_password.tpl.php';

/**
 * 'Forgot Password Complete' email template.
 * Default: 'new_password.tpl.php'
 */
$config['email']['email_template_forgot_password_complete'] = 'new_password.tpl.php';

/**
 * 'Update Email' email template.
 * Default: 'update_email_address.tpl.php'
 */
$config['email']['email_template_update_email'] = 'update_email_address.tpl.php';

#####
// MESSAGE SETTINGS
#####

```

```

/**
 * Message Delimiter Settings
 * Define status and error message delimiters to style auth messages.
 * @param: string
 *
 * Example: ['status_prefix'] = '<p class="status_msg">', ['status_suffix'] = '</p>'
 */

// Message Start Delimiter
$config['messages']['delimiters']['status_prefix'] = '<p class="status_msg">';

// Message End Delimiter
$config['messages']['delimiters']['status_suffix'] = '</p>';

// Error Start Delimiter
$config['messages']['delimiters']['error_prefix'] = '<p class="error_msg">';

// Error End Delimiter
$config['messages']['delimiters']['error_suffix'] = '</p>';

#####

/**
 * Message Visibility
 * Define which status and error messages are returned as public or admin messages, or
which messages are not returned at all.
 * Public messages are intended to be displayed to public and admin users, whilst admin
messages are intended for admin users only.
 *
 * Example:
 * Public and Admin message =
$config['messages']['target_user']['account_creation_successful'] = 'public';
 * Admin Only message = $config['messages']['target_user']['account_creation_successful']
= 'admin';
 * Do NOT set public or admin message =
$config['messages']['target_user']['account_creation_successful'] = FALSE;
 */

// Account Creation
$config['messages']['target_user']['account_creation_successful'] = 'public';
$config['messages']['target_user']['account_creation_unsuccessful'] = 'public';
$config['messages']['target_user']['account_creation_duplicate_email'] = 'public';
$config['messages']['target_user']['account_creation_duplicate_username'] = 'public';
$config['messages']['target_user']['account_creation_duplicate_identity'] = 'public';
$config['messages']['target_user']['account_creation_insufficient_data'] = 'public';

// Password
$config['messages']['target_user']['password_invalid'] = 'public';
$config['messages']['target_user']['password_change_successful'] = 'public';
$config['messages']['target_user']['password_change_unsuccessful'] = 'public';
$config['messages']['target_user']['password_token_invalid'] = 'public';
$config['messages']['target_user']['email_new_password_successful'] = 'public';
$config['messages']['target_user']['email_forgot_password_successful'] = 'public';
$config['messages']['target_user']['email_forgot_password_unsuccessful'] = 'public';

// Activation
$config['messages']['target_user']['activate_successful'] = 'public';
$config['messages']['target_user']['activate_unsuccessful'] = 'public';
$config['messages']['target_user']['deactivate_successful'] = 'public';
$config['messages']['target_user']['deactivate_unsuccessful'] = 'public';
$config['messages']['target_user']['activation_email_successful'] = 'public';
$config['messages']['target_user']['activation_email_unsuccessful'] = 'public';
$config['messages']['target_user']['account_requires_activation'] = 'public';
$config['messages']['target_user']['account_already_activated'] = 'public';
$config['messages']['target_user']['email_activation_email_successful'] = 'public';
$config['messages']['target_user']['email_activation_email_unsuccessful'] = 'public';

// Login / Logout
$config['messages']['target_user']['login_successful'] = 'public';
$config['messages']['target_user']['login_unsuccessful'] = 'public';

```

```

$config['messages']['target_user']['logout_successful'] = 'public';
$config['messages']['target_user']['login_details_invalid'] = 'public';
$config['messages']['target_user']['captcha_answer_invalid'] = 'public';
$config['messages']['target_user']['login_attempts_exceeded'] = 'public';
$config['messages']['target_user']['login_session_expired'] = 'public';
$config['messages']['target_user']['account_suspended'] = 'public';

// Account Changes
$config['messages']['target_user']['update_successful'] = 'public';
$config['messages']['target_user']['update_unsuccessful'] = 'public';
$config['messages']['target_user']['delete_successful'] = 'public';
$config['messages']['target_user']['delete_unsuccessful'] = 'public';

// Form Validation
$config['messages']['target_user']['form_validation_duplicate_identity'] = 'public';
$config['messages']['target_user']['form_validation_duplicate_email'] = 'public';
$config['messages']['target_user']['form_validation_duplicate_username'] = 'public';

/* End of file flexi_auth.php */
/* Location: ./system/application/config/flexi_auth.php */

/application/config/routes.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
/*
| -----
| URI ROUTING
| -----
| This file lets you re-map URI requests to specific controller functions.
|
| Typically there is a one-to-one relationship between a URL string
| and its corresponding controller class/method. The segments in a
| URL normally follow this pattern:
|
|     example.com/class/method/id/
|
| In some instances, however, you may want to remap this relationship
| so that a different class/function is called than the one
| corresponding to the URL.
|
| Please see the user guide for complete details:
|
|     http://codeigniter.com/user_guide/general/routing.html
|
| -----
| RESERVED ROUTES
| -----
|
| There are two reserved routes:
|
|     $route['default_controller'] = 'welcome';
|
| This route indicates which controller class should be loaded if the
| URI contains no data. In the above example, the "welcome" class
| would be loaded.
|
|     $route['404_override'] = 'errors/page_missing';
|
| This route will tell the Router what URI segments to use if those provided
| in the URL cannot be matched to a valid route.
|
*/

$route['default_controller'] = 'index';

/* End of file routes.php */
/* Location: ./application/config/routes.php */

```

LIBRARIES

```
/application/libraries/auth.php  
<?php
```

```
class Auth {  
    public function __construct()  
    {  
        $this->auth = new stdClass;  
    }  
}  
?>
```

MODELS

```
/application/models/change_entries.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Change_entries extends CI_Model {
    public function __construct(){
        parent::__construct();
    }

    public function record_count() {
        return $this->db->count_all('changes');
    }

    public function fetch_entries($limit, $start) {
        $this->db->limit($limit,$start);
        $this->db->order_by('change_id','asc');
        $query = $this->db->get('changes');

        if ($query->num_rows() > 0) {
            foreach ($query->result() as $row) {
                $data[] = $row;
            }
            return $data;
        }
        return false;
    }

    public function get_offset() {
        $page = $this->input->offset('page');
        if (!$page) {
            $offset = 0;
        }
        else {
            $offset = $page;
        }
        return $offset;
    }
}

/* End of file: dictionary_entries.php */

/application/models/checked_model.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Checked_model extends CI_Model {
    function __construct(){
        parent::__construct();
    }

    function is_checked($chkname,$value)
    {
        if(!empty($_POST[$chkname]))
        {
            foreach($_POST[$chkname] as $chkval)
            {
                if($chkval == $value)
                {
                    return true;
                }
            }
        }
        return false;
    }
}

/* End of file: checked_model.php */

/application/models/dictionary_entries.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
```

```

class Dictionary_entries extends CI_Model {
    public function __construct(){
        parent::__construct();
    }

    public function record_count() {
        return $this->db->count_all('dictionary');
    }

    public function letter_record_count($start_letter) {
        $this->db->like('dictionary_term',$start_letter,'after');
        $this->db->from('dictionary');
        return $this->db->count_all_results();
    }

    public function fetch_entries($limit, $start) {
        $this->db->limit($limit,$start);
        $this->db->order_by('dictionary_term','asc');
        $query = $this->db->get('dictionary');

        if ($query->num_rows() > 0) {
            foreach ($query->result() as $row) {
                $data[] = $row;
            }
            return $data;
        }
        return false;
    }

    public function fetch_entries_by_letter($limit, $start, $start_letter) {
        $this->db->limit($limit,$start);
        $this->db->order_by('dictionary_term','asc');
        $this->db->like('dictionary_term',$start_letter,'after');
        $query = $this->db->get('dictionary');

        if ($query->num_rows() > 0) {
            foreach ($query->result() as $row) {
                $data[] = $row;
            }
            return $data;
        }
        return false;
    }

    public function get_offset() {
        $page = $this->input->offset('page');
        if (!$page) {
            $offset = 0;
        }
        else {
            $offset = $page;
        }
        return $offset;
    }
}

```

/* End of file: dictionary_entries.php */

/application/models/grammar_entries.php

<?php if (! defined('BASEPATH')) exit('No direct script access allowed');

```

class Grammar_entries extends CI_Model {
    public function __construct(){
        parent::__construct();
    }

    public function record_count() {
        return $this->db->count_all('grammar');
    }
}

```

```

public function letter_record_count($start_letter) {
    $this->db->like('grammar_rule',$start_letter,'after');
    $this->db->from('grammar');
    return $this->db->count_all_results();
}

public function fetch_entries($limit, $start) {
    $this->db->limit($limit,$start);
    $query = $this->db->get('grammar');

    if ($query->num_rows() > 0) {
        foreach ($query->result() as $row) {
            $data[] = $row;
        }
        return $data;
    }
    return false;
}

public function fetch_entries_by_letter($limit, $start, $start_letter) {
    $this->db->limit($limit,$start);
    $this->db->order_by('grammar_rule','asc');
    $this->db->like('grammar_rule',$start_letter,'after');
    $query = $this->db->get('grammar');

    if ($query->num_rows() > 0) {
        foreach ($query->result() as $row) {
            $data[] = $row;
        }
        return $data;
    }
    return false;
}

public function get_offset() {
    $page = $this->input->offset('page');
    if (!$page) {
        $offset = 0;
    }
    else {
        $offset = $page;
    }
    return $offset;
}
}

```

/* End of file: grammar_entries.php */

/application/models/instruction_entries.php

<?php if (! defined('BASEPATH')) exit('No direct script access allowed');

```

class Instruction_entries extends CI_Model {
    public function __construct(){
        parent::__construct();
    }

    public function record_count() {
        return $this->db->count_all('bank_instructions');
    }

    public function fetch_entries($limit, $start) {
        $this->db->limit($limit,$start);
        $query = $this->db->get('bank_instructions');

        if ($query->num_rows() > 0) {
            foreach ($query->result() as $row) {
                $data[] = $row;
            }
            return $data;
        }
    }
}

```

```

        return false;
    }

    public function get_offset() {
        $page = $this->input->offset('page');
        if (!page) {
            $offset = 0;
        }
        else {
            $offset = $page;
        }
        return $offset;
    }
}

/* End of file: instruction_entries.php */

/application/models/lesson_entries.php

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Lesson_entries extends CI_Model {
    public function __construct(){
        parent::__construct();
    }

    public function record_count() {
        return $this->db->count_all('lessons');
    }

    public function fetch_entries_without_page() {
        $query = $this->db->get('lessons');

        if ($query->num_rows() > 0) {
            foreach ($query->result() as $row) {
                $data[] = $row;
            }
            return $data;
        }
        return false;
    }

    public function fetch_entries($limit, $start) {
        $this->db->limit($limit,$start);
        $query = $this->db->get('lessons');

        if ($query->num_rows() > 0) {
            foreach ($query->result() as $row) {
                $data[] = $row;
            }
            return $data;
        }
        return false;
    }

    public function get_offset() {
        $page = $this->input->offset('page');
        if (!page) {
            $offset = 0;
        }
        else {
            $offset = $page;
        }
        return $offset;
    }
}

/* End of file: lesson_entries.php */

/application/models/link_model.php

```



```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Link_model extends CI_Model {
    public function __construct(){
        parent::__construct();
    }

    public function record_count() {
        return $this->db->count_all('links');
    }

    public function approved_count() {
        $this->db->where('is_approved','1');
        $this->db->from('links');
        return $this->db->count_all_results();
    }

    public function record_count_by_user() {
        $user_to_select = $this->flexi_auth->get_user_identity();
        $this->db->where('contributed_by',$user_to_select);
        $this->db->from('links');
        return $this->db->count_all_results();
    }

    public function fetch_entries($limit, $start) {
        $this->db->limit($limit,$start);
        $query = $this->db->get('links');

        if ($query->num_rows() > 0) {
            foreach ($query->result() as $row) {
                $data[] = $row;
            }
            return $data;
        }
        return false;
    }

    public function fetch_approved_entries($limit, $start) {
        $query = $this->db->get_where('links',array('is_approved' => '1'), $limit,
    $start);

        if ($query->num_rows() > 0) {
            foreach ($query->result() as $row) {
                $data[] = $row;
            }
            return $data;
        }
        return false;
    }

    public function fetch_all_user_entries() {
        $user_to_select = $this->flexi_auth->get_user_identity();
        $this->db->where('contributed_by',$user_to_select);
        $query = $this->db->get('links');

        if ($query->num_rows() > 0) {
            foreach ($query->result() as $row) {
                $data[] = $row;
            }
            return $data;
        }
        return false;
    }

    public function fetch_user_entries($limit, $start) {
        $user_to_select = $this->flexi_auth->get_user_identity();
        $this->db->where('contributed_by',$user_to_select);
        $this->db->limit($limit,$start);
        $query = $this->db->get('links');
    }
}

```

```

        if ($query->num_rows() > 0) {
            foreach ($query->result() as $row) {
                $data[] = $row;
            }
            return $data;
        }
        return false;
    }
}

public function get_offset() {
    $page = $this->input->offset('page');
    if (!$page) {
        $offset = 0;
    }
    else {
        $offset = $page;
    }
    return $offset;
}
}

}

/* End of file: link_model.php */

/application/models/phrasebook_entries.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Phrasebook_entries extends CI_Model {
    public function __construct(){
        parent::__construct();
    }

    public function record_count() {
        return $this->db->count_all('phrasebook');
    }

    public function letter_record_count($start_letter) {
        $this->db->like('dictionary_term',$start_letter,'after');
        $this->db->from('dictionary');
        return $this->db->count_all_results();
    }

    public function fetch_entries($limit, $start) {
        $this->db->limit($limit,$start);
        $this->db->order_by('phrasebook_phrase','asc');
        $query = $this->db->get('phrasebook');

        if ($query->num_rows() > 0) {
            foreach ($query->result() as $row) {
                $data[] = $row;
            }
            return $data;
        }
        return false;
    }

    public function fetch_entries_by_letter($limit, $start, $start_letter) {
        $this->db->limit($limit,$start);
        $this->db->order_by('phrasebook_phrase','asc');
        $this->db->like('phrasebook_phrase',$start_letter,'after');
        $query = $this->db->get('phrasebook');

        if ($query->num_rows() > 0) {
            foreach ($query->result() as $row) {
                $data[] = $row;
            }
            return $data;
        }
        return false;
    }
}

```

```

    public function get_offset() {
        $page = $this->input->offset('page');
        if (!$page) {
            $offset = 0;
        }
        else {
            $offset = $page;
        }
        return $offset;
    }
}

/* End of file: phrasebook_entries.php */

/application/models/progress_model.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Progress_model extends CI_Model {
    public function __construct(){
        parent::__construct();
    }

    public function record_count() {
        $user_id = $this->flexi_auth->get_user_id();
        $this->db->where('user_id',$user_id);
        $this->db->from('quizzes_taken');
        return $this->db->count_all_results();
    }

    public function get_entries() {
        $query = $this->db->get('quizzes_taken');

        if ($query->num_rows() > 0) {
            foreach ($query->result() as $row) {
                $data[] = $row;
            }
            return $data;
        }
        return false;
    }
}

/* End of file: progress_model.php */

/application/models/question_entries.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Question_entries extends CI_Model {
    public function __construct(){
        parent::__construct();
    }

    public function record_count() {
        return $this->db->count_all('question_bank');
    }

    public function fetch_entries($limit, $start) {
        $this->db->limit($limit,$start);
        $query = $this->db->get('question_bank');

        if ($query->num_rows() > 0) {
            foreach ($query->result() as $row) {
                $data[] = $row;
            }
            return $data;
        }
        return false;
    }

    public function get_offset() {

```

```

        $page = $this->input->offset('page');
        if (!page) {
            $offset = 0;
        }
        else {
            $offset = $page;
        }
        return $offset;
    }
}

/* End of file: question_entries.php */

/application/models/quiz_model.php

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Quiz_model extends CI_Model {
    public function __construct(){
        parent::__construct();
    }

    public function generate_entries() {
        $this->db->order_by('question_no','random');
        $this->db->limit(10);
        $query = $this->db->get('question_bank');

        foreach ($query->result() as $row) {
            $data[] = $row;
        }
        return $data;
    }

    public function generate_entries_by_lesson($lesson) {
        $this->db->order_by('question_no','random');
        $query = $this->db->get_where('question_bank',array('lesson_id' => $lesson), 10);

        foreach ($query->result() as $row) {
            $data[] = $row;
        }
        return $data;
    }

    public function fetch_instructions($id) {
        $this->db->where('lesson_id',$id);
        $query = $this->db->get('bank_instructions');

        if ($query->num_rows() > 0) {
            foreach ($query->result() as $row) {
                $data[] = $row;
            }
            return $data;
        }
        return false;
    }

    public function check_correct_answer($correct_answer,$answer_to_check) {
        if ($answer_to_check == $correct_answer) {
            return true;
        }
    }
}

/* End of file: quiz_model.php */

```

CONTROLLERS

/application/controllers/about.php

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class About extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    public function index() {
        $data['pageTitle'] = "About FILLES";
        if ($this->flexi_auth->in_group('users')) {
            $this->load->view('header_user',$data);
        }
        else if ($this->flexi_auth->in_group('contributors')) {
            $this->load->view('header_contributor', $data);
        }
        else {
            $this->load->view('header', $data);
        }
        $this->load->view('about_view');
        $this->load->view('footer');
    }
}

/* End of file: about.php */
```

/application/controllers/add_admin.php

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_admin extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->library('form_validation');
    }
    public function index() {
        $data['pageTitle'] = "Add System Admin";
        $data['message'] = "";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth->
in_group('sysadmin'))) {
            redirect('/', 'refresh');
        }
        $this->load->view('header_sysadmin', $data);
        $this->load->view('add_admin_view', $data);
        $this->load->view('footer');
    }
}

/* End of file: add_admin.php */
```

/application/controllers/add_admin_process.php

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_admin_process extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->library('form_validation');
    }

    public function index() {
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth->
in_group('sysadmin'))) {
            redirect('/', 'refresh');
        }
        else {
            if ($this->input->post('submit_admin')) {
```

```

$username = strtolower($this->input->post('username'));
$password = $this->input->post('password');
$first_name = $this->input->post('first_name');
$last_name = $this->input->post('last_name');
$email = $this->input->post('email');

$additional_data = array(

'userinfo_first_name' => $first_name,

'userinfo_last_name' => $last_name
);

$group = 1;

$reg_staff = $this->flexi_auth-
>insert_user($email,$username,$password,$additional_data,$group,TRUE);

if (!$reg_staff) {
    $data['message'] = "System Admin was not
successfully added.";
    redirect('add_admin',$data);
}
else {
    redirect('add_admin_success');
}
}
else {
    $this->data['message'] = (validation_errors() ?
validation_errors() : ($this->ion_auth->errors() ? $this->ion_auth->errors() : $this->session-
>flashdata('message')));

    $this->data['first_name'] = array(
        'name' => 'first_name',
        'id' => 'first_name',
        'type' => 'text',
        'value' => $this->form_validation-
>set_value('first_name'),
    );
    $this->data['last_name'] = array(
        'name' => 'last_name',
        'id' => 'last_name',
        'type' => 'text',
        'value' => $this->form_validation-
>set_value('last_name'),
    );
    $this->data['email'] = array(
        'name' => 'email',
        'id' => 'email',
        'type' => 'text',
        'value' => $this->form_validation-
>set_value('email'),
    );
    $this->data['password'] = array(
        'name' => 'password',
        'id' => 'password',
        'type' => 'password',
        'value' => $this->form_validation-
>set_value('password'),
    );
    $this->data['confirm_password'] = array(
        'name' => 'confirm_password',
        'id' => 'confirm_password',
        'type' => 'password',
        'value' => $this->form_validation-
>set_value('confirm_password'),
    );
    redirect('add_admin',$this->data);
}
}
}

```

```

    }

/* End of file: add_admin_process.php */

/application/controllers/add_admin_success.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_admin_success extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->library('form_validation');
    }
    public function index() {
        $data['pageTitle'] = "Add System Admin Successful";
        $data['message'] = "";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('sysadmin'))) {
            redirect('/', 'refresh');
        }
        $this->load->view('header_sysadmin', $data);
        $this->load->view('add_admin_success_view', $data);
        $this->load->view('footer');
    }
}

/* End of file: add_admin_success.php */

/application/controllers/add_bank_question.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_bank_question extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index($msg = NULL) {
        $data['pageTitle'] = "Add Bank Question";
        $this->load->library('form_validation');
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('add_bank_question_view', $msg);
            $this->load->view('footer');
        }
    }
}

/* End of file: add_bank_question.php */

/application/controllers/add_content_admin.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_content_admin extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->library('form_validation');
    }
    public function index() {
        $data['pageTitle'] = "Add Content Admin";
        $data['message'] = "";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('sysadmin'))) {
            redirect('/', 'refresh');
        }
        $this->load->view('header_sysadmin', $data);
        $this->load->view('add_content_admin_view', $data);
        $this->load->view('footer');
    }
}

```

```

    }

/* End of file: add_content_admin.php */

/application/controllers/add_content_admin_process.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_content_admin_process extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->library('form_validation');
    }

    public function index() {
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('sysadmin')) {
            redirect('/', 'refresh');
        }
        else {
            if ($this->input->post('submit_content_admin')) {
                $username = strtolower($this->input->post('username'));
                $password = $this->input->post('password');
                $first_name = $this->input->post('first_name');
                $last_name = $this->input->post('last_name');
                $email = $this->input->post('email');

                $additional_data = array(
                    'userinfo_first_name' => $first_name,
                    'userinfo_last_name' => $last_name
                );

                $group = 2;

                $reg_staff = $this->flexi_auth-
>insert_user($email, $username, $password, $additional_data, $group, TRUE);

                if (!$reg_staff) {
                    $data['message'] = "Content Admin was not
successfully added.";

                    redirect('add_content_admin', $data);
                }
                else {
                    redirect('add_content_admin_success');
                }
            }
            else {
                $this->data['message'] = (validation_errors() ?
validation_errors() : ($this->ion_auth->errors() ? $this->ion_auth->errors() : $this->session-
>flashdata('message')));

                $this->data['first_name'] = array(
                    'name' => 'first_name',
                    'id' => 'first_name',
                    'type' => 'text',
                    'value' => $this->form_validation-
>set_value('first_name'),
                );
                $this->data['last_name'] = array(
                    'name' => 'last_name',
                    'id' => 'last_name',
                    'type' => 'text',
                    'value' => $this->form_validation-
>set_value('last_name'),
                );
                $this->data['email'] = array(
                    'name' => 'email',
                    'id' => 'email',
                    'type' => 'text',
                    'value' => $this->form_validation-
>set_value('email'),
                );
                $this->data['password'] = array(
                    'name' => 'password',

```



```

        'id'    => 'password',
        'type' => 'password',
        'value' => $this->form_validation-
>set_value('password'),
    );
    $this->data['confirm_password'] = array(
        'name' => 'confirm_password',
        'id'   => 'confirm_password',
        'type' => 'password',
        'value' => $this->form_validation-
>set_value('confirm_password'),
    );

    redirect('add_content_admin',$this->data);
}
}
}

/* End of file: add_content_admin_process.php */

/application/controllers/add_content_admin_success.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_content_admin_success extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->library('form_validation');
    }
    public function index() {
        $data['pageTitle'] = "Add Content Admin Successful";
        $data['message'] = "";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('sysadmin'))) {
            redirect('/', 'refresh');
        }
        $this->load->view('header_sysadmin', $data);
        $this->load->view('add_content_admin_success_view',$data);
        $this->load->view('footer');
    }
}

/* End of file: add_content_admin_success.php */

/application/controllers/add_contributor.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_contributor extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->library('form_validation');
    }

    function index() {
        if ($this->input->post('submit_contributor')) {
            $username = strtolower($this->input->post('username'));
            $password = $this->input->post('password');
            $first_name = $this->input->post('first_name');
            $last_name = $this->input->post('last_name');
            $email = $this->input->post('email');

            $additional_data = array(
                'userinfo_first_name' => $first_name,
                'userinfo_last_name' => $last_name
            );

            $group = 4;

            $reg_contributor = $this->flexi_auth-
>insert_user($email,$username,$password,$additional_data,$group,TRUE);

            if ($reg_contributor) {
                redirect('add_contributor_success');
            }
        }
    }
}

```

```

        else {
            $data['message'] = "Contributor was not successfully
added.";
            redirect('contributor_signup',$data);
        }
    }
    else {
        $this->data['message'] = (validation_errors() ? validation_errors()
: ($this->ion_auth->errors() ? $this->ion_auth->errors() : $this->session-
>flashdata('message')));

        $this->data['first_name'] = array(
            'name' => 'first_name',
            'id'   => 'first_name',
            'type' => 'text',
            'value' => $this->form_validation->set_value('first_name'),
        );
        $this->data['last_name'] = array(
            'name' => 'last_name',
            'id'   => 'last_name',
            'type' => 'text',
            'value' => $this->form_validation->set_value('last_name'),
        );
        $this->data['email'] = array(
            'name' => 'email',
            'id'   => 'email',
            'type' => 'text',
            'value' => $this->form_validation->set_value('email'),
        );
        $this->data['password'] = array(
            'name' => 'password',
            'id'   => 'password',
            'type' => 'password',
            'value' => $this->form_validation->set_value('password'),
        );
        $this->data['confirm_password'] = array(
            'name' => 'confirm_password',
            'id'   => 'confirm_password',
            'type' => 'password',
            'value' => $this->form_validation-
>set_value('confirm_password'),
        );

        redirect('contributor_signup',$this->data);
    }
}
}
}

```

/* End of file: add_contributor.php */

/application/controllers/add_contributor_success.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_contributor_success extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    public function index() {
        $data['pageTitle'] = "Contributor Signup Submitted";
        $this->load->view('header', $data);
        $this->load->view('add_contributor_success_view');
        $this->load->view('footer');
    }
}

```

/* End of file: add_contributor_success.php */

/application/controllers/add_dictionary_entry.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_dictionary_entry extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
}

```

```

    }

    function index() {
        $data['pageTitle'] = "Add Dictionary Entry";
        $this->load->library('form_validation');
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('add_dictionary_entry_view');
            $this->load->view('footer');
        }
    }
}

}

/* End of file: add_dictionary_entry.php */

/application/controllers/add_dictionary_entry_process.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_dictionary_entry_process extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->library('upload');
    }

    function index() {
        $this->load->library('form_validation');
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {

            $term = $this->input->post('term');
            $part_of_speech = $this->input->post('part_of_speech');
            $pronunciation = $this->input->post('pronunciation');
            $meaning = $this->input->post('meaning');
            $category = $this->input->post('category');
            $usage = $this->input->post('usage');

            if (!empty($_FILES['userfile']['name'])) {
                $config['upload_path'] = './images/';
                $config['allowed_types'] = 'gif|jpg|png';
                $config['max_size'] = '2048';

                $this->upload->initialize($config);

                if ($this->upload->do_upload('userfile')) {
                    $image_data_array = $this->upload->data();
                    $image_link =
$image_data_array['full_path'];
                }
                else {
                    $error = $this->upload->display_errors();
                    $data['pageTitle'] = "Add Dictionary Entry

Unsuccessful";

                    $this->load->view('header_staff', $data);
                    $this->load-
>view('add_dictionary_entry', $error);
                }
            }

            if (!empty($_FILES['userfile1']['name'])) {
                $config['upload_path'] = './sounds/';
                $config['allowed_types'] = 'wav';
                $config['max_size'] = '2048';

                $this->upload->initialize($config);

```

```

        if ($this->upload->do_upload('userfile1')) {
            $audio_data_array = $this->upload->data();
            $audio_link =
$audio_data_array['full_path'];
        }
        else {
            $error = $this->upload->display_errors();
            $data['pageTitle'] = "Add Dictionary Entry
Unsuccessful";
            $this->load->view('header_staff', $data);
            $this->load-
>view('add_dictionary_entry',$error);
        }
    }

    $dictionary_data = array('dictionary_term' => $term,
'dictionary_meaning' => $meaning, 'dictionary_partofspeech' => $part_of_speech,
'dictionary_pronunciation' => $pronunciation, 'dictionary_usage' => $usage,
'dictionary_image_link' => $image_link, 'dictionary_audio_link' => $audio_link,
'dictionary_category' => $category);
    $query = $this->db->insert('dictionary',$dictionary_data);
    $query2 = $this->db-
>get_where('dictionary',array('dictionary_term' => $term));
    $new_row = $query2->row();
    if ($query) {
        $new_row_no = $new_row->dictionary_id;
        redirect('view_dictionary_entry?id=' . $new_row_no);
    }
    else {
        $data['pageTitle'] = "Add Dictionary Entry
Unsuccessful";
        $this->load->view('header_staff', $data);
        $this->load->view('add_dictionary_entry','Dictionary
entry already exists.');
```

}
 \$this->load->view('footer');

}
 }
}
}
}

/* End of file: add_dictionary_entry_process.php */

/application/controllers/add_grammar_entry.php
<?php if (! defined('BASEPATH')) exit('No direct script access allowed');
class Add_grammar_entry extends CI_Controller {
 function __construct() {
 parent::__construct();
 }

 function index() {
 \$data['pageTitle'] = "Add Grammar Rule";
 \$this->load->library('form_validation');
 if (!((\$this->flexi_auth->is_logged_in()) || (!\$this->flexi_auth-
>in_group('staff')))) {
 redirect('/', 'refresh');
 }
 else {
 \$this->load->view('header_staff', \$data);
 \$this->load->view('add_grammar_entry_view');
 \$this->load->view('footer');

}
 }
}

/* End of file: add_grammar_entry.php */

/application/controllers/add_grammar_entry.php
<?php if (! defined('BASEPATH')) exit('No direct script access allowed');

```

class Add_grammar_entry_process extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        $this->load->library('form_validation');
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff')) {
            redirect('/', 'refresh');
        }
        else {
            $rule = $this->input->post('rule');
            $usage = $this->input->post('usage');
            $example = $this->input->post('example');
            $category = $this->input->post('category');

            $grammar_data = array('grammar_rule' => $rule,
'grammar_usage' => $usage, 'grammar_example' => $example, 'grammar_category' => $category);
            $query = $this->db->insert('grammar', $grammar_data);
            if ($query) {
                $data['pageTitle'] = "Adding of Grammar Entry
Successful";

                $this->load->view('header_staff', $data);

                $this->load->view('add_grammar_entry_success_view');
            }
            else {
                $data['pageTitle'] = "Adding of Grammar Entry
Unsuccessful";

                $this->load->view('header_staff', $data);

                $this->load->view('add_grammar_entry', 'Grammar rule
entry already exists.');
```

```

        }
        $this->load->view('footer');
```

```

    }
}

/* End of file: add_grammar_entry_process.php */

/application/controllers/add_guides.php
```

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_guides extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        $data['pageTitle'] = "Add Guide Contents";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff')) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('add_guides_view');
            $this->load->view('footer');
```

```

        }
    }
}

/* End of file: add_guides.php */
```

```

/application/controllers/add_instruction.php
```

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_instruction extends CI_Controller {
    function __construct() {
        parent::__construct();
```

```

    }

    function process() {
        $text = $this->input->post('text');
        $lesson_id = $this->input->post('lesson_id');

        $instruction_data = array('instruction_text' => $text, 'lesson_id' =>
$lesson_id);

        $query = $this->db->insert('bank_instructions',$instruction_data);
        $query2 = $this->db-
>get_where('bank_instructions',array('instruction_text' => $text));
        if ($query) {
            $row = $query2->row();
            $instruction_no = $row->instruction_id;
            redirect('view_instruction?id=' . $instruction_no);
        }
        else {
            $data['pageTitle'] = "Query Failed";
            $this->load->view('header_staff', $data);
            $this->load->view('query_failure_view');
            $this->load->view('footer');
        }
    }

    function index() {
        $data['pageTitle'] = "Add Instruction";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('add_instruction_view');
            $this->load->view('footer');
        }
    }
}

/* End of file: add_instruction.php */

/application/controllers/add_lesson.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_lesson extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        $data['pageTitle'] = "Add Lesson";
        $data['error'] = "";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('add_lesson_view');
            $this->load->view('footer');
        }
    }
}

/* End of file: add_lesson.php */

/application/controllers/add_lesson_process.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_lesson_process extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
}

```

```

    }

    function index() {
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $config['upload_path'] = "./lessons/";
            $config['allowed_types'] = "pdf";
            $config['max_size'] = '2048';

            $this->load->library('upload', $config);

            $title = $this->input->post('title');
            $description = $this->input->post('description');
            if (!$this->upload->do_upload()) {
                $data['pageTitle'] = "Upload Error";
                $error = array('error' => $this->upload->display_errors());
                $this->load->view('header_staff', $data);
                $this->load->view('add_lesson_view', $error);
                $this->load->view('footer');
            }
            else {
                $upload_array = $this->upload->data();
                $content = $upload_array['full_path'];

                $lesson_data = array('lesson_title' => $title,
'lesson_description' => $description, 'lesson_content' => $content);
                $query = $this->db->insert('lessons', $lesson_data);
                $query2 = $this->db-
>get_where('lessons', array('lesson_title' => $title));
                $new_row = $query2->row();
                if ($query) {
                    $record_no = $new_row->lesson_id;
                    redirect('lesson_entry?id=' . $record_no);
                }
                else {
                    $data['pageTitle'] = "Add Lesson Unsuccessful";
                    $this->load->view('header_staff', $data);
                    $this->load->view('add_lesson_view', 'A lesson with
the same title already exists.');
```

```

        $this->form_validation->set_message('_check_math_captcha', 'Enter a valid math
captcha response.');
```

```

        return FALSE;
    }
}

public function index() {
    $config["operation"] = "random";
    $config["question_format"] = "random";
    $this->mathcaptcha->init($config);

    $data['math_captcha_question'] = $this->mathcaptcha->get_question();

    $this->form_validation->set_rules('math_captcha', 'Math CAPTCHA',
'required|callback__check_math_captcha');

    $group = array('contributors','staff');
    if ($this->flexi_auth->in_group($group)) {
        $data['pageTitle'] = "Add Link";
        if($this->flexi_auth->in_group('contributors')) {
            $this->load->view('header_contributor', $data);
        }
        else if ($this->flexi_auth->in_group('staff')) {
            $this->load->view('header_staff', $data);
        }
        if ($this->form_validation->run() == FALSE)
        {
            $this->load->view('add_link_view',$data);
        }
        else
        {
            $this->load->view('add_link_view',$data);
        }
        $this->load->view('footer');
    }
    else {
        redirect('/');
    }
}

}

/* End of file: add_link.php */

/application/controllers/add_link_process.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_link_process extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->library('form_validation');
    }

    function index() {
        $group = array('contributors','staff');
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group($group))) {
            redirect('/', 'refresh');
        }
        else {
            $title = $this->input->post('title');
            $description = $this->input->post('description');
            $url = $this->input->post('url');
            $contributor = $this->flexi_auth->get_user_identity();

            $link_data = array('link_title' => $title, 'link_description' =>
$description, 'link_url' => $url, 'is approved' => '0', 'contributed_by' => $contributor);
            $query = $this->db->insert('links',$link_data);
            if ($query) {
                $data['pageTitle'] = "Add Link Submitted for Approval";
                if($this->flexi_auth->in_group('contributors')) {
                    $this->load->view('header_contributor', $data);
                }
            }
        }
    }
}

```



```

        else if ($this->flexi_auth->in_group('staff')) {
            $this->load->view('header_staff', $data);
        }
        $this->load->view('add_link_success_view');
    }
    else {
        $data['pageTitle'] = "Add Link Unsuccessful";
        if($this->flexi_auth->in_group('contributors')) {
            $this->load->view('header_contributor', $data);
        }
        else if ($this->flexi_auth->in_group('staff')) {
            $this->load->view('header_staff', $data);
        }
        $this->load->view('add_link_view','A link with the same
title already exists.');
```

```

    }
    $this->load->view('footer');
}
}

}

/* End of file: add_link_process.php */
```

/application/controllers/add_phrasebook_entry.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_phrasebook_entry extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        $data['pageTitle'] = "Add Phrasebook Entry";
        $data['error'] = "";
        $this->load->library('form_validation');
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('add_phrasebook_entry_view',$data);
            $this->load->view('footer');
        }
    }
}

/* End of file: add_phrasebook_entry.php */
```

/application/controllers/add_phrasebook_entry_process.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_phrasebook_entry_process extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $config['upload_path'] = "./sounds/";
            $config['allowed_types'] = "wav";
            $config['max_size'] = '2048';

            $this->load->library('upload',$config);

            $phrase = $this->input->post('phrase');
            $pronunciation = $this->input->post('pronunciation');
```

```

        $translation = $this->input->post('translation');
        $category = $this->input->post('category');
        if (!empty($_FILES['userfile']['name'])) {
            if ($this->upload->do_upload()) {
                $upload_array = $this->upload->data();
                $audio_link = $upload_array['full_path'];
                $phrasebook_data = array('phrasebook_phrase' =>
$phrase, 'phrasebook_pronunciation' => $pronunciation, 'phrasebook_translation' => $translation,
'phrasebook_category' => $category, 'phrasebook_audio_link' => $audio_link);
                $query = $this->db-
>insert('phrasebook',$phrasebook_data);
                $query2 = $this->db-
>get_where('phrasebook',array('phrasebook_phrase' => $phrase));
                if ($query) {
                    $row = $query2->row();
                    $new_record_no = $row->phrasebook_id;
                    redirect('view_phrasebook_entry?id=' .
$new_record_no);
                }
                else {
                    $data['pageTitle'] = "Phrasebook Entry Not
Successfully Created";
                    $this->load->view('header_staff',$data);
                    $this->load-
>view('add_phrasebook_entry','Phrasebook entry already exists.');
```

```

class Add_question_process extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth->in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $question_type = $this->input->post('question_type');
            $question_name = $this->input->post('question_name');
            $choice1 = $this->input->post('choice1');
            $choice2 = $this->input->post('choice2');
            $choice3 = $this->input->post('choice3');
            $choice4 = $this->input->post('choice4');
            $correct_answer = $this->input->post('correct_answer');
            $lesson_id = $this->input->post('lesson_id');

            $question_data = array('question_type' => $question_type,
'question_title' => $question_name, 'choice_a' => $choice1, 'choice_b' => $choice2, 'choice_c' =>
$choice3, 'choice_d' => $choice4, 'correct_answer' => $correct_answer, 'lesson_id' =>
$lesson_id);

            $query = $this->db->insert('question_bank', $question_data);
            if ($query) {
                $row = $query->row();
                $question_id = $row->question_no;
                redirect('view_question?id=' . $question_id);
            }
            else {
                $data['pageTitle'] = "Add Question Unsuccessful";
                $this->load->view('header_staff', $data);
                $this->load->view('add_bank_question_view', 'A question with
the same title already exists.');
```

```

        }
    }
}

/* End of file: add_question_process.php */
```

```

/application/controllers/add_staff.php
```

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_staff extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->library('form_validation');
    }
    public function index() {
        $data['pageTitle'] = "Add SWF Staff";
        $data['message'] = "";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth->in_group('sysadmin'))) {
            redirect('/', 'refresh');
        }
        $this->load->view('header_admin', $data);
        $this->load->view('add_staff_view', $data);
        $this->load->view('footer');
    }
}

/* End of file: add_staff.php */
```

```

/application/controllers/add_staff_process.php
```

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_staff_process extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->library('form_validation');
```

```

    }

    public function index() {
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('sysadmin'))) {
            redirect('/', 'refresh');
        }
        else {
            if ($this->input->post('submit_staff')) {
                $username = strtolower($this->input->post('username'));
                $password = $this->input->post('password');
                $first_name = $this->input->post('first_name');
                $last_name = $this->input->post('last_name');
                $email = $this->input->post('email');

                $additional_data = array(
                    'userinfo_first_name' => $first_name,
                    'userinfo_last_name' => $last_name
                );

                $group = 3;

                $reg_staff = $this->flexi_auth-
>insert_user($email, $username, $password, $additional_data, $group, TRUE);

                if (!$reg_staff) {
                    $data['message'] = "Staff member was not
successfully added.";

                    redirect('add_staff', $data);
                }
                else {
                    redirect('add_staff_success');
                }
            }
            else {
                $this->data['message'] = (validation_errors() ?
validation_errors() : ($this->ion_auth->errors() ? $this->ion_auth->errors() : $this->session-
>flashdata('message')));

                $this->data['first_name'] = array(
                    'name' => 'first_name',
                    'id' => 'first_name',
                    'type' => 'text',
                    'value' => $this->form_validation-
>set_value('first_name'),
                );
                $this->data['last_name'] = array(
                    'name' => 'last_name',
                    'id' => 'last_name',
                    'type' => 'text',
                    'value' => $this->form_validation-
>set_value('last_name'),
                );
                $this->data['email'] = array(
                    'name' => 'email',
                    'id' => 'email',
                    'type' => 'text',
                    'value' => $this->form_validation-
>set_value('email'),
                );
                $this->data['password'] = array(
                    'name' => 'password',
                    'id' => 'password',
                    'type' => 'password',
                    'value' => $this->form_validation-
>set_value('password'),
                );
                $this->data['confirm_password'] = array(
                    'name' => 'confirm_password',
                    'id' => 'confirm_password',
                    'type' => 'password',

```

```

        'value' => $this->form_validation-
>set_value('confirm_password'),
        );
        redirect('add_staff',$this->data);
    }
}
}

/* End of file: add_staff_process.php */

/application/controllers/add_staff_success.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_staff_success extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    public function index() {
        $data['pageTitle'] = "Add Staff Success";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('sysadmin'))) {
            redirect('/', 'refresh');
        }
        $this->load->view('header_admin', $data);
        $this->load->view('add_staff_success_view');
        $this->load->view('footer');
    }
}

/* End of file: add_staff_success.php */

/application/controllers/add_user_group.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Add_user_group extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    public function process() {
        $group = $this->input->post('user_group');
        $desc = $this->input->post('user_description');
        $is_admin = FALSE;

        $insert = $this->flexi_auth->insert_group($group, $desc, $is_admin);

        if ($insert) {
            $this->load->view('header_sysadmin', $data);
            $this->load->view('add_user_group_success_view');
            $this->load->view('footer');
        }
        else {
            $this->load->view('header_sysadmin', $data);
            $this->load->view('add_user_group_failure_view');
            $this->load->view('footer');
        }
    }

    public function index() {
        $data['pageTitle'] = "Add User Group";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>is_admin())) {
            redirect('/', 'refresh');
        }

        $this->load->view('header_sysadmin', $data);
        $this->load->view('add_user_group_view');
        $this->load->view('footer');
    }
}
}

```

```

/* End of file: add_user_group.php */

/application/controllers/ajax_search.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Ajax_search extends CI_Controller {
    public function __construct(){
        parent::__construct();
    }

    public function do_phrase_search_staff() {
        if($this->input->get('keyword')) {
            $keyword = trim($this->input->get('keyword'));
            $keyword = $this->db->escape_str($keyword);

            $this->db-
>select('phrasebook_phrase','phrasebook_translation','phrasebook_category');
            $query = $this->db->query("SELECT * FROM phrasebook WHERE
phrasebook_phrase LIKE '%$keyword%' OR phrasebook_translation LIKE '%$keyword%'");

            if ($query->num_rows() != 0) {
                echo "<table>
                    <thead>\n
                        <th>Phrase</th>\n
                        <th>English translation</th>\n
                        <th>Category/Tags</th>\n
                        <th>Edit Phrase</th>\n
                    </thead>";
                foreach ($query->result() as $row) {
                    $edit_link = $row->phrasebook_id;
                    echo "<tr>
                        <td>$row->phrasebook_phrase</td>\n
                        <td>$row->phrasebook_translation</td>\n
                        <td>$row->phrasebook_category</td>\n
                        <td><a
href='edit_phrase?id=$edit_link'>Edit</a></td>\n
                    </tr>";
                }
            }
            else {
                echo 'No results for: "' . $this->input->get('keyword') .
''';
            }
        }
        else {
            echo 'Parameter missing';
        }
    }

    public function do_dictionary_search_staff() {
        if($this->input->get('keyword')) {
            $keyword = trim($this->input->get('keyword'));
            $keyword = $this->db->escape_str($keyword);

            $this->db-
>select('dictionary_term','dictionary_partofspeech','dictionary_meaning','dictionary_usage','dict
ionary_category');
            $query = $this->db->query("SELECT * FROM dictionary WHERE
dictionary_term LIKE '%$keyword%' OR dictionary_meaning LIKE '%$keyword%'");

            if ($query->num_rows() != 0) {
                echo "<table>
                    <thead>\n
                        <th>Term</th>\n
                        <th>Part Of Speech</th>\n
                        <th>Meaning</th>\n
                        <th>Usage</th>\n
                        <th>Category</th>\n
                        <th>Edit Entry</th>\n
                    </thead>";
                foreach ($query->result() as $row) {

```

```

        $edit_link = $row->dictionary_id;
        echo "<tr>
            <td>$row->dictionary_term</td>\n
            <td>$row->dictionary_partofspeech</td>\n
            <td>$row->dictionary_meaning</td>\n
            <td>$row->dictionary_usage</td>\n
            <td>$row->dictionary_category</td>\n
            <td><a
href='edit_dictionary?id=$edit_link'>Edit</a></td>\n
            </tr>";
    }
    }
    else {
        echo 'No results for: "' . $this->input->get('keyword') .
''';
    }
    }
    else {
        echo 'Parameter missing';
    }
}

public function do_phrase_search() {
    if($this->input->get('keyword')) {
        $keyword = trim($this->input->get('keyword'));
        $keyword = $this->db->escape_str($keyword);

        $this->db->
>select('phrasebook_phrase','phrasebook_translation','phrasebook_category');
        $query = $this->db->query("SELECT * FROM phrasebook WHERE
phrasebook_phrase LIKE '%$keyword%' OR phrasebook_translation LIKE '%$keyword%'");

        if ($query->num_rows() != 0) {
            echo "<table>
                <thead>\n
                    <th>Phrase</th>\n
                    <th>English translation</th>\n
                    <th>Category/Tags</th>\n
                </thead>";
            foreach ($query->result() as $row) {
                $edit_link = $row->phrasebook_id;
                echo "<tr>
                    <td>$row->phrasebook_phrase</td>\n
                    <td>$row-
>phrasebook_translation</td>\n
                    <td>$row->phrasebook_category</td>\n
                </tr>";
            }
        }
        else {
            echo 'No results for: "' . $this->input->get('keyword') .
''';
        }
    }
    else {
        echo 'Parameter missing';
    }
}

public function do_dictionary_search() {
    if($this->input->get('keyword')) {
        $keyword = trim($this->input->get('keyword'));
        $keyword = $this->db->escape_str($keyword);

        $this->db->
>select('dictionary_term','dictionary_partofspeech','dictionary_meaning','dictionary_usage','dict
ionary_category');
        $query = $this->db->query("SELECT * FROM dictionary WHERE
dictionary_term LIKE '%$keyword%' OR dictionary_meaning LIKE '%$keyword%'");

```

```

        if ($query->num_rows() != 0) {
            echo "<table>
                <thead>\n
                    <th>Term</th>\n
                    <th>Part Of Speech</th>\n
                    <th>Meaning</th>\n
                    <th>Usage</th>\n
                    <th>Category</th>\n
                </thead>";
            foreach ($query->result() as $row) {
                $edit_link = $row->dictionary_id;
                echo "<tr>
                    <td>$row->dictionary_term</td>\n
                    <td>$row-
>dictionary_partofspeech</td>\n
                    <td>$row->dictionary_meaning</td>\n
                    <td>$row->dictionary_usage</td>\n
                    <td>$row->dictionary_category</td>\n
                </tr>";
            }
        }
        else {
            echo 'No results for: "' . $this->input->get('keyword') .
''';
        }
    }
    else {
        echo 'Parameter missing';
    }
}

public function do_grammar_search() {
    if($this->input->get('keyword')) {
        $keyword = trim($this->input->get('keyword'));
        $keyword = $this->db->escape_str($keyword);

        $query = $this->db->query("SELECT * FROM grammar WHERE grammar_rule
LIKE '%$keyword%'");

        if ($query->num_rows() != 0) {
            echo "<table>
                <thead>\n
                    <th>Rule</th>\n
                    <th>Usage</th>\n
                    <th>Example</th>\n
                    <th>Category</th>\n
                </thead>";
            foreach ($query->result() as $row) {
                $edit_link = $row->grammar_id;
                echo "<tr>
                    <td>$row->grammar_rule</td>\n
                    <td>$row->grammar_usage</td>\n
                    <td>$row->grammar_example</td>\n
                    <td>$row->grammar_category</td>\n
                </tr>";
            }
        }
        else {
            echo 'No results for: "' . $this->input->get('keyword') .
''';
        }
    }
    else {
        echo 'Parameter missing';
    }
}

}

/* End of file: ajax_search.php */

```



```

/application/controllers/approve_links.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
    class Approve_links extends CI_Controller {
        function __construct() {
            parent::__construct();
            $this->load->model('link_model');
        }

        function index() {
            $data['pageTitle'] = "Approve Links";
            if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('contentadmin'))) {
                redirect('/');
            }
            else {
                $this->load->view('header_admin', $data);
                $this->load->view('approve_links_view');
                $this->load->view('footer');
            }
        }
    }

}

/* End of file: approve_links.php */

/application/controllers/approve_links_process.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
    class Approve_links_process extends CI_Controller {
        function __construct() {
            parent::__construct();
        }

        function index() {
            if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('contentadmin'))) {
                redirect('/');
            }
            else {
                $data['pageTitle'] = "Approval of Links Processing...";
                $this->load->view('header_admin', $data);
                $this->load->model('Checked_model','check');
                $titles_to_check = $this->input->post('title');
                $title_count = count($titles_to_check);
                $j = 0;
                if ($title_count > 0) {
                    $this->db->like('link_title',$titles_to_check[0],'none');
                    for ($j = 1; $j < $title_count; $j++) {
                        $this->db-
>or_like('link_title',$titles_to_check[$j],'none');
                    }
                }
                else {
                    $this->db->like('link_title',$titles_to_check,'none');
                }
                $query = $this->db->get('links');
                if ($query->num_rows() > 0) {
                    $selected_data = array('is_selected' => '1');
                    $select_alter_query = $this->db-
>update('links',$selected_data);
                    foreach ($query->result() as $row)
                    {
                        if($this->check->is_checked('approved','1'))

                        $link = $row->link_title;
                        $data = array('is_approved' => '1');
                        $this->db->where('link_title',$link);
                        $alter_query = $this->db-
>update('links',$data);
                        $alter_result = $alter_query;
                        if($alter_result) {

```

```

$this->db->affected_rows();

        $data["links_approved"] =
        $data["links_rejected"] = 0;
    }
    else {
        if ($this->check-
            $reject_link = $row-
            $reject_data =
            $this->db-
            $reject_alter_query =
            $reject_alter_result =
        }
        $this->load-
        $this->load->view('footer');
    }
}
if ($this->check->is_checked('approved', '-
    $link = $row->link_title;
    $data = array('is_approved' => '-1');
    $this->db->where('link_title', $link);
    $alter_query = $this->db-
    $alter_result = $alter_query;
    if($alter_result) {
        $data["links_approved"] = 0;
        $data["links_rejected"] =
        $this->load-
        $this->load->view('footer');
    }
}
else {
    if ($this->check-
        $reject_link = $row-
        $reject_data =
        $this->db-
        $reject_alter_query = $this-
        $reject_alter_result =
        $this->load-
        $this->load->view('footer');
    }
    $this->load-
    $this->load->view('footer');
}
}
}
}
else {
    $this->load->view('no_links_selected_view');
    $this->load->view('footer');
}
$deselect = array('is_selected' => '0');

```

```

        $this->db->update('links',$deselect);
    }
}

/* End of file: approve_links_process.php */

/application/controllers/bank_maintenance.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Bank_maintenance extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        $data['pageTitle'] = "Question Bank Maintenance";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('bank_maintenance_view');
            $this->load->view('footer');
        }
    }
}

/* End of file: bank_maintenance.php */

/application/controllers/change_detail.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Change_detail extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->model('lesson_entries');
    }

    function index() {
        $group = array('staff','contentadmin');
        if (!$this->flexi_auth->in_group($group)) {
            redirect('/', 'refresh');
        }
        else {
            $data['pageTitle'] = "View Change Request";
            if ($this->flexi_auth->in_group('contentadmin')) {
                $this->load->view('header_admin',$data);
            }
            else {
                $this->load->view('header_staff', $data);
            }
            $this->load->view('change_detail_view');
            $this->load->view('footer');
        }
    }
}

/* End of file: change_detail.php */

/application/controllers/change_password.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Change_password extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    public function index() {
        $data['pageTitle'] = "Password Change Form";
        if ($this->flexi_auth->is_admin()){
            $this->load->view('header_sysadmin',$data);
        }
    }
}

```

```

    }
    else if ($this->flexi_auth->in_group('contentadmin')) {
        $this->load->view('header_admin', $data);
    }
    else if ($this->flexi_auth->in_group('contributors')) {
        $this->load->view('header_contributor', $data);
    }
    else if ($this->flexi_auth->in_group('staff')) {
        $this->load->view('header_staff', $data);
    }
    else {
        $this->load->view('header_user', $data);
    }
    $this->load->view('change_password_view');
    $this->load->view('footer');
}
}

/* End of file: change_password.php */

/application/controllers/change_password_process.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Change_password_process extends CI_Controller{
    function __construct(){
        parent::__construct();
    }

    public function index(){
        $data['pageTitle'] = "Change Password Process";
        if ($this->flexi_auth->is_admin()){
            $this->load->view('header_sysadmin', $data);
        }
        else if ($this->flexi_auth->in_group('contentadmin')) {
            $this->load->view('header_admin', $data);
        }
        else if ($this->flexi_auth->in_group('contributors')) {
            $this->load->view('header_contributor', $data);
        }
        else if ($this->flexi_auth->in_group('staff')) {
            $this->load->view('header_staff', $data);
        }
        else if ($this->flexi_auth->in_group('users')) {
            $this->load->view('header_user', $data);
        }
        else {
            redirect('/');
        }
        $current_password = $this->input->post('current_password');
        $new_password = $this->input->post('new_password');

        $user_identity = $this->flexi_auth->get_user_identity();
        $change_pw = $this->flexi_auth-
>change_password($user_identity, $current_password, $new_password);

        if ($change_pw) {
            $data['pageTitle'] = "Change Password Successful";
            if ($this->flexi_auth->is_admin()){
                $this->load->view('header_sysadmin', $data);
            }
            else if ($this->flexi_auth->in_group('contentadmin')) {
                $this->load->view('header_admin', $data);
            }
            else if ($this->flexi_auth->in_group('contributors')) {
                $this->load->view('header_contributor', $data);
            }
            else if ($this->flexi_auth->in_group('staff')) {
                $this->load->view('header_staff', $data);
            }
            else if ($this->flexi_auth->in_group('users')) {
                $this->load->view('header_user', $data);
            }
        }
    }
}

```

```

        $this->load->view('change_password_success_view');
        $this->load->view('footer');
    }
    else {
        $data['pageTitle'] = "Change Password Unsuccessful";
        if ($this->flexi_auth->is_admin()){
            $this->load->view('header_sysadmin',$data);
        }
        else if ($this->flexi_auth->in_group('contentadmin')) {
            $this->load->view('header_admin',$data);
        }
        else if ($this->flexi_auth->in_group('contributors')) {
            $this->load->view('header_contributor',$data);
        }
        else if ($this->flexi_auth->in_group('staff')) {
            $this->load->view('header_staff',$data);
        }
        else if ($this->flexi_auth->in_group('users')) {
            $this->load->view('header_user',$data);
        }
        $this->load->view('change_password_failure_view');
        $this->load->view('footer');
    }
}
}

```

/* End of file: change_password_process.php */

/application/controllers/changes.php

```

<?php
class Changes extends CI_Controller
{
    public function __construct() {
        parent::__construct();
        $this->load->helper("url");
        $this->load->model("change_entries");
        $this->load->library("pagination");
    }

    public function view_changes() {
        $config["base_url"] = site_url("changes/view_changes");
        $config["total_rows"] = $this->change_entries->record_count();
        $config["per_page"] = 20;
        $config["uri_segment"] = 3;
        $choice = $config["total_rows"] / $config["per_page"];
        $config["num_links"] = round($choice);

        $this->pagination->initialize($config);

        $page = ($this->uri->segment(3))? $this->uri->segment(3) : 0;
        $data["pageTitle"] = "View Changes";
        $data["results"] = $this->change_entries->fetch_entries($config["per_page"], $page);
        $data["links"] = $this->pagination->create_links();

        if ($this->flexi_auth->in_group('contentadmin')) {
            $this->load->view('header_admin',$data);
        }
        else {
            $this->load->view('header_staff', $data);
        }
        $this->load->view("changes_view", $data);
        $this->load->view("footer");
    }

    public function index() {
        $group = array('contentadmin','staff');
        if ($this->flexi_auth->in_group($group)) {
            redirect($config["base_url"] . "changes/view_changes");
        }
        else {
            redirect('/');
        }
    }
}

```

```

    }
}

/* End of file: changes.php */

/application/controllers/check_link_status.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Check_link_status extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->model('link_model');
    }

    function index() {
        $user_records = $this->link_model->record_count_by_user();
        $data['results'] = $this->link_model->fetch_all_user_entries();
        $data['pageTitle'] = "Check Link Status";

        $group = array('contributors','staff');
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group($group))) {
            redirect('/', 'refresh');
        }
        else if ($this->flexi_auth->in_group('contributors')) {
            $this->load->view('header_contributor', $data);
            $this->load->view('check_link_status_view', $data);
            $this->load->view('footer');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('check_link_status_view', $data);
            $this->load->view('footer');
        }
    }
}

```

```
/* End of file: check_link_status.php */
```

```

/application/controllers/confirm_change_deletion.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Confirm_change_deletion extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function delete() {
        $id = $this->input->get('id');
        $this->db->where('change_id', $id);
        $query = $this->db->delete('changes');
        if($query) {
            $data["pageTitle"] = "Entry Deleted";
            if($this->flexi_auth->in_group('contentadmin')) {
                $this->load->view('header_admin', $data);
            }
            else if ($this->flexi_auth->in_group('staff')) {
                $this->load->view('header_staff', $data);
            }
            $this->load->view('confirmed_deletion_entry_view');
        }
        else {
            $data['pageTitle'] = "Delete Error";
            if($this->flexi_auth->in_group('contentadmin')) {
                $this->load->view('header_admin', $data);
            }
            else if ($this->flexi_auth->in_group('staff')) {
                $this->load->view('header_staff', $data);
            }
            $this->load->view('delete_error_view');
        }
        $this->load->view('footer');
    }
}

```

```

    }

    function index() {
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth->in_group(array('contentadmin','staff')))) {
            redirect('/', 'refresh');
        }
        else {
            $data["pageTitle"] = "Confirm Deletion?";

            if($this->flexi_auth->in_group('contentadmin')) {
                $this->load->view('header_admin', $data);
            }
            else if ($this->flexi_auth->in_group('staff')) {
                $this->load->view('header_staff', $data);
            }
            $id = $this->input->get('id');
            if ($id) {
                $data['id'] = $id;
                $this->load->view('confirm_change_deletion_view', $data);
                $this->load->view('footer');
            }
            else {
                $this->load->view('parameter_missing_view');
                $this->load->view('footer');
            }
        }
    }
}

}

/* End of file: confirm_change_deletion.php */

/application/controllers/confirm_dictionary_deletion.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Confirm_dictionary_deletion extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function delete() {
        $id = $this->input->get('id');
        $this->db->where('dictionary_id', $id);
        $query = $this->db->delete('dictionary');
        if($query) {
            $data["pageTitle"] = "Entry Deleted";
            $this->load->view('header_staff', $data);
            $this->load->view('confirmed_deletion_guide_entry_view');
        }
        else {
            $data['pageTitle'] = "Delete Error";
            $this->load->view('header_staff', $data);
            $this->load->view('delete_error_view');
        }
        $this->load->view('footer');
    }

    function index() {
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth->in_group('staff')))) {
            redirect('/', 'refresh');
        }
        else {
            $data["pageTitle"] = "Confirm Deletion?";
            $this->load->view('header_staff', $data);
            $id = $this->input->get('id');
            if ($id) {
                $data['id'] = $id;

```

```

        $this->load-
>view('confirm_dictionary_deletion_view',$data);
        $this->load->view('footer');
    }
    else {
        $this->load->view('parameter_missing_view');
        $this->load->view('footer');
    }
}
}

/* End of file: confirm_dictionary_deletion.php */

/application/controllers/confirm_grammar_deletion.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Confirm_grammar_deletion extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function delete() {
        $id = $this->input->get('id');
        $this->db->where('grammar_id',$id);
        $query = $this->db->delete('grammar');
        if($query) {
            $data["pageTitle"] = "Entry Deleted";
            $this->load->view('header_staff',$data);
            $this->load->view('confirmed_deletion_guide_entry_view');
        }
        else {
            $data['pageTitle'] = "Delete Error";
            $this->load->view('header_staff',$data);
            $this->load->view('delete_error_view');
        }
        $this->load->view('footer');
    }

    function index() {
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $data["pageTitle"] = "Confirm Deletion?";
            $this->load->view('header_staff',$data);
            $id = $this->input->get('id');
            if ($id) {
                $data['id'] = $id;
                $this->load->view('confirm_grammar_deletion_view',$data);
                $this->load->view('footer');
            }
            else {
                $this->load->view('parameter_missing_view');
                $this->load->view('footer');
            }
        }
    }
}

/* End of file: confirm_grammar_deletion.php */

/application/controllers/confirm_instruction_deletion.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Confirm_instruction_deletion extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function delete() {

```



```

$id = $this->input->get('id');
$this->db->where('instruction_no',$id);
$query = $this->db->delete('bank_instructions');
if($query) {
    $data["pageTitle"] = "Entry Deleted";
    $this->load->view('header_staff',$data);
    $this->load->view('confirmed_deletion_instruction_view');
}
else {
    $data['pageTitle'] = "Delete Error";
    $this->load->view('header_staff',$data);
    $this->load->view('delete_error_view');
}
$this->load->view('footer');
}

function index() {
    if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
        redirect('/', 'refresh');
    }
    else {
        $data["pageTitle"] = "Confirm Deletion?";
        $this->load->view('header_staff',$data);
        $id = $this->input->get('id');
        if ($id) {
            $data['id'] = $id;
            $this->load-
>view('confirm_instruction_deletion_view',$data);
            $this->load->view('footer');
        }
        else {
            $this->load->view('parameter_missing_view');
            $this->load->view('footer');
        }
    }
}
}
}

```

/* End of file: confirm_instruction_deletion.php */

/application/controllers/confirm_link_deletion.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Confirm_link_deletion extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function delete() {
        $id = $this->input->get('id');
        $this->db->where('link_id',$id);
        $query = $this->db->delete('links');
        if($query) {
            $data["pageTitle"] = "Entry Deleted";
            if($this->flexi_auth->in_group('contributors')) {
                $this->load->view('header_contributor', $data);
            }
            else if ($this->flexi_auth->in_group('staff')) {
                $this->load->view('header_staff', $data);
            }
            $this->load->view('confirmed_deletion_guide_entry_view');
        }
        else {
            $data['pageTitle'] = "Delete Error";
            if($this->flexi_auth->in_group('contributors')) {
                $this->load->view('header_contributor', $data);
            }
            else if ($this->flexi_auth->in_group('staff')) {
                $this->load->view('header_staff', $data);
            }
        }
    }
}

```

```

        $this->load->view('delete_error_view');
    }
    $this->load->view('footer');
}

function index() {
    if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group(array('contributors','staff')))) {
        redirect('/', 'refresh');
    }
    else {
        $data["pageTitle"] = "Confirm Deletion?";

        if($this->flexi_auth->in_group('contributors')) {
            $this->load->view('header_contributor', $data);
        }
        else if ($this->flexi_auth->in_group('staff')) {
            $this->load->view('header_staff', $data);
        }
        $id = $this->input->get('id');
        if ($id) {
            $data['id'] = $id;
            $this->load->view('confirm_link_deletion_view', $data);
            $this->load->view('footer');
        }
        else {
            $this->load->view('parameter_missing_view');
            $this->load->view('footer');
        }
    }
}

}

}

}

/* End of file: confirm_link_deletion.php */

```

/application/controllers/confirm_phrase_deletion.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Confirm_phrase_deletion extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function delete() {
        $id = $this->input->get('id');
        $this->db->where('phrasebook_id',$id);
        $query = $this->db->delete('phrasebook');
        if($query) {
            $data["pageTitle"] = "Entry Deleted";
            $this->load->view('header_staff', $data);
            $this->load->view('confirmed_deletion_guide_entry_view');
        }
        else {
            $data['pageTitle'] = "Delete Error";
            $this->load->view('header_staff', $data);
            $this->load->view('delete_error_view');
        }
        $this->load->view('footer');
    }

    function index() {
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff')))) {
            redirect('/', 'refresh');
        }
        else {
            $data["pageTitle"] = "Confirm Deletion?";
            $this->load->view('header_staff', $data);
            $id = $this->input->get('id');

```

```

        if ($id) {
            $data['id'] = $id;
            $this->load->view('confirm_phrase_deletion_view',$data);
            $this->load->view('footer');
        }
        else {
            $this->load->view('parameter_missing_view');
            $this->load->view('footer');
        }
    }
}

}

}

/* End of file: confirm_phrase_deletion.php */

/application/controllers/confirm_question_deletion.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Confirm_question_deletion extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function delete() {
        $id = $this->input->get('id');
        $this->db->where('question_no',$id);
        $query = $this->db->delete('question_bank');
        if($query) {
            $data["pageTitle"] = "Entry Deleted";
            $this->load->view('header_staff',$data);
            $this->load->view('confirmed_deletion_question_view');
        }
        else {
            $data['pageTitle'] = "Delete Error";
            $this->load->view('header_staff',$data);
            $this->load->view('delete_error_quiz_view');
        }
        $this->load->view('footer');
    }

    function index() {
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $data["pageTitle"] = "Confirm Deletion?";
            $this->load->view('header_staff',$data);
            $id = $this->input->get('id');
            if ($id) {
                $data['id'] = $id;
                $this->load->view('confirm_question_deletion_view',$data);
                $this->load->view('footer');
            }
            else {
                $this->load->view('parameter_missing_view');
                $this->load->view('footer');
            }
        }
    }
}

}

/* End of file: confirm_question_deletion.php */

/application/controllers/contributor_signup.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Contributor_signup extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    public function index() {

```

```

        $data['pageTitle'] = "Contributor Signup Form";
        $data['message'] = "";
        $this->load->view('header', $data);
        $this->load->view('contributor_signup_view',$data);
        $this->load->view('footer');
    }
}

/* End of file: contributor_signup.php */

/application/controllers/delete_guides.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Delete_guides extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        $data['pageTitle'] = "Delete Guide Contents";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('delete_guides_view');
            $this->load->view('footer');
        }
    }
}

/* End of file: delete_guides.php */

/application/controllers/delete_lesson.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Delete_lesson extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->model('lesson_entries');
        $this->load->library('pagination');
    }

    function index() {
        $segment = $this->uri->total_segments();
        $config = array();
        $config["base_url"] = site_url("delete_lesson/index");
        $config["total_rows"] = $this->lesson_entries->record_count();
        $config["per_page"] = 10;
        $config["uri_segment"] = $segment;
        $choice = $config["total_rows"] / $config["per_page"];
        $config["num_links"] = round($choice);

        $this->pagination->initialize($config);

        $page = ($this->uri->segment($segment)) ? $this->uri->segment($segment) :
0;

        $data['pageTitle'] = "Delete Lesson Entries";
        $data["results"] = $this->lesson_entries-
>fetch_entries($config["per_page"],$page);
        $data["links"] = $this->pagination->create_links();

        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('delete_lesson_view');
            $this->load->view('footer');
        }
    }
}

```

```

        }
    }
}

/* End of file: delete_lesson.php */

/application/controllers/delete_user.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Delete_user extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function confirm() {
        $id = $this->input->get('id');
        $query = $this->flexi_auth->delete_user('id');
        if($query) {
            $data["pageTitle"] = "Entry Deleted";
            $this->load->view('header_sysadmin',$data);
            $this->load->view('user_deleted_view');
        }
        else {
            $data['pageTitle'] = "Delete Error";
            $this->load->view('header_sysadmin',$data);
            $this->load->view('delete_error_auth_view');
        }
        $this->load->view('footer');
    }

    function index() {
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>is_admin())) {
            redirect('/', 'refresh');
        }
        else {
            $data["pageTitle"] = "Confirm Deletion?";
            $this->load->view('header_sysadmin',$data);
            $id = $this->input->get('id');
            if ($id) {
                $data['id'] = $id;
                $this->load->view('delete_user_view',$data);
                $this->load->view('footer');
            }
            else {
                $this->load->view('parameter_missing_view');
                $this->load->view('footer');
            }
        }
    }
}

/* End of file: delete_user.php */

```

```

/application/controllers/delete_user_group.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Delete_user_group extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function confirm() {
        $id = $this->input->get('id');
        $query = $this->flexi_auth->delete_group($id);
        if($query) {
            $data["pageTitle"] = "Entry Deleted";
            $this->load->view('header_sysadmin',$data);
            $this->load->view('user_deleted_view');
        }
    }
}

```

```

    }
    else {
        $data['pageTitle'] = "Delete Error";
        $this->load->view('header_sysadmin',$data);
        $this->load->view('delete_error_auth_view');
    }
    $this->load->view('footer');
}

function index() {
    if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>is_admin())) {
        redirect('/', 'refresh');
    }
    else {
        $data["pageTitle"] = "Confirm Deletion?";
        $this->load->view('header_sysadmin',$data);
        $id = $this->input->get('id');
        if ($id) {
            $data['id'] = $id;
            $this->load->view('delete_user_group_view',$data);
            $this->load->view('footer');
        }
        else {
            $this->load->view('parameter_missing_view');
            $this->load->view('footer');
        }
    }
}
}

}

/* End of file: delete_user_group.php */

/application/controllers/dictionary.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Dictionary extends CI_Controller {
    public function __construct() {
        parent::__construct();
        $this->load->model('dictionary_entries');
        $this->load->library('pagination');
    }

    public function view_entries($start_letter) {
        if ($start_letter) {
            $segment = $this->uri->total_segments();
            $config = array();
            $config["base_url"] = site_url("dictionary/view_entries/" .
$start_letter . "/" );
            $config["total_rows"] = $this->dictionary_entries-
>letter_record_count($start_letter);
            $config["per_page"] = 10;
            $config["uri_segment"] = $segment;
            $choice = $config["total_rows"] / $config["per_page"];
            $config["num_links"] = round($choice);

            $this->pagination->initialize($config);

            $page = ($this->uri->segment($segment)) ? $this->uri-
>segment($segment) : 0;

            $data['pageTitle'] = "View Dictionary Entries by Letter: " .
$start_letter;
            $data["results"] = $this->dictionary_entries-
>fetch_entries_by_letter($config["per_page"], $page, $start_letter);
            $data["links"] = $this->pagination->create_links();

            if ($this->flexi_auth->is_logged_in()) {
                $this->load->view('header_user_tables',$data);
            }
            else if ($this->flexi_auth->in_group('staff')) {

```

```

        $this->load->view('header_staff_tables', $data);
    }
    else if ($this->flexi_auth->in_group('contributors')) {
        $this->load->view('header_contributor_tables', $data);
    }
    else {
        $this->load->view('header_tables', $data);
    }
    $this->load->view('dictionary_view', $data);
    $this->load->view('footer');
}
else {
    $data['pageTitle'] = "Parameter Missing!";
    if ($this->flexi_auth->is_logged_in()) {
        $this->load->view('header_user_tables', $data);
    }
    else {
        $this->load->view('header_tables', $data);
    }
    $this->load->view('parameter_missing_view');
    $this->load->view('footer');
}
}

public function index() {
    $data['pageTitle'] = "View Dictionary Entries";
    if ($this->flexi_auth->is_logged_in()) {
        $this->load->view('header_user_tables', $data);
    }
    else if ($this->flexi_auth->in_group('staff')) {
        $this->load->view('header_staff_tables', $data);
    }
    else {
        $this->load->view('header_tables', $data);
    }
    $this->load->view('dictionary_index_view');
    $this->load->view('footer');
}
}

/* End of file: dictionary.php */

/application/controllers/differences.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Differences extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    public function index() {
        $data['pageTitle'] = "Differences Between User and Contributor Accounts";
        $this->load->view('header', $data);
        $this->load->view('differences_view');
        $this->load->view('footer');
    }
}

/* End of file: differences.php */

/application/controllers/edit_dictionary.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_dictionary extends CI_Controller {
    public function __construct(){
        parent::__construct();
    }

    public function index() {
        $data['pageTitle'] = "Edit Dictionary Entry";

```

```

        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('edit_dictionary_view');
            $this->load->view('footer');
        }
    }
}

/* End of file: edit_dictionary.php */

/application/controllers/edit_dictionary_process.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_dictionary_process extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->library('upload');
    }

    function index() {
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $id = $this->input->post('dictionary_id');
            $term = $this->input->post('term');
            $partofspeech = $this->input->post('partofspeech');
            $pronunciation = $this->input->post('pronunciation');
            $meaning = $this->input->post('meaning');
            $usage = $this->input->post('usage');

            if (!empty($_FILES['userfile']['name'])) {
                $config['upload_path'] = './images/';
                $config['allowed_types'] = 'gif|jpg|png';
                $config['max_size'] = '2048';

                $this->upload->initialize($config);

                if ($this->upload->do_upload('userfile')) {
                    $image_data_array = $this->upload->data();
                    $image_link =
$image_data_array['full_path'];
                }
                else {
                    echo $this->upload->display_errors();
                }
            }

            if (!empty($_FILES['userfile1']['name'])) {
                $config['upload_path'] = './sounds/';
                $config['allowed_types'] = 'wav';
                $config['max_size'] = '2048';

                $this->upload->initialize($config);

                if ($this->upload->do_upload('userfile1')) {
                    $audio_data_array = $this->upload->data();
                    $audio_link =
$audio_data_array['full_path'];
                }
                else {
                    echo $this->upload->display_errors();
                }
            }

            $category = $this->input->post('category');

            if ($audio_link && $image_link) {

```



```

        $data_to_update = array(
'dictionary_term' => $term,
'dictionary_partofspeech' => $partofspeech,
'dictionary_pronunciation' => $pronunciation,
'dictionary_meaning' => $meaning,
'dictionary_usage' => $usage,
'dictionary_image_link' => $image_link,
'dictionary_audio_link' => $audio_link,
'dictionary_category' => $category
);
    }
    else if ($audio_link) {
        $data_to_update = array(
'dictionary_term' => $term,
'dictionary_partofspeech' => $partofspeech,
'dictionary_pronunciation' => $pronunciation,
'dictionary_meaning' => $meaning,
'dictionary_usage' => $usage,
'dictionary_audio_link' => $audio_link,
'dictionary_category' => $category
);
    }
    else if ($image_link) {
        $data_to_update = array(
'dictionary_term' => $term,
'dictionary_partofspeech' => $partofspeech,
'dictionary_pronunciation' => $pronunciation,
'dictionary_meaning' => $meaning,
'dictionary_usage' => $usage,
'dictionary_image_link' => $image_link,
'dictionary_category' => $category
);
    }
    else {
        $data_to_update = array(
'dictionary_term' => $term,
'dictionary_partofspeech' => $partofspeech,
'dictionary_pronunciation' => $pronunciation,
'dictionary_meaning' => $meaning,
'dictionary_usage' => $usage,

```

```

'dictionary_category' => $category
);
}

$this->db->where('dictionary_id',$id);
$query = $this->db->update('dictionary',$data_to_update);
if($query) {
    redirect('view_dictionary_entry?id=' . $id);
}
else {
    $data["pageTitle"] = "Editing Unsuccessful";
    $this->load->view('header_staff',$data);
    $this->load->view('edit_unsuccessful_view');
}
$this->load->view('footer');
}
}
}

/* End of file: edit_dictionary_process.php */

/application/controllers/edit_grammar.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_grammar extends CI_Controller {
    public function __construct(){
        parent::__construct();
        $this->load->model('grammar_entries');
        $this->load->library('pagination');
    }

    public function index() {
        $segment = $this->uri->total_segments();
        $config = array();
        $config["base_url"] = site_url("edit_grammar/index");
        $config["total_rows"] = $this->grammar_entries->record_count();
        $config["per_page"] = 10;
        $config["uri_segment"] = $segment;
        $choice = $config["total_rows"] / $config["per_page"];
        $config["num_links"] = round($choice);

        $this->pagination->initialize($config);

        $page = ($this->uri->segment($segment)) ? $this->uri->segment($segment) :
0;

        $data['pageTitle'] = "View Grammar Rule Entries For Editing";
        $data["results"] = $this->grammar_entries-
>fetch_entries($config["per_page"],$page);
        $data["links"] = $this->pagination->create_links();

        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff')) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('edit_grammar_view',$data);
            $this->load->view('footer');
        }
    }
}

/* End of file: edit_grammar.php */

/application/controllers/edit_grammar_rule.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_grammar_rule extends CI_Controller {
    public function __construct(){
        parent::__construct();

```

```

    }

    public function index() {
        $data['pageTitle'] = "Edit Grammar Rule";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('edit_grammar_rule_view');
            $this->load->view('footer');
        }
    }
}

```

/* End of file: edit_grammar_rule.php */

/application/controllers/edit_grammar_rule_process.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_grammar_rule_process extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $id = $this->input->post('grammar_id');
            $rule = $this->input->post('rule');
            $usage = $this->input->post('usage');
            $example = $this->input->post('example');
            $category = $this->input->post('category');

            $data_to_update = array(
                'grammar_rule' => $rule,
                'grammar_usage' => $usage,
                'grammar_example' => $example,
                'grammar_category' => $category
            );

            $this->db->where('grammar_id', $id);
            $query = $this->db->update('grammar', $data_to_update);

            if($query) {
                redirect();
            }
            else {
                $data["pageTitle"] = "Editing Unsuccessful";
                $this->load->view('header_staff', $data);
                $this->load->view('edit_unsuccessful_view');
            }
            $this->load->view('footer');
        }
    }
}

```

/* End of file: edit_grammar_rule_process.php */

/application/controllers/edit_guides.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_guides extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        $data['pageTitle'] = "Edit Guide Contents";
    }
}

```

```

        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('edit_guides_view');
            $this->load->view('footer');
        }
    }
}

}

}

/* End of file: edit_guides.php */

/application/controllers/edit_instruction.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_instruction extends CI_Controller {
    public function __construct(){
        parent::__construct();
        $this->load->model('instruction_entries');
        $this->load->library('pagination');
    }

    public function process() {
        $instruction_id = $this->input->post('instruction_id');
        $text = $this->input->post('text');
        $lesson_id = $this->input->post('lesson_id');

        $data_to_update = array('instruction_text' => $text, 'lesson_id' =>
$lesson_id);

        $this->db->where('instruction_id',$instruction_id);
        $query = $this->db->update('bank_instructions',$data_to_update);
        if ($query) {
            redirect('view_instruction?id=' . $instruction_id);
        }
        else {
            $data['pageTitle'] = "Query Failed";
            $this->load->view('header_staff', $data);
            $this->load->view('query_failure_view');
            $this->load->view('footer');
        }
    }

    public function index() {
        $segment = $this->uri->total_segments();
        $config = array();
        $config["base_url"] = site_url("edit_instruction/index");
        $config["total_rows"] = $this->instruction_entries->record_count();
        $config["per_page"] = 10;
        $config["uri_segment"] = $segment;
        $choice = $config["total_rows"] / $config["per_page"];
        $config["num_links"] = round($choice);

        $this->pagination->initialize($config);

        $page = ($this->uri->segment($segment)) ? $this->uri->segment($segment) :
0;

        $data['pageTitle'] = "View Instruction Entries For Editing";
        $data["results"] = $this->instruction_entries-
>fetch_entries($config["per_page"], $page);
        $data["links"] = $this->pagination->create_links();

        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);

```

```

        $this->load->view('edit_instruction_view',$data);
        $this->load->view('footer');
    }
}

}

/* End of file: edit_instruction.php */

/application/controllers/edit_instruction_entry.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_instruction_entry extends CI_Controller {
    public function __construct(){
        parent::__construct();
    }

    public function index() {
        $data['pageTitle'] = "Edit Instruction Entry";

        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff')) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff',$data);
            $this->load->view('edit_instruction_entry_view');
            $this->load->view('footer');
        }
    }
}

/* End of file: edit_instruction_entry.php */

/application/controllers/edit_lesson.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_lesson extends CI_Controller {
    public function __construct(){
        parent::__construct();
        $this->load->model('lesson_entries');
        $this->load->library('pagination');
    }

    public function index() {
        $segment = $this->uri->total_segments();
        $config = array();
        $config["base_url"] = site_url("edit_lesson/index");
        $config["total_rows"] = $this->lesson_entries->record_count();
        $config["per_page"] = 10;
        $config["uri_segment"] = $segment;
        $choice = $config["total_rows"] / $config["per_page"];
        $config["num_links"] = round($choice);

        $this->pagination->initialize($config);

        $page = ($this->uri->segment($segment)) ? $this->uri->segment($segment) :
0;

        $data['pageTitle'] = "View Lesson Entries For Editing";
        $data["results"] = $this->lesson_entries-
>fetch_entries($config["per_page"],$page);
        $data["links"] = $this->pagination->create_links();

        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff')) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('edit_lesson_view',$data);
            $this->load->view('footer');
        }
    }
}

```

```

    }

}

/* End of file: edit_lesson.php */

/application/controllers/edit_lesson_process.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_lesson_process extends CI_Controller {
    public function __construct(){
        parent::__construct();
    }

    public function index() {
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth->in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $config['upload_path'] = "./lessons/";
            $config['allowed_types'] = "pdf";
            $config['max_size'] = '2048';

            $this->load->library('upload', $config);

            $title = $this->input->post('title');
            $description = $this->input->post('description');
            if (!$this->upload->do_upload()) {
                $data['pageTitle'] = "Upload Error";
                $error = array('error' => $this->upload->display_errors());
                $this->load->view('header_staff', $data);
                $this->load->view('add_lesson_view', $error);
                $this->load->view('footer');
            }
            else {
                $upload_array = $this->upload->data();
                $content = $upload_array['full_path'];

                $data_to_update = array(
                    'lesson_title' => $title,
                    'lesson_description' => $description,
                    'lesson_content' => $content
                );

                $this->db->where('lesson_id', $lesson_id);
                $query = $this->db->update('lessons', $data_to_update);
                if($query) {
                    redirect('lesson_entry?id=' . $lesson_id);
                }
                else {
                    $data['pageTitle'] = "Lesson Editing Unsuccessful";
                    $this->load->view('header_staff', $data);
                    $this->load->view('edit_lesson_failure_view');
                    $this->load->view('footer');
                }
            }
        }
    }
}

/* End of file: edit_lesson_process.php */

```

```

/application/controllers/edit_lesson_proper.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_lesson_proper extends CI_Controller {
    public function __construct(){
        parent::__construct();
    }

    public function index() {
        $data['pageTitle'] = "Edit Lesson Entry";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff')) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('edit_lesson_proper_view');
            $this->load->view('footer');
        }
    }
}

/* End of file: edit_lesson_proper.php */

/application/controllers/edit_link.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_link extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->model('link_model');
        $this->load->library('pagination');
    }

    function index() {
        $segment = $this->uri->total_segments();
        $config = array();
        $config["base_url"] = site_url("edit_link/index");
        $config["total_rows"] = $this->link_model->record_count_by_user();
        $config["per_page"] = 10;
        $config["uri_segment"] = $segment;
        $choice = $config["total_rows"] / $config["per_page"];
        $config["num_links"] = round($choice);

        $this->pagination->initialize($config);

        $page = ($this->uri->segment($segment)) ? $this->uri->segment($segment) :
0;

        $data['pageTitle'] = "Edit Links";
        $data["results"] = $this->link_model-
>fetch_user_entries($config["per_page"], $page);
        $data["links"] = $this->pagination->create_links();

        $group = array('contributors', 'staff');
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group($group)) {
            redirect('/', 'refresh');
        }
        else {
            if($this->flexi_auth->in_group('contributors')) {
                $this->load->view('header_contributor', $data);
            }
            else if ($this->flexi_auth->in_group('staff')) {
                $this->load->view('header_staff', $data);
            }
            $this->load->view('edit_link_view', $data);
            $this->load->view('footer');
        }
    }
}
}

```

```

/* End of file: edit_link.php */

/application/controllers/edit_link_entry.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_link_entry extends CI_Controller {
    public function __construct(){
        parent::__construct();
    }

    public function index() {
        $data['pageTitle'] = "Edit Link Entry";

        $group = array('contributors','staff');
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group($group))) {
            redirect('/', 'refresh');
        }
        else {
            if ($this->flexi_auth->in_group('contributors')) {
                $this->load->view('header_contributor',$data);
                $this->load->view('edit_link_entry_view');
                $this->load->view('footer');
            }
            else if ($this->flexi_auth->in_group('staff')) {
                $this->load->view('header_staff',$data);
                $this->load->view('edit_link_entry_view');
                $this->load->view('footer');
            }
        }
    }
}

/* End of file: edit_link_entry.php */

/application/controllers/edit_link_process.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_link_process extends CI_Controller {
    public function __construct(){
        parent::__construct();
    }

    public function index() {
        $group = array('contributors','staff');
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group($group))) {
            redirect('/', 'refresh');
        }
        else {
            $link_id = $this->input->post('link_id');
            $title = $this->input->post('title');
            $description = $this->input->post('description');
            $url = $this->input->post('url');

            $data_to_update = array(
                'link_title' => $title,
                'link_description' => $description,
                'link_url' => $url
            );

            $this->db->where('link_id',$link_id);
            $query = $this->db->update('links',$data_to_update);
            if($query) {
                redirect('view_link?id=' . $link_id);
            }
            else {
                $data['pageTitle'] = "Link Editing Unsuccessful";
                if($this->flexi_auth->in_group('contributors')) {
                    $this->load->view('header_contributor', $data);
                }
                else if ($this->flexi_auth->in_group('staff')) {

```



```

        $this->load->view('header_staff', $data);
    }
    $this->load->view('edit_link_failure_view');
    $this->load->view('footer');
}
}
}
}

/* End of file: edit_link_process.php */

/application/controllers/edit_phrase.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_phrase extends CI_Controller {
    public function __construct(){
        parent::__construct();
    }

    public function index() {
        $data['pageTitle'] = "Edit Phrase";
        $error = "";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff')) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('edit_phrase_view', $data);
            $this->load->view('footer');
        }
    }
}

/* End of file: edit_phrase.php */

/application/controllers/edit_phrase_process.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_phrase_process extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff')) {
            redirect('/', 'refresh');
        }
        else {
            $config['upload_path'] = "./sounds/";
            $config['allowed_types'] = "wav";
            $config['max_size'] = '2048';

            $this->load->library('upload', $config);

            $id = $this->input->post('phrase_id');
            $phrase = $this->input->post('phrase');
            $pronunciation = $this->input->post('pronunciation');
            $translation = $this->input->post('translation');
            $category = $this->input->post('category');

            if (!empty($_FILES['userfile']['name'])) {
                if (!$this->upload->do_upload()) {
                    $data['pageTitle'] = "Upload Error";
                    $error = array('error' => $this->upload-
>display_errors());

                    $this->load->view('header_staff', $data);
                    $this->load->view('edit_phrase_view', $error);
                    $this->load->view('footer');
                }
                else {
                    $upload_array = $this->upload->data();

```

```

        $audio_link = $upload_array['full_path'];

        $data_to_update = array(

'phrasebook_phrase' => $phrase,

'phrasebook_translation' => $translation,

'phrasebook_category' => $category,

'phrasebook_audio_link' => $audio_link

        );

    }
    $this->db->where('phrasebook_id',$id);
    $query = $this->db->update('phrasebook',$data_to_update);
    if($query) {
        redirect('view_phrasebook_entry?id=' . $id);
    }
    else {
        $data["pageTitle"] = "Editing Unsuccessful";
        $this->load->view('header_staff',$data);
        $this->load->view('edit_unsuccessful_view');
    }
    $this->load->view('footer');
}
else {
    $data_to_update = array(

'phrasebook_phrase' =>
$phrase,

'phrasebook_translation' =>
$translation,

'phrasebook_category' =>
$category

    );

    $this->db->where('phrasebook_id',$id);
    $query = $this->db->update('phrasebook',$data_to_update);
    if($query) {
        redirect('view_phrasebook_entry?id=' . $id);
    }
    else {
        $data["pageTitle"] = "Editing Unsuccessful";
        $this->load->view('header_staff',$data);
        $this->load->view('edit_unsuccessful_view');
        $this->load->view('footer');
    }
}
}
}
}
}


```

```

/* End of file: edit_phrase_process.php */

```

```

/application/controllers/edit_question.php

```

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_question extends CI_Controller {
    public function __construct(){
        parent::__construct();
        $this->load->model('question_entries');
        $this->load->library('pagination');
    }

    public function index() {
        $segment = $this->uri->total_segments();
        $config = array();
        $config["base_url"] = site_url("edit_question/index");
        $config["total_rows"] = $this->question_entries->record_count();
        $config["per_page"] = 10;
        $config["uri_segment"] = $segment;
    }
}

```

```

        $choice = $config["total_rows"] / $config["per_page"];
        $config["num_links"] = round($choice);

        $this->pagination->initialize($config);

        $page = ($this->uri->segment($segment)) ? $this->uri->segment($segment) :
0;

        $data['pageTitle'] = "View Question Entries For Editing";
        $data["results"] = $this->question_entries-
>fetch_entries($config["per_page"],$page);
        $data["links"] = $this->pagination->create_links();

        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('edit_question_view', $data);
            $this->load->view('footer');
        }
    }
}

/* End of file: edit_question.php */

/application/controllers/edit_question_process.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_question_process extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $id = $this->input->post('question_no');
            $title = $this->input->post('title');
            $type = $this->input->post('type');
            $choice_a = $this->input->post('choice_a');
            $choice_b = $this->input->post('choice_b');
            $choice_c = $this->input->post('choice_c');
            $choice_d = $this->input->post('choice_d');
            $correct_answer = $this->input->post('correct_answer');
            $lesson_id = $this->input->post('lesson_id');
            $data_to_update = array(
                'question_title' => $title,
                'question_type' => $type,
                'choice_a' => $choice_a,
                'choice_b' => $choice_b,
                'choice_c' => $choice_c,
                'choice_d' => $choice_d,
                'correct_answer' => $correct_answer,
                'lesson_id' => $lesson_id
            );

            $this->db->where('question_no',$id);
            $query = $this->db->update('question_bank',$data_to_update);
            if($query) {
                redirect('view_question?id=' . $id);
            }
            else {
                $data['pageTitle'] = "Question Not Successfully Edited";
                $this->load->view('header_staff',$data);
                $this->load->view('edit_question_unsuccessful_view');
            }
        }
    }
}

```

```

        $this->load->view('footer');
    }
}

/* End of file: edit_question_process.php */

/application/controllers/edit_question_proper.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Edit_question_proper extends CI_Controller {
    public function __construct(){
        parent::__construct();
    }

    public function index() {
        $data['pageTitle'] = "Edit Question Entry";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff',$data);
            $this->load->view('edit_question_proper_view');
            $this->load->view('footer');
        }
    }
}

/* End of file: edit_question_proper.php */

/application/controllers/forgot_password.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Forgot_password extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    public function index() {
        $data['pageTitle'] = "Password Reset Form";
        $data['message'] = "";
        $this->load->library('form_validation');
        $this->load->view('header', $data);
        $this->load->view('forgot_password_view',$data);
        $this->load->view('footer');
    }
}

/* End of file: forgot_password.php */

/application/controllers/forgot_password_process.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Forgot_password_process extends CI_Controller{
    function __construct(){
        parent::__construct();
        $this->load->library('form_validation');
    }

    public function complete($user_id = FALSE, $token = FALSE) {
        // forgotten_password_complete() will validate the token exists and reset the
password.
        // To ensure the new password is emailed to the user, set the 4th argument of
forgotten_password_complete() to 'TRUE' (The 3rd arg manually sets a new password so set as
'FALSE').
        // If successful, the password will be reset and emailed to the user.
        $this->flexi_auth->forgotten_password_complete($user_id, $token, FALSE, TRUE);

        // Set a message to the CIflashdata so that it is available after the page
redirect.
        $this->session->set_flashdata('message', $this->flexi_auth->get_messages());

        redirect('login');
    }
}

```

```

public function index(){
    $this->form_validation->set_rules('identity','Identity','required');

    $identity = $this->input->post('identity');
    $forgotten = $this->flexi_auth->forgotten_password($identity);

    if ($forgotten) { //if there were no errors
        $this->data['pageTitle'] = "Password Reset Successful";
        $this->load->view('header',$data);
        $this->load->view('forgotten_password_success_view');
        $this->load->view('footer');
    }
    else {
        $this->data['pageTitle'] = "Password Reset Error";
        $this->data['message'] = validation_errors('<p class="error_msg">',
'</p>');
        $this->data['message'] = (empty($this->data['message'])) ? $this-
>flexi_auth->get_messages() : $this->data['message'];
        $this->load->view('header',$this->data);
        $this->load->view('forgot_password_view',$this->data);
        $this->load->view('footer');
    }
}

}

/* End of file: forgot_password_process.php */

/application/controllers/grammar.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Grammar extends CI_Controller {
    public function __construct() {
        parent::__construct();
        $this->load->model('grammar_entries');
        $this->load->library('pagination');
    }

    public function view_entries($start_letter) {
        if ($start_letter) {
            $segment = $this->uri->total_segments();
            $config = array();
            $config["base_url"] = site_url("grammar/view_entries/" .
$start_letter . "/"");
            $config["total_rows"] = $this->grammar_entries-
>letter_record_count($start_letter);
            $config["per_page"] = 10;
            $config["uri_segment"] = $segment;
            $choice = $config["total_rows"] / $config["per_page"];
            $config["num_links"] = round($choice);

            $this->pagination->initialize($config);

            $page = ($this->uri->segment($segment)) ? $this->uri-
>segment($segment) : 0;

            $data['pageTitle'] = "View Grammar Rule Entries";
            $data["results"] = $this->grammar_entries-
>fetch_entries_by_letter($config["per_page"],$page,$start_letter);
            $data["links"] = $this->pagination->create_links();

            if ($this->flexi_auth->is_logged_in()) {
                $this->load->view('header_user_tables',$data);
            }
            else if ($this->flexi_auth->in_group('staff')) {
                $this->load->view('header_staff_tables', $data);
            }
            else if ($this->flexi_auth->in_group('contributors')) {
                $this->load->view('header_contributor_tables', $data);
            }
            else {

```

```

        $this->load->view('header_tables', $data);
    }
    $this->load->view('grammar_view',$data);
    $this->load->view('footer');
}
else {
    if ($this->flexi_auth->is_logged_in()) {
        $this->load->view('header_user_tables',$data);
    }
    else {
        $this->load->view('header_tables', $data);
    }
    $this->load->view('parameter_missing_view');
    $this->load->view('footer');
}
}

public function index() {
    $data['pageTitle'] = "View Grammar Rules";
    if ($this->flexi_auth->is_logged_in()) {
        $this->load->view('header_user_tables',$data);
    }
    else if ($this->flexi_auth->in_group('staff')) {
        $this->load->view('header_staff_tables', $data);
    }
    else {
        $this->load->view('header_tables', $data);
    }
    $this->load->view('grammar_index_view');
    $this->load->view('footer');
}
}

/* End of file: grammar.php */

/application/controllers/guide_maintenance.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Guide_maintenance extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        $data['pageTitle'] = "Guide Maintenance";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('guide_maintenance_view');
            $this->load->view('footer');
        }
    }
}

/* End of file: guide_maintenance.php */

/application/controllers/guides.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Guides extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        $data['pageTitle'] = "Guide Portal";
        if ($this->flexi_auth->is_logged_in()) {
            $this->load->view('header_user',$data);

```

```

    }
    else if ($this->flexi_auth->in_group('staff')) {
        $this->load->view('header_staff',$data);
    }
    else if ($this->flexi_auth->in_group('contributors')) {
        $this->load->view('header_contributor',$data);
    }
    else {
        $this->load->view('header', $data);
    }
    $this->load->view('guides_view');
    $this->load->view('footer');
}

}

/* End of file: guides.php */

/application/controllers/index.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Index extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    public function index() {
        if (!$this->flexi_auth->is_logged_in()) {
            $data['pageTitle'] = "Mabuhay!";
            $this->load->view('header', $data);
            $this->load->view('index_view');
            $this->load->view('footer');
        }
        else {
            if ($this->flexi_auth->is_admin()){
                redirect('index_sysadmin');
            }
            else if ($this->flexi_auth->in_group('contentadmin')) {
                redirect('index_admin');
            }
            else if ($this->flexi_auth->in_group('contributors')) {
                redirect('index_contributor');
            }
            else if ($this->flexi_auth->in_group('staff')) {
                redirect('index_staff');
            }
            else {
                redirect('index_user');
            }
        }
    }
}

/* End of file: index.php */

/application/controllers/index_admin.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Index_admin extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    public function index() {
        if ($this->flexi_auth->in_group('contentadmin')) {
            $user_id = $this->flexi_auth->get_user_id();
            $name = $this->flexi_auth->get_user_by_id($user_id)->row();
            $data['pageTitle'] = "Mabuhay!";
            $data['name'] = $name->userinfo_first_name;

            $this->load->view('header_admin', $data);
            $this->load->view('index_admin_view',$data);
            $this->load->view('footer');
        }
    }
}

```

```

        else {
            redirect('');
        }
    }
}

/* End of file: index_admin.php */

/application/controllers/index_contributor.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Index_contributor extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    public function index() {
        if ($this->flexi_auth->in_group('contributors')) {
            $user_id = $this->flexi_auth->get_user_id();
            $name = $this->flexi_auth->get_user_by_id($user_id)->row();
            $data['pageTitle'] = "Mabuhay!";
            $data['name'] = $name->userinfo_first_name;

            $this->load->view('header_contributor', $data);
            $this->load->view('index_contributor_view', $data);
            $this->load->view('footer');
        }
        else {
            redirect('');
        }
    }
}

/* End of file: index_contributor.php */

/application/controllers/index_staff.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Index_staff extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    public function index() {
        if ($this->flexi_auth->in_group('staff')) {
            $user_id = $this->flexi_auth->get_user_id();
            $name = $this->flexi_auth->get_user_by_id($user_id)->row();
            $data['pageTitle'] = "Mabuhay!";
            $data['name'] = $name->userinfo_first_name;
            $this->load->view('header_staff', $data);
            $this->load->view('index_staff_view', $data);
            $this->load->view('footer');
        }
        else {
            redirect('/', 'refresh');
        }
    }
}

/* End of file: index_staff.php */

/application/controllers/index_sysadmin.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Index_sysadmin extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    public function index() {
        if ($this->flexi_auth->is_admin()) {
            $user_id = $this->flexi_auth->get_user_id();
            $name = $this->flexi_auth->get_user_by_id($user_id)->row();
            $data['pageTitle'] = "Mabuhay!";
            $data['name'] = $name->userinfo_first_name;

```



```

        $this->load->view('header_sysadmin', $data);
        $this->load->view('index_sysadmin_view', $data);
        $this->load->view('footer');
    }
    else {
        redirect('');
    }
}
}

/* End of file: index_sysadmin.php */

```

/application/controllers/index_user.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Index_user extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    public function index() {
        if ($this->flexi_auth->is_logged_in()) {
            $user_id = $this->flexi_auth->get_user_id();
            $name = $this->flexi_auth->get_user_by_id($user_id)->row();
            $data['pageTitle'] = "Mabuhay!";
            $data['name'] = $name->userinfo_first_name;
            $this->load->view('header_user', $data);
            $this->load->view('index_user_view', $data);
            $this->load->view('footer');
        }
        else {
            redirect('');
        }
    }
}

/* End of file: index_user.php */

```

/application/controllers/lesson_delete.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Lesson_delete extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function delete() {
        $id = $this->input->get('id');
        $this->db->where('lesson_id', $id);
        $query = $this->db->delete('lesson');
        if($query) {
            $data['pageTitle'] = "Lesson Deleted";
            $this->load->view('header_staff', $data);
            $this->load->view('lesson_deleted_view');
        }
        else {
            $data['pageTitle'] = "Delete Error";
            $this->load->view('header_staff', $data);
            $this->load->view('delete_error_lesson_view');
        }
        $this->load->view('footer');
    }

    function index() {
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth->
        >in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $data['pageTitle'] = "Delete Lesson?";

```



```
/* End of file: lesson_maintenance.php */
```

```
/application/controllers/lessons.php
```

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Lessons extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        $data['pageTitle'] = "Lesson Portal";
        if (!$this->flexi_auth->is_logged_in()) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_user', $data);
            $this->load->view('lessons_view');
            $this->load->view('footer');
        }
    }
}
}
```

```
/* End of file: lessons.php */
```

```
/application/controllers/link_maintenance.php
```

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Link_maintenance extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        $data['pageTitle'] = "Link Maintenance";
        if (!$this->flexi_auth->is_logged_in()) {
            redirect('/', 'refresh');
        }
        else if ($this->flexi_auth->in_group('contributors')) {
            $this->load->view('header_contributor', $data);
            $this->load->view('link_maintenance_view');
            $this->load->view('footer');
        }
        else if ($this->flexi_auth->in_group('staff')) {
            $this->load->view('header_staff', $data);
            $this->load->view('link_maintenance_view');
            $this->load->view('footer');
        }
    }
}
}
```

```
/* End of file: link_maintenance.php */
```

```
/application/controllers/links.php
```

```
<?php
class Links extends CI_Controller
{
    public function __construct() {
        parent::__construct();
        $this->load->helper("url");
        $this->load->model("link_model");
        $this->load->library("pagination");
    }

    public function view_links() {
        $config["base_url"] = site_url("links/view_links");
        $config["total_rows"] = $this->link_model->approved_count();
        $config["per_page"] = 20;
        $config["uri_segment"] = 3;
    }
}
```

```

$choice = $config["total_rows"] / $config["per_page"];
$config["num_links"] = round($choice);

$this->pagination->initialize($config);

$page = ($this->uri->segment(3))? $this->uri->segment(3) : 0;
$data["pageTitle"] = "Link Exchange";
$data["results"] = $this->link_model->fetch_approved_entries($config["per_page"], $page);
$data["links"] = $this->pagination->create_links();

if ($this->flexi_auth->is_logged_in()) {
    $this->load->view('header_user_tables',$data);
}
else {
    $this->load->view('header_tables', $data);
}
$this->load->view("links_view", $data);
$this->load->view("footer");
}

public function index() {
    redirect($config["base_url"] . "links/view_links");
}
}

/* End of file: links.php */

/application/controllers/list_delete_dictionary.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class List_delete_dictionary extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->model('dictionary_entries');
        $this->load->library('pagination');
    }

    function index() {
        $segment = $this->uri->total_segments();
        $config = array();
        $config["base_url"] = site_url("list_delete_dictionary/index");
        $config["total_rows"] = $this->dictionary_entries->record_count();
        $config["per_page"] = 10;
        $config["uri_segment"] = $segment;
        $choice = $config["total_rows"] / $config["per_page"];
        $config["num_links"] = round($choice);

        $this->pagination->initialize($config);

        $page = ($this->uri->segment($segment)) ? $this->uri->segment($segment) :
0;

        $data['pageTitle'] = "Delete Dictionary Rule Entries";
        $data["results"] = $this->dictionary_entries-
>fetch_entries($config["per_page"], $page);
        $data["links"] = $this->pagination->create_links();

        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('list_delete_dictionary_view',$data);
            $this->load->view('footer');
        }
    }
}

}

/* End of file: list_delete_dictionary.php */

```

```

/application/controllers/list_delete_grammar.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
    class List_delete_grammar extends CI_Controller {
        function __construct() {
            parent::__construct();
            $this->load->model('grammar_entries');
            $this->load->library('pagination');
        }

        function index() {
            $segment = $this->uri->total_segments();
            $config = array();
            $config["base_url"] = site_url("list_delete_grammar/index");
            $config["total_rows"] = $this->grammar_entries->record_count();
            $config["per_page"] = 10;
            $config["uri_segment"] = $segment;
            $choice = $config["total_rows"] / $config["per_page"];
            $config["num_links"] = round($choice);

            $this->pagination->initialize($config);

            $page = ($this->uri->segment($segment)) ? $this->uri->segment($segment) :
0;

            $data['pageTitle'] = "Delete Grammar Rule Entries";
            $data["results"] = $this->grammar_entries-
>fetch_entries($config["per_page"], $page);
            $data["links"] = $this->pagination->create_links();

            if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
                redirect('/', 'refresh');
            }
            else {
                $this->load->view('header_staff', $data);
                $this->load->view('list_delete_grammar_view', $data);
                $this->load->view('footer');
            }
        }
    }

}

/* End of file: list_delete_grammar.php */

```

```

/application/controllers/list_delete_instruction.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
    class List_delete_instruction extends CI_Controller {
        function __construct() {
            parent::__construct();
            $this->load->model('instruction_entries');
            $this->load->library('pagination');
        }

        function index() {
            $segment = $this->uri->total_segments();
            $config = array();
            $config["base_url"] = site_url("list_delete_instruction/index");
            $config["total_rows"] = $this->instruction_entries->record_count();
            $config["per_page"] = 10;
            $config["uri_segment"] = $segment;
            $choice = $config["total_rows"] / $config["per_page"];
            $config["num_links"] = round($choice);

            $this->pagination->initialize($config);

            $page = ($this->uri->segment($segment)) ? $this->uri->segment($segment) :
0;

            $data['pageTitle'] = "Delete Instructions from the Bank";
            $data["results"] = $this->instruction_entries-
>fetch_entries($config["per_page"], $page);

```

```

        $data["links"] = $this->pagination->create_links();

        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('list_delete_instruction_view', $data);
            $this->load->view('footer');
        }
    }
}

/* End of file: list_delete_instruction.php */

/application/controllers/list_delete_link.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class List_delete_link extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->model('link_model');
        $this->load->library('pagination');
    }

    function index() {
        $segment = $this->uri->total_segments();
        $config = array();
        $config["base_url"] = site_url("list_delete_link/index");
        $config["total_rows"] = $this->link_model->record_count_by_user();
        $config["per_page"] = 10;
        $config["uri_segment"] = $segment;
        $choice = $config["total_rows"] / $config["per_page"];
        $config["num_links"] = round($choice);

        $this->pagination->initialize($config);

        $page = ($this->uri->segment($segment)) ? $this->uri->segment($segment) :
0;

        $data['pageTitle'] = "Delete Links";
        $data["results"] = $this->link_model-
>fetch_entries($config["per_page"], $page);
        $data["links"] = $this->pagination->create_links();

        $group = array('contributors', 'staff');
        if (!$this->flexi_auth->in_group($group)) {
            redirect('/', 'refresh');
        }
        else {
            if($this->flexi_auth->in_group('contributors')) {
                $this->load->view('header_contributor', $data);
            }
            else if ($this->flexi_auth->in_group('staff')) {
                $this->load->view('header_staff', $data);
            }
            $this->load->view('list_delete_link_view', $data);
            $this->load->view('footer');
        }
    }
}

/* End of file: list_delete_link.php */

/applications/controllers/list_delete_phrasebook.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class List_delete_phrasebook extends CI_Controller {
    function __construct() {
        parent::__construct();

```

```

        $this->load->model('phrasebook_entries');
        $this->load->library('pagination');
    }

    function index() {
        $segment = $this->uri->total_segments();
        $config = array();
        $config["base_url"] = site_url("list_delete_phrasebook/index");
        $config["total_rows"] = $this->phrasebook_entries->record_count();
        $config["per_page"] = 10;
        $config["uri_segment"] = $segment;
        $choice = $config["total_rows"] / $config["per_page"];
        $config["num_links"] = round($choice);

        $this->pagination->initialize($config);

        $page = ($this->uri->segment($segment)) ? $this->uri->segment($segment) :
0;

        $data['pageTitle'] = "Delete Phrasebook Entries";
        $data["results"] = $this->phrasebook_entries-
>fetch_entries($config["per_page"], $page);
        $data["links"] = $this->pagination->create_links();

        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('list_delete_phrasebook_view', $data);
            $this->load->view('footer');
        }
    }
}

/* End of file: list_delete_phrasebook.php */

/application/controllers/list_delete_question.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class List_delete_question extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->model('question_entries');
        $this->load->library('pagination');
    }

    function index() {
        $segment = $this->uri->total_segments();
        $config = array();
        $config["base_url"] = site_url("list_delete_question/index");
        $config["total_rows"] = $this->question_entries->record_count();
        $config["per_page"] = 10;
        $config["uri_segment"] = $segment;
        $choice = $config["total_rows"] / $config["per_page"];
        $config["num_links"] = round($choice);

        $this->pagination->initialize($config);

        $page = ($this->uri->segment($segment)) ? $this->uri->segment($segment) :
0;

        $data['pageTitle'] = "Delete Questions from the Bank";
        $data["results"] = $this->question_entries-
>fetch_entries($config["per_page"], $page);
        $data["links"] = $this->pagination->create_links();

        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
    }
}

```

```

        else {
            $this->load->view('header_staff', $data);
            $this->load->view('list_delete_question_view',$data);
            $this->load->view('footer');
        }
    }
}

/* End of file: list_delete_question.php */

/application/controllers/list_dictionary_entry.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class List_dictionary_entry extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->model('dictionary_entries');
        $this->load->library('pagination');
    }

    function index() {
        $segment = $this->uri->total_segments();
        $config = array();
        $config["base_url"] = site_url("list_dictionary_entry/index");
        $config["total_rows"] = $this->dictionary_entries->record_count();
        $config["per_page"] = 10;
        $config["uri_segment"] = $segment;
        $choice = $config["total_rows"] / $config["per_page"];
        $config["num_links"] = round($choice);

        $this->pagination->initialize($config);

        $page = ($this->uri->segment($segment)) ? $this->uri->segment($segment) :
0;

        $data['pageTitle'] = "Edit Dictionary Rule Entries";
        $data["results"] = $this->dictionary_entries-
>fetch_entries($config["per_page"],$page);
        $data["links"] = $this->pagination->create_links();

        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('list_dictionary_entry_view',$data);
            $this->load->view('footer');
        }
    }
}

/* End of file: list_dictionary_entry.php */

```

```

/application/controllers/list_phrasebook_entry.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class List_phrasebook_entry extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->model('phrasebook_entries');
        $this->load->library('pagination');
    }

    function index() {
        $segment = $this->uri->total_segments();
        $config = array();
        $config["base_url"] = site_url("list_phrasebook_entry/index");
        $config["total_rows"] = $this->phrasebook_entries->record_count();
    }
}

```



```

        $config["per_page"] = 10;
        $config["uri_segment"] = $segment;
        $choice = $config["total_rows"] / $config["per_page"];
        $config["num_links"] = round($choice);

        $this->pagination->initialize($config);

        $page = ($this->uri->segment($segment)) ? $this->uri->segment($segment) :
0;

        $data['pageTitle'] = "Edit Phrasebook Entries";
        $data["results"] = $this->phrasebook_entries-
>fetch_entries($config["per_page"], $page);
        $data["links"] = $this->pagination->create_links();

        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('list_phrasebook_entry_view', $data);
            $this->load->view('footer');
        }
    }

}

/* End of file: list_phrasebook_entry.php */

```

/application/controllers/login.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Login extends CI_Controller{

    function __construct(){
        parent::__construct();
    }

    public function index(){
        $this->load->library('form_validation');
        $data['pageTitle'] = "User Login Form";
        $data['message'] = "";

        $this->load->view('header', $data);
        $this->load->view('login_view', $data);
        $this->load->view('footer');
    }

}

/* End of file: login.php */

```

/application/controllers/login_process.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Login_process extends CI_Controller{

    function __construct(){
        parent::__construct();
    }

    public function index() {
        $this->load->library('form_validation');
        $data['pageTitle'] = "User Login Form";

        $this->form_validation->set_rules('identity', 'Username', 'required');
        $this->form_validation->set_rules('password', 'Password', 'required');

        $identity = $this->input->post('identity');
        $password = $this->input->post('password');
        $remember = (bool) $this->input->post('remember_user');
    }
}

```

```

        if ($remember) {
            $login = $this->flexi_auth->login($identity,$password,$remember);
            if(!$login) {
                $this->data['pageTitle'] = "Login Incorrect";
                $this->data['message'] = validation_errors('<p class="error_msg">',
'</p>');
                $this->data['message'] = (empty($this->data['message'])) ? $this-
>flexi_auth->get_messages() : $this->data['message'];
                $this->load->view('header', $this->data);
                $this->load->view('login_view',$this->data);
                $this->load->view('footer');
            }
            else {
                redirect('/', 'refresh');
            }
        }
        else {
            $login = $this->flexi_auth->login($identity,$password);
            if(!$login) {
                $this->data['pageTitle'] = "Login Incorrect";
                $this->data['message'] = validation_errors('<p class="error_msg">',
'</p>');
                $this->data['message'] = (empty($this->data['message'])) ? $this-
>flexi_auth->get_messages() : $this->data['message'];
                $this->load->view('header', $this->data);
                $this->load->view('login_view',$this->data);
                $this->load->view('footer');
            }
            else {
                redirect('/', 'refresh');
            }
        }
    }
}

```

/* End of file: login_process.php */

/application/controllers/logout.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Logout extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    public function index() {
        $this->flexi_auth->logout();
        redirect('/');
    }
}

```

/* End of file: logout.php */

/application/controllers/maintain_user_groups.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Maintain_user_groups extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    public function index() {
        $data['pageTitle'] = "Maintain User Groups";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>is_admin())) {
            redirect('/', 'refresh');
        }
        $data['results'] = $this->flexi_auth->get_groups_result();
        $this->load->view('header_sysadmin', $data);
        $this->load->view('maintain_user_groups_view',$data);
        $this->load->view('footer');
    }
}

```

```

/* End of file: maintain_user_groups.php */

/application/controllers/maintain_users.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Maintain_users extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    public function index() {
        $data['pageTitle'] = "Maintain Users";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>is_admin())) {
            redirect('/', 'refresh');
        }

        $data['results'] = $this->flexi_auth->get_custom_user_data_result();
        $this->load->view('header_sysadmin', $data);
        $this->load->view('maintain_users_view', $data);
        $this->load->view('footer');
    }
}

/* End of file: maintain_users.php */

/application/controllers/organize_bank.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Organize_bank extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        $data['pageTitle'] = "Organize Question Bank";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff')) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('organize_bank_view');
            $this->load->view('footer');
        }
    }
}

/* End of file: organize_bank.php */

/application/controllers/phrasebook.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Phrasebook extends CI_Controller {
    public function __construct() {
        parent::__construct();
        $this->load->model('phrasebook_entries');
        $this->load->library('pagination');
    }

    public function view_entries($start_letter) {
        if ($start_letter) {
            $segment = $this->uri->total_segments();
            $config = array();
            $config["base_url"] = site_url("phrasebook/view_entries/".
$start_letter . "/");
            $config["total_rows"] = $this->phrasebook_entries-
>letter_record_count($start_letter);
            $config["per_page"] = 10;
            $config["uri_segment"] = $segment;
            $choice = $config["total_rows"] / $config["per_page"];
            $config["num_links"] = round($choice);

            $this->pagination->initialize($config);

```

```

>segment($segment) : 0;

$page = ($this->uri->segment($segment)) ? $this->uri-

$data['pageTitle'] = "View Phrasebook Entries by Letter: " .

$start_letter;

$data["results"] = $this->phrasebook_entries-
>fetch_entries_by_letter($config["per_page"], $page, $start_letter);
$data["links"] = $this->pagination->create_links();

if ($this->flexi_auth->is_logged_in()) {
    $this->load->view('header_user_tables', $data);
}
else if ($this->flexi_auth->in_group('staff')) {
    $this->load->view('header_staff_tables', $data);
}
else if ($this->flexi_auth->in_group('contributors')) {
    $this->load->view('header_contributor_tables', $data);
}
else {
    $this->load->view('header_tables', $data);
}
$this->load->view('phrasebook_view', $data);
$this->load->view('footer');
}
else {
    $data['pageTitle'] = "Parameter Missing!";
    if ($this->flexi_auth->is_logged_in()) {
        $this->load->view('header_user_tables', $data);
    }
    else {
        $this->load->view('header_tables', $data);
    }
    $this->load->view('parameter_missing_view');
    $this->load->view('footer');
}
}

public function index() {
    $data['pageTitle'] = "View Phrasebook Entries";
    if ($this->flexi_auth->is_logged_in()) {
        $this->load->view('header_user_tables', $data);
    }
    else if ($this->flexi_auth->in_group('staff')) {
        $this->load->view('header_staff_tables', $data);
    }
    else {
        $this->load->view('header_tables', $data);
    }
    $this->load->view('phrasebook_index_view');
    $this->load->view('footer');
}
}

/* End of file: phrasebook.php */

/application/controllers/quiz_process.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Quiz_process extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->model('quiz_model');
    }

    function check_correct_answers($correct, $answered, $entries) {
        $no_correct = 0;
        for ($j = 0; $j < $entries; $j++) {

```

```

        $is_correct = $this->quiz_model-
>check_correct_answer($correct[$j], $answered[$j]);
        if ($is_correct) {
            $no_correct++;
        }
    }

    return $no_correct;
}

function display_right_or_wrong ($correct, $answered, $entries) {
    $correct_array = array();
    for ($j = 0; $j < $entries; $j++) {
        $is_correct = $this->quiz_model-
>check_correct_answer($correct[$j], $answered[$j]);
        if ($is_correct) {
            $correct_array[] = $j+1 . " . " . $answered[$j] . " is
correct. ";
        }
        else {
            $correct_array[] = $j+1 . " . " . $answered[$j] . " is
wrong. The correct answer is " . $correct[$j] . ".";
        }
    }
    return $correct_array;
}

function index() {
    if (!$this->flexi_auth->is_logged_in()) {
        redirect('/', 'refresh');
    }
    else {
        $data['pageTitle'] = "Quiz Results";
        $user_id = $this->flexi_auth->get_user_id();

        $correct_answers = $this->session->userdata('correct');
        $no_of_questions = $this->session->userdata('entries');
        $choices = $this->input->post('choice');
        $lesson_id = $this->input->post('lesson_id');

        $no_of_correct_answers = $this-
>check_correct_answers($correct_answers, $choices, $no_of_questions);

        $insert_into_test_records = array(
            'user_id' => $user_id,
            'no_correct' => $no_of_correct_answers,
            'no_of_questions' => $no_of_questions,
            'lesson_id' => $lesson_id
        );

        $evaluated_answers = $this-
>display_right_or_wrong($correct_answers, $choices, $no_of_questions);

        $data['right_wrong'] = $evaluated_answers;
        $data['correct'] = $no_of_correct_answers;
        $data['questions'] = $no_of_questions;

        $this->db->insert('quizzes_taken', $insert_into_test_records);

        $this->load->view('header_user', $data);
        $this->load->view('quiz_results_view', $data);
        $this->load->view('footer');
    }
}

}

/* End of file: quiz_process.php */

/application/controllers/reg_user.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

```

```

class Reg_user extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->library('form_validation');
    }

    function index() {
        if ($this->input->post('submit_reg_user')) {
            $username = strtolower($this->input->post('username'));
            $password = $this->input->post('password');
            $first_name = $this->input->post('first_name');
            $last_name = $this->input->post('last_name');
            $email = $this->input->post('email');

            $additional_data = array(

                'userinfo_first_name' => $first_name,

                'userinfo_last_name' => $last_name

            );

            $group = 5;

            $reg_user = $this->flexi_auth-
>insert_user($email,$username,$password,$additional_data,$group,TRUE);

            if ($reg_user) {
                redirect('user_signup_success');
            }
            else {
                $data['message'] = "User was not successfully added.";
                redirect('user_signup',$data);
            }
        }

        else {
            $this->data['message'] = (validation_errors() ? validation_errors()
: ($this->ion_auth->errors() ? $this->ion_auth->errors() : $this->session-
>flashdata('message')));

            $this->data['first_name'] = array(
                'name' => 'first_name',
                'id' => 'first_name',
                'type' => 'text',
                'value' => $this->form_validation->set_value('first_name'),
            );
            $this->data['last_name'] = array(
                'name' => 'last_name',
                'id' => 'last_name',
                'type' => 'text',
                'value' => $this->form_validation->set_value('last_name'),
            );
            $this->data['email'] = array(
                'name' => 'email',
                'id' => 'email',
                'type' => 'text',
                'value' => $this->form_validation->set_value('email'),
            );
            $this->data['password'] = array(
                'name' => 'password',
                'id' => 'password',
                'type' => 'password',
                'value' => $this->form_validation->set_value('password'),
            );
            $this->data['confirm_password'] = array(
                'name' => 'confirm_password',
                'id' => 'confirm_password',
                'type' => 'password',
                'value' => $this->form_validation-
>set_value('confirm_password'),
            );
        }
    }
}

```

```

        redirect('user_signup',$this->data);
    }
}

/* End of file: reg_user.php */

/application/controllers/reset_password.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Reset_password extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->library('form_validation');
    }
    public function index($code = NULL) {
        if (!$code)
        {
            show_404();
        }

        $user = $this->ion_auth->forgotten_password_check($code);

        if ($user)
        {
            //if the code is valid then display the password reset form

            $this->form_validation->set_rules('new', 'New Password',
'required|min_length[' . $this->config->item('min_password_length', 'ion_auth') . ']|max_length['
. $this->config->item('max_password_length', 'ion_auth') . ']|matches[new_confirm]');
            $this->form_validation->set_rules('new_confirm', 'Confirm New Password',
'required');

            if ($this->form_validation->run() == false)
            {
                //display the form

                //set the flash data error message if there is one
                $this->data['message'] = (validation_errors()) ?
validation_errors() : $this->session->flashdata('message');

                $this->data['min_password_length'] = $this->config-
>item('min_password_length', 'ion_auth');
                $this->data['new_password'] = array(
                    'name' => 'new',
                    'id' => 'new',
                    'type' => 'password',
                    'pattern' => '^.{'.$this-
>data['min_password_length'].'}.*$',
                );
                $this->data['new_password_confirm'] = array(
                    'name' => 'new_confirm',
                    'id' => 'new_confirm',
                    'type' => 'password',
                    'pattern' => '^.{'.$this-
>data['min_password_length'].'}.*$',
                );
                $this->data['user_id'] = array(
                    'name' => 'user_id',
                    'id' => 'user_id',
                    'type' => 'hidden',
                    'value' => $user->id,
                );
                $this->data['csrf'] = $this->_get_csrf_nonce();
                $this->data['code'] = $code;

                //render
                $this->load->view('reset_password_view', $this->data);
            }
            else
            {
                // do we have a valid request?

```

```

>input->post('user_id'))
    if ($this->_valid_csrf_nonce() === FALSE || $user->id != $this-
    {
        //something fishy might be up
        $this->ion_auth->clear_forgotten_password_code($code);

        show_error('This form post did not pass our security
checks.');
```

```

    }
    else
    {
        // finally change the password
        $identity = $user->{$this->config->item('identity',
'ion_auth')});

        $change = $this->ion_auth->reset_password($identity, $this-
>input->post('new'));

        if ($change)
        {
            //if the password was successfully changed
            $this->session->set_flashdata('message', $this-
>ion_auth->messages());

            $this->logout();
        }
        else
        {
            $this->session->set_flashdata('message', $this-
>ion_auth->errors());

            redirect('reset_password' . $code, 'refresh');
        }
    }
}
}
else
{
    //if the code is invalid then send them back to the forgot password page
    $this->session->set_flashdata('message', $this->ion_auth->errors());
    redirect("forgot_password", 'refresh');
}
}
}

/* End of file: reset_password.php */

```

/application/controllers/search_dictionary_entry.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Search_dictionary_entry extends CI_Controller {
    public function __construct() {
        parent::__construct();
        $this->load->library('form_validation');
    }

    public function index() {
        $data['pageTitle'] = "Search for Dictionary Entries";
        if ($this->flexi_auth->in_group('staff')) {
            $this->load->view('header_staff_dictionary_ajax',$data);
            $this->load->view('search_dictionary_view');
            $this->load->view('footer');
        }
        else if ($this->flexi_auth->in_group('users')) {
            $this->load->view('header_user_dictionary_ajax',$data);
            $this->load->view('search_dictionary_view');
            $this->load->view('footer');
        }
        else {
            $this->load->view('header_dictionary_ajax',$data);
            $this->load->view('search_dictionary_view');
            $this->load->view('footer');
        }
    }
}

```



```

    }
}

/* End of file: search_dictionary_entry.php */

/application/controller/search_grammar_entry.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Search_grammar_entry extends CI_Controller {
    public function __construct() {
        parent::__construct();
        $this->load->library('form_validation');
    }

    public function index() {
        $data['pageTitle'] = "Search for Grammar Rules";
        if ($this->flexi_auth->in_group('staff')) {
            $this->load->view('header_staff_grammar_ajax',$data);
            $this->load->view('search_grammar_view');
            $this->load->view('footer');
        }
        else if ($this->flexi_auth->in_group('users')) {
            $this->load->view('header_user_grammar_ajax',$data);
            $this->load->view('search_grammar_view');
            $this->load->view('footer');
        }
        else {
            $this->load->view('header_grammar_ajax',$data);
            $this->load->view('search_grammar_view');
            $this->load->view('footer');
        }
    }
}

```

/* End of file: search_grammar_entry.php */

```

/application/controllers/search_phrasebook_entry.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Search_phrasebook_entry extends CI_Controller {
    public function __construct() {
        parent::__construct();
        $this->load->library('form_validation');
    }

    public function index() {
        $data['pageTitle'] = "Search for Phrasebook Entries";
        if ($this->flexi_auth->in_group('staff')) {
            $this->load->view('header_staff_phrase_ajax',$data);
            $this->load->view('search_for_phrase_view');
            $this->load->view('footer');
        }
        else if ($this->flexi_auth->in_group('users')) {
            $this->load->view('header_user_phrase_ajax',$data);
            $this->load->view('search_for_phrase_view');
            $this->load->view('footer');
        }
        else {
            $this->load->view('header_phrase_ajax',$data);
            $this->load->view('search_for_phrase_view');
            $this->load->view('footer');
        }
    }
}

```

/* End of file: search_phrasebook_entry.php */

```

/application/controllers/suggest_changes.php
<?php
class Suggest_changes extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
}

```

```

        $this->load->library('email');
        $this->load->library('mathcaptcha');
        $this->load->library('form_validation');
    }

    function _check_math_captcha($str)
    {
        if ($this->mathcaptcha->check_answer($str))
        {
            return TRUE;
        }
        else
        {
            $this->form_validation->set_message('_check_math_captcha', 'Enter a valid math
captcha response.');
```

captcha response.');

```

            return FALSE;
        }
    }

    public function process() {
        $sender_id = $this->flexi_auth->get_user_id();

        $guide_type = $this->input->post('guide_type');
        $message = $this->input->post('message');

        $data_to_add = array(
            'sender_id' => $sender_id,
            'type' => $guide_type,
            'content' => $message
        );

        $query = $this->db->insert('changes',$data_to_add);

        if($query) {
            $data['pageTitle'] = "Change Request Successful";
            if ($this->flexi_auth->in_group('contributors')) {
                $this->load->view('header_contributor', $data);
            }
            else {
                $this->load->view('header_user', $data);
            }
            $this->load->view('change_request_sent');
            $this->load->view('footer');
        }
        else {
            $data['pageTitle'] = "Change Request Unsuccessful";
            if ($this->flexi_auth->in_group('contributors')) {
                $this->load->view('header_contributor', $data);
            }
            else {
                $this->load->view('header_user', $data);
            }
            $this->load->view('change_request_unsuccessful');
            $this->load->view('footer');
        }
    }

    public function index()
    {
        $config["operation"] = "random";
        $config["question_format"] = "random";
        $this->mathcaptcha->init($config);

        $data['math_captcha_question'] = $this->mathcaptcha->get_question();

        $this->form_validation->set_rules('math_captcha', 'Math CAPTCHA',
'required|callback__check_math_captcha');
```

'required|callback__check_math_captcha');

```

        $data['pageTitle'] = "Suggest Guide Changes";
        if (!$this->flexi_auth->in_group(array('users','contributors')) {
            redirect('/', 'refresh');
```

redirect('/', 'refresh');

```

    }
    else {
    if ($this->flexi_auth->in_group('contributors')) {
        $this->load->view('header_contributor', $data);
    }
    else {
        $this->load->view('header_user', $data);
    }
    if ($this->form_validation->run() == FALSE)
    {
        $this->load->view('suggest_changes_view',$data);
    }
    else
    {
        $this->load->view('suggest_changes_view',$data);
    }
    $this->load->view('footer');
    }
}
}

```

/* End of file: suggest_changes.php */

/application/controllers/take_quiz.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Take_quiz extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->model('quiz_model');
    }

    function index() {
        if (!$this->flexi_auth->is_logged_in()) {
            redirect('/', 'refresh');
        }
        else {
            $data['pageTitle'] = "Take Quiz";
            $this->load->view('header_user_ajax', $data);

            $lesson = $this->input->get('id');

            $data['instructions'] = $this->quiz_model-
>fetch_instructions($lesson);
            $data["results"] = $this->quiz_model-
>generate_entries_by_lesson($lesson);

            $this->load->view('take_quiz_view', $data);
            $this->load->view('footer');
        }
    }
}
}

```

/* End of file: take_quiz.php */

/application/controllers/update_user.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Update_user extends CI_Controller {
    public function __construct(){
        parent::__construct();
    }

    public function process() {
        $username = $this->input->post('username');
        $user_id = $this->input->post('user_id');
        $email = $this->input->post('email');
        $first_name = $this->input->post('first_name');
        $last_name = $this->input->post('last_name');

        $user_data = array(
            'uacc_email' => $email,

```

```

        'userinfo_first_name' => $first_name,
        'userinfo_last_name' => $last_name
    );

    $update = $this->flexi_auth->update_user($user_id,$user_data);

    if ($update) {
        $data['pageTitle'] = "Updating of User Details Successful";
        $this->load->view('header_sysadmin',$data);
        $this->load->view('update_user_success_view');
        $this->load->view('footer');
    }
    else {
        $data['pageTitle'] = "Updating of User Details Unsuccessful";
        $this->load->view('header_sysadmin',$data);
        $this->load->view('update_user_failure_view');
        $this->load->view('footer');
    }
}

public function index() {
    $data['pageTitle'] = "Update User Details";
    if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>is_admin())) {
        redirect('/', 'refresh');
    }
    else {
        $this->load->view('header_sysadmin',$data);
        $this->load->view('update_user_view');
        $this->load->view('footer');
    }
}
}

/* End of file: update_user.php */

/application/controllers/update_user_group.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Update_user_group extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    public function process() {
        $group_id = $this->input->post('group_id');
        $name = $this->input->post('group_name');
        $description = $this->input->post('group_desc');

        $group_data = array(
            'ugrp_name'=> $name,
            'ugrp_desc' => $description
        );

        $result = $this->flexi_auth->update_group($group_id,$group_data);
        if ($update) {
            $data['pageTitle'] = "Updating of User Group Details Successful";
            $this->load->view('header_sysadmin',$data);
            $this->load->view('update_user_group_success_view');
            $this->load->view('footer');
        }
        else {
            $data['pageTitle'] = "Updating of User Group Details Unsuccessful";
            $this->load->view('header_sysadmin',$data);
            $this->load->view('update_user_group_failure_view');
            $this->load->view('footer');
        }
    }

    public function index() {
        $data['pageTitle'] = "Update User Group Details";

```

```

        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>is_admin())) {
            redirect('/', 'refresh');
        }

        $this->load->view('header_sysadmin', $data);
        $this->load->view('update_user_group_view');
        $this->load->view('footer');
    }
}

```

/* End of file: update_user_group.php */

```

/application/controllers/user_signup.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class User_signup extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->library('form_validation');
    }
    public function index() {
        $data['pageTitle'] = "User Signup Form";
        $data['message'] = "";
        $this->load->view('header', $data);
        $this->load->view('user_signup_view');
        $this->load->view('footer');
    }
}

```

/* End of file: user_signup.php */

```

/application/controllers/user_signup_success.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class User_signup_success extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    public function index() {
        $data['pageTitle'] = "User Signup Success";
        $this->load->view('header', $data);
        $this->load->view('user_signup_success_view');
        $this->load->view('footer');
    }
}

```

/* End of file: user_signup_success.php */

```

/application/controllers/view_dictionary_entry.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class View_dictionary_entry extends CI_Controller {
    public function __construct() {
        parent::__construct();
    }

    public function index() {
        $data['pageTitle'] = "View Dictionary Entry";
        if ($this->flexi_auth->in_group('staff')) {
            $this->load->view('header_staff', $data);
        }
        else if ($this->flexi_auth->in_group('contributors')) {
            $this->load->view('header_contributor', $data);
        }
        else if ($this->flexi_auth->in_group('users')) {
            $this->load->view('header_user', $data);
        }
        else {
            $this->load->view('header', $data);
        }
    }
}

```

```

        $this->load->view('view_dictionary_entry_view');
        $this->load->view('footer');
    }
}

/* End of file: view_dictionary_entry.php */

/application/controllers/view_grammar.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class View_grammar extends CI_Controller {
    public function __construct() {
        parent::__construct();
    }

    public function index() {
        $data['pageTitle'] = "View Grammar Rule Entry";
        if ($this->flexi_auth->in_group('staff')) {
            $this->load->view('header_staff',$data);
        }
        else if ($this->flexi_auth->in_group('contributors')) {
            $this->load->view('header_contributor',$data);
        }
        else if ($this->flexi_auth->in_group('users')) {
            $this->load->view('header_user',$data);
        }
        else {
            $this->load->view('header',$data);
        }
        $this->load->view('view_grammar_view');
        $this->load->view('footer');
    }
}

/* End of file: view_grammar.php */

/application/controllers/view_instruction.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class View_instruction extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        $data['pageTitle'] = "View Instruction";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff')) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff',$data);
            $this->load->view('view_instruction_view');
            $this->load->view('footer');
        }
    }
}

/* End of file: view_instruction.php */

/application/controllers/view_lessons.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class View_lessons extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->model('lesson_entries');
    }

    function index() {
        $data['pageTitle'] = "View Lessons";
        $data["results"] = $this->lesson_entries->fetch_entries_without_page();
    }
}

```

```

        if (!$this->flexi_auth->is_logged_in()) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_user', $data);
            $this->load->view('view_lessons_view', $data);
            $this->load->view('footer');
        }
    }
}

}

/* End of file: view_lessons.php */

/application/controllers/view_link.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class View_link extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        $data['pageTitle'] = "View Link";
        if ($this->flexi_auth->in_group('staff')) {
            $this->load->view('header_staff', $data);
        }
        else if ($this->flexi_auth->in_group('contributors')) {
            $this->load->view('header_contributor', $data);
        }
        else if ($this->flexi_auth->in_group('users')) {
            $this->load->view('header_user', $data);
        }
        else {
            $this->load->view('header', $data);
        }
        $this->load->view('view_link_view');
        $this->load->view('footer');
    }
}

/* End of file: view_link.php */

/application/controllers/view_phrasebook_entry.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class View_phrasebook_entry extends CI_Controller {
    public function __construct() {
        parent::__construct();
    }

    public function index() {
        $data['pageTitle'] = "View Phrasebook Entry";
        if ($this->flexi_auth->in_group('staff')) {
            $this->load->view('header_staff', $data);
        }
        else if ($this->flexi_auth->in_group('contributors')) {
            $this->load->view('header_contributor', $data);
        }
        else if ($this->flexi_auth->in_group('users')) {
            $this->load->view('header_user', $data);
        }
        else {
            $this->load->view('header', $data);
        }
        $this->load->view('view_phrasebook_entry_view');
        $this->load->view('footer');
    }
}

/* End of file: view_phrasebook_entry.php */

/application/controllers/view_progress.php

```

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class View_progress extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->model('progress_model');
    }

    function index() {
        $data['pageTitle'] = "View Progress";
        $data["results"] = $this->progress_model->get_entries();

        if (!$this->flexi_auth->is_logged_in()) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_user', $data);
            $this->load->view('view_progress_view', $data);
            $this->load->view('footer');
        }
    }
}

/* End of file: view_progress.php */

/application/controllers/view_question.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class View_question extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    function index() {
        $data['pageTitle'] = "View Question";
        if ((!$this->flexi_auth->is_logged_in()) || (!$this->flexi_auth-
>in_group('staff'))) {
            redirect('/', 'refresh');
        }
        else {
            $this->load->view('header_staff', $data);
            $this->load->view('view_question_view');
            $this->load->view('footer');
        }
    }
}

/* End of file: view_question.php */

```


VIEWS

about_view.php

```
<div id="main" role="main">
  <article class="main_class">
    <section>
      <h1>Why a new system for learning the Filipino
language?</h1>
      <p>The Filipino language is one of the fastest-growing
languages spoken in countries where Filipinos have migrated in recent years (e.g. Canada), but
its language sustainability among people abroad through the generations will be a key problem in
the future. In addition, the children and other further descendants of Filipinos who had settled
in the United States have no inclination to learn the culture and language of their ancestors.
Furthermore, there are also tourists from around the world who have to tackle the language
barrier once they are here since there are not too many commercial solutions for the Filipino
language in the language-learning marketplace.</p>
      <h1>Why is this system comprehensive?</h1>
      <p>The author's goal is to broaden the importance of the
Filipino language to wherever there are Filipinos around the world. Although there are resources
on the Filipino language on the Internet, these are scattered and often cover only one aspect of
the language. This system brings all aspects of the language, together with lessons that explain
the language further, into one comprehensive information system that can be of importance to
anyone who desires to learn the Filipino language.</p>
    </section>
    <section>
      <h1>CREDITS</h1>
      <ul>
        <li>Some content in the guides adapted from a list
found at hillmanwonders.com</li>
      </ul>
    </section>
  </article>
</div>
```

add_admin_success_view.php

```
<div id="main" role="main">
  <article class="regcomplete_main">
    <h3>Registration is now complete.</h3>
    <h4>Advise the admin to now log in with their chosen username and
password.</h4>
  </article>
</div>
```

add_admin_view.php

```
<div id="main" role="main">
  <article class="signup_main">
    <section id="signup_form">
      <h3>Before adding another system admin, make sure that he/she has
not had an account before. If he/she has forgotten their password, direct them to the Forgot
Password link in the Login screen.</h3>
      <h4>All fields are required.</h4>
      <div id="infoMessage"><?php echo $message;?></div>
      <?php echo form_open('add_admin_process'); ?>
      <p>
        <label for="username">
          Username: <?php echo
form_input('username', set_value('username')); ?>
        </label>
      </p>
      <p>
        <label for="password">
          Password: <?php echo
form_password('password', set_value('password')); ?>
        </label>
      </p>
      <p>
        <label for="confirm_password">
          Confirm Password: <?php echo
form_password('confirm_password', set_value('confirm_password')); ?>
      </p>
    </section>
  </article>
</div>
```

```

        </p>
        <p>
            <label for="first_name">
                First Name: <?php echo
form_input('first_name', set_value('first_name')); ?>
            </label>
        </p>
        <p>
            <label for="last_name">
                Last Name: <?php echo
form_input('last_name', set_value('last_name')); ?>
            </label>
        </p>
        <p>
            <label for="email">
                Email Address: <?php echo
form_input('email', set_value('email')); ?>
            </label>
        </p>
        <p>
            <?php echo form_submit('submit_admin', 'Add
Content Admin') ?>
            <?php echo form_reset('reset', 'Reset Form')
?>
        </p>
    </form>
</section>
</article>
</div>

```

```

add_bank_question_view.php
    <div id="main" class="bank" role="main">
        <?php echo validation_errors(); ?>
        <?php echo form_open("add_question_process"); ?>
        <p><label for="question_type">Type of Question:</label>
            <input type="radio" name="question_type" value="true_false" />
            <label for="true_false">True or False</label>
            <input type="radio" name="question_type" value="multiple_choice" />
            <label for="multiple_choice">Multiple Choice</label></p>
        <p><label for="question_name">Name of Question:</label> <input type="text"
name="question_name" size="50" /></p>
        <p><label for="choice1">Choice 1:</label> <input type="text"
name="choice1" size="25" /></p>
        <p><label for="choice2">Choice 2:</label> <input type="text"
name="choice2" size="25" /></p>
        <p>Not needed for True or False:</p>
        <p><label for="choice3">Choice 3:</label> <input type="text"
name="choice3" size="25" /></p>
        <p><label for="choice4">Choice 4:</label> <input type="text"
name="choice4" size="25" /></p>
        <p><label for="correct_answer">Correct Answer (required for all
questions):</label> <input type="text" name="correct_answer" size="25" /></p>
        <p><label for="lesson_id">Lesson Number (required for all
questions):</label> <input type="text" name="lesson_id" size="15" /></p>
        <p><?php echo form_submit('submit_bank', 'Add Question to Bank'); ?>
        <?php echo form_reset('reset_form', 'Reset Form'); ?></p>
    </form>
</div>

    <div id="related" class="sidebar" role="complementary">
    <aside>
        <h4>Help</h4>
        <p>To add a question, make sure that the question doesn't already exist
yet.</p>
        <p>Fill in all necessary fields, making sure that the spelling is correct
for each of these.</p>
        <p>HTML is allowed in the questions and answers.</p>
        <p>After making sure that the fields are complete, click Submit.</p>
    </aside>
</div>

```

add_content_admin_success_view.php

```
<div id="main" role="main">
    <article class="regcomplete_main">
        <h3>Registration is now complete.</h3>
        <h4>Advise the content admin to now log in with their chosen username and
password.</h4>
    </article>
</div>
```

add_content_admin_view.php

```
<div id="main" role="main">
    <article class="signup_main">
        <section id="signup_form">
            <h3>Before adding another content admin, make sure that he/she has
not had an account before. If he/she has forgotten their password, direct them to the Forgot
Password link in the Login screen.</h3>
            <h4>All fields are required.</h4>
            <div id="infoMessage"><?php echo $message;?></div>
            <?php echo form_open('add_content_admin_process'); ?>
            <p>
                <label for="username">
                    Username: <?php echo
form_input('username', set_value('username')); ?>
                </label>
            </p>
            <p>
                <label for="password">
                    Password: <?php echo
form_password('password', set_value('password')); ?>
                </label>
            </p>
            <p>
                <label for="confirm_password">
                    Confirm Password: <?php echo
form_password('confirm_password', set_value('confirm_password')); ?>
                </label>
            </p>
            <p>
                <label for="first_name">
                    First Name: <?php echo
form_input('first_name', set_value('first_name')); ?>
                </label>
            </p>
            <p>
                <label for="last_name">
                    Last Name: <?php echo
form_input('last_name', set_value('last_name')); ?>
                </label>
            </p>
            <p>
                <label for="email">
                    Email Address: <?php echo
form_input('email', set_value('email')); ?>
                </label>
            </p>
            <p>
                <?php echo
form_submit('submit_content_admin', 'Add Content Admin') ?>
                <?php echo form_reset('reset', 'Reset Form')
?>
            </p>
        </form>
    </section>
</article>
</div>
```

add_contributor_success_view.php

```
<div id="main" role="main">
    <article class="regcomplete_main">
```

```

                <h3>Registration is now complete.</h3>
                <h4>Thank you for signing up for a contributor account at FILLES.</h4>
            </article>
        </div>

add_dictionary_entry_view.php
<div id="main" class="guide" role="main">
    <?php echo $error; ?>
    <?php echo form_open_multipart('add_dictionary_process'); ?>
        <input type="hidden" name="MAX_FILE_SIZE" value="2097152" />
        <p>Dictionary Term: <input type="text" name="term" size="100" /> </p>
        <p>Part of Speech: <input type="text" name="part_of_speech" size="75"
    /></p>
        <p>Pronunciation: <input type="text" name="pronunciation" size="60" /></p>
        <p>Dictionary Meaning: <input type="text" name="meaning" size="93" /> </p>
        <p>Category: <input type="text" name="category" size="60" /></p>
        <p>Word Usage: <input type="text" name="usage" size="93" /> </p>
        <p>Link to Image: <input type="file" name="userfile" size="50" /></p>
        <p>Link to Audio Pronunciation: <input type="file" name="userfile1"
size="50" /></p>
        <?php echo form_submit('submit_dictionary','Add Entry To Database'); ?>
        <?php echo form_reset('reset_dictionary','Reset Form'); ?>
    </form>
</div>
<div id="related" class="sidebar" role="complementary">
    <aside>
        <h4>INSTRUCTIONS:</h4>
        <ul>
            <li>The dictionary is meant for basic Filipino terms only. No slang,
scientific or derogatory terms allowed.</li>
            <li>After making sure that the fields are complete, click Submit.</li>
            <li>Tip: Pronunciation is represented by spoken syllables, not IPA symbols
(for compatibility with other browsers and tablets).</li>
            <li>Supported file types: Audio - .wav only (for cross-browser
compatibility); Image - .png, .jpg, .gif</li>
        </ul>
    </aside>
</div>

add_grammar_entry_success_view.php
<div id="main" role="main">
    <article class="main_class">
        <h3>Adding of grammar rule entry successful!</h3>
        <h4>To add more entries, click <a href="add_grammar_entry">here</a>.</h4>
        <h4>Return to <a href="index_staff">home page</a>.</h4>
    </article>
</div>

add_grammar_entry_view.php
<div id="main" class="guide" role="main">
    <?php echo validation_errors(); ?>
    <?php echo form_open('add_grammar_entry_process'); ?>
        <p>Grammar Rule: <input type="text" name="rule" size="100" /> </p>
        <p>Grammar Usage: <input type="text" name="usage" size="100" /> </p>
        <p>Grammar Example: <input type="text" name="example" size="100" /> </p>
        <p>Category: <input type="text" name="category" size="50"> </p>
        <?php echo form_submit('submit_grammar','Add Entry To Database'); ?>
        <?php echo form_reset('reset_grammar','Reset Form'); ?>
    </form>
</div>
<div id="related" class="sidebar" role="complementary">
    <aside>
        <h4>INSTRUCTIONS:</h4>
        <ol>
            <li>To add a grammar rule, make sure that the grammar rule doesn't exist
yet.</li>
            <li>Fill in all fields, making sure that the spelling is correct.</li>
            <li>After making sure that the fields are complete, click Submit.</li>
        </ol>
    </aside>
</div>

```

add_guides_view.php

```
<div id="main" class="guide" role="main">
  <h3>What section of the Guides would you like to add entries to?</h3>
  <ul>
    <li><a href="add_dictionary_entry">Dictionary</a></li>
    <li><a href="add_grammar_entry">Grammar</a></li>
    <li><a href="add_phrasebook_entry">Phrasebook</a></li>
  </ul>
</div>
<div id="related" class="sidebar" role="complementary">
  <aside>
    <h4>INSTRUCTIONS:</h4>
    <ul>
      <li>The Guides in FILLES are specialized. Each Guide has its own set of
fields and requirements.</li>
      <li>Follow the instructions in each guide for guide-specific
information.</li>
    </ul>
  </aside>
</div>
```

add_instruction_view.php

```
<div id="main" class="links" role="main">
  <?php echo form_open('add_instruction/process'); ?>
  <p>Instruction Text: <input type="text" name="text" size="100" /> </p>
  <p>Lesson ID: <input type="text" name="lesson_id" size="15" /> </p>
  <?php echo form_submit('submit_instruction','Add Instruction To
Database'); ?>
  <?php echo form_reset('reset_instruction','Reset Form'); ?>
  <?php echo form_close(); ?>
</div>

<div id="related" class="sidebar" role="complementary">
  <aside>
    <p>Need help coordinating an instruction to a lesson? <a
href="edit_lesson" target="_new">Click here for the lesson list</a>.</p>
  </aside>
</div>
```

add_lesson_view.php

```
<div id="main" class="lesson" role="main">
  <?php echo $error; ?>
  <?php echo form_open_multipart('add_lesson_process'); ?>
  <input type="hidden" name="MAX_FILE_SIZE" value="2097152" />
  <p>Lesson Title: <input type="text" name="title" size="100" /> </p>
  <p>Lesson Description: <input type="text" name="description" size="93" />
</p>
  <p>Content of the Lesson: <input type="file" name="userfile" size="75"
/></p>
  <?php echo form_submit('submit_lesson','Add Lesson To Database'); ?>
  <?php echo form_reset('reset_lesson','Reset Form'); ?>
</form>
</div>
<div id="related" class="sidebar" role="complementary">
  <aside>
    <h4>INSTRUCTIONS:</h4>
    <ul>
      <li>Before submitting, make sure that the lesson is in PDF format before
submitting.</li>
      <li>After making sure that the lesson is complete, click Submit.</li>
    </ul>
  </aside>
</div>
```

add_link_success_view.php

```
<div id="main" role="main">
  <article class="main_class">
    <h3>Link submitted for approval.</h3>
    <h3>Wait for the administrator's email if your link has been approved.</h3>
  </article>
```

```

</div>

add_link_view.php
<div id="main" class="links" role="main">
  <?php echo validation_errors(); ?>
  <?php echo form_open('add_link_process'); ?>
  <p>Link Title: <input type="text" name="title" size="100" /> </p>
  <p>Link Description: <input type="text" name="description" size="93" />
</p>
  <p>URL of the Link: <input type="text" name="url" size="100" /> </p>
  <?php echo $math_captcha_question; ?>
  <?php echo form_input('math_captcha'); ?>
  <br /><br />
  <?php echo form_submit('submit_link','Add Link To Database'); ?>
  <?php echo form_reset('reset_link','Reset Form'); ?>
  <?php echo form_close(); ?>
</div>
<div id="related" class="sidebar" role="complementary">
  <aside>
    <h4>INSTRUCTIONS:</h4>
    <ol>
      <li>To add a link, please input the contents of the link in the
appropriate boxes/fields. The link number is automatically generated, but the contents of that
link stored in the link number can be edited later.</li>
      <li>After making sure that the link is complete and the captcha is
answered, click Submit.</li>
    </ol>
  </aside>
</div>

add_phrasebook_entry_view.php
<div id="main" class="guide" role="main">
  <?php echo $error; ?>
  <?php echo form_open_multipart('add_phrasebook_entry_process'); ?>
  <input type="hidden" name="MAX_FILE_SIZE" value="2097152" />
  <p>Phrase: <input type="text" name="phrase" size="100" /> </p>
  <p>Phrase's Translation (in English): <input type="text"
name="translation" size="93" /> </p>
  <p>Pronunciation: <input type="text" name="pronunciation" size="60" /></p>
  <p>Category: <input type="text" name="category" size="60" /></p>
  <p>Link to Audio Pronunciation: <input type="file" name="userfile"
size="50" /> </p>
  <?php echo form_submit('submit_phrasebook','Add Entry To Database'); ?>
  <?php echo form_reset('reset_phrasebook','Reset Form'); ?>
</form>
</div>
<div id="related" class="sidebar" role="complementary">
  <aside>
    <h4>INSTRUCTIONS:</h4>
    <ol>
      <li>The phrasebook is meant for basic Filipino phrases only. No slang or
advanced phrases please.</li>
      <li>Audio pronunciation, if any, is only supported in .WAV format.</li>
      <li>Fill in all fields, and when you are sure of the correct spelling and
grammar, click Submit.</li>
    </ol>
  </aside>
</div>

add_staff_success_view.php
<div id="main" role="main">
  <article class="main_class">
    <p>Staff member has been added to the list.</p>
    <p>Advise him/her to log in with the given username now.</p>
  </article>
</div>

add_staff_view.php
  <div id="main" role="main">
    <article class="signup_main">
      <section id="signup_form">

```

<h3>Before adding a staff member, make sure that he/she has not had an account before. If he/she has forgotten their password, direct them to the Forgot Password link in the Login screen.</h3>

```

    <h4>All fields are required.</h4>
    <div id="infoMessage"><?php echo $message;?></div>
    <?php echo form_open('add_staff_process'); ?>
        <p>
            <label for="username">
                Username: <?php echo
form_input('username', set_value('username')); ?>
            </label>
        </p>
        <p>
            <label for="password">
                Password: <?php echo
form_password('password', set_value('password')); ?>
            </label>
        </p>
        <p>
            <label for="confirm_password">
                Confirm Password: <?php echo
form_password('confirm_password', set_value('confirm_password')); ?>
            </label>
        </p>
        <p>
            <label for="first_name">
                First Name: <?php echo
form_input('first_name', set_value('first_name')); ?>
            </label>
        </p>
        <p>
            <label for="last_name">
                Last Name: <?php echo
form_input('last_name', set_value('last_name')); ?>
            </label>
        </p>
        <p>
            <label for="email">
                Email Address: <?php echo
form_input('email', set_value('email')); ?>
            </label>
        </p>
        <?php echo form_submit('submit_staff', 'Add
Staff') ?>
        <?php echo form_reset('reset', 'Reset Form')
?>
    </p>
</form>
</section>
</article>
</div>

```

add_user_group_failure_view.php

```

<div id="main" role="main">
    <article class="main_class">
        <h3>Adding of user group unsuccessful.</h3>
    </article>
</div>

```

add_user_group_success_view.php

```

<div id="main" role="main">
    <article class="main_class">
        <h3>Adding of user group successful.</h3>
    </article>
</div>

```

add_user_group_view.php

```

<div id="main" role="main">
    <article class="signup_main">
        <h4>All fields are required.</h4>

```

```

        <?php echo form_open('add_user_group/process'); ?>
        <p>
            <label for="user_group">
                Group Name: <?php echo
form_input('user_group', set_value('user_group')); ?>
            </label>
        </p>
        <p>
            <label for="group_description">
                Group Description: <?php echo
form_input('group_description', set_value('group_description')); ?>
            </label>
        </p>
        <p>
            <?php echo form_submit('submit_group', 'Add
User Group') ?>
            <?php echo form_reset('reset', 'Reset Form')
?>
        </p>
    </form>
</article>
</div>

<div id="main" role="main">
    <article class="main_class">
        <?php
            $this->db->where('is_approved','0');
            $query = $this->db->get('links');

            if ($query->num_rows() == 0) {
                echo '<h2>No links to be approved.</h2>';
            }
            else {
                echo form_open('approve_links_process');
                $i = 0;
                echo '<table class="link_table" cellspacing="1"
border="1">';

                echo "<th>\n
                    <td>Link Title</td>\n
                    <td>Link Description</td>\n
                    <td>Link URL</td>\n
                    <td>Contributed by</td>\n
                    <td>Approve Link</td>\n
                    <td>Reject Link</td>\n
                    </th>";

                foreach ($query->result() as $row) {
                    $link_title = $row->link_title;
                    echo "<input type='hidden' name='title[$i]'
value='$link_title' />\n";

                    echo "<tr>\n";
                    echo "<td></td>\n";
                    echo "<td>$link_title</td>\n";
                    echo "<td>$row->link_description</td>\n";
                    echo "<td>$row->link_url</td>\n";
                    echo "<td>$row->contributed_by</td>\n";
                    echo "<td><input type='radio'
name='approved[$i]' value='1' /></td>\n";

                    echo "<td><input type='radio'
name='approved[$i]' value='-1' /></td>\n";

                    echo "</tr>";
                    $i++;
                }
                echo "</table>";
                echo "<br>";
                echo "<center>";
                echo form_submit('approve_links', 'Approve Links');
                echo form_reset('reset', 'Reset Form');
                echo "</center>";
                echo '</form>';
            }
        }
    }
}

```



```

        ?>
    </article>
</div>

<div id="related" class="sidebar" role="complementary">
    <aside>
        <h4>To approve a link, click on the first radio button. It will be
reflected in the public link exchange.</h4>
        <h4>To reject a link, click on the second radio button. It will
forever be prevented from public viewing.</h4>
    </aside>
</div>

bank_maintenance_view.php
<div id="main" class="bank" role="main">
    <ul>
        <li><a href="add_bank_question">Add Questions</a></li>
        <li><a href="edit_question">Edit Questions</a></li>
        <li><a href="organize_bank">Organize Instructions</a></li>
        <li><a href="list_delete_question">Delete Questions</a></li>
    </ul>
</div>

<div id="related" class="sidebar" role="complementary">
<aside>
    <h4>Help</h4>
    <p>To add questions to the Bank, click on <b>Add Questions</b>.</p>
    <p>To edit questions in the Bank, click on <b>Edit Questions</b>.</p>
    <p>To maintain the instructions for each lesson in the Bank, click on
<b>Organize Instructions</b>.</p>
    <p>If you need to delete any questions from the Bank, click on <b>Delete
Questions</b>.</p>
</aside>
</div>

change_detail_view.php
<div id="main" role="main">
    <article class="main_class">
        <?php
            if($this->input->get('id')) {
                $change_id = $this->input->get('id');
                $query = $this->db->get_where('changes',array('change_id'
=> $change_id));

                if ($query->num_rows() > 0) {
                    $row = $query->row();
                    $type = $row->type;
                    $sender_id = $row->sender_id;
                    $sender_data = $this->flexi_auth-
>get_user_by_id($sender_id)->row();

                    $sender = $sender_data->userinfo_first_name . " " .
$sender_data->userinfo_last_name . " (" . $sender_data->uacc_username . ")";
                    $content = $row->content;
                    $timestamp = $row->timestamp;

                    echo '<h4>Type: ' . $type . '</h4>';
                    echo '<h4>Sender: ' . $sender . '</h4>';
                    echo '<h4>Date Submitted: ' . $timestamp . '</h4>';

                    echo "<h4>Message: </h4>";
                    echo '<pre>' . $content . '</pre>';

                    echo "<p><a
href='confirm_change_deletion?id=$change_id'>Delete This Change Request</a></p>";
                }
                else {
                    echo '<p>No such change request with that ID
exists.</p>';
                }
            }
            else {
                echo '<p>Parameter missing!</p>';
            }
        }
    </article>
</div>

```

```

        }
    ?>
</article>
</div>

```

change_password_failure_view.php

```

<div id="main" role="main">
    <article class="main_class">
        <p>Password not successfully changed.</p>
    </article>
</div>

```

change_password_success_view.php

```

<div id="main" role="main">
    <article class="main_class">
        <p>Password successfully changed. Log out and log in with the new password to make
sure that it's OK.</p>
    </article>
</div>

```

change_password_view.php

```

<div id="main" role="main">
    <article class="main_class">
        <section id="change_pw_form">
            <h1>Change Your Password</h1>
            <p>To proceed, enter your existing password and a new password (entered
twice).</p>
            <?php echo form_open('change_password_process'); ?>
            <p>Current Password: <?php echo form_password('current_password');
?></p>
            <p>New Password: <?php echo form_password('new_password'); ?></p>
            <p>Confirm New Password: <?php echo
form_password('confirm_new_password'); ?></p>
            <?php echo form_submit('submit_change_pw','Change Password'); ?>
            <?php echo form_close(); ?>
        </section>
    </article>
</div>

```

change_request_sent.php

```

<div id="main" role="main">
    <article class="main_class">
        <p>Change request filed to be read by the SWF Staff.</p>
    </article>
</div>

```

change_request_unsuccessful.php

```

<div id="main" role="main">
    <article class="main_class">
        <p>Sorry, change request was not successfully filed. You can go back one page to
retrieve your request, if in case you need to resend it later.</p>
    </article>
</div>

```

changes_view.php

```

<div id="main" role="main">
    <article class="main_class">
        <table border="1">
            <?php
                if ($this->change_entries->record_count() == 0) {
                    echo '<h2>No changes to review.</h2>';
                }
                else {
                    echo '<thead>';
                    echo '<th>Number</th>';
                    echo "&<th>Sender</th>";
                    echo "<th>Change Type</th>";

```

```

        echo "<th>Delete?</th>";
        echo '</thead>';
        echo '<tbody>';
        foreach($results as $data) {
            $detail_link = site_url('change_detail?id=' . $data-
>change_id);
            $delete_link = site_url('confirm_change_deletion?id=' .
            $data->change_id);
            $sender_identity = $this->flexi_auth->get_user_by_id($data-
>sender_id)->row();
            echo "<tr>
                <td><a href='$detail_link'>$data-
                <td>$sender_identity->uacc_username</td>\n
                <td>$data->type</td>\n
                <td><a href='$delete_link'>Delete</a></td>\n
            </tr>";
        }
        echo '</tbody>';
        echo '</table>';
        ?>
        <p><?php echo $links; ?></p>
    </article>
</div>

```

check_link_status_view.php

```

    <div id="main" role="main">
        <article class="links">
            <table border="1">
                <?php
                    if ($this->link_model->record_count_by_user() == 0)
                    {
                        echo '<h2>No links defined so far...</h2>';
                    }
                    else {
                        echo '<thead>';
                        echo '<th>Link Title</th>';
                        echo '<th>Description</th>';
                        echo '<th class="{sorter:'
false}">URL</th>';
                        echo '<th>Approved*?</th>';
                        echo '<th>Edit Link?</th>';
                        echo '</thead>';
                        echo '<tbody>';
                        foreach($results as $data) {
                            echo "<tr>
                                <td>$data->link_title</td>\n
                                <td>$data-
                                <td>$data->link_url</td>\n
                                <td>$data->is_approved</td>\n
                                <td><a
                                href='edit_link_entry?id=$data->link_id'>Edit</a></td>\n
                                </tr>";
                            }
                        echo '</tbody>';
                    }
                ?>
            </table>
            <p>*Key to approval status: 1 - Approved; 0 - Not yet approved; -1
- Rejected*</p>
        </article>
    </div>

```

confirm_change_deletion_view.php

```

<div id="main" role="main">
    <article class="guide">
        <h3>Are you sure you really want to delete the entry?</h3>
        <p><a href="confirm_change_deletion/delete?id=<?php echo $id; ?>">Yes</a>

```

```

        <a href="changes">No</a></p>
    </article>
</div>

confirm_dictionary_deletion_view.php
<div id="main" role="main">
    <article class="guide">
        <h3>Are you sure you really want to delete the entry?</h3>
        <p><a href="confirm_dictionary_deletion/delete?id=?php echo $id; ?">Yes</a>
            <a href="list_delete_dictionary">No</a></p>
    </article>
</div>

confirm_grammar_deletion_view.php
<div id="main" role="main">
    <article class="guide">
        <h3>Are you sure you really want to delete the entry?</h3>
        <p><a href="confirm_grammar_deletion/delete?id=?php echo $id; ?">Yes</a>
            <a href="list_delete_grammar">No</a></p>
    </article>
</div>

confirm_instruction_deletion_view
<div id="main" role="main">
    <article class="guide">
        <h3>Are you sure you really want to delete the entry?</h3>
        <p><a href="confirm_instruction_deletion/delete?id=?php echo $id; ?">Yes</a>
            <a href="list_delete_instruction">No</a></p>
    </article>
</div>

confirm_link_deletion_view
<div id="main" role="main">
    <article class="guide">
        <h3>Are you sure you really want to delete the entry?</h3>
        <p><a href="confirm_link_deletion/delete?id=?php echo $id; ?">Yes</a>
            <a href="list_delete_link">No</a></p>
    </article>
</div>

confirm_phrase_deletion_view
<div id="main" role="main">
    <article class="guide">
        <h3>Are you sure you really want to delete the entry?</h3>
        <p><a href="confirm_phrase_deletion/delete?id=?php echo $id; ?">Yes</a>
            <a href="list_delete_phrasebook">No</a></p>
    </article>
</div>

confirm_question_deletion_view
<div id="main" role="main">
    <article class="guide">
        <h3>Are you sure you really want to delete the entry?</h3>
        <p><a href="confirm_question_deletion/delete?id=?php echo $id; ?">Yes</a>
            <a href="list_delete_question">No</a></p>
    </article>
</div>

confirmed_deletion_guide_entry.php
<div id="main" role="main">
    <article class="guide">
        <h3>Deletion of guide entry successful.</h3>
        <p>Return to the <a href="guide_maintenance">guide maintenance</a> page.</p>
    </article>
</div>

confirmed_deletion_instruction_view.php
<div id="main" role="main">
    <article class="guide">
        <h3>Deletion of instruction successful.</h3>

```

```

        <p>Return to the <a href="bank_maintenance">bank maintenance</a> page.</p>
    </article>
</div>

confirmed_deletion_question_view.php
<div id="main" role="main">
    <article class="guide">
        <h3>Deletion of question successful.</h3>
        <p>Return to the <a href="bank_maintenance">bank maintenance</a> page.</p>
    </article>
</div>

contributor_signup_view.php
    <div id="main" role="main">
        <article class="signup_main">
            <section id="signup_form">
                <h3>If you have forgotten your password, go back to the
Login screen and request that your password be reset.</h3>
                <h4>All fields are required.</h4>
                <div id="infoMessage"><?php echo $message;?></div>
                <center>
                    <table cellpadding="1">
                        <tr>
                            <th>User Login Information
                            </th>
                            <td>
                                Username:
                            </td>
                        </tr>
                        <tr>
                            <td>
                                <?php echo
form_input('username', set_value('username')); ?>
                            </td>
                        </tr>
                        <tr>
                            <td>
                                Password:
                            </td>
                            <td>
                                <?php echo
form_password('password', set_value('password')); ?>
                            </td>
                        </tr>
                        <tr>
                            <td>
                                Confirm Password:
                            </td>
                            <td>
                                <?php echo
form_password('confirm_password', set_value('confirm_password')); ?>
                            </td>
                        </tr>
                    </table>
                </center>
                <th>User Information</th>
                <tr>
                    <td>
                        First Name:
                    </td>
                    <td>
                        <?php echo
form_input('first_name', set_value('first_name')); ?>
                    </td>
                </tr>
                <tr>
                    <td>
                        Last Name:
                    </td>
                    <td>
                        <?php echo
form_input('last_name', set_value('last_name')); ?>
                    </td>
                </tr>
            </section>
        </article>
    </div>

```

```

        </td>
    </tr>
    <tr>
        <td>
            Email Address:
        </td>
        <td>
            <?php echo
form_input('email', set_value('email')); ?>
        </td>
    </tr>
</tr>
</table>

        <center>
            <?php echo
form_submit('submit_contributor', 'Sign up for a contributor account') ?>
            <?php echo
form_reset('reset', 'Reset Form') ?>
        </center>
    </form>
</center>
</section>
</article>
</div>

delete_error_auth_view.php
    <h4>There was an error in deleting the data.</h4>

delete_error_lesson_view.php
    <h4>There was an error in deleting the data.</h4>
    <p>Return to the <a href="lesson_maintenance">lesson maintenance</a> page.</p>

delete_error_quiz_view.php
    <h4>There was an error in deleting the data.</h4>
    <p>Return to the <a href="bank_maintenance">bank maintenance</a> page.</p>

delete_error_view.php
    <h4>There was an error in deleting the data.</h4>

delete_guides_view.php
<div id="main" class="guide" role="main">
    <h3>What section of the Guides would you like to delete entries from?</h3>
    <ul>
        <li><a href="list_delete_dictionary">Dictionary</a></li>
        <li><a href="list_delete_grammar">Grammar</a></li>
        <li><a href="list_delete_phrasebook">Phrasebook</a></li>
    </ul>
</div>
<div id="related" class="sidebar" role="complementary">
    <aside>
        <h4>INSTRUCTIONS:</h4>
        <ul>
            <li>The Guides in FILLES are specialized. Each Guide has its own set of
fields and requirements.</li>
            <li>Follow the instructions in each guide for guide-specific
information.</li>
        </ul>
    </aside>
</div>

delete_lesson_view.php
    <div id="main" role="main">
        <article class="lesson">
            <table border="1">
                <?php
                    if ($this->lesson_entries->record_count() == 0) {
                        echo '<h2>No lessons defined so
far...</h2>';
                    }
                else {

```

```

        echo '<thead>';
        echo '<th>Lesson Name</th>';
        echo '<th>Delete Entry?</th>';
echo '</thead>';
echo '<tbody>';
foreach($results as $data) {
    echo "<tr>
        <td>$data-
>lesson_title</td>\n
        <td><a
href='lesson_delete?id=$data->lesson_id'>Delete</a></td>\n
        </tr>";
    }
echo '</tbody>';
}
}
?>
</table>
<p><?php echo $links; ?></p>
</article>
</div>

<div id="related" class="sidebar" role="complementary">
    <aside>
        <h4>To delete a lesson from the list, click on the "Delete" link of
that row where the entry is located.</h4>
    </aside>
</div>

delete_user_group_view.php
<div id="main" role="main">
    <article class="users">
        <h3>Are you sure you really want to delete the entry?</h3>
        <p><a href="delete_user_group/confirm?id=<?php echo $id; ?>">Yes</a>
        <a href="maintain_user_groups">No</a></p>
    </article>
</div>

delete_user_view.php
<div id="main" role="main">
    <article class="users">
        <h3>Are you sure you really want to delete the entry?</h3>
        <p><a href="delete_user/confirm?id=<?php echo $id; ?>">Yes</a>
        <a href="maintain_users">No</a></p>
    </article>
</div>

dictionary_index_view.php
<div id="main" role="main">
    <article class="guide">
        <h3>To view the dictionary entries starting with that particular letter, select a
letter below.</h3>
        <p>
            <?php
                $range = range('A','Z');
                foreach ($range as $char) {
                    echo "<a href=' " . site_url("dictionary/view_entries/" .
$char) . "'> " . "$char</a> ";
                }
            ?>
        </p>
    </article>
</div>

dictionary_view.php
    <div id="main" role="main">
        <article class="guide">
            <table class="tablesorter" cellspacing="3">

```

```

<?php
if ($this->dictionary_entries->record_count() == 0) {
    echo '<h2>No dictionary terms defined so
far...</h2>';
}
else {
    echo '<thead>';
    echo '<th>Dictionary Term</th>';
    echo '<th>Part of Speech</th>';
    echo '<th class="{sorter: false}">Syllabic
Pronunciation</th>';
    echo '<th class="{sorter: false}">Meaning</th>';
    echo '<th class="{sorter: false}">Usage</th>';
    echo '<th>Category</th>';
    echo '<th>Image</th>';
    echo '<th>Audio Link</th>';
    echo '</thead>';
    echo '<tbody>';
    foreach($results as $data) {
        $dictionary_link =
site_url('view_dictionary_entry?id=' . $data->dictionary_id);
        if ($data->dictionary_image_link) {
            echo "<tr>
                <td><a href='{$dictionary_link}'>$data-
>dictionary_term</a></td>\n
                <td>$data-
>dictionary_partofspeech</td>\n
                <td>$data-
>dictionary_pronunciation</td>\n
                <td>$data->dictionary_meaning</td>\n
                <td>$data->dictionary_usage</td>\n
                <td>$data->dictionary_category</td>\n
                <td><a href='{$data-
>dictionary_image_link'><img src='{$data->dictionary_image_link' width='25%' height='25%'
border='1' /></a></td>\n
                <td>N/A</td>\n
            </tr>";
        }
        else if ($data->dictionary_audio_link) {
            echo "<tr>
                <td><a href='{$dictionary_link}'>$data-
>dictionary_term</a></td>\n
                <td>$data-
>dictionary_partofspeech</td>\n
                <td>$data-
>dictionary_pronunciation</td>\n
                <td>$data->dictionary_meaning</td>\n
                <td>$data->dictionary_usage</td>\n
                <td>$data->dictionary_category</td>\n
                <td>N/A</td>\n
                <td><audio controls>
                    <source src='{$data-
>dictionary_audio_link'>
                    <embed src='{$data-
>dictionary_audio_link'>
                    </audio></td>\n
            </tr>";
        }
        else if (($data->dictionary_image_link) && ($data-
>dictionary_audio_link)) {
            echo "<tr>
                <td><a href='{$dictionary_link}'>$data-
>dictionary_term</a></td>\n
                <td>$data-
>dictionary_partofspeech</td>\n
                <td>$data-
>dictionary_pronunciation</td>\n
                <td>$data->dictionary_meaning</td>\n
                <td>$data->dictionary_usage</td>\n
            </tr>";
        }
    }
}

```



```

                                <td>${data->dictionary_category}</td>\n
                                <td><a href='${data-
>dictionary_image_link'><img src='${data->dictionary_image_link' width='25%' height='25%'
border='1' /></a></td>\n
                                <td><audio controls>
                                <source src='${data-
>dictionary_audio_link'>
                                <embed src='${data-
>dictionary_audio_link'>
                                </audio></td>\n
                                </tr>";
                                }
                                else {
                                echo "<tr>
                                <td><a href='${dictionary_link}'>${data-
>dictionary_term</a></td>\n
                                <td>${data-
>dictionary_partofspeech</td>\n
                                <td>${data-
>dictionary_pronunciation</td>\n
                                <td>${data->dictionary_meaning}</td>\n
                                <td>${data->dictionary_usage}</td>\n
                                <td>${data->dictionary_category}</td>\n
                                <td>N/A</td>\n
                                <td>N/A</td>\n
                                </tr>";
                                }
                                }
                                echo '</tbody>';
                                }
                                ?>
                                </table>
                                <p><?php echo $links; ?></p>
                                <p>
                                <?php
                                $range = range('A','Z');
                                foreach ($range as $char) {
                                echo "<a href=' " .
site_url("dictionary/view_entries/" . $char) . "'> " . "$char</a> ";
                                }
                                ?>
                                </p>
                                </article>
                                </div>

```

differences_view.php

```

<div id="main" role="main">
  <article class="difference">
    <h3>Are you still unsure what will be your role in the system?</h3>
    <p>This table will help you decide.</p>
    <table border="1" cellspacing="2" cellpadding="5">
      <thead>
        <th></th>
        <th>Users</th>
        <th>Contributors</th>
      </thead>
      <tbody>
        <tr>
          <td></td>
          <td>Role in the system</td>
          <td>Use the system and learn</td>
          <td>Can only contribute links or suggest
guide changes</td>
        </tr>
        <tr>
          <td></td>
          <td>Can suggest guide changes</td>
          <td></td>
          <td></td>
        </tr>
        <tr>
          <td></td>
          <td></td>
          <td></td>
          <td></td>
        </tr>
      </tbody>
    </table>
  </article>
</div>

```

```

and quizzes)</td>
<td>Can utilize the system proper (lessons
<td></td>
<td></td>
</tr>
<tr>
the learners</td>
<td>Can add links for further resources by
<td></td>
<td></td>
</tr>
<tr>
<td>Can view guides</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cost to join</td>
<td>Free</td>
<td>Free</td>
</tr>
<tr>
<td>Better for first-timers?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Better for fluent and daily users?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
<p><a href="user_signup">Sign up for a user account now!</a><br />
<a href="contributor_signup">Sign up for a contributor account
now!</a></p>
</div>
</article>
</div>

```

edit_dictionary_view.php

```

<div id="main" role="main">
<article class="guide_edit">
<?php
if($this->input->get('id')) {
$dictionary_id = $this->input->get('id');
$query = $this->db-
>get_where('dictionary',array('dictionary_id' => $dictionary_id));

if ($query->num_rows() > 0) {
$row = $query->row();
$term = $row->dictionary_term;
$partofspeech = $row-

$pronunciation = $row-

$meaning = $row->dictionary_meaning;
$usage = $row->dictionary_usage;
$category = $row->dictionary_category;

$term_form_data = array(

'name' => 'term',

'value' => $term,

'size' => '100',

'style' => 'width:50%',

```

```

);

'name' => 'partofspeech',
'value' => $partofspeech,
'size' => '100',
'style' => 'width:50%',
);

$part_of_speech_form_data = array(

'name' => 'pronunciation',
'value' => $pronunciation,
'size' => '100',
'style' => 'width:50%',
);

$pronunciation_form_data = array(

'name' => 'meaning',
'value' => $meaning,
'size' => '100',
'style' => 'width:50%',
);

$meaning_form_data = array(

'name' => 'usage',
'value' => $usage,
'size' => '100',
'style' => 'width:50%',
);

$usage_form_data = array(

'name' => 'category',
'value' => $category,
'size' => '100',
'style' => 'width:50%',
);

$category_form_data = array(

form_open_multipart("edit_dictionary_process");

echo
echo form_hidden('MAX_FILE_SIZE','2097152');
echo '<p>Term: ';
echo form_input($term_form_data);
echo '</p>';
echo '<p>Part Of Speech: ';
echo form_input($part_of_speech_form_data);
echo '</p>';
echo '<p>Pronunciation: ';
echo form_input($pronunciation_form_data);
echo '</p>';
echo '<p>Meaning: ';

```

```

        echo form_input($meaning_form_data);
        echo '</p>';
        echo '<p>Usage: ';
        echo form_input($usage_form_data);
        echo '</p>';
        echo '<p>Image Link: ';
        echo form_upload('userfile');
        echo '</p>';

        echo '<p>Audio Pronunciation Link: ';
        echo form_upload('userfile1');
        echo '</p>';

        echo '<p>Category: ';
        echo form_input($category_form_data);
        echo '</p>';
        echo
form_submit('submit_dictionary_revision', 'Finalize Edits');
        echo
form_reset('cancel_dictionary_revision', 'Cancel Revision');
        echo
form_hidden('dictionary_id',$dictionary_id);
    }
    else {
        $this->load->view('no_record_view');
    }
}
else {
    $this->load->view('parameter_missing_view');
}
?>
</article>
</div>

<div id="related" class="sidebar" role="complementary">
    <aside>
        <center>
            <ul>
                <li>To edit the dictionary entry, make sure that you
have at least one necessary modification for it to succeed.</li>
                <li>Tip: Pronunciation is represented by spoken
syllables, not IPA symbols (for compatibility with other browsers and tablets).</li>
                <li>Supported file types: Audio - .wav only (for
cross-browser compatibility); Image - .png, .jpg, .gif</li>
                <li>After making sure that you have changed the
entry, click Finalize Edits.</li>
            </ul>
        </center>
    </aside>
</div>

edit_grammar_rule_view.php
    <div id="main" role="main">
        <article class="guide_edit">
            <?php
                if($this->input->get('id')) {
                    $grammar_id = $this->input->get('id');
                    $query = $this->db-
>get_where('grammar',array('grammar_id' => $grammar_id));

                    if ($query->num_rows() > 0) {
                        $row = $query->row();
                        $rule = $row->grammar_rule;
                        $usage = $row->grammar_usage;
                        $example = $row->grammar_example;
                        $category = $row->grammar_category;

                        $rule_form_data = array(

'name' => 'rule',

```

```

'value' => $rule,
'size' => '100',
'style' => 'width:50%',
);

'name' => 'usage',
'value' => $usage,
'size' => '100',
'style' => 'width:50%',
);

'name' => 'example',
'value' => $example,
'size' => '100',
'style' => 'width:50%',
);

'name' => 'category',
'value' => $category,
'size' => '100',
'style' => 'width:50%',
);

echo form_open("edit_grammar_rule_process");
echo '<p>Term: ';
echo form_input($rule_form_data);
echo '</p>';
echo '<p>Usage: ';
echo form_input($usage_form_data);
echo '</p>';
echo '<p>Example: ';
echo form_input($example_form_data);
echo '</p>';
echo '<p>Category: ';
echo form_input($category_form_data);
echo '</p>';

echo form_submit('submit_grammar_revision',
echo form_reset('cancel_grammar_revision',
echo form_hidden('grammar_id',$grammar_id);
}
else {
    $this->load->view('no_record_view');
}
}
else {
    $this->load->view('parameter_missing_view');
}
}
?>

'Finalize Edits');
'Cancel Revision');

```

```

        </article>
    </div>

    <div id="related" class="sidebar" role="complementary">
        <aside>
            <center>
                <ul>
                    <li>To edit the grammar rule entry, make sure that
you have at least one necessary modification for it to succeed.</li>
                    <li>After making sure that you have changed the
entry, click Finalize Edits.</li>
                </ul>
            </center>
        </aside>
    </div>
edit_grammar_view.php
    <div id="main" role="main">
        <article class="guide">
            <table class="tablesorter">
                <?php
                    if ($this->grammar_entries->record_count() == 0) {
                        echo '<h2>No grammar rules defined so
far...</h2>';
                    }
                    else {
                        echo '<thead>';
                        echo '<th>Grammar Rule</th>';
                        echo "&<th>Rule Usage</th>";
                        echo '<th>Example of the Rule in Use</th>';
                        echo '<th>Rule
Category/Classification</th>';

                        echo '<th>Edit Rule</th>';
                        echo '</thead>';
                        echo '<tbody>';
                        foreach($results as $data) {
                            $edit_link =
site_url('edit_grammar_rule?id=' . $data->grammar_id);
                            echo "<tr>
                                <td>$data-
                                <td>$data-
                                <td>$data-
                                <td>$data-
                                <td><a
                                </tr>";
                            }
                        echo '</tbody>';
                        echo '</table>';
                    }
                <?php echo $links; ?></p>
            </article>
        </div>

edit_guides_view.php
    <div id="main" class="guide" role="main">
        <h3>What section of the Guides would you like to edit entries in?</h3>
        <ul>
            <li>Dictionary: <a href="search_dictionary_entry">Search</a> <a
href="list_dictionary_entry">List</a> </li>
            <li><a href="edit_grammar">Grammar</a></li>
            <li>Phrasebook: <a href="search_phrasebook_entry">Search</a> <a
href="list_phrasebook_entry">List</a></li>
        </ul>
    </div>
    <div id="related" class="sidebar" role="complementary">

```

```

        <aside>
            <h4>INSTRUCTIONS:</h4>
            <ul>
                <li>The Guides in FILLES are specialized. Each Guide has its own set of
fields and requirements.</li>
                <li>Follow the instructions in each guide for guide-specific
information.</li>
            </ul>
        </aside>
    </div>

edit_instruction_entry_view.php
    <div id="main" role="main">
        <article class="quiz">
            <?php
                if($this->input->get('id')) {
                    $instruction_id = $this->input->get('id');
                    $query = $this->db-
>get_where('bank_instructions',array('instruction_id' => $instruction_id));

                    if ($query->num_rows() > 0) {
                        $row = $query->row();
                        $text = $row->instruction_text;
                        $lesson_id = $row->lesson_id;

                        $text_form_data = array(

                            'name' => 'text',

                            'value' => $text,

                            'size' => '100',

                            'style' => 'width:50%',

                        );

                        $lessonid_form_data = array(

                            'name' => 'lesson_id',

                            'value' => $lesson_id,

                            'size' => '15',

                            'style' => 'width:10%'

                        );

                        echo form_open("edit_instruction_entry/process");
                        echo '<p>Text: ';
                        echo form_input($text_form_data);
                        echo '</p>';
                        echo '<p>Lesson ID: ';
                        echo form_input($lessonid_form_data);
                        echo '</p>';
                        echo form_submit('submit_instruction_revision',

'Finalize Edits');

                        echo form_reset('cancel_instruction_revision',

'Cancel Revision');

                        echo form_hidden('instruction_id',$instruction_id);

                        $delete_url =
site_url('confirm_instruction_deletion?id=' . $instruction_id);
                        echo "<a href='$delete_url'>Delete Instruction</a>";
                    }
                    else {
                        $this->load->view('no_record_view');
                    }
                }
            }
            else {
                $this->load->view('parameter_missing_view');
            }
        }
    }

```

```

    }
    ?>
  </article>
</div>

<div id="related" class="sidebar" role="complementary">
  <aside>
    <center>
      <ul>
        <li>To edit the instruction, make sure that you have
at least one necessary modification for it to succeed.</li>
        <li>After making sure that you have changed the
entry, click Finalize Edits.</li>
      </ul>
    </center>
  </aside>
</div>

```

edit_instruction_view.php

```

<div id="main" role="main">
  <article class="quiz">
    <table border="1">
      <?php
        if ($this->instruction_entries->record_count() == 0)
        {
          echo '<h2>No instructions defined so
far...</h2>';
        }
        else {
          echo '<thead>';
          echo '<th>Instruction Text</th>';
          echo '<th>Edit Instruction</th>';
          echo '</thead>';
          echo '<tbody>';
          foreach($results as $data) {
            $edit_link =
            site_url('edit_instruction_entry?id=' . $data->instruction_id);
            echo "<tr>
              <td>$data-
              <td><a
              </tr>";
          }
          echo '</tbody>';
        }
      <?php echo $links; ?></p>
    </table>
  </article>
</div>

```

edit_lesson_failure_view.php

```

<div id="main" role="main">
  <article class="main_class">
    <h4>Either you did not change the requested data, or there was an error in
updating the data.</h4>
    <p>Return to the <a href="lesson_maintenance">lesson maintenance</a> page.</p>
  </article>
</div>

```

edit_lesson_proper_view.php

```

<div id="main" role="main">
  <article class="lesson">
    <?php
      if($this->input->get('id')) {

```



```

        $lesson_id = $this->input->get('id');
        $query = $this->db-
>get_where('lessons',array('lesson_id' => $lesson_id));

        if ($query->num_rows() > 0) {
            $row = $query->row();
            $title = $row->lesson_title;
            $description = $row->lesson_description;

            $title_form_data = array(

                'name' => 'title',

                'value' => $title,

                'size' => '100',

                'style' => 'width:50%',

            );

            $description_form_data = array(

                'name' => 'description',

                'value' => $description,

                'size' => '100',

                'style' => 'width:50%',

            );

            form_open_multipart('edit_lesson_process');

            echo

            echo form_hidden('MAX_FILE_SIZE','2097152');
            echo '<p>Title: ';
            echo form_input($title_form_data);
            echo '</p>';
            echo '<p>Description: ';
            echo form_input($description_form_data);
            echo '</p>';
            echo '<p>Content: ';
            echo form_upload('userfile');
            echo '</p>';
            echo form_submit('submit_lesson_revision',

            echo form_reset('cancel_lesson_revision',

            echo form_hidden('lesson_id',$lesson_id);
        }
        else {
            $this->load->view('no_record_view');
        }
    }
    else {
        $this->load->view('parameter_missing_view');
    }
}
?>
</article>
</div>

<div id="related" class="sidebar" role="complementary">
    <aside>
        <center>
            <ul>
                <li>To edit the lesson, make sure that you have at
least one necessary modification for it to succeed.</li>
                <li>After making sure that you have changed the
entry, click Finalize Edits.</li>
            </ul>
        </center>

```

```

        </aside>
    </div>

edit_lesson_view.php
    <div id="main" role="main">
        <article class="lesson">
            <table border="1">
                <?php
                    if ($this->lesson_entries->record_count() == 0) {
                        echo '<h2>No lessons defined so
far...</h2>';
                    }
                    else {
                        echo '<thead>';
                        echo '<th>Lesson Name</th>';
                        echo "&<th class>Lesson Description</th>";
                        echo '<th class>Edit Lesson</th>';
                        echo '</thead>';
                        echo '<tbody>';
                        foreach($results as $data) {
                            $edit_link =
site_url('edit_lesson_proper?id=' . $data->lesson_id);
                            echo "<tr>
                                <td>$data->
lesson_title</td>\n
                                <td>$data->
lesson_description</td>\n
                                <td><a
href='$edit_link'>Edit</a></td>\n
                                </tr>\n";
                        }
                    }
                </table>
                <p><?php echo $links; ?></p>
            </article>
        </div>

edit_link_entry_view.php
    <div id="main" role="main">
        <article class="lesson">
            <?php
                if($this->input->get('id')) {
                    $link_id = $this->input->get('id');
                    $query = $this->db->
get_where('links',array('link_id' => $link_id));

                    if ($query->num_rows() > 0) {
                        $row = $query->row();
                        $title = $row->link_title;
                        $description = $row->link_description;
                        $url = $row->link_url;
                        $contributor = $row->contributed_by;

                        $title_form_data = array(

                            'name' => 'title',

                            'value' => $title,

                            'size' => '100',

                            'style' => 'width:50%',

                        );

                        $description_form_data = array(

                            'name' => 'description',

                            'value' => $description,

```

```

'size' => '100',
'style' => 'width:50%',
);

$url_form_data = array(

'name' => 'url',
'value' => $url,
'size' => '100',
'style' => 'width:100%'
);

echo form_open("edit_link_process");
echo '<p>Title: ';
echo form_input($title_form_data);
echo '</p>';
echo '<p>Description: ';
echo form_input($description_form_data);
echo '</p>';
echo '<p>URL: ';
echo form_input($url_form_data);
echo '</p>';
echo form_submit('submit_link_revision',

echo form_reset('cancel_link_revision',

echo form_hidden('link_id',$link_id);

$current_user = $this->flexi_auth-

if ($current_user === $contributor) {
    echo '<a
href="confirm_link_deletion?id=$link_id">Delete Link</a>';
    }
else {
    $this->load->view('no_record_view');
}
else {
    $this->load->view('parameter_missing_view');
}
?>
</article>
</div>

<div id="related" class="sidebar" role="complementary">
<aside>
<center>
<ul>
<li>To edit the link, make sure that you have at
least one necessary modification for it to succeed.</li>
<li>After making sure that you have changed the
entry, click Finalize Edits.</li>
</ul>
</center>
</aside>
</div>

edit_link_failure_view.php
<div id="main" role="main">
<article class="main_class">
<h3>Sorry, there was an error in editing the link.</h3>
</article>
</div>

```

```

edit_link_view.php
    <div id="main" role="main">
        <article class="links">
            <table border="1">
                <?php
                    if ($this->link_model->record_count_by_user() == 0)
                    {
                        echo '<h2>No links defined so far...</h2>';
                    }
                    else {
                        echo '<thead>';
                        echo '<th>Link Title</th>';
                        echo '<th>Description</th>';
                        echo '<th class="{sorter:'
false}">URL</th>';
                        echo '<th>Edit Link?</th>';
                        echo '</thead>';
                        echo '<tbody>';
                        foreach($results as $data) {
                            echo "<tr>
                                <td>$data->link_title</td>\n
                                <td>$data-
>link_description</td>\n
                                <td>$data->link_url</td>\n
                                <td><a
href='edit_link_entry?id=$data->link_id'>Edit</a></td>\n
                                </tr>";
                            }
                        echo '</tbody>';
                    }
                ?>
            </table>
            <p><?php echo $links; ?></p>
        </article>
    </div>

    <div id="related" class="sidebar" role="complementary">
        <aside>
            <h4>To edit a link, click on the "Edit" link of that row where the
entry is located.</h4>
        </aside>
    </div>
edit_phrase_view.php
    <div id="main" role="main">
        <article class="guide_edit">
            <?php
                echo $error;
                if($this->input->get('id')) {
                    $phrase_id = $this->input->get('id');
                    $query = $this->db-
>get_where('phrasebook',array('phrasebook_id' => $phrase_id));

                    if ($query->num_rows() > 0) {
                        $row = $query->row();
                        $phrase = $row->phrasebook_phrase;
                        $pronunciation = $row-
>phrasebook_pronunciation;

                        $translation = $row->phrasebook_translation;
                        $category = $row->phrasebook_category;

                        $phrase_form_data = array(

                            'name' => 'phrase',
                            'value' => $phrase,
                            'size' => '100',
                            'style' => 'width:50%',

```

```

);
    'name' => 'pronunciation',
    'value' => $pronunciation,
    'size' => '100',
    'style' => 'width:50%',
);
    'name' => 'translation',
    'value' => $translation,
    'size' => '100',
    'style' => 'width:50%',
);
    'name' => 'category',
    'value' => $category,
    'size' => '60',
    'style' => 'width:25%',
);

form_open_multipart("edit_phrase_process");
    echo
    echo form_hidden('MAX_FILE_SIZE','2097152');
    echo '<p>Phrase: ';
    echo form_input($phrase_form_data);
    echo '</p>';
    echo '<p>Pronunciation: ';
    echo form_input($pronunciation_form_data);
    echo '</p>';

    echo '<p>Translation: ';
    echo form_input($translation_form_data);
    echo '</p>';

    echo '<p>Category: ';
    echo form_input($category_form_data);
    echo '</p>';
    echo '<p>Link to Audio Pronunciation: ';
    echo form_upload('userfile');
    echo '</p>';
    echo
form_submit('submit_phrasebook_revision', 'Finalize Edits');
    echo
form_reset('cancel_phrasebook_revision', 'Cancel Revision');
    echo form_hidden('phrase_id',$phrase_id);
    }
    else {
        $this->load->view('no_record_view');
    }
}
else {
    $this->load->view('parameter_missing_view');
}
?>
</article>

```

```

</div>

<div id="related" class="sidebar" role="complementary">
  <aside>
    <center>
      <ul>
        <li>To edit the phrase, make sure that you have at
least one necessary modification for it to succeed.</li>
        <li>Audio pronunciation, if any, is only supported
in .WAV format.</li>
        <li>After making sure that you have changed the
phrase, click Finalize Edits.</li>
      </ul>
    </center>
  </aside>
</div>

```

edit_question_proper_view.php

```

<div id="main" role="main">
  <article class="guide_edit">
    <?php
      if($this->input->get('id')) {
        $question_no = $this->input->get('id');
        $query = $this->db-
>get_where('question_bank',array('question_no' => $question_no));

        if ($query->num_rows() > 0) {
          $row = $query->row();
          $title = $row->question_title;
          $type = $row->question_type;
          $choice_a = $row->choice_a;
          $choice_b = $row->choice_b;
          $choice_c = $row->choice_c;
          $choice_d = $row->choice_d;
          $choice_correctanswer = $row-
>correct_answer;

          $lesson_id = $row->lesson_id;

          $title_form_data = array(

            'name' => 'title',
            'value' => $title,
            'size' => '100',
            'style' => 'width:50%',
          );

          $type_form_data = array(

            'name' => 'type',
            'value' => $type,
            'size' => '75',
            'style' => 'width:50%',
          );

          $choicea_form_data = array(

            'name' => 'choice_a',
            'value' => $choice_a,
            'size' => '75',
            'style' => 'width:30%',
          );

```

```

        'name' => 'choice_b',
        'value' => $choice_b,
        'size' => '75',
        'style' => 'width:30%',
    );

    'name' => 'choice_c',
    'value' => $choice_c,
    'size' => '75',
    'style' => 'width:30%',
);

    'name' => 'choice_d',
    'value' => $choice_d,
    'size' => '75',
    'style' => 'width:30%',
);

    'name' => 'correct_answer',
    'value' => $choice_correctanswer,
    'size' => '75',
    'style' => 'width:30%',
);

    'name' => 'lesson_id',
    'value' => $lesson_id,
    'size' => '25',
    'style' => 'width:10%'
);

only): '

$choiceb_form_data = array(

$choicec_form_data = array(

$choiced_form_data = array(

$correctanswer_form_data = array(

$lessonid_form_data = array(

echo form_open("edit_question_process");
echo '<p>Title: ' ;
echo form_input($title_form_data);
echo '</p>';
echo '<p>Type (multiple_choice or true_false

echo form_input($type_form_data);
echo '</p>';
echo '<p>Choice A: ' ;
echo form_input($choicea_form_data);
echo '</p>';
echo '<p>Choice B: ' ;
echo form_input($choiceb_form_data);
echo '</p>';
echo '<p>Choice C: ' ;

```

```

        echo form_input($choicec_form_data);
        echo '</p>';
        echo '<p>Choice D: ';
        echo form_input($choiced_form_data);
        echo '</p>';
        echo '<p>Correct Answer: ';
        echo form_input($correctanswer_form_data);
        echo '</p>';
        echo '<p>Lesson where the question will be

shown: ';

        echo form_input($lessonid_form_data);
        echo '</p>';
        echo form_submit('submit_question_revision',

'Finalize Edits');

        echo form_reset('cancel_question_revision',

'Cancel Revision');

        echo

form_hidden('question_no',$question_no);
    }
    else {
        $this->load->view('no_record_view');
    }
}
else {
    $this->load->view('parameter_missing_view');
}
?>
</article>
</div>

<div id="related" class="sidebar" role="complementary">
    <aside>
        <center>
            <ul>
                <li>To edit the question, make sure that you have at
least one necessary modification for it to succeed.</li>
                <li>HTML is allowed in the questions and
answers.</li>
                <li>After making sure that you have changed the
entry, click Finalize Edits.</li>
            </ul>
        </center>
    </aside>
</div>

edit_question_unsuccessful_view.php
<div id="main" role="main">
    <article class="main_class">
        <h4>Either you did not change the requested data, or there was an error in
updating the data.</h4>
        <p>Return to the <a href="bank_maintenance">bank maintenance</a> page.</p>
    </article>
</div>

edit_question_view.php
    <div id="main" role="main">
        <article class="quiz">
            <table border="1">
                <?php
                    if ($this->question_entries->record_count() == 0) {
                        echo '<h2>No questions defined so
far...</h2>';
                    }
                    else {
                        echo '<thead>';
                        echo '<th>Question Title</th>';
                        echo '</thead>';
                        echo '<tbody>';

```



```

                                foreach($results as $data) {
                                $edit_link =
site_url('view_question?id=' . $data->question_no);
                                echo "<tr>
                                <td><a
href='$edit_link'>$data->question_title</a></td>\n
                                </tr>";
                                }
                                }
                                echo '</tbody>';
                                ?>
                                </table>
                                <p><?php echo $links; ?></p>
                                </article>
                                </div>

edit_unsuccessful_view.php
<div class="guide">
    <h4>Either you did not change the requested data, or there was an error in updating the
data.</h4>
    <p>Return to the <a href="guide_maintenance">guide maintenance</a> page.</p>
</div>

footer.php
</div>
</body>
</html>

forgot_password.php
    <div id="main" role="main">
        <article class="forgot_main">
            <section id="forgot_form">
                <h1>Forgot Password</h1>
                <p>Please enter your email address or username so we can
send you an email to reset your password.</p>

                <div id="infoMessage"><?php echo $message;?></div>
                <?php echo form_open("forgot_password_process");?>

                <p>
                    Email Address or Username: <br />
                    <?php echo form_input($identity);?>
                </p>

                <p><?php echo form_submit('submit', 'Submit');?></p>

                <?php echo form_close();?>
            </section>
        </article>
    </div>

forgotten_password_success_view.php
    <div id="main" role="main">
        <article class="forgot_main">
            <h3>Password Reset</h3>
            <p>Please check your email for instructions on how to reset your
password.</p>
        </article>
    </div>

grammar_index_view.php
<div id="main" role="main">
    <article class="guide">
        <h3>To view the grammar rules starting with that particular letter, select a
letter below.</h3>
        <p>
            <?php
                $range = range('A','Z');
                foreach ($range as $char) {

```

```

    $char) . "'> " . "$char</a> ";
    }
    ?>
    </p>
  </article>
</div>

grammar_view.php
  <div id="main" role="main">
    <article class="guide">
      <table class="tablesorter" cellspacing="3">
        <?php
          if ($this->grammar_entries->record_count() == 0) {
            echo '<h2>No grammar rules defined so
far...</h2>';
          }
          else {
            echo '<thead>';
            echo '<th>Grammar Rule</th>';
            echo '<th>Rule Usage</th>';
            echo '<th>Example of the Rule in Use</th>';
            echo '<th>Rule
Category/Classification</th>';

            echo '</thead>';
            echo '<tbody>';
            foreach($results as $data) {
              $view_link =
site_url('view_grammar?id=' . $data->grammar_id);
              echo "<tr>
                <td><a
href='$view_link'$>$data->grammar_rule</a></td>\n
                <td>$data-
>grammar_usage</td>\n
                <td>$data-
>grammar_example</td>\n
                <td>$data-
>grammar_category</td>\n
                </tr>";
            }
            echo '</tbody>';
          }
          echo '</table>';
        ?>
        <p><?php echo $links; ?></p>
        <p>
          <?php
            $range = range('A','Z');
            foreach ($range as $char) {
              echo "<a href=' " .
site_url("dictionary/view_entries/" . $char) . "'> " . "$char</a> ";
            }
          ?>
        </p>
      </article>
    </div>

guide_maintenance_view.php
  <div id="main" class="guide" role="main">
    <ul>
      <li><a href="add_guides">Add Guide Contents</a></li>
      <li><a href="edit_guides">Edit Guide Contents</a></li>
      <li><a href="delete_guides">Delete Guide Contents</a></li>
      <li><a href="guides">View Guides</a></li>
    </ul>
  </div>

  <div id="related" class="sidebar" role="complementary">
  <aside>
    <h4>Help</h4>
    <p>To add guides, click on <b>Add Guide Contents</b>.</p>
  </aside>
  </div>

```

```

                <p>To edit guides, click on <b>Edit Guide Contents</b>.</p>
                <p>If you need to delete any content from the Guides, click on <b>Delete
Guide Contents</b>.</p>
            </aside>
        </div>

```

guides_view.php

```

<div id="main" role="main">
    <article class="main_class">
        <h2>Which of the three guides would you like to view?</h2>
        <ul>
            <li>Dictionary: <a href="search_dictionary_entry">Search</a> <a
href="dictionary">List</a></li>
            <li>Grammar: <a href="search_grammar_entry">Search</a> <a
href="grammar">List</a></li>
            <li>Phrasebook: <a href="search_phrasebook_entry">Search</a> <a
href="phrasebook">List</a></li>
        </ul>
    </article>
</div>

```

header.php

```

<html lang="en">
<head>
    <title>FILLES: <?php echo $pageTitle; ?></title>
    <link rel="stylesheet" href="/filles/css/filles.css" />
    <link rel="shortcut icon" href="/filles/images/favicon.ico" type="image/x-icon">
    <link rel="icon" href="/filles/images/favicon.ico" type="image/x-icon">
</head>
<body>
<div id="container">
    <div id="page">
        <header id="masthead" role="titlebanner">
            <div>
                <div class="logo_header">
                    
                </div>
                <nav role="navigation">
                    <ul id="nav_tabs">
                        <li><a href='<?php echo site_url('index'); ?>'>Home
page</a></li>
                        <li><a href='<?php echo site_url('about');
?>'>About</a></li>
                        <li><a href='<?php echo site_url('guides');
?>'>Guides</a></li>
                        <li><a href='<?php echo site_url('links'); ?>'>Link
Exchange</a></li>
                        <li><a href='<?php echo site_url('login'); ?>'>Sign
Up/Log In</a></li>
                    </ul>
                </nav>
            </div>
        </header>
    </div>

```

header_admin.php

```

<html lang="en">
<head>
    <title>FILLES: <?php echo $pageTitle; ?></title>
    <link rel="stylesheet" href="/filles/css/filles.css" />
    <link rel="shortcut icon" href="/filles/images/favicon.ico" type="image/x-icon">
    <link rel="icon" href="/filles/images/favicon.ico" type="image/x-icon">
</head>
<body>
<div id="container">
    <div id="page">
        <header id="masthead" role="titlebanner">
            <div>
                <div class="logo_header">

```

```

                                
                                </div>
                                <nav role="navigation">
                                    <ul id="nav_tabs">
                                        <li><a href="<?php echo site_url('index_admin');
?>">Home page</a></li>
                                        <li><a href="<?php echo site_url('guides');
?>">Guides</a></li>
                                        <li><a href="<?php echo site_url('approve_links');
?>">Approve Links</a></li>
                                        <li><a href="<?php echo site_url('changes');
?>">Change Requests</a></li>
                                        <li><a href="<?php echo site_url('change_password');
?>">Change Password</a></li>
                                        <li><a href="<?php echo site_url('logout');
?>">Logout</a></li>
                                    </ul>
                                </nav>
                                </div>
                            </header>

```

header_contributor.php

```

<html lang="en">
<head>
    <title>FILLES: <?php echo $pageTitle; ?></title>
    <link rel="stylesheet" href="/filles/css/filles.css" />
    <link rel="shortcut icon" href="/filles/images/favicon.ico" type="image/x-icon">
    <link rel="icon" href="/filles/images/favicon.ico" type="image/x-icon">
</head>
<body>
<div id="container">
    <div id="page">
        <header id="masthead" role="titlebanner">
            <div>
                <div class="logo_header">
                    
                </div>
                <nav role="navigation">
                    <ul id="nav_tabs">
                        <li><a href="<?php echo
site_url('index_contributor'); ?>">Home page</a></li>
                        <li><a href="<?php echo site_url('about');
?>">About</a></li>
                        <li><a href="<?php echo site_url('suggest_changes');
?>">Suggest Guide Changes</a></li>
                        <li><a href="<?php echo
site_url('link_maintenance'); ?>">Add/Edit Links</a></li>
                        <li><a href="<?php echo site_url('change_password');
?>">Change Password</a></li>
                        <li><a href="<?php echo site_url('logout');
?>">Logout</a></li>
                    </ul>
                </nav>
            </div>
        </header>

```

header_contributor_tables.php

```

<html lang="en">
<head>
    <title>FILLES: <?php echo $pageTitle; ?></title>
    <link rel="stylesheet" href="/filles/css/filles.css" />
    <link rel="shortcut icon" href="/filles/images/favicon.ico" type="image/x-icon">
    <link rel="icon" href="/filles/images/favicon.ico" type="image/x-icon">
    <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
    <script src="/filles/scripts/jquery.metadata.js"></script>
    <script src="/filles/scripts/jquery.tablesorter.js"></script>
    <script>
        $(document).ready(function()
        {

```

```

        // extend the default setting to always sort on the first column
        $.tablesorter.defaults.sortList = [[0,0]];

        // call the tablesorter plugin
        $("table").tablesorter();

    });
</script>
</head>
<body>
<div id="container">
    <div id="page">
        <header id="masthead" role="titlebanner">
            <div>
                <div class="logo_header">
                    
                </div>
                <nav role="navigation">
                    <ul id="nav_tabs">
                        <li><a href="<?php echo
site_url('index_contributor'); ?>">Home page</a></li>
                        <li><a href="<?php echo site_url('about');
?>">About</a></li>
                        <li><a href="<?php echo site_url('suggest_changes');
?>">Suggest Guide Changes</a></li>
                        <li><a href="<?php echo
site_url('link_maintenance'); ?>">Add/Edit Links</a></li>
                        <li><a href="<?php echo site_url('change_password');
?>">Change Password</a></li>
                        <li><a href="<?php echo site_url('logout');
?>">Logout</a></li>
                    </ul>
                </nav>
            </div>
        </header>

```

header_dictionary_ajax.php

```

<html lang="en">
<head>
    <title>FILLES: <?php echo $pageTitle; ?></title>
    <link rel="stylesheet" href="/filles/css/filles.css" />
    <link rel="shortcut icon" href="/filles/images/favicon.ico" type="image/x-icon">
    <link rel="icon" href="/filles/images/favicon.ico" type="image/x-icon">
    <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
    <script src="/filles/scripts/jquery.watermark.js"></script>
    <script type="text/javascript">
        $(document).ready(function() {
            $("#search_input").watermark("Begin typing to search");
            $("#search_input").keyup(function() {
                var search_input = $(this).val();
                var dataString = 'keyword=' + search_input;
                if (search_input.length > 2) {
                    $.ajax({
                        type: "GET",
                        url: "ajax_search/do_dictionary_search",
                        data: dataString,
                        success: function(server_response)
                        {
                            $('#result_data').html(server_response).show();
                            $('#span#phrase_title').html(search_input);
                        }
                    });
                }
            });
            return false;
        });
    </script>
</head>
<body>

```

```

<div id="container">
  <div id="page">
    <header id="masthead" role="titlebanner">
      <div>
        <div class="logo_header">
          
        </div>
        <nav role="navigation">
          <ul id="nav_tabs">
            <li><a href='<?php echo site_url('index'); ?>'>Home
page</a></li>
            <li><a href='<?php echo site_url('about');
?>'>About</a></li>
            <li><a href='<?php echo site_url('guides');
?>'>Guides</a></li>
            <li><a href='<?php echo site_url('links'); ?>'>Link
Exchange</a></li>
            <li><a href='<?php echo site_url('login'); ?>'>Sign
Up/Log In</a></li>
          </ul>
        </nav>
      </div>
    </header>
  </div>

```

header_grammar_ajax.php

```

<html lang="en">
<head>
  <title>FILLES: <?php echo $pageTitle; ?></title>
  <link rel="stylesheet" href="/filles/css/filles.css" />
  <link rel="shortcut icon" href="/filles/images/favicon.ico" type="image/x-icon">
  <link rel="icon" href="/filles/images/favicon.ico" type="image/x-icon">
  <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
  <script src="/filles/scripts/jquery.watermark.js"></script>
  <script type="text/javascript">
    $(document).ready(function() {
      $("#search_input").watermark("Begin typing to search");
      $("#search_input").keyup(function() {
        var search_input = $(this).val();
        var dataString = 'keyword=' + search_input;
        if (search_input.length > 2) {
          $.ajax({
            type: "GET",
            url: "ajax_search/do_grammar_search",
            data: dataString,
            success: function(server_response)
            {
              $('#result_data').html(server_response).show();
              $('#span#phrase_title').html(search_input);
            }
          });
        }
        return false;
      });
    });
  </script>
</head>
<body>
<div id="container">
  <div id="page">
    <header id="masthead" role="titlebanner">
      <div>
        <div class="logo_header">
          
        </div>
        <nav role="navigation">
          <ul id="nav_tabs">
            <li><a href='<?php echo site_url('index'); ?>'>Home
page</a></li>

```

```

?>'>About</a></li>
?>'>Guides</a></li>
Exchange</a></li>
Up/Log In</a></li>
</ul>
</nav>
</div>
</header>

```

header_phrase_ajax.php

```

<html lang="en">
<head>
<title>FILES: <?php echo $pageTitle; ?></title>
<link rel="stylesheet" href="/filles/css/filles.css" />
<link rel="shortcut icon" href="/filles/images/favicon.ico" type="image/x-icon">
<link rel="icon" href="/filles/images/favicon.ico" type="image/x-icon">
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
<script src="/filles/scripts/jquery.watermark.js"></script>
<script type="text/javascript">
$(document).ready(function() {
    $("#search_input").watermark("Begin typing to search");
    $("#search_input").keyup(function() {
        var search_input = $(this).val();
        var dataString = 'keyword=' + search_input;
        if (search_input.length > 2) {
            $.ajax({
                type: "GET",
                url: "ajax_search/do_grammar_search",
                data: dataString,
                success: function(server_response)
                {
                    $('#result_data').html(server_response).show();
                    $('#span#phrase_title').html(search_input);
                }
            });
        }
        return false;
    });
});
</script>
</head>
<body>
<div id="container">
<div id="page">
<header id="masthead" role="titlebanner">
<div>
<div class="logo_header">

</div>
<nav role="navigation">
<ul id="nav_tabs">
<li><a href='<?php echo site_url('index'); ?>'>Home
page</a></li>
?>'>About</a></li>
?>'>Guides</a></li>
Exchange</a></li>
Up/Log In</a></li>
</ul>
</nav>

```

```
    </div>
</header>
```

header_staff.php

```
<html lang="en">
<head>
    <title>FILLES: <?php echo $pageTitle; ?></title>
    <link rel="stylesheet" href="/filles/css/filles.css" />
    <link rel="shortcut icon" href="/filles/images/favicon.ico" type="image/x-icon">
    <link rel="icon" href="/filles/images/favicon.ico" type="image/x-icon">
</head>
<body>
<div id="container">
    <div id="page">
        <header id="masthead" role="titlebanner">
            <div>
                <div class="logo_header">
                    
                </div>
                <nav role="navigation">
                    <ul id="nav_tabs">
                        <li><a href="<?php echo site_url('index_staff');
?>">Home page</a></li>
                        <li><a href="<?php echo
site_url('lesson_maintenance'); ?>">Maintain Lessons</a></li>
                        <li><a href="<?php echo
site_url('guide_maintenance'); ?>">Maintain Guides</a></li>
                        <li><a href="<?php echo
site_url('bank_maintenance'); ?>">Maintain Questions</a></li>
                        <li><a href="<?php echo site_url('changes');
?>">Change Requests</a></li>
                        <li><a href="<?php echo
site_url('link_maintenance'); ?>">Maintain Links</a></li>
                        <li><a href="<?php echo site_url('change_password');
?>">Change Password</a></li>
                        <li><a href="<?php echo site_url('logout');
?>">Logout</a></li>
                    </ul>
                </nav>
            </div>
        </div>
    </header>
```

header_staff_dictionary_ajax.php

```
<html lang="en">
<head>
    <title>FILLES: <?php echo $pageTitle; ?></title>
    <link rel="stylesheet" href="/filles/css/filles.css" />
    <link rel="shortcut icon" href="/filles/images/favicon.ico" type="image/x-icon">
    <link rel="icon" href="/filles/images/favicon.ico" type="image/x-icon">
    <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
    <script src="/filles/scripts/jquery.watermark.js"></script>
    <script type="text/javascript">
        $(document).ready(function() {
            $("#search_input").watermark("Begin typing to search");
            $("#search_input").keyup(function() {
                var search_input = $(this).val();
                var dataString = 'keyword=' + search_input;
                if (search_input.length > 2) {
                    $.ajax({
                        type: "GET",
                        url: "ajax_search/do_dictionary_search_staff",
                        data: dataString,
                        success: function(server_response)
                        {
                            $('#result_data').html(server_response).show();
                            $('#span#phrase_title').html(search_input);
                        }
                    });
                }
            });
        }
    </script>
```



```

        return false;
    });
</script>
</head>
<body>
<div id="container">
    <div id="page">
        <header id="masthead" role="titlebanner">
            <div>
                <div class="logo_header">
                    
                </div>
                <nav role="navigation">
                    <ul id="nav_tabs">
                        <li><a href="<?php echo site_url('index_staff');
?>">Home page</a></li>
                        <li><a href="<?php echo
site_url('lesson_maintenance'); ?>">Maintain Lessons</a></li>
                        <li><a href="<?php echo
site_url('guide_maintenance'); ?>">Maintain Guides</a></li>
                        <li><a href="<?php echo
site_url('bank_maintenance'); ?>">Maintain Questions</a></li>
                        <li><a href="<?php echo site_url('changes');
?>">Change Requests</a></li>
                        <li><a href='<?php echo
site_url("link_maintenance"); ?>'>Maintain Links</a></li>
                        <li><a href='<?php echo site_url("change_password");
?>'>Change Password</a></li>
                        <li><a href='<?php echo site_url("logout");
?>'>Logout</a></li>
                    </ul>
                </nav>
            </div>
        </header>

```

header_staff_phrase_ajax.php

```

<html lang="en">
<head>
    <title>FILLES: <?php echo $pageTitle; ?></title>
    <link rel="stylesheet" href="/filles/css/filles.css" />
    <link rel="shortcut icon" href="/filles/images/favicon.ico" type="image/x-icon">
    <link rel="icon" href="/filles/images/favicon.ico" type="image/x-icon">
    <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
    <script src="/filles/scripts/jquery.watermark.js"></script>
    <script type="text/javascript">
        $(document).ready(function() {
            $("#search_input").watermark("Begin typing to search");
            $("#search_input").keyup(function() {
                var search_input = $(this).val();
                var dataString = 'keyword=' + search_input;
                if (search_input.length > 2) {
                    $.ajax({
                        type: "GET",
                        url: "ajax_search/do_phrase_search_staff",
                        data: dataString,
                        success: function(server_response)
                        {
                            $('#result_data').html(server_response).show();
                            $('#span#phrase_title').html(search_input);
                        }
                    });
                }
            });
            return false;
        });
    </script>
</head>
<body>

```

```

<div id="container">
  <div id="page">
    <header id="masthead" role="titlebanner">
      <div>
        <div class="logo_header">
          
        </div>
        <nav role="navigation">
          <ul id="nav_tabs">
            <li><a href="<?php echo site_url('index_staff');
?>">Home page</a></li>
            <li><a href="<?php echo
site_url('lesson_maintenance'); ?>">Maintain Lessons</a></li>
            <li><a href="<?php echo
site_url('guide_maintenance'); ?>">Maintain Guides</a></li>
            <li><a href="<?php echo
site_url('bank_maintenance'); ?>">Maintain Questions</a></li>
            <li><a href="<?php echo site_url('changes');
?>">Change Requests</a></li>
            <li><a href='<?php echo
site_url("link_maintenance"); ?>'>Maintain Links</a></li>
            <li><a href='<?php echo site_url("change_password");
?>'>Change Password</a></li>
            <li><a href='<?php echo site_url("logout");
?>'>Logout</a></li>
          </ul>
        </nav>
      </div>
    </header>

```

header_sysadmin.php

```

<html lang="en">
<head>
  <title>FILLES: <?php echo $pageTitle; ?></title>
  <link rel="stylesheet" href="/filles/css/filles.css" />
  <link rel="shortcut icon" href="/filles/images/favicon.ico" type="image/x-icon">
  <link rel="icon" href="/filles/images/favicon.ico" type="image/x-icon">
</head>
<body>
<div id="container">
  <div id="page">
    <header id="masthead" role="titlebanner">
      <div>
        <div class="logo_header">
          
        </div>
        <nav role="navigation">
          <ul id="nav_tabs">
            <li><a href="<?php echo site_url('index_admin');
?>">Home page</a></li>
            <li><a href="<?php echo site_url('add_admin');
?>">Add System Admin</a></li>
            <li><a href="<?php echo
site_url('add_content_admin'); ?>">Add Content Admin</a></li>
            <li><a href="<?php echo site_url('add_staff');
?>">Add Staff Member</a></li>
            <li><a href="<?php echo site_url('maintain_users');
?>">Maintain Users</a></li>
            <li><a href="<?php echo
site_url('maintain_user_groups'); ?>">Maintain Groups</a></li>
            <li><a href="<?php echo site_url('change_password');
?>">Change Password</a></li>
            <li><a href="<?php echo site_url('logout');
?>">Logout</a></li>
          </ul>
        </nav>
      </div>
    </header>

```

header_tables.php

```
<html lang="en">
<head>
  <title>FILLES: <?php echo $pageTitle; ?></title>
  <link rel="stylesheet" href="/filles/css/filles.css" />
  <link rel="shortcut icon" href="/filles/images/favicon.ico" type="image/x-icon">
  <link rel="icon" href="/filles/images/favicon.ico" type="image/x-icon">
  <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
  <script src="/filles/scripts/jquery.metadata.js"></script>
  <script src="/filles/scripts/jquery.tablesorter.js"></script>
  <script>
    $(document).ready(function()
    {
      // extend the default setting to always sort on the first column
      $.tablesorter.defaults.sortList = [[0,0]];

      // call the tableorter plugin
      $("table").tablesorter();

    });
  </script>
</head>
<body>
<div id="container">
  <div id="page">
    <header id="masthead" role="titlebanner">
      <div>
        <div class="logo_header">
          
        </div>
        <nav role="navigation">
          <ul id="nav_tabs">
            <li><a href='<?php echo site_url("index"); ?>'>Home
page</a></li>
            <li><a href='<?php echo site_url("about");
?>'>About</a></li>
            <li><a href='<?php echo site_url("guides");
?>'>Guides</a></li>
            <li><a href='<?php echo site_url("links"); ?>'>Link
Exchange</a></li>
            <li><a href='<?php echo site_url("login"); ?>'>Sign
Up/Log In</a></li>
          </ul>
        </nav>
      </div>
    </header>
  </div>
</div>
</body>
</html>
```

header_user.php

```
<html lang="en">
<head>
  <title>FILLES: <?php echo $pageTitle; ?></title>
  <link rel="stylesheet" href="/filles/css/filles.css" />
  <link rel="shortcut icon" href="/filles/images/favicon.ico" type="image/x-icon">
  <link rel="icon" href="/filles/images/favicon.ico" type="image/x-icon">
</head>
<body>
<div id="container">
  <div id="page">
    <header id="masthead" role="titlebanner">
      <div>
        <div class="logo_header">
          
        </div>
        <nav role="navigation">
          <ul id="nav_tabs">
            <li><a href='<?php echo site_url('index_user');
?>'>Home page</a></li>
            <li><a href='<?php echo site_url('about');
?>'>About</a></li>
          </ul>
        </nav>
      </div>
    </header>
  </div>
</div>
</body>
</html>
```

```

?>'>Lessons</a></li>
?>'>Guides</a></li>
Exchange</a></li>
?>'>Suggest Guide Changes</a></li>
?>">Change Password</a></li>
?>'>Logout</a></li>
</ul>
</nav>
</div>
</header>

```

header_user_ajax.php

```

<html lang="en">
<head>
  <title>FILLES: <?php echo $pageTitle; ?></title>
  <link rel="stylesheet" href="/filles/css/filles.css" />
  <link rel="shortcut icon" href="/filles/images/favicon.ico" type="image/x-icon">
  <link rel="icon" href="/filles/images/favicon.ico" type="image/x-icon">
  <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
</head>
<body>
<div id="container">
  <div id="page">
    <header id="masthead" role="titlebanner">
      <div>
        <div class="logo_header">
          
        </div>
        <nav role="navigation">
          <ul id="nav_tabs">
            <li><a href='<?php echo site_url('index_user');
?>'>Home page</a></li>
            <li><a href='<?php echo site_url('about');
?>'>About</a></li>
            <li><a href='<?php echo site_url('lessons');
?>'>Lessons</a></li>
            <li><a href='<?php echo site_url('guides');
?>'>Guides</a></li>
            <li><a href='<?php echo site_url('links'); ?>'>Link
Exchange</a></li>
            <li><a href='<?php echo site_url('suggest_changes');
?>'>Suggest Guide Changes</a></li>
            <li><a href="<?php echo site_url('change_password');
?>">Change Password</a></li>
            <li><a href='<?php echo site_url('logout');
?>'>Logout</a></li>
          </ul>
        </nav>
      </div>
    </div>
  </div>
</div>

```

header_user_dictionary_ajax.php

```

<html lang="en">
<head>
  <title>FILLES: <?php echo $pageTitle; ?></title>
  <link rel="stylesheet" href="/filles/css/filles.css" />
  <link rel="shortcut icon" href="/filles/images/favicon.ico" type="image/x-icon">
  <link rel="icon" href="/filles/images/favicon.ico" type="image/x-icon">
  <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
  <script src="/filles/scripts/jquery.watermark.js"></script>
  <script type="text/javascript">
    $(document).ready(function() {
      $("#search_input").watermark("Begin typing to search");
    });
  </script>

```

```

        $("#search_input").keyup(function() {
            var search_input = $(this).val();
            var dataString = 'keyword=' + search_input;
            if (search_input.length > 2) {
                $.ajax({
                    type: "GET",
                    url: "ajax_search/do_dictionary_search",
                    data: dataString,
                    success: function(server_response)
                    {

                        $('#result_data').html(server_response).show();
                        $('#span#phrase_title').html(search_input);
                    }
                });
            }
            return false;
        });
    });
</script>
</head>
<body>
<div id="container">
    <div id="page">
        <header id="masthead" role="titlebanner">
            <div>
                <div class="logo_header">
                    
                </div>
                <nav role="navigation">
                    <ul id="nav_tabs">
                        <li><a href='<?php echo site_url('index_user');
?>'>Home page</a></li>
                        <li><a href='<?php echo site_url('about');
?>'>About</a></li>
                        <li><a href='<?php echo site_url('lessons');
?>'>Lessons</a></li>
                        <li><a href='<?php echo site_url('guides');
?>'>Guides</a></li>
                        <li><a href='<?php echo site_url('links'); ?>'>Link
Exchange</a></li>
                        <li><a href='<?php echo site_url('suggest_changes');
?>'>Suggest Guide Changes</a></li>
                        <li><a href="<?php echo site_url('change_password');
?>">Change Password</a></li>
                        <li><a href='<?php echo site_url('logout');
?>'>Logout</a></li>
                    </ul>
                </nav>
            </div>
        </header>

```

header_user_grammar_ajax.php

```

<html lang="en">
<head>
    <title>FILLES: <?php echo $pageTitle; ?></title>
    <link rel="stylesheet" href="/filles/css/filles.css" />
    <link rel="shortcut icon" href="/filles/images/favicon.ico" type="image/x-icon">
    <link rel="icon" href="/filles/images/favicon.ico" type="image/x-icon">
    <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
    <script type="text/javascript">
        $(document).ready(function() {
            $("#search_input").watermark("Begin typing to search");
            $("#search_input").keyup(function() {
                var search_input = $(this).val();
                var dataString = 'keyword=' + search_input;
                if (search_input.length > 2) {
                    $.ajax({
                        type: "GET",
                        url: "ajax_search/do_grammar_search",

```

```

        data: dataString,
        success: function(server_response)
        {
            $('#result_data').html(server_response).show();
            $('#span#phrase_title').html(search_input);
        }
    });
}
return false;
});
});
</script>
</head>
<body>
<div id="container">
    <div id="page">
        <header id="masthead" role="titlebanner">
            <div>
                <div class="logo_header">
                    
                </div>
                <nav role="navigation">
                    <ul id="nav_tabs">
                        <li><a href='<?php echo site_url('index_user');
                        <li><a href='<?php echo site_url('about');
                        <li><a href='<?php echo site_url('lessons');
                        <li><a href='<?php echo site_url('guides');
                        <li><a href='<?php echo site_url('links'); ?>'>Link
                        <li><a href='<?php echo site_url('suggest_changes');
                        <li><a href='<?php echo site_url('change_password');
                        <li><a href='<?php echo site_url('logout');
                    </ul>
                </nav>
            </div>
        </header>

```

header_user_phrase_ajax.php

```

<html lang="en">
<head>
    <title>FILLES: <?php echo $pageTitle; ?></title>
    <link rel="stylesheet" href="/filles/css/filles.css" />
    <link rel="shortcut icon" href="/filles/images/favicon.ico" type="image/x-icon">
    <link rel="icon" href="/filles/images/favicon.ico" type="image/x-icon">
    <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
    <script src="/filles/scripts/jquery.watermark.js"></script>
    <script type="text/javascript">
        $(document).ready(function() {
            $('#search_input').watermark("Begin typing to search");
            $('#search_input').keyup(function() {
                var search_input = $(this).val();
                var dataString = 'keyword=' + search_input;
                if (search_input.length > 2) {
                    $.ajax({
                        type: "GET",
                        url: "ajax_search/do_phrase_search",
                        data: dataString,
                        success: function(server_response)
                        {
                            $('#result_data').html(server_response).show();
                        }
                    });
                }
            });
        });
    </script>

```

```

        $('span#phrase_title').html(search_input);
    }
    });
}
return false;
});
});
</script>
</head>
<body>
<div id="container">
    <div id="page">
        <header id="masthead" role="titlebanner">
            <div>
                <div class="logo_header">
                    
                </div>
                <nav role="navigation">
                    <ul id="nav_tabs">
                        <li><a href='<?php echo site_url('index_user');
?>'>Home page</a></li>
                        <li><a href='<?php echo site_url('about');
?>'>About</a></li>
                        <li><a href='<?php echo site_url('lessons');
?>'>Lessons</a></li>
                        <li><a href='<?php echo site_url('guides');
?>'>Guides</a></li>
                        <li><a href='<?php echo site_url('links'); ?>'>Link
Exchange</a></li>
                        <li><a href='<?php echo site_url('suggest_changes');
?>'>Suggest Guide Changes</a></li>
                        <li><a href='<?php echo site_url('change_password');
?>'>Change Password</a></li>
                        <li><a href='<?php echo site_url('logout');
?>'>Logout</a></li>
                    </ul>
                </nav>
            </div>
        </header>
    </div>
</div>

```

header_user_tables.php

```

<html lang="en">
<head>
    <title>FILLES: <?php echo $pageTitle; ?></title>
    <link rel="stylesheet" href="/filles/css/filles.css" />
    <link rel="shortcut icon" href="/filles/images/favicon.ico" type="image/x-icon">
    <link rel="icon" href="/filles/images/favicon.ico" type="image/x-icon">
    <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
    <script src="/filles/scripts/jquery.metadata.js"></script>
    <script src="/filles/scripts/jquery.tablesorter.js"></script>
    <script>
        $(document).ready(function()
        {
            // extend the default setting to always sort on the first column
            $.tablesorter.defaults.sortList = [[0,0]];

            // call the tableorter plugin
            $("table").tablesorter();
        });
    </script>
</head>
<body>
<div id="container">
    <div id="page">
        <header id="masthead" role="titlebanner">
            <div>
                <div class="logo_header">

```

```

        
    </div>
    <nav role="navigation">
        <ul id="nav_tabs">
            <li><a href='<?php echo site_url('index_user');
?>'>Home page</a></li>
            <li><a href='<?php echo site_url('about');
?>'>About</a></li>
            <li><a href='<?php echo site_url('lessons');
?>'>Lessons</a></li>
            <li><a href='<?php echo site_url('guides');
?>'>Guides</a></li>
            <li><a href='<?php echo site_url('links'); ?>'>Link
Exchange</a></li>
            <li><a href='<?php echo site_url('suggest_changes');
?>'>Suggest Guide Changes</a></li>
            <li><a href='<?php echo site_url('change_password');
?>'>Change Password</a></li>
            <li><a href='<?php echo site_url('logout');
?>'>Logout</a></li>
        </ul>
    </nav>
</div>
</header>

```

index_admin_view.php

```

    <div id="main" role="main">
        <div class="main_class">
            <h1>Mabuhay! Welcome <?php echo $name; ?>!</h1>
            <section>
                <p>Welcome to FILLES, an open-source comprehensive learning
system developed for learning the Filipino language.</p>
                <p>Unlike many commercial learning systems, FILLES is
targeted towards younger learners, particularly children of Filipinos who migrated before the
children were born or Filipino migrants who migrated before they started schooling, and also
Filipino balikbayans who have forgotten the grasp of the language.</p>
                <p>It is with this endeavor that the sustainability of the
Filipino language among migrants around the world will be realized.</p>
            </section>
        </div>
    </div>

    <div id="related" class="sidebar" role="complementary">
    <aside>
        <p>This SP is developed with the Sentro ng Wikang Filipino - UP
Manila.</p>
        <p>For content concerns, contact the SWF.</p>
        <p>For technical concerns, contact the author <?php echo
safe_mailto('bonarae@gmail.com', 'here'); ?>.</p>
        <p>To ensure full compatibility with HTML5, Firefox or Opera is
recommended.</p>
    </aside>
    </div>

```

index_contributor_view.php

```

    <div id="main" role="main">
        <div class="main_class">
            <h1>Mabuhay! Welcome <?php echo $name; ?>!</h1>
            <section>
                <p>Welcome to FILLES, an open-source comprehensive learning
system developed for learning the Filipino language.</p>
                <p>Unlike many commercial learning systems, FILLES is
targeted towards younger learners, particularly children of Filipinos who migrated before the
children were born or Filipino migrants who migrated before they started schooling, and also
Filipino balikbayans who have forgotten the grasp of the language.</p>
                <p>It is with this endeavor that the sustainability of the
Filipino language among migrants around the world will be realized.</p>
            </section>

```



```

    </div>
</div>

<div id="related" class="sidebar" role="complementary">
<aside>
    <p>This SP is developed with the Sentro ng Wikang Filipino - UP
Manila.</p>
    <p>For content concerns, contact the SWF.</p>
    <p>For technical concerns, contact the author <?php echo
safe_mailto('bonarae@gmail.com', 'here'); ?>.</p>
    <p>To ensure full compatibility with HTML5, Firefox or Opera is
recommended.</p>
</aside>
</div>

index_staff_view.php
<div id="main" role="main">
    <div class="main_class">
        <h1>Mabuhay! Welcome <?php echo $name; ?>!</h1>
        <section>
            <p>Welcome to FILLES, an open-source comprehensive learning
system developed for learning the Filipino language.</p>
            <p>Unlike many commercial learning systems, FILLES is
targeted towards younger learners, particularly children of Filipinos who migrated before the
children were born or Filipino migrants who migrated before they started schooling, and also
Filipino balikbayans who have forgotten the grasp of the language.</p>
            <p>It is with this endeavor that the sustainability of the
Filipino language among migrants around the world will be realized.</p>
        </section>
    </div>
</div>

<div id="related" class="sidebar" role="complementary">
<aside>
    <p>This SP is developed with the Sentro ng Wikang Filipino - UP
Manila.</p>
    <p>For content concerns, contact the SWF.</p>
    <p>For technical concerns, contact the author <?php echo
safe_mailto('bonarae@gmail.com', 'here'); ?>.</p>
    <p>To ensure full compatibility with HTML5, Firefox or Opera is
recommended.</p>
</aside>
</div>

index_sysadmin_view.php
<div id="main" role="main">
    <div class="main_class">
        <h1>Mabuhay! Welcome <?php echo $name; ?>!</h1>
        <section>
            <p>Welcome to FILLES, an open-source comprehensive learning
system developed for learning the Filipino language.</p>
            <p>Unlike many commercial learning systems, FILLES is
targeted towards younger learners, particularly children of Filipinos who migrated before the
children were born or Filipino migrants who migrated before they started schooling, and also
Filipino balikbayans who have forgotten the grasp of the language.</p>
            <p>It is with this endeavor that the sustainability of the
Filipino language among migrants around the world will be realized.</p>
        </section>
    </div>
</div>

<div id="related" class="sidebar" role="complementary">
<aside>
    <p>This SP is developed with the Sentro ng Wikang Filipino - UP
Manila.</p>
    <p>For content concerns, contact the SWF.</p>
    <p>For technical concerns, contact the author <?php echo
safe_mailto('bonarae@gmail.com', 'here'); ?>.</p>
    <p>To ensure full compatibility with HTML5, Firefox or Opera is
recommended.</p>
</aside>
</div>

```

```
</aside>
</div>
```

index_user_view.php

```
<div id="main" role="main">
  <div class="main_class">
    <h1>Mabuhay! Welcome <?php echo $name; ?>!</h1>
    <section>
      <p>Welcome to FILLES, an open-source comprehensive learning
system developed for learning the Filipino language.</p>
      <p>Unlike many commercial learning systems, FILLES is
targeted towards younger learners, particularly children of Filipinos who migrated before the
children were born or Filipino migrants who migrated before they started schooling, and also
Filipino balikbayans who have forgotten the grasp of the language.</p>
      <p>It is with this endeavor that the sustainability of the
Filipino language among migrants around the world will be realized.</p>
    </section>
  </div>
</div>

<div id="related" class="sidebar" role="complementary">
<aside>
  <p>This SP is developed with the Sentro ng Wikang Filipino - UP
Manila.</p>
  <p>For content concerns, contact the SWF.</p>
  <p>For technical concerns, contact the author <?php echo
safe_mailto('bonarae@gmail.com', 'here'); ?>.</p>
  <p>To ensure full compatibility with HTML5, Firefox or Opera is
recommended.</p>
</aside>
</div>
```

index_view.php

```
<div id="main" role="main">
  <div class="main_class">
    <h1>Mabuhay! Welcome!</h1>
    <section>
      <p>Welcome to FILLES, an open-source comprehensive learning
system developed for learning the Filipino language.</p>
      <p>Unlike many commercial learning systems, FILLES is
targeted towards younger learners, particularly children of Filipinos who migrated before the
children were born or Filipino migrants who migrated before they started schooling, and also
Filipino balikbayans who have forgotten the grasp of the language.</p>
      <p>It is with this endeavor that the sustainability of the
Filipino language among migrants around the world will be realized.</p>
    </section>
  </div>
</div>

<div id="related" class="sidebar" role="complementary">
<aside>
  <p>This SP is developed with the Sentro ng Wikang Filipino - UP
Manila.</p>
  <p>For content concerns, contact the SWF.</p>
  <p>For technical concerns, contact the author <?php echo
safe_mailto('bonarae@gmail.com', 'here'); ?>.</p>
  <p>To ensure full compatibility with HTML5, Firefox or Opera is
recommended.</p>
</aside>
</div>
```

lesson_delete_view.php

```
<div id="main" role="main">
```

```

<article class="guide">
    <h3>Are you sure you really want to delete the lesson?</h3>
    <p><a href="lesson_delete/delete?id=?php echo $id; ?>">Yes</a>
        <a href="delete_lesson">No</a></p>
</article>
</div>

lesson_deleted_view.php
    <h3>Deletion of lesson entry successful.</h3>
    <p>Return to the <a href="lesson_maintenance">lesson maintenance</a> page.</p>

lesson_maintenance_view.php
    <div id="main" class="lesson" role="main">
        <ul>
            <li><a href="add_lesson">Add Lessons</a></li>
            <li><a href="edit_lesson">Edit Lessons</a></li>
            <li><a href="delete_lesson">Delete Lessons</a></li>
        </ul>
    </div>

    <div id="related" class="sidebar" role="complementary">
    <aside>
        <h4>Help</h4>
        <p>To add lessons, click on <b>Add Lessons</b>.</p>
        <p>To edit lessons, click on <b>Edit Lessons</b>.</p>
        <p>To delete any duplicate or unwanted lessons, click on <b>Delete
Lessons</b>.</p>
    </aside>
    </div>

lesson_view.php
    <div id="main" role="main">
        <article class="lesson">
            <?php
                if($this->input->get('id')) {
                    $lesson_id = $this->input->get('id');
                    $query = $this->db->get_where('lessons',array('lesson_id'
=> $lesson_id));

                    if ($query->num_rows() > 0) {
                        $row = $query->row();
                        $title = $row->lesson_title;
                        $description = $row->lesson_description;
                        $content = $row->lesson_content;

                        echo '<h2>' . $title . '</h2>';
                        echo '<h4>' . $description . '</h4>';

                        echo "<object width='100%' height='700px'
data='$content' type='application/pdf'>
                                <p>It appears you don't have a PDF plugin
                                for this browser.
                                No biggie... you can <a
                                href='$content'>click here to
                                download the PDF file.</a></p>
                                </object>";

                                echo "<h4><a href='take_quiz?id=$lesson_id'>Take a
quiz for this lesson</a></h4>";
                            }
                            else {
                                echo '<p>No lesson with that ID exists.</p>';
                            }
                        }
                        else {
                            echo '<p>Parameter missing!</p>';
                        }
                    }
                <?>
            </article>
        </div>

```

lessons_view.php

```
<div id="main" role="main">
  <article class="main_class">
    <h2>What task are you going to do today?</h2>
    <ul>
      <li><a href="view_lessons">View lessons</a></li>
      <li>To take a test, visit a lesson first and take a quiz specific
to that lesson.</li>
      <li><a href="view_progress">View your progress</a></li>
    </ul>
  </article>
</div>
```

link_deleted_view.php

```
<h3>Deletion of link successful.</h3>
<p>Return to the <a href="link_maintenance">link maintenance</a> page.</p>
```

link_maintenance_view.php

```
<div id="main" class="links" role="main">
  <ul>
    <li><a href="add_link">Add Links</a></li>
    <li><a href="edit_link">Edit Links</a></li>
    <li><a href="check_link_status">Check Status of Submitted
Links</a></li>
    <li><a href="list_delete_link">Delete Links</a></li>
  </ul>
</div>

<div id="related" class="sidebar" role="complementary">
<aside>
  <h4>Help</h4>
  <p>To add a link, click on <b>Add Links</b>.</p>
  <p>To edit your links, click on <b>Edit Links</b>.</p>
  <p>If you need to delete any of your links from the Exchange, click on
<b>Delete Links</b>.</p>
</aside>
</div>
```

links_approved_view.php

```
<div id="main" role="main">
  <article class="main_class">
    <?php
      echo '<h3>' . $links_approved . ' links approved.</h3>';
      if ($links_rejected) {
        echo '<h3>' . $links_rejected . ' links rejected.</h3>';
      }
    <?>
    <p><center><a href="index_admin">Return home</a></center></p>
  </article>
</div>
```

links_view.php

```
<div id="main" role="main">
  <article class="links">
    <table class="tablesorter">
      <?php
        if ($this->link_model->approved_count() == 0) {
          echo '<h2>No links defined so far...</h2>';
        }
        else {
          echo '<thead>';
          echo '<th>Link Title</th>';
          echo "<th class='{sorter:"
false}'>Description</th>";
          echo '</thead>';
          echo '<tbody>';
```

```

        foreach($results as $data) {
            $view_link = site_url('view_link?id='
. $data->link_id);
            echo "<tr>
                <td><a
href='$view_link'>$data->link_title</a></td>\n
                <td>$data->link_description</td>\n
            </tr>";
        }
        echo '</tbody>';
        echo '</table>';
        ?>
        <p><?php echo $links; ?></p>
    </article>
</div>

links_delete_dictionary_view.php
    <div id="main" role="main">
        <article class="guide">
            <table border="1">
                <?php
                    if ($this->dictionary_entries->record_count() == 0)
                    {
                        echo '<h2>No dictionary entries defined so
far...</h2>';
                    }
                    else {
                        echo '<thead>';
                        echo '<th>Entry Name</th>';
                        echo '<th>Part of Speech</th>';
                        echo '<th class="{sorter:
false}">Meaning</th>';
                        echo '<th>Category</th>';
                        echo '<th>Delete Entry?</th>';
                        echo '</thead>';
                        echo '<tbody>';
                        foreach($results as $data) {
                            echo "<tr>
                                <td>$data->
dictionary_term</td>\n
                                <td>$data->
dictionary_partofspeech</td>\n
                                <td>$data->
dictionary_meaning</td>\n
                                <td>$data->
dictionary_category</td>\n
                                <td><a
href='confirm_dictionary_deletion?id=$data->dictionary_id'>Delete</a></td>\n
                            </tr>";
                        }
                        echo '</tbody>';
                    }
                ?>
            </table>
            <p><?php echo $links; ?></p>
        </article>
    </div>

    <div id="related" class="sidebar" role="complementary">
        <aside>
            <h4>To delete an entry from the dictionary, click on the "Delete"
link of that row where the entry is located.</h4>
        </aside>
    </div>

list_delete_grammar_view.php
    <div id="main" role="main">
        <article class="guide">
            <table border="1">

```

```

        <?php
            if ($this->grammar_entries->record_count() == 0) {
                echo '<h2>No grammar rules defined so
far...</h2>';
            }
            else {
                echo '<thead>';
                echo '<th>Grammar Rule</th>';
                echo '<th class="{sorter:
false}">Usage</th>';
                echo '<th class="{sorter:
false}">Example</th>';
                echo '<th class="{sorter:
false}">Delete Rule?</th>';
                echo '</thead>';
                echo '<tbody>';
                foreach($results as $data) {
                    echo "<tr>
                        <td>$data-
>grammar_phrase</td>\n
                        <td>$data-
>grammar_usage</td>\n
                        <td>$data-
>grammar_example</td>\n
                        <td><a
href='confirm_grammar_deletion?id=$data->grammar_id'>Delete</a></td>\n
                        </tr>";
                    }
                echo '</tbody>';
            }
        ?>
    </table>
    <p><?php echo $links; ?></p>
</article>
</div>

<div id="related" class="sidebar" role="complementary">
    <aside>
        <h4>To delete a grammar rule, click on the "Delete" link of that
row where the entry is located.</h4>
    </aside>
</div>

list_delete_instruction_view.php
<div id="main" role="main">
    <article class="quiz">
        <table border="1">
            <?php
                if ($this->instruction_entries->record_count() == 0)
                {
                    echo '<h2>No instructions defined so
far...</h2>';
                }
                else {
                    echo '<thead>';
                    echo '<th>Instruction Text</th>';
                    echo '<th>Lesson ID</th>';
                    echo '<th>Delete Instruction?</th>';
                    echo '</thead>';
                    echo '<tbody>';
                    foreach($results as $data) {
                        echo "<tr>
                            <td>$data-
>instruction_text</td>\n
                            <td>$data->lesson_id</td>\n
                            <td><a
href='confirm_instruction_deletion?id=$data->instruction_no'>Delete</a></td>\n
                            </tr>";
                        }
                    echo '</tbody>';
                }
            ?>
        </table>
    </article>
</div>

```

```

        </table>
    </article>
</div>

list_delete_link_view.php
<div id="main" role="main">
    <article class="links">
        <table border="1">
            <?php
                if ($this->link_model->record_count_by_user() == 0)
                {
                    echo '<h2>No links defined so far...</h2>';
                }
                else {
                    echo '<thead>';
                    echo '<th>Link Title</th>';
                    echo '<th>Description</th>';
                    echo '<th class="{sorter:'
false}">URL</th>';
                    echo '<th>Delete Link?</th>';
                    echo '</thead>';
                    echo '<tbody>';
                    foreach($results as $data) {
                        echo "<tr>
                            <td>$data->link_title</td>\n
                            <td>$data-
>link_description</td>\n
                            <td>$data->link_url</td>\n
                            <td><a
href='confirm_link_deletion?id=$data->link_id'>Delete</a></td>\n
                            </tr>";
                        }
                    echo '</tbody>';
                }
            ?>
        </table>
        <p><?php echo $links; ?></p>
    </article>
</div>

<div id="related" class="sidebar" role="complementary">
    <aside>
        <h4>To delete a link from the database, click on the "Delete" link
of that row where the entry is located.</h4>
    </aside>
</div>

list_delete_phrasebook_view.php
<div id="main" role="main">
    <article class="guide">
        <table border="1">
            <?php
                if ($this->phrasebook_entries->record_count() == 0)
                {
                    echo '<h2>No phrases defined so
far...</h2>';
                }
                else {
                    echo '<thead>';
                    echo '<th>Phrase</th>';
                    echo '<th class="{sorter:'
false}">English translation</th>';
                    echo '<th>Delete Phrase?</th>';
                    echo '</thead>';
                    echo '<tbody>';
                    foreach($results as $data) {
                        echo "<tr>
                            <td>$data-
>phrasebook_phrase</td>\n
                            <td>$data-
>phrasebook_translation</td>\n

```

```

                                <td><a
href='confirm_phrase_deletion?id=$data->phrasebook_id'>Delete</a></td>\n
                                </tr>";
                                }
                                echo '</tbody>';
                                }
                                ?>
                                </table>
                                <p><?php echo $links; ?></p>
                                </article>
                                </div>

                                <div id="related" class="sidebar" role="complementary">
                                <aside>
                                <h4>To delete a phrase, click on the "Delete" link of that row
where the entry is located.</h4>
                                </aside>
                                </div>

```

list_delete_question_view.php

```

                                <div id="main" role="main">
                                <article class="quiz">
                                <table border="1">
                                <?php
                                if ($this->question_entries->record_count() == 0) {
                                echo '<h2>No questions defined so
far...</h2>';
                                }
                                else {
                                echo '<thead>';
                                echo '<th>Question Name</th>';
                                echo '<th>Type</th>';
                                echo '<th>Correct Answer</th>';
                                echo '<th>Delete Question?</th>';
                                echo '</thead>';
                                echo '<tbody>';
                                foreach($results as $data) {
                                echo "<tr>
                                <td>$data-
                                <td>$data-
                                <td>$data-
                                <td><a
href='confirm_question_deletion?id=$data->question_no'>Delete</a></td>\n
                                </tr>";
                                }
                                echo '</tbody>';
                                }
                                ?>
                                </table>
                                </article>
                                </div>

                                <div id="related" class="sidebar" role="complementary">
                                <aside>
                                <h4>To delete a question from the question bank, click on the
"Delete" link of that row where the entry is located.</h4>
                                </aside>
                                </div>

```

list_dictionary_entry_view.php

```

                                <div id="main" role="main">
                                <article class="guide">
                                <table border="1">
                                <?php

```



```

        if ($this->dictionary_entries->record_count() == 0)
        {
            echo '<h2>No dictionary entries defined so
far...</h2>';
        }
        else {
            echo '<thead>';
            echo '<th>Entry Name</th>';
            echo '<th>Part of Speech</th>';
            echo '<th>Meaning</th>';
            echo '<th>Usage</th>';
            echo '<th>Category</th>';
            echo '<th>Edit Entry?</th>';
            echo '</thead>';
            echo '<tbody>';
            foreach($results as $data) {
                $edit_link =
site_url('edit_dictionary?id=' . $data->dictionary_id);
                echo "<tr>
                <td>$data-
                <td>$data-
                <td>$data-
                <td>$data-
                <td>$data-
                <td>$data-
                <td><a
                </tr>";
            }
            echo '</tbody>';
        }
    }
    </table>
    <p><?php echo $links; ?></p>
</article>
</div>

```

list_phrasebook_entry_view.php

```

    <div id="main" role="main">
        <article class="guide">
            <table border="1">
                <?php
                    if ($this->phrasebook_entries->record_count() == 0)
                    {
                        echo '<h2>No phrases defined so
far...</h2>';
                    }
                    else {
                        echo '<thead>';
                        echo '<th>Phrase</th>';
                        echo '<th>English translation</th>';
                        echo '<th>Edit Phrase?</th>';
                        echo '</thead>';
                        echo '<tbody>';
                        foreach($results as $data) {
                            $edit_link =
site_url('edit_phrase?id=' . $data->phrasebook_id);
                            echo "<tr>
                            <td>$data-
                            <td>$data-
                            <td><a
                            </tr>";
                        }
                    }
                }
            }
        }
    }

```

```

        echo '</tbody>';
    }
    ?>
</table>
<p><?php echo $links; ?></p>
</article>
</div>

login_view.php
<div id="main" role="main">
    <div class="login_main">
        <div id="login_form">
            <h4>If you already have an account for this system, log in
below.</h4>

            <div id="infoMessage"><?php echo $message; ?></div>
            <?php echo form_open('login_process'); ?>
            <p>
                <label for="identity">
                    Username: <?php echo form_input('identity',
set_value('identity')); ?>
                </label>
            </p>
            <p>
                <label for="password">
                    Password: <?php echo
form_password('password', set_value('password')); ?>
                </label>
            </p>
            <p>
                <label for="remember">Remember Me:</label>
                <?php echo form_checkbox('remember_user', '1', FALSE,
'id="remember");?>
            </p>
            <p>
                <?php echo form_submit('submit_login', 'Log me
in!'); ?>
            </p>
            <?php echo form_close(); ?>
        </div>
        <div id="login_forgotpassword">
            <h4 class="forgot_password"><a href="forgot_password"
title="Resets your password">Forgot password?</a></h4>
        </div>
        <div id="login_signup">
            <h4 class="user_signup"><a href="user_signup" title="Sign
up here">New to this system?</a><br />
                <a href="contributor_signup" title="Sign up for a
contributor account here">Want to become a contributor?</a></h4>
            <h4><a href="differences">Differences between the user and
the contributor types of accounts</a></h4>
        </div>
    </div>
</div>

maintain_user_groups_view.php
<div id="main" role="main">
    <article class="users">
        <table border="1">
            <?php
                echo '<thead>';
                echo '<th>Group Name</th>';
                echo '<th>Description</th>';
                echo '<th>Update Group?</th>';
                echo '<th>Delete Group?</th>';
            echo '</thead>';
            echo '<tbody>';
            foreach($results as $data) {
                echo "<tr>
                    <td>$data->ugrp_name</td>\n
                    <td>$data->ugrp_desc</td>\n

```

```

>ugrp_id'>Update</a></td>\n
>ugrp_id'>Delete</a></td>\n
    </tr>";
    }
    echo '</tbody>';
?>
</table>
<h4>To add a group not listed in the groups above, click <a
href="add_user_group">here</a>.</h4>
</article>
</div>

```

maintain_users_view.php

```

<div id="main" role="main">
  <article class="users">
    <table border="1">
      <?php
        echo '<thead>';
        echo '<th>Username</th>';
        echo '<th>User Email</th>';
        echo "<th>User's Name</th>";
        echo '<th>Member Since</th>';
        echo '<th>Update User?</th>';
        echo '<th>Delete User?</th>';
        echo '</thead>';
        echo '<tbody>';
        foreach($results as $data) {
          $name_array = array($data->
userinfo_first_name,$data->userinfo_last_name);
          $name = implode(" ",$name_array);
          echo "<tr>
            <td>$data->uacc_username</td>\n
            <td>$data->uacc_email</td>\n
            <td>$name</td>\n
            <td>$data->uacc_date_added</td>\n
            <td><a href='update_user?id=$data->
uacc_id'>Update</a></td>\n
            <td><a href='delete_user?id=$data->
uacc_id'>Delete</a></td>\n
          </tr>";
        }
        echo '</tbody>';
      ?>
    </table>
  </article>
</div>

```

no_links_approved_view.php

```

<div id="main" role="main">
  <article class="main_class">
    <?php
      if ($links_approved) {
        echo '<p>' . $links_approved . ' links approved.</p>';
      }
      ?>
      <p><?php echo $links_rejected; ?> links rejected.</p>
      <p><center><a href="index_admin">Return home</a></center></p>
    </article>
  </div>

```

no_links_selected_view.php

```

<div id="main" role="main">
  <article class="main_class">
    <h2>No links are selected for approval.</h2>
  </article>
</div>

```

no_record_view.php

```

<div id="main" role="main">
    <article class="main_class">
        <p>No record bearing that record ID exists.</p>
    </article>
</div>

organize_bank_view.php
<div id="main" role="main">
    <article class="quiz">
        <h3>To organize the bank's instructions, please select an option below: </h3>
        <ul>
            <li><a href="add_instruction">Add instruction for the quiz</a></li>
            <li><a href="edit_instruction">Edit existing instruction for the
quiz</a></li>
            <li><a href="delete_instruction">Delete instruction for the quiz</a></li>
        </ul>
    </article>
</div>

parameter_missing_view.php
<div id="main" role="main">
    <article class="main_class">
        <h3>Parameter missing!</h3>
    </article>
</div>

phrasebook_index_view.php
<div id="main" role="main">
    <article class="guide">
        <h3>To view the phrasebook entries starting with that particular letter, select a
letter below.</h3>
        <p>
            <?php
                $range = range('A','Z');
                foreach ($range as $char) {
                    echo "<a href=' " . site_url("phrasebook/view_entries/" .
$char) . "'> " . "$char</a> ";
                }
            ?>
        </p>
    </article>
</div>

phrasebook_view.php
<div id="main" role="main">
    <article class="guide">
        <table class="tablesorter" cellspacing="3">
            <?php
                if ($this->phrasebook_entries->record_count() == 0)
                {
                    echo '<h2>No phrases defined so
far...</h2>';
                }
                else {
                    echo '<thead>';
                    echo '<th>Phrase</th>';
                    echo '<th class="{sorter:
false}>Syllabic Pronunciation</th>';
                    echo '<th class="{sorter:
false}>English translation</th>';
                    echo '<th>Category/Tags</th>';
                    echo '<th>Audio Link</th>';
                    echo '</thead>';
                    echo '<tbody>';
                    foreach($results as $data) {
                        $phrase_link =
site_url('view_phrasebook_entry?id=' . $data->phrasebook_id);
                        if ($data->phrasebook_audio_link) {
                            echo "<tr>
                                <td><a
href='$phrase_link'>$data->phrasebook_phrase</a></td>\n

```

```

>phrasebook_pronunciation</td>\n
>phrasebook_translation</td>\n
>phrasebook_category</td>\n
src='$data->phrasebook_audio_link'>
src='$data->phrasebook_audio_link'>
    </audio></td>\n
}
else {
    echo "<tr>
        <td><a
            href='$phrase_link'>$data->phrasebook_phrase</a></td>\n
            <td>$data-
                >phrasebook_pronunciation</td>\n
                <td>$data-
                    >phrasebook_translation</td>\n
                    <td>$data-
                        >phrasebook_category</td>\n
                        <td>N/A</td>\n
                    </tr>";
        }
    }
    echo '</tbody>';
}
?>
</table>
<p><?php echo $links; ?></p>
<p>
    <?php
        $range = range('A','Z');
        foreach ($range as $char) {
            echo "<a href=' " .
                site_url("phrasebook/view_entries/" . $char) . "'> " . "$char</a> ";
        }
    ?>
</p>
</article>
</div>

```

query_failure_view.php

```

<div id="main" role="main">
    <article class="main_class">
        <h3>Query failed!</h3>
    </article>
</div>

```

quiz_results_view.php

```

<div id="main" role="main">
    <article class="quiz">
        <center>
            <?php
                $percentage = ($correct/$questions) * 100;
            ?>
            <h2>You have taken a quiz, and here are your results: </h2>

            <?php
                foreach ($right_wrong as $row) {
                    echo "<h3>$row</h3>";
                }
            ?>

            <h3>Your Score: <?php echo $correct . "/" . $questions; ?> </h3>
            <h3>For a total of <?php echo $percentage; ?> percent. </h3>

```

```

                <h4><a href="index_user">Return to home page</a></h4>
            </center>
        </article>
    </div>

reset_password_view.php
<h1>Change Password</h1>

<div id="infoMessage"><?php echo $message;?></div>

<?php echo form_open('reset_password' . $code);?>

    <p>
        New Password (at least <?php echo $min_password_length;?> characters long): <br />
        <?php echo form_input($new_password);?>
    </p>

    <p>
        Confirm New Password: <br />
        <?php echo form_input($new_password_confirm);?>
    </p>

    <?php echo form_input($user_id);?>
    <?php echo form_hidden($csrf); ?>

    <p><?php echo form_submit('submit', 'Change');?></p>

<?php echo form_close();?>

search_dictionary_view.php
    <div id="main" role="main">
        <article class="guide_edit">
            <div class="search_inputbox">
                <input type="text" id="search_input"
name="dictionary_query" />
            </div>
            <h3><strong>Search results for: </strong> <span
id="dictionary_title">Keyword</span></h3>
            <div id="result_data">
                </div>
            </table>
        </article>
    </div>

    <div id="related" class="sidebar" role="complementary">
        <aside>
            <center>
                <ul>
                    <li>As you type, the search field will return a list
of Filipino terms listed in the system's dictionary that contain the entire word or part of the
word being searched for.</li>
                </ul>
            </center>
        </aside>
    </div>

search_for_phrase_view.php
    <div id="main" role="main">
        <article class="guide_edit">
            <div class="search_inputbox">
                <input type="text" id="search_input" name="phrase_query" />
            </div>
            <h3><strong>Search results for: </strong> <span
id="phrase_title">Keyword</span></h3>
            <div id="result_data">
                </div>
            </table>
        </article>
    </div>

```

```

        <div id="related" class="sidebar" role="complementary">
            <aside>
                <center>
                    <ul>
                        <li>As you type, the search field will return a list
of Filipino phrases that contain the entire word or part of the word being searched for.</li>
                    </ul>
                </center>
            </aside>
        </div>

```

```

search_grammar_view.php
    <div id="main" role="main">
        <article class="guide_edit">
            <div class="search_inputbox">
                <input type="text" id="search_input" name="grammar_query"
/>
            </div>
            <h3><strong>Search results for: </strong> <span
id="grammar_title">Keyword</span></h3>
            <div id="result_data">
                </div>
            </table>
        </article>
    </div>

```

```

suggest_changes_view.php
<div id="main" class="links" role="main">
    <?php echo validation_errors(); ?>
    <?php echo form_open('suggest_changes/process'); ?>
    <p>Guide to be checked:
        <?php echo form_radio('guide_type','dictionary'); ?>Dictionary
        <?php echo form_radio('guide_type','grammar'); ?>Grammar
        <?php echo form_radio('guide_type','phrasebook'); ?>Phrasebook
        <?php echo form_radio('guide_type','general'); ?>General
    </p>
    <p>Suggestion/Concern
    </p>
    <p>Message:<br /> <textarea name="message" rows="10" cols="80"></textarea>
    </p>
    <?php echo $math_captcha_question; ?>
    <?php echo form_input('math_captcha'); ?>
    <br /><br />
    <?php echo form_submit('submit_email','Send Email'); ?>
    <?php echo form_reset('reset_email','Reset Form'); ?>
    <?php echo form_close(); ?>
</div>

```

```

<div id="related" class="sidebar" role="complementary">
    <aside>
        <h4>INSTRUCTIONS:</h4>
        <ul>
            <li>To suggest any guide changes to the SWF staff, specify the guide to be
checked for review, and elaborate what changes to be made in the message body.</li>
            <li>Each email will be reviewed by the staff.</li>
            <li>If you need access to the guides, they are available <a href="guides"
target="_blank">here</a>.</li>
        </ul>
    </aside>
</div>

```

```

take_quiz_view.php
    <div id="main" role="main">
        <article class="quiz">
            <?php
                if ($this->input->get('id')) {
                    $lesson_id = $this->input->get('id');
                    foreach ($instructions as $ins) {
                        echo "<h3>" . $ins->instruction_text . "</h3>";
                    }
                    echo form_open('quiz_process');
                    $i = 1;
                    $j = 0;

```

```

        $choice = array();
        $correct = array();
        $correct_answers = 0;
        foreach ($results as $data) {
            echo "<div>\n";
            echo "<p>\n";
            echo "$i" . ". \n";
            echo "$data->question_title\n";
            echo "</p>\n";
            echo "<h4>\n";
            echo "A. <input type='radio' name='choice[$j]'
value='$data->choice_a' /> $data->choice_a" . "<br />\n";
            echo "B. <input type='radio' name='choice[$j]'
value='$data->choice_b' /> $data->choice_b" . "<br />\n";
            if ($data->choice_c) {
                echo "C. <input type='radio'
name='choice[$j]' value='$data->choice_c' /> $data->choice_c" . "<br />\n";
            }
            if ($data->choice_d) {
                echo "D. <input type='radio'
name='choice[$j]' value='$data->choice_d' /> $data->choice_d" . "<br />\n";
            }
            echo "</h4>";
            echo "</div>";
            $correct[$j] = $data->correct_answer;
            $i++;
            $j++;
        }
        $entries = array('entries' => $j);
        $correct_answers = array('correct' => $correct);
        $this->session->set_userdata($entries);
        $this->session->set_userdata($correct_answers);
        // checking of answers will only be done once the answering
is complete

        echo form_submit('submit_quiz','Submit Quiz');
        echo form_reset('reset_quiz','Start Over');
        echo form_hidden('lesson_id',$lesson_id);
        echo form_close();
    }
    else {
        $this->load->view('parameter_missing_view');
    }
?>
</article>
</div>

<div id="related" class="sidebar" role="complementary">
<aside>
    <h4>The quiz will display a maximum of ten questions, whose types are
either multiple choice or true or false.</h4>
    <h4>To answer a question, click the radio button beside the letter of your
choice.</h4>
    <h4>Be sure to answer all questions. If you press Submit Quiz without
answering all entries, the browser will alert you and will force you to finish your quiz.</h4>
</aside>
</div>
<script type="text/javascript">

    // checks if the quiz is completely answered, if not, an alert will be issued to
force the user to complete the quiz
    function checkEntries() {
        var number_of_problems = $.parseJSON(<?php echo json_encode(json_encode($i
- 1)); ?>);
        var answered_questions = $( "input:checked" ).length;
        if (answered_questions != number_of_problems) {
            return false;
        }
        return true;
    }

    $("form").submit(function(event) {

```



```

        event.preventDefault();
        var readyToSubmit = checkEntries();
        var answered = new Array();
        var correctAnswersArray = new Array();
        var numberCorrect = 0;
        answered = $.parseJSON(<?php echo json_encode(json_encode($choice)); ?>);
        correctAnswers = $.parseJSON(<?php echo
json_encode(json_encode($correct)); ?>);
        if (!readyToSubmit) {
            alert("You only answered " + answered_questions + " questions out
of " + number_of_problems + " questions. Please complete the quiz first before submitting.");
            return false;
        }
        else {
            return true;
        }
    });
</script>

```

update_user_failure_view.php

```

<div id="main" role="main">
    <article class="main_class">
        <h3>Updating of user details unsuccessful.</h3>
    </article>
</div>

```

update_user_group_failure_view.php

```

<div id="main" role="main">
    <article class="main_class">
        <h3>Updating of user group details unsuccessful.</h3>
    </article>
</div>

```

update_user_group_success_view.php

```

<div id="main" role="main">
    <article class="main_class">
        <h3>Updating of user group details successful.</h3>
    </article>
</div>

```

update_user_group_view.php

```

    <div id="main" role="main">
        <article class="users">
            <?php
                if($this->input->get('id')) {
                    $sql_where = 'ugrp_id' = 'id';
                    $result = $this->flexi_auth-
>get_groups(FALSE,$sql_where)->row();
                    if ($result) {
                        $group_name = $result->ugrp_name;
                        $group_desc = $result->ugrp_desc;

                        $group_form_data = array(
=> 'group_name',
=> $group_name,
=> '60',
=> 'width:33%',
                                'name'
                                'value'
                                'size'
                                'style'
                                );
                        $description_form_data = array(
=> 'group_desc',
=> $group_desc,
=> '60',
                                'name'
                                'value'
                                'size'

```

```

=> 'width:33%',
                                                    'style'
                                                    );

form_open("update_user_group/process");
                                                    echo
                                                    echo '<p>Group Name: ';
                                                    echo form_input($group_form_data);
                                                    echo '</p>';

                                                    echo '<p>Last Name: ';
                                                    echo

form_input($description_form_data);
                                                    echo '</p>';

                                                    echo
form_submit('update_user_group_details', 'Update User Group');
                                                    echo
form_reset('cancel_user_group_details', 'Cancel Update');
                                                    echo form_hidden('group_id', $result-
>ugrp_id);
                                                    }
                                                    else {
                                                    $this->load->view('no_record_view');
                                                    }
                                                    }
                                                    else {
                                                    $this->load->view('parameter_missing_view');
                                                    }
                                                    }
                                                    ?>
                                                    </article>
                                                    </div>

update_user_success_view.php
<div id="main" role="main">
    <article class="main_class">
        <h3>Updating of user details successful.</h3>
    </article>
</div>

update_user_view.php
<div id="main" role="main">
    <article class="users">
        <?php
            if($this->input->get('id')) {
                $userid_to_get = $this->input->get('id');
                $result = $this->flexi_auth-
>get_user_by_id($userid_to_get)->row();

                if ($result) {
                    $username = $result->uacc_username;
                    $email = $result->uacc_email;
                    $first_name = $result->userinfo_first_name;
                    $last_name = $result->userinfo_last_name;

                    $username_form_data = array(
                                                    'name'
=> 'username',
                                                    'value'
=> $username,
                                                    'size'
=> '60',
                                                    'style'
=> 'width:33%',
                                                    'readonly' => 'true',
                                                    'disabled' => 'true'
                                                    );
                    $email_form_data = array(

```

```

=> 'email',
=> $email,
=> '100',
=> 'width:33%',
    'readonly' => 'true',
    'disabled' => 'true'
);
$firstname_form_data = array(
=> 'first_name',
=> $first_name,
=> '60',
=> 'width:33%',
    'readonly' => 'true',
    'disabled' => 'true'
);
$lastname_form_data = array(
=> 'last_name',
=> $last_name,
=> '60',
=> 'width:33%',
    'readonly' => 'true',
    'disabled' => 'true'
);

echo form_open("update_user/process");
echo '<p>Username: ';
echo form_input($username_form_data);
echo '</p>';
echo '<p>Email: ';
echo form_input($email_form_data);
echo '</p>';
echo '<p>First Name: ';
echo form_input($firstname_form_data);
echo '</p>';
echo '<p>Last Name: ';
echo form_input($lastname_form_data);
echo '</p>';
echo form_submit('update_user_details',

'Update User Details');
'Cancel Update');
>uacc_id);
}
else {
    $this->load->view('no_record_view');
}
}
else {
    $this->load->view('parameter_missing_view');
}
}
?>
</article>

```

```

        </div>

user_deleted_view.php
    <h4>User successfully deleted.</h4>

user_signup_success_view.php
    <div id="main" role="main">
        <article class="regcomplete_main">
            <h3>Registration is now complete.</h3>
            <h4>Click here to <a href="login">log in</a> now.</h4>
        </article>
    </div>

user_signup_view.php
    <div id="main" role="main">
        <article class="signup_main">
            <section id="signup_form">
                <h3>Signing-up is easy.</h3>
                <h4>All fields are required.</h4>
                <div id="infoMessage"><?php echo $message;?></div>
                <?php echo form_open('reg_user'); ?>
                <center>
                    <table cellpadding="1">
                        <tr>
                            <th>User Login Information</th>
                            <tr>
                                <td>
                                    Username:
                                </td>
                                <td>
                                    <?php echo form_input('username',
set_value('username')); ?>
                                </td>
                            </tr>
                            <tr>
                                <td>
                                    Password:
                                </td>
                                <td>
                                    <?php echo form_password('password',
set_value('password')); ?>
                                </td>
                            </tr>
                            <tr>
                                <td>
                                    Confirm Password:
                                </td>
                                <td>
                                    <?php echo
form_password('confirm_password', set_value('confirm_password')); ?>
                                </td>
                            </tr>
                            <tr>
                                <th>User Details</th>
                                <tr>
                                    <td>
                                        First Name:
                                    </td>
                                    <td>
                                        <?php echo form_input('first_name',
set_value('first_name')); ?>
                                    </td>
                                </tr>
                                <tr>
                                    <td>
                                        Last Name:
                                    </td>
                                    <td>

```

```

set_value('last_name')); ?>
</td>
</tr>
<tr>
<td>
Email Address:
</td>
<td>
<?php echo form_input('email',
set_value('email')); ?>
</td>
</tr>
</tr>
</table>
</center>
<center>
<?php echo form_submit('submit_reg_user',
'Sign me up!') ?>
<?php echo form_reset('reset', 'Reset Form')
?>
</center>
<?php echo form_close(); ?>
</section>
</article>
</div>

```

view_dictionary_entry_view.php

```

<div id="main" role="main">
  <article class="guide">
    <?php
      if($this->input->get('id')) {
        $dictionary_no = $this->input->get('id');
        $query = $this->db->get_where('dictionary',array('dictionary_id' =>
$dictionary_no));

        if ($query->num_rows() > 0) {
          $row = $query->row();
          $term = $row->dictionary_term;
          $part_of_speech = $row->dictionary_partofspeech;
          $pronunciation = $row->dictionary_pronunciation;
          $meaning = $row->dictionary_meaning;
          $usage = $row->dictionary_usage;
          $image_link = $row->dictionary_image_link;
          $audio_link = $row->dictionary_audio_link;
          $category = $row->dictionary_category;

          echo "<h3>Dictionary Entry Detail</h3>";
          echo "<p>Term: <strong>" . $term . "</strong></p>";
          echo "<p>Part of Speech: <strong>" . $part_of_speech .
"</strong></p>";

          echo "<p>Pronunciation:" . $pronunciation . "</p>";
          echo "<p>Meaning: " . $meaning . "</p>";
          echo "<p>Usage: " . $usage . "</p>";
          if ($image_link) {
            echo "<img src='$image_link' width='350'
height='350' title='Image of $term' />";
          }
          if ($audio_link) {
            echo "<audio controls>
              <source src='$audio_link'>
              <embed src='$audio_link'>
            </audio>";
          }
          echo "<p>Category: " . $category . "</p>";

          $if_staff = $this->flexi_auth->in_group('staff');
          if ($if_staff) {
            echo "<p><a
href='edit_dictionary?id=$dictionary_no'>Edit Entry</a></p>\n";
          }
        }
      }
    </?php
  </article>
</div>

```

```

        }
        else {
            $this->load->view('no_record_view');
        }
    }
    else {
        $this->load->view('parameter_missing_view');
    }
}
?>
</article>
</div>

```

view_grammar_view.php

```

<div id="main" role="main">
    <article class="guide">
        <?php
            if($this->input->get('id')) {
                $grammar_id = $this->input->get('id');
                $query = $this->db->get_where('grammar',array('grammar_id' =>
                    $grammar_id));

                if ($query->num_rows() > 0) {
                    $row = $query->row();
                    $rule = $row->grammar_rule;
                    $usage = $row->grammar_usage;
                    $example = $row->grammar_example;
                    $category = $row->grammar_category;

                    echo "<h3>Grammar Rule Entry Detail</h3>";
                    echo "<p>Rule: <strong>" . $rule . "</strong></p>";
                    echo "<p>Usage: <strong>" . $usage . "</strong></p>";
                    echo "<p>Example: <strong>" . $example . "</strong></p>";
                    echo "<p>Category: " . $category . "</p>";

                    $if_staff = $this->flexi_auth->in_group('staff');
                    if ($if_staff) {
                        echo "<p><a
href='edit_grammar_rule?id=$grammar_id'>Edit Entry</a></p>\n";
                    }
                }
                else {
                    $this->load->view('no_record_view');
                }
            }
            else {
                $this->load->view('parameter_missing_view');
            }
        }
    }
?>
</article>
</div>

```

view_instruction_view.php

```

<div id="main" role="main">
    <article class="quiz">
        <?php
            if($this->input->get('id')) {
                $instruction_no = $this->input->get('id');
                $query = $this->db->
>get_where('bank_instructions',array('instructions_id' => $instruction_no));

                if ($query->num_rows() > 0) {
                    $row = $query->row();
                    $text = $row->instruction_text;
                    $lesson_id = $row->lesson_id;

                    echo "<h3>Instruction Detail</h3>";
                    echo "<p>Text: " . $text . "</p>";
                    echo "<p>Lesson ID: " . $lesson_id . "</p>";
                }
            }
        }
    }
?>
</article>
</div>

```

```

        echo "<p><a href='edit_instruction?id=${instruction_no}'>Edit
Instruction</a></p>\n";
    }
    else {
        $this->load->view('no_record_view');
    }
}
else {
    $this->load->view('parameter_missing_view');
}
?>
</article>
</div>

```

view_lessons_view.php

```

<div id="main" role="main">
    <article class="lesson">
        <table>
            <?php
                if ($this->lesson_entries->record_count() == 0) {
                    echo '<h2>No lessons defined so
far...</h2>';
                }
                else {
                    echo '<thead>';
                    echo '<th>Lesson Name</th>';
                    echo "<th>Lesson Description</th>";
                    echo '</thead>';
                    echo '<tbody>';
                    foreach($results as $data) {
                        echo "<tr>
                            <td><a
href='lesson_entry?id=${data->lesson_id}'>${data->lesson_title}</a></td>\n
                            <td>${data-
>lesson_description}</td>\n
                                </tr>";
                    }
                    echo '</tbody>';
                }
            echo '</table>';
        ?>
    </article>
</div>

```

view_link_view.php

```

<div id="main" role="main">
    <article class="links">
        <?php
            if($this->input->get('id')) {
                $link_no = $this->input->get('id');
                $query = $this->db->get_where('links',array('link_id' =>
$link_no));

                if ($query->num_rows() > 0) {
                    $row = $query->row();
                    $title = $row->link_title;
                    $description = $row->link_description;
                    $url = $row->link_url;
                    $contributor = $row->contributed_by;

                    echo "<h3>Link Detail</h3>";
                    echo "<p>Link Title: " . $title . "</p>";
                    echo "<p>Description: " . $description . "</p>";
                    echo "<p>Contributed by: " . $contributor . "</p>";
                    echo "<p>URL: <a href='" . $url . "'>$url</a></p>";
                }
            }
        ?>
    </article>
</div>

```

```

        $current_user = $this->flexi_auth->get_user_identity();
        if ($current_user === $contributor) {
            echo "<p><a href='edit_link?id=$link_no'>Edit
Link</a></p>\n";
        }
    }
    else {
        $this->load->view('no_record_view');
    }
}
else {
    $this->load->view('parameter_missing_view');
}
?>
</article>
</div>

```

view_phrasebook_entry_view.php

```

<div id="main" role="main">
    <article class="guide">
        <?php
            if($this->input->get('id')) {
                $phrasebook_no = $this->input->get('id');
                $query = $this->db->get_where('phrasebook',array('phrasebook_id' =>
$phrasebook_no));

                if ($query->num_rows() > 0) {
                    $row = $query->row();
                    $term = $row->phrasebook_phrase;
                    $pronunciation = $row->phrasebook_pronunciation;
                    $translation = $row->phrasebook_translation;
                    $audio_link = $row->phrasebook_audio_link;
                    $category = $row->phrasebook_category;

                    echo "<h3>Phrasebook Entry Detail</h3>";
                    echo "<p>Term: <strong>" . $term . "</strong></p>";
                    echo "<p>Pronunciation:" . $pronunciation . "</p>";
                    echo "<p>Translation: " . $translation . "</p>";
                    if ($audio_link) {
                        echo "<audio controls>
                            <source src='$audio_link'>
                            <embed src='$audio_link'>
                        </audio>";
                    }
                    echo "<p>Category: " . $category . "</p>";

                    $if_staff = $this->flexi_auth->in_group('staff');
                    if ($if_staff) {
                        echo "<p><a
href='edit_phrase?id=$phrasebook_no'>Edit Entry</a></p>\n";
                    }
                }
            }
            else {
                $this->load->view('no_record_view');
            }
        }
        else {
            $this->load->view('parameter_missing_view');
        }
    }
?>
</article>
</div>

```



```

view_progress_view.php
    <div id="main" role="main">
        <article class="lesson">
            <table>
                <?php
                    if ($this->progress_model->record_count() == 0) {
                        echo '<h2>You have not taken any quizzes so
far...</h2>';
                    }
                    else {
                        $i = 1;
                        $j = 0;
                        echo '<thead>';
                        echo '<th>Quiz Number</th>';
                        echo '<th>Lesson Number</th>';
                        echo '<th>Number Correct</th>';
                        echo '<th>Number of Items</th>';
                        echo '<th>Percent</th>';
                        echo '<th>Date Taken</th>';
                        echo '</thead>';
                        echo '<tbody>';
                        foreach($results as $data) {
                            $percentage_decimal[$j] = ($data-
>no_correct) / ($data->no_of_questions);
                            $date_taken = date($data-
>timestamp_completed);
                            $percentagetimes100[$j] =
$percentage_decimal[$j] * 100;
                            echo "<tr>
                                <td>$i</td>\n
                                <td>$data->lesson_id</td>\n
                                <td>$data->no_correct</td>\n
                                <td>$data-
>no_of_questions</td>\n
                                <td>$percentagetimes100[$j]</td>\n
                                <td>$date_taken</td>\n
                                </tr>";
                            $i++;
                        }
                        echo '</tbody>';
                    }
                <?php
            </table>
        </article>
    </div>

    <div id="related" class="sidebar" role="complementary">
        <aside>
            <h4>The progress of all your quizzes ever taken are recorded and
presented here.</h4>
        </aside>
    </div>

```

```

view_question_view.php
<div id="main" role="main">
    <article class="question">
        <?php
            if($this->input->get('id')) {
                $question_no = $this->input->get('id');
                $query = $this->db->get_where('question_bank',array('question_no'
=> $question_no));

                if ($query->num_rows() > 0) {
                    $row = $query->row();
                    $title = $row->question_title;
                    $type = $row->question_type;
                    $choice_a = $row->choice_a;
                    $choice_b = $row->choice_b;
                    $choice_c = $row->choice_c;
                }
            }
        </?php
    </article>
</div>

```

```

        $choice_d = $row->choice_d;
        $choice_correctanswer = $row->correct_answer;
        $lesson_id = $row->lesson_id;

        echo "<p>Question Title: " . $title . "</p>";
        echo "<p>Question Type: " . $type . "</p>";
        echo "<p>Choice A: " . $choice_a . "</p>";
        echo "<p>Choice B: " . $choice_b . "</p>";
        echo "<p>Choice C: " . $choice_c . "</p>";
        echo "<p>Choice D: " . $choice_d . "</p>";
        echo "<p>Correct Answer: " . $choice_correctanswer .

"</p>";

        echo "<p>Lesson ID: " . $lesson_id . "</p>";

        echo "<p><a
href='edit_question_proper?id=$question_no'>Edit Question</a></p>\n";
    }
    else {
        $this->load->view('no_record_view');
    }
}
else {
    $this->load->view('parameter_missing_view');
}
?>
</article>
</div>

```

SQL SCRIPT

```

-- phpMyAdmin SQL Dump
-- version 3.5.1
-- http://www.phpmyadmin.net
--
--
-- Host: localhost
-- Generation Time: Apr 04, 2013 at 01:59 PM
-- Server version: 5.5.24-log
-- PHP Version: 5.4.3

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

SET time_zone = "+00:00";

```

```

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;

/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;

/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

/*!40101 SET NAMES utf8 */;

--

-- Database: `filles`

--

CREATE DATABASE `filles` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci;

USE `filles`;

-----

--

-- Table structure for table `bank_instructions`

--

CREATE TABLE IF NOT EXISTS `bank_instructions` (

  `instruction_id` int(2) NOT NULL AUTO_INCREMENT COMMENT 'Instruction ID',

  `instruction_text` varchar(1024) NOT NULL COMMENT 'Instruction Text',

  `lesson_id` int(2) NOT NULL COMMENT 'Lesson ID of the Instruction',

  PRIMARY KEY (`instruction_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Instructions for display in the Question Bank'
AUTO_INCREMENT=7 ;

--

-- Dumping data for table `bank_instructions`

--

```

```

INSERT INTO `bank_instructions` (`instruction_id`, `instruction_text`, `lesson_id`) VALUES

(1, 'Select the word that reflects the statement.', 1),

(2, 'Select TRUE if the sentence is correct and FALSE if the sentence is incorrect.', 2),

(3, 'Choose the letter of the missing marker ang or si. The plural form of ang is ang mga and si
is sina.', 3),

(4, 'Complete the following sentences by choosing the letter of the appropriate pronouns
(indicated by their English equivalents).', 4),

(5, 'Choose the letter of the correct demonstrative substitute.', 5),

(6, 'Choose the letter of the correct plural adjective subject. ', 6);

```

```

-- -----

```

```

--

```

```

-- Table structure for table `changes`

```

```

--

```

```

CREATE TABLE IF NOT EXISTS `changes` (
  `change_id` int(5) unsigned NOT NULL AUTO_INCREMENT COMMENT 'Change #',
  `sender_id` int(8) NOT NULL COMMENT 'ID # of the Sender who suggested the Change',
  `type` enum('dictionary','phrasebook','grammar','general') NOT NULL COMMENT 'Type of
Suggestion',
  `content` varchar(4096) NOT NULL COMMENT 'Content of the Suggestion',
  `timestamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`change_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

```

```

--

```

```

-- Dumping data for table `changes`

```

```

--

```

```
INSERT INTO `changes` (`change_id`, `sender_id`, `type`, `content`, `timestamp`) VALUES
(1, 6, 'general', 'test', '2013-04-01 02:41:59');
```

```
-- -----
```

```
--
```

```
-- Table structure for table `ci_sessions`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `ci_sessions` (
  `session_id` varchar(40) NOT NULL DEFAULT '0',
  `ip_address` varchar(16) NOT NULL DEFAULT '0',
  `user_agent` varchar(120) DEFAULT NULL,
  `last_activity` int(10) unsigned NOT NULL DEFAULT '0',
  `user_data` text NOT NULL,
  PRIMARY KEY (`session_id`),
  KEY `last_activity` (`last_activity`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Dumping data for table `ci_sessions`
```

```
--
```

```
INSERT INTO `ci_sessions` (`session_id`, `ip_address`, `user_agent`, `last_activity`,
`user_data`) VALUES
('247a85099f669d14dbad076885f7fcb1', '127.0.0.1', 'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:20.0)
Gecko/20100101 Firefox/20.0', 1365083700,
'a:2:{s:9:"user_data";s:0:"";s:10:"flexi_auth";a:7:{s:15:"user_identifier";b:0;s:7:"user_id";b:0;
s:5:"admin";b:0;s:5:"group";b:0;s:10:"privileges";b:0;s:22:"logged_in_via_password";b:0;s:19:"log
in_session_token";b:0;}}');
```

```

-----

--

-- Table structure for table `dictionary`

--

CREATE TABLE IF NOT EXISTS `dictionary` (

  `dictionary_id` int(5) unsigned NOT NULL AUTO_INCREMENT COMMENT 'Dictionary Entry ID',

  `dictionary_term` varchar(100) NOT NULL COMMENT 'Dictionary Term',

  `dictionary_partofspeech` varchar(200) DEFAULT NULL COMMENT 'Term's Part of Speech',

  `dictionary_pronunciation` varchar(256) DEFAULT NULL COMMENT 'Syllabic pronunciation of the term',

  `dictionary_meaning` varchar(1000) NOT NULL COMMENT 'Dictionary Meaning',

  `dictionary_usage` varchar(1000) DEFAULT NULL COMMENT 'Sample Usage of the Dictionary Term',

  `dictionary_image_link` varchar(1024) DEFAULT NULL COMMENT 'Link to Image of Term',

  `dictionary_audio_link` varchar(1024) DEFAULT NULL COMMENT 'Audio Pronunciation Link of the Term',

  `dictionary_category` varchar(100) DEFAULT NULL COMMENT 'Category of the Entry',

  PRIMARY KEY (`dictionary_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Dictionary Entry Index' AUTO_INCREMENT=49 ;

--

-- Dumping data for table `dictionary`

--

INSERT INTO `dictionary` (`dictionary_id`, `dictionary_term`, `dictionary_partofspeech`,
`dictionary_pronunciation`, `dictionary_meaning`, `dictionary_usage`, `dictionary_image_link`,
`dictionary_audio_link`, `dictionary_category`) VALUES

(1, 'maya', 'noun (pangngalan)', 'MAH-ya', 'The former national bird of the Philippines. Its
common name in English is European tree sparrow.', 'Maraming mga maya sa kapaligiran ng
Maynila.', NULL, NULL, NULL),

```

(2, 'bayan', 'noun (pangngalan)', 'BAH-yan', 'town, city, country, town proper', 'Nasa bayan ang munisipyo.', NULL, NULL, NULL),

(3, 'baso', 'noun (pangngalan)', 'BA-soh', 'drinking glass', 'Nasaan ang baso mo?', NULL, NULL, NULL),

(4, 'gripo', 'noun (pangngalan)', 'GRI-po', 'water faucet', "'Wag tayo maginom ng tubig na yari nang direktso sa gripo.', NULL, NULL, NULL),

(5, 'Paalam', 'pang-abay (adverb)', 'PAH-al-am', 'Goodbye', 'Paalam na po, Kuya Luis.', NULL, '/filles/sounds/paalam.wav', 'Greetings'),

(6, 'Mabuti', 'Pang-uri (Adjective)', 'mah-BU-ti', 'I am fine', 'Mabuti naman ako, Regie.', NULL, '/filles/sounds/mabuti.wav', 'Greetings'),

(7, 'Paki-', 'pang-abay (adverb)', 'PAH-ki', 'Please', 'Pakigabay mo si Jillian sa gawaing-bahay niya.', NULL, '/filles/sounds/paki.wav', 'Civilities'),

(8, 'Salamat', 'verb', NULL, 'Thank you', 'Salamat po sa inyong panglilibre, Tito Jun.', NULL, '/filles/sounds/salamat.wav', 'Civilities'),

(9, 'Kaliwa', 'pang-uri (adjective)', NULL, 'turn left', 'Kaliwa ka sa may bandang munisipyo.', NULL, '/filles/sounds/kaliwa.wav', 'Directions'),

(10, 'Kanan', 'adjective (pang-uri)', NULL, 'turn right', 'Pag galing kang EDSA, kumanaan ka kung pupunta ka ng TriNoma.', NULL, '/filles/sounds/kanan.wav', 'Directions'),

(11, 'Diretso', 'adjective (pang-uri)', NULL, 'straight ahead', 'Diretso ka sa dulo para makapunta ka sa chapel.', NULL, '/filles/sounds/diretso.wav', 'Directions'),

(12, 'Taas', 'adjective (pang-uri)', NULL, 'up', 'Nasa taas ka po ba, Steve?', NULL, '/filles/sounds/taas.wav', 'Directions'),

(13, 'Baba', 'adjective (pang-uri)', NULL, 'down', 'Nasa baba si Tita Carol, sa salas namin.', NULL, '/filles/sounds/baba.wav', 'Directions'),

(14, 'Dito', 'adjective (pang-uri)', NULL, 'Here', 'Dito ako nakatira sa Tandang Sora.', NULL, '/filles/sounds/dito.wav', 'Directions'),

(15, 'Diyan', 'adjective (pang-uri)', NULL, 'there (near direction)', 'Diyan ka ba nagtatrabaho sa Makati, James?', NULL, '/filles/sounds/diyan.wav', 'Directions'),

(16, 'Doon', 'adjective (pang-uri)', NULL, 'There (far direction)', 'Doon siya sa may kusina.', NULL, '/filles/sounds/doon.wav', 'Directions'),

(17, 'Kabila', 'pang-uri (adjective)', NULL, 'Other side', 'Nasa kabila ang sakayan ng mga tricycle.', NULL, '/filles/sounds/kabila.wav', 'Directions'),

(18, 'Pera', 'Noun (pangngalan)', NULL, 'money', 'Meron akong pera na naka-impok sa bangko.', NULL, '/filles/sounds/pera.wav', 'Money'),

(19, 'Magkano?', 'pronoun', NULL, 'How much?', 'Magkano ba ang kwintas na ito?', NULL, '/filles/sounds/magkano.wav', 'Money'),

(20, 'Mahal (money)', 'adjective (pang-uri)', NULL, 'Expensive', 'Mahal ang pagkain na inorder mo, Ate Sheila.', NULL, '/filles/sounds/mahal.wav', 'Money'),

(21, 'Mura', 'adjective (pang-uri)', NULL, 'Cheap', 'Mura ang mga pang-regalo doon sa Divisoria.', NULL, '/filles/sounds/mura.wav', 'Money'),

(22, 'Ilan?', 'pronoun', NULL, 'How many?', 'Ilang pirasong kwintas ba ang kailangan kong ibili para may diskwento ako?', NULL, '/filles/sounds/ilan.wav', 'Money'),

(23, 'Isa', 'adjective (pang-uri)', NULL, 'One', 'Isa lang ba ang bibili mo, Karl?', NULL, '/filles/sounds/isa.wav', 'Count'),

(24, 'Dalawa', 'adjective (pang-uri)', NULL, 'Two', 'Dalawa lang ang anak namin ni Miriam.', NULL, '/filles/sounds/dalawa.wav', 'Count'),

(25, 'Tatlo', 'adjective (pang-uri)', NULL, 'Three', 'Tatlo ang malalaking unibersidad sa kapaligiran ng Katipunan Avenue.', NULL, '/filles/sounds/tatlo.wav', 'Count'),

(26, 'Malaki', 'adjective (pang-uri)', NULL, 'Big', 'Masyadong malaki ang binili mong jacket, Franco.', NULL, '/filles/sounds/malaki.wav', 'Size'),

(27, 'Maliit', 'adjective (pang-uri)', NULL, 'Small', 'Maliit pa lang ang apo ko, si Chloe.', NULL, '/filles/sounds/maliit.wav', 'Size'),

(28, 'Marami', 'adjective (pang-uri)', NULL, 'Many', 'Marami akong tinitindang bagay dito sa sukung tindahan ko.', NULL, '/filles/sounds/marami.wav', 'Amount'),

(29, 'Konti', 'adjective (pang-uri)', NULL, 'Few', 'Konti lang ang mga kamaganak ko sa Davao City.', NULL, '/filles/sounds/konti.wav', 'Amount'),

(30, 'Ano?', 'Pronoun', NULL, 'What?', 'Ano ang ibig sabihin ng wika?', NULL, '/filles/sounds/ano.wav', 'Questions'),

(31, 'Sino?', 'Pronoun', NULL, 'Who?', 'Sino ba ang bisor ng grupong ito?', NULL, '/filles/sounds/sino.wav', 'Questions'),

(32, 'Kailan?', 'Pronoun', NULL, 'When?', 'Kailan ba ang Mahal na Araw ngayong taon?', NULL, '/filles/sounds/kailan.wav', 'Questions'),

(33, 'Bakit?', 'Pronoun', NULL, 'Why?', 'Bakit kaya walang nyebe ang Pilipinas?', NULL, '/filles/sounds/bakit.wav', 'Questions'),

(34, 'Paano?', 'Pronoun', NULL, 'How?', 'Paano ba ang pag-ibig sa sariling bayan pag umalis ka ng bansa?', NULL, '/filles/sounds/paano.wav', 'Questions'),

(35, 'Saan?', 'Pronoun', NULL, 'Where?', 'Saan ang botika sa poblacion?', NULL, '/filles/sounds/saan.wav', 'Questions'),

(36, 'Alin?', 'Pronoun', NULL, 'Which?', 'Alin ba ang pinakamaganda sa mga buko pie na binebenta dito ngayon?', NULL, '/filles/sounds/alin.wav', 'Questions'),

(37, 'Oo / opo', 'Adjective (pang-uri)', NULL, 'Yes', 'Opo, masarap ang bangus sa Dagupan.', NULL, '/filles/sounds/oo_opo.wav', 'Answers'),

(38, 'Hindi', 'Adjective (pang-uri)', NULL, 'No', 'Hindi ko masabi kung matamis ang alak na ito.', NULL, '/filles/sounds/hindi.wav', 'Answers'),

(39, 'Hinto!', 'Verb (pandiwa)', NULL, 'Stop!', 'Hinto na sa pagsagot, Mark!', NULL, '/filles/sounds/hinto.wav', 'Imperative Words'),

(40, 'Masarap', 'Adjective (pang-uri)', NULL, 'Delicious', 'Masarap ang mga mangga sa Pilipinas, hinog man o hilaw.', NULL, '/filles/sounds/masarap.wav', 'Descriptions'),

(41, 'Mainit', 'Adjective (pang-uri)', NULL, 'Hot', 'Mag-ingat, mainit ang giniling na kape.', NULL, '/filles/sounds/mainit.wav', 'Descriptions'),


```

(42, 'Malamig', 'Adjective (pang-uri)', NULL, 'Cold', 'Malamig doon sa Tagaytay at sa Baguio.',
NULL, '/filles/sounds/malamig.wav', 'Descriptions'),

(43, 'Sige', 'Adverb (pang-abay)', NULL, 'Okay', 'Sige po, pupunta po ako sa Greenhills ngayon.',
NULL, '/filles/sounds/sige.wav', 'Answers'),

(44, 'Maganda', 'Adjective (pang-uri)', NULL, 'Beautiful', 'Maganda ang mga makikita sa mga
probinsya.', NULL, '/filles/sounds/maganda.wav', 'Descriptions'),

(45, 'Ako', 'pronoun', NULL, 'I', 'Ako ay isang Pilipino.', NULL, '/filles/sounds/ako.wav',
'Descriptions'),

(46, 'Tayo', 'pronoun', NULL, 'We', 'Tayo'y magkaisa para maunlad ang bansa natin.', NULL,
'/filles/sounds/tayo.wav', 'Descriptions'),

(47, 'Siya', 'pronoun', NULL, 'He / she', 'Nandito siya sa tabi mo.', NULL,
'/filles/sounds/siya.wav', 'Descriptions'),

(48, 'Sila', 'pronoun', NULL, 'They', 'Nandoon sila sa Batangas, kung saan sila nakatira.', NULL,
'/filles/sounds/sila.wav', 'Descriptions');

```

```

-- -----

```

```

--

```

```

-- Table structure for table `grammar`

```

```

--

```

```

CREATE TABLE IF NOT EXISTS `grammar` (
  `grammar_id` int(4) unsigned NOT NULL AUTO_INCREMENT COMMENT 'Grammar Rule ID',
  `grammar_rule` varchar(250) NOT NULL COMMENT 'Grammar Rule',
  `grammar_usage` varchar(512) NOT NULL COMMENT 'Grammar Rule's Usage',
  `grammar_example` varchar(512) NOT NULL COMMENT 'Grammar Rule's Example',
  `grammar_category` varchar(50) DEFAULT NULL COMMENT 'Grammar Rule's Category',
  PRIMARY KEY (`grammar_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Grammar Rule Index' AUTO_INCREMENT=2 ;

```

```

--

```

```

-- Dumping data for table `grammar`

```

```

--

INSERT INTO `grammar` (`grammar_id`, `grammar_rule`, `grammar_usage`, `grammar_example`,
`grammar_category`) VALUES

(1, 'Nang vs. ng', 'Nang - use it as a verb or an adverb; ng - use it as an adjective modifier',
'Nang - nang sumapit ang panahon; ng - ulat ng panahon', 'Usage');

-----

--

-- Table structure for table `lessons`

--

CREATE TABLE IF NOT EXISTS `lessons` (

  `lesson_id` int(2) unsigned NOT NULL AUTO_INCREMENT COMMENT 'Lesson ID',

  `lesson_title` varchar(200) NOT NULL COMMENT 'Lesson Title',

  `lesson_description` varchar(1024) DEFAULT NULL COMMENT 'Lesson Description',

  `lesson_content` varchar(1024) NOT NULL COMMENT 'Link to Lesson Content',

  PRIMARY KEY (`lesson_id`),

  KEY `lesson_id` (`lesson_id`),

  KEY `lesson_id_2` (`lesson_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Lessons' AUTO_INCREMENT=4 ;

--

-- Dumping data for table `lessons`

--

INSERT INTO `lessons` (`lesson_id`, `lesson_title`, `lesson_description`, `lesson_content`)
VALUES

(1, 'Introduction to the Filipino Language', 'A simple introduction to the language.',
'/filles/lessons/Introduction.pdf'),

```

```
(2, 'Simple Sentences', 'An introduction on how sentences are constructed in the Filipino language.', '/filles/lessons/Lesson1.pdf'),
```

```
(3, 'The Subject', 'This lesson covers the use of the subject in Filipino language.', '/filles/lessons/Lesson2.pdf');
```

```
-- -----
```

```
--
```

```
-- Table structure for table `links`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `links` (  
  `link_id` int(5) unsigned NOT NULL AUTO_INCREMENT COMMENT 'Link ID',  
  `link_title` varchar(128) NOT NULL COMMENT 'Link Title',  
  `link_description` varchar(1024) DEFAULT NULL COMMENT 'Link Description',  
  `link_url` varchar(1024) NOT NULL COMMENT 'Link URL',  
  `is_approved` int(1) NOT NULL COMMENT 'Is Approved?',  
  `is_selected` int(1) NOT NULL COMMENT 'Is the Link Selected for Maintenance?',  
  `contributed_by` varchar(50) NOT NULL COMMENT 'Contributor',  
  `timestamp_created` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT 'Created on',  
  PRIMARY KEY (`link_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Link Exchange' AUTO_INCREMENT=1 ;
```

```
-- -----
```

```
--
```

```
-- Table structure for table `phrasebook`
```

```
--
```

```

CREATE TABLE IF NOT EXISTS `phrasebook` (

  `phrasebook_id` int(4) unsigned NOT NULL AUTO_INCREMENT COMMENT 'Phrasebook ID',

  `phrasebook_phrase` varchar(500) NOT NULL COMMENT 'Phrasebook Phrase',

  `phrasebook_pronunciation` varchar(256) DEFAULT NULL COMMENT 'Syllabic Pronunciation of the
Term',

  `phrasebook_translation` varchar(500) NOT NULL COMMENT 'English Translation of Phrase',

  `phrasebook_category` varchar(256) DEFAULT NULL COMMENT 'Phrasebook Category/Tags',

  `phrasebook_audio_link` varchar(1024) DEFAULT NULL COMMENT 'Audio Link of the Phrase',

  PRIMARY KEY (`phrasebook_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Phrasebook Entry Index' AUTO_INCREMENT=15 ;

--

-- Dumping data for table `phrasebook`

--

INSERT INTO `phrasebook` (`phrasebook_id`, `phrasebook_phrase`, `phrasebook_pronunciation`,
`phrasebook_translation`, `phrasebook_category`, `phrasebook_audio_link`) VALUES

(1, 'Kumusta na?', 'kum-u-stah nah', 'How are you?', 'Greetings', '/filles/sounds/kumusta.wav'),

(2, 'Magandang umaga!', NULL, 'Good morning!', 'Greetings',
'/filles/sounds/magandang_umaga.wav'),

(3, 'Magandang gabi!', NULL, 'Good evening', 'Greetings', '/filles/sounds/magandang_gabi.wav'),

(4, 'Barya lang po sa umaga', NULL, 'A reminder posted in jeepneys that they usually do not have
change for larger denominations in the morning.', 'Civilities', '/filles/sounds/barya_lang.wav'),

(5, 'Sandali lang!', NULL, 'Wait a minute!', 'Civilities', '/filles/sounds/sandali_lang.wav'),

(6, 'Walang anuman', NULL, 'You're welcome', 'Civilities', '/filles/sounds/walang_anuman.wav'),

(7, 'Saan ang', NULL, 'Where is', 'Directions', '/filles/sounds/saan_ang.wav'),

(8, 'Di bale', NULL, 'It doesn't matter', 'Answers', '/filles/sounds/di_bale.wav'),

(9, 'Hindi ko alam!', NULL, 'I don't know!', 'Answers', '/filles/sounds/hindi_ko_alam.wav'),

(10, 'Meron ka bang', NULL, 'Do you have', 'Questions', '/filles/sounds/meron_ka_bang.wav'),

(11, 'Gusto mo', NULL, 'Do you like', 'Questions', '/filles/sounds/gusto_mo.wav'),

(12, 'Gusto ko', NULL, 'I would like', 'Questions', '/filles/sounds/gusto_ko.wav'),

```

```
(13, 'Mahal kita', NULL, 'I love you', 'Expressions', '/filles/sounds/mahal_kita.wav'),
(14, 'Ikaw ako', NULL, 'You are myself', 'Wala lang', '/filles/sounds/ako.wav');
```

```
-- -----
```

```
--
```

```
-- Table structure for table `question_bank`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `question_bank` (
  `question_no` int(4) unsigned NOT NULL AUTO_INCREMENT COMMENT 'Question Number',
  `question_type` enum('true_false','multiple_choice') NOT NULL COMMENT 'Question Type',
  `question_title` varchar(512) NOT NULL COMMENT 'Question Name',
  `choice_a` varchar(50) NOT NULL COMMENT 'Choice A',
  `choice_b` varchar(50) NOT NULL COMMENT 'Choice B',
  `choice_c` varchar(50) DEFAULT NULL COMMENT 'Choice C',
  `choice_d` varchar(50) DEFAULT NULL COMMENT 'Choice D',
  `correct_answer` varchar(50) NOT NULL COMMENT 'Correct Answer',
  `lesson_id` int(2) DEFAULT NULL,
  PRIMARY KEY (`question_no`),
  UNIQUE KEY `question_no` (`question_no`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Question Bank' AUTO_INCREMENT=49 ;
```

```
--
```

```
-- Dumping data for table `question_bank`
```

```
--
```

```
INSERT INTO `question_bank` (`question_no`, `question_type`, `question_title`, `choice_a`,
`choice_b`, `choice_c`, `choice_d`, `correct_answer`, `lesson_id`) VALUES
```

(1, 'true_false', 'During the Spanish era in the Philippines, the Spanish language influenced the current state of the various Philippine languages.', 'True', 'False', '', '', 'True', 1),

(2, 'multiple_choice', 'What year did Manuel L. Quezon declare Tagalog as the basis for the Filipino language?', '1934', '1935', '1939', '1937', '1935', 1),

(3, 'true_false', 'The Filipino language follows the sentence order similar to English, that is, in the form of Subject then Predicate.', 'True', 'False', '', '', 'False', 1),

(4, 'multiple_choice', 'Nagsalita ____ babae.', 'ang', 'ang mga', 'si', 'sina', 'ang', 3),

(5, 'multiple_choice', 'Lumaban _____ sundalo. (plural)', 'ang', 'ang mga', 'si', 'sina', 'ang mga', 3),

(6, 'multiple_choice', 'Umiyak _____ bata.', 'ang', 'ang mga', 'si', 'sina', 'ang', 3),

(7, 'multiple_choice', 'Guro _____ Damian.', 'ang', 'ang mga', 'si', 'sina', 'si', 3),

(8, 'multiple_choice', 'Nasa garahe _____ Nanay.', 'ang', 'ang mga', 'si', 'sina', 'si', 3),

(9, 'multiple_choice', 'Masaya ____ Marsha at Nina.', 'ang', 'ang mga', 'si', 'sina', 'sina', 3),

(10, 'multiple_choice', 'Nagsulat _____ estudyante. (plural)', 'ang', 'ang mga', 'si', 'sina', 'ang mga', 3),

(11, 'multiple_choice', 'Nagwalis _____ kasambahay.', 'ang', 'ang mga', 'si', 'sina', 'ang', 3),

(12, 'multiple_choice', 'Nabasag _____ baso.', 'ang', 'ang mga', 'si', 'sina', 'ang', 3),

(13, 'multiple_choice', 'Sa Huwebes _____ pagsusulit.', 'ang', 'ang mga', 'si', 'sina', 'ang', 3),

(14, 'true_false', 'Subjectless sentences and existentials have fewer structure types than sentences with a subject.', 'True', 'False', '', '', 'True', 2),

(15, 'true_false', 'The typical simple sentence in Tagalog has a subject and a predicate.', 'True', 'False', '', '', 'True', 2),

(16, 'true_false', 'The normal order of the sentence elements is Predicate then Subject.', 'True', 'False', '', '', 'True', 2),

(17, 'true_false', 'Examples of the Subjectless Sentences are <i>Gusto</i> Sentences and Phenomenal Sentences.', 'True', 'False', '', '', 'True', 2),

(18, 'true_false', 'In Subjectless Sentences, none of the participants in the event is subject or focus of the sentence.', 'True', 'False', '', '', 'True', 2),

(19, 'true_false', '<i>May</i> / <i>Mayroon</i> sentences are often identified as existential sentences.', 'True', 'False', '', '', 'True', 2),

(20, 'true_false', '<i>Gusto</i> is the negative form of <i>may</i> / <i>mayroon</i>.', 'True', 'False', '', '', 'False', 2),

(21, 'true_false', '<i>Wala</i> means non-possession or non-existence.', 'True', 'False', '', '', 'True', 2),

(22, 'true_false', '<i>Mayroon</i> is a combination of <i>may</i> and <i>roon</i>, the latter being a locative demonstrative.', 'True', 'False', '', '', 'True', 2),

(23, 'multiple_choice', '(they) Gulong-gulo _____.', 'kami', 'tayo', 'kayo', 'sila', 'sila', 4),

(24, 'multiple_choice', 'We (exclusive) Gulong-gulo _____.', 'kami', 'tayo', 'kayo', 'sila', 'kami', 4),

(25, 'multiple_choice', '(she) Nasa eskwelahan _____.', 'ako', 'ikaw', 'siya', 'sila', 'siya', 4),

(26, 'multiple_choice', '(he [exclusive]) Nasa eskwelahan _____.', 'ako', 'ikaw', 'siya', 'sila', 'siya', 4),

(27, 'multiple_choice', '(you [singular]) Natulog _____.', 'ako', 'ikaw', 'siya', 'sila', 'ikaw', 4),

(28, 'multiple_choice', '(you [plural]) Natulog _____.', 'ako', 'ikaw', 'siya', 'sila', 'sila', 4),

(29, 'multiple_choice', '(I) Doktor _____.', 'ako', 'ikaw', 'siya', 'sila', 'ako', 4),

(30, 'multiple_choice', '(we [inclusive]) Doktor _____.', 'kami', 'tayo', 'kayo', 'sila', 'tayo', 4),

(31, 'multiple_choice', '(He and she) Naghapunan _____.', 'kami', 'tayo', 'kayo', 'sila', 'sila', 4),

(32, 'multiple_choice', '(you [singular] and I) Naghapunan _____.', 'kami', 'tayo', 'kayo', 'sila', 'tayo', 4),

(33, 'multiple_choice', 'Ginisa (<i>gulay</i> plural)', 'ang mga ito', 'ang mga iyan', 'ang mga iyon', 'all of the above', 'all of the above', 5),

(34, 'multiple_choice', 'Hugasan mo _____. (<i>plato</i> plural)', 'ang mga ito', 'ang mga iyan', 'ang mga iyon', 'all of the above', 'all of the above', 5),

(35, 'multiple_choice', 'Natupi _____. (<i>papel</i>)', 'ito', 'ang mga iyan', 'ang mga iyon', 'all of the above', 'ito', 5),

(36, 'multiple_choice', 'Abutin mo _____. (<i>bunga</i>)', 'ang mga ito', 'ang mga iyan', 'iyon', 'all of the above', 'iyon', 5),

(37, 'multiple_choice', 'Nasa bahay _____. (<i>papeles</i> plural)', 'ito', 'ang mga iyan', 'ang mga iyon', 'all of the above', 'ang mga iyon', 5),

(38, 'multiple_choice', 'Nanalo ang masipag na bata.', 'ang mga masipag', 'masisipag', 'mamasipag', 'all of the above', 'masisipag', 6),

(39, 'multiple_choice', 'Bukas dadating ang matalinong estudyante.', 'ang mga matatalinong', 'matatalinong', 'mamatalinong', 'all of the above', 'matatalino', 6),

(40, 'multiple_choice', 'Maglalaro ang matangkad na manlalaro.', 'ang mga matangkad', 'matatangkad', 'mamatangkad', 'all of the above', 'matatangkad', 6),

(41, 'multiple_choice', 'Magtuturo ang mabuting guro.', 'ang mga mabuting', 'mabubuting', 'mamabuting', 'all of the above', 'mabubuting', 6),

(42, 'multiple_choice', 'Kakain ang masiba.', 'ang mga masiba', 'masisiba', 'mamasiba', 'all of the above', 'masisiba', 6),

(43, 'multiple_choice', 'Parurusahan ang masama.', 'ang mga masama', 'masasama', 'mamasama', 'all of the above', 'masasama', 6),

```
(44, 'multiple_choice', 'Babagsak ang mayabang.', 'ang mga mayabang', 'mayayabang', 'mamayabang',  
'all of the above', 'mayayabang', 6),
```

```
(45, 'multiple_choice', 'Pinupuri ang magalang na anak.', 'ang mga magalang', 'magagalang',  
'mamagalang', 'all of the above', 'magagalang', 6),
```

```
(46, 'multiple_choice', 'Iniwasan ang mataong lugar.', 'ang mga mataong', 'matataong',  
'mamataong', 'all of the above', 'matataong', 6),
```

```
(47, 'multiple_choice', 'Kinagigiliwan ang mahinhin na dalaga.', 'ang mga mahinhin',  
'mahihinhin', 'mamahinhin', 'all of the above', 'mahihinhin', 6),
```

```
(48, 'multiple_choice', 'What is the Filipino for verb?', 'Panghalip', 'Pandiwa', 'Pangngalan',  
'Pang-uri', 'Pandiwa', NULL);
```

```
-- -----
```

```
--
```

```
-- Table structure for table `quizzes_taken`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `quizzes_taken` (  
  
  `quiz_id` int(6) unsigned NOT NULL AUTO_INCREMENT COMMENT 'Quiz ID',  
  
  `user_id` mediumint(8) NOT NULL COMMENT 'ID of User who took the quiz',  
  
  `no_correct` int(2) NOT NULL COMMENT 'Number of Correct Answers',  
  
  `no_of_questions` int(2) NOT NULL COMMENT 'Number of Questions in that quiz',  
  
  `lesson_id` int(2) NOT NULL COMMENT 'Lesson where the Quiz was based on',  
  
  `timestamp_completed` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT 'Timestamp When Quiz  
Was Finished',  
  
  PRIMARY KEY (`quiz_id`),  
  
  KEY `user_id` (`user_id`)  
  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Table of Quizzes Taken' AUTO_INCREMENT=1 ;
```

```
-- -----
```

```
--
```



```

-- Table structure for table `user_account_info`
--

CREATE TABLE IF NOT EXISTS `user_account_info` (
  `userinfo_id` int(8) unsigned NOT NULL AUTO_INCREMENT,
  `userinfo_uacc_fk` int(8) NOT NULL,
  `userinfo_first_name` varchar(256) DEFAULT NULL,
  `userinfo_last_name` varchar(128) DEFAULT NULL,
  PRIMARY KEY (`userinfo_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='User Information' AUTO_INCREMENT=6 ;

--

-- Dumping data for table `user_account_info`
--

INSERT INTO `user_account_info` (`userinfo_id`, `userinfo_uacc_fk`, `userinfo_first_name`,
`userinfo_last_name`) VALUES

(1, 1, 'System', 'Administrator'),

(2, 2, 'Content', 'Administrator'),

(3, 3, 'SWF', 'Staff'),

(4, 4, 'Outside', 'Contributor'),

(5, 5, 'Registered', 'User');

-----

--

-- Table structure for table `user_accounts`
--

```

```

CREATE TABLE IF NOT EXISTS `user_accounts` (

  `uacc_id` int(11) unsigned NOT NULL AUTO_INCREMENT,

  `uacc_group_fk` smallint(5) unsigned NOT NULL,

  `uacc_email` varchar(100) NOT NULL,

  `uacc_username` varchar(15) NOT NULL,

  `uacc_password` varchar(60) NOT NULL,

  `uacc_ip_address` varchar(40) NOT NULL,

  `uacc_salt` varchar(40) NOT NULL,

  `uacc_activation_token` varchar(40) NOT NULL,

  `uacc_forgotten_password_token` varchar(40) NOT NULL,

  `uacc_forgotten_password_expire` datetime NOT NULL,

  `uacc_update_email_token` varchar(40) NOT NULL,

  `uacc_update_email` varchar(100) NOT NULL,

  `uacc_active` tinyint(1) unsigned NOT NULL,

  `uacc_suspend` tinyint(1) unsigned NOT NULL,

  `uacc_fail_login_attempts` smallint(5) NOT NULL,

  `uacc_fail_login_ip_address` varchar(40) NOT NULL,

  `uacc_date_fail_login_ban` datetime NOT NULL COMMENT 'Time user is banned until due to repeated failed logins',

  `uacc_date_last_login` datetime NOT NULL,

  `uacc_date_added` datetime NOT NULL,

  PRIMARY KEY (`uacc_id`),

  UNIQUE KEY `uacc_id` (`uacc_id`),

  KEY `uacc_group_fk` (`uacc_group_fk`),

  KEY `uacc_email` (`uacc_email`),

  KEY `uacc_username` (`uacc_username`),

  KEY `uacc_fail_login_ip_address` (`uacc_fail_login_ip_address`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ;

```

```

--
-- Dumping data for table `user_accounts`
--

INSERT INTO `user_accounts` (`uacc_id`, `uacc_group_fk`, `uacc_email`, `uacc_username`,
`uacc_password`, `uacc_ip_address`, `uacc_salt`, `uacc_activation_token`,
`uacc_forgotten_password_token`, `uacc_forgotten_password_expire`, `uacc_update_email_token`,
`uacc_update_email`, `uacc_active`, `uacc_suspend`, `uacc_fail_login_attempts`,
`uacc_fail_login_ip_address`, `uacc_date_fail_login_ban`, `uacc_date_last_login`,
`uacc_date_added`) VALUES

(1, 1, 'bonarae@gmail.com', 'admin',
'$2a$08$pI37S/Oaac6Gzzj815S4qOcobndLGPysv89pleBoqbKFnxv79yza', '127.0.0.1', 'YT2RpJnR3Y', '',
'', '0000-00-00 00:00:00', '', '', 1, 0, 0, '', '0000-00-00 00:00:00', '2013-04-02 13:21:37',
'2013-02-11 22:40:03'),

(2, 2, 'boeinghelmsman@yahoo.com', 'contentadmin',
'$2a$08$nGKlNpKIe0r/wHL3bif3yeveeRRIHsELYD7iAPb1FBdetcIlnPKxq', '127.0.0.1', 'jm8HtxVq4D', '',
'', '0000-00-00 00:00:00', '', '', 1, 0, 0, '', '0000-00-00 00:00:00', '2013-04-02 13:21:47',
'2013-02-13 11:59:07'),

(3, 3, 'kenilworth_maroon_asiam@yahoo.com', 'swfstaff',
'$2a$08$eVNzhysKSx5jqdhlnoVNEexThZncWz/J7PvSak7wtPSPjy2j5O5qm', '127.0.0.1', 'M9MWj8WYgf', '',
'', '0000-00-00 00:00:00', '', '', 1, 0, 0, '', '0000-00-00 00:00:00', '2013-04-04 13:17:22',
'2013-02-13 13:38:48'),

(4, 4, 'mormongator@hotmail.com', 'contributor',
'$2a$08$GRkAzxv4z8dcRC/PZ.czP.xb16bk5NUKM4VpscpTJ7oPFxXgBG6iO', '127.0.0.1', 'p3wFcHJmWc', '',
'', '0000-00-00 00:00:00', '', '', 1, 0, 0, '', '0000-00-00 00:00:00', '2013-04-02 12:00:19',
'2013-02-14 04:08:38'),

(5, 5, '200703380@post.upm.edu.ph', 'reguser',
'$2a$08$VD0SObozGw2NyLmpkbvPkeiqeNLpElwBw0k0eekSpdyV/Xp8uImVy', '127.0.0.1', '7cc3GmSZKN', '',
'', '0000-00-00 00:00:00', '', '', 1, 0, 0, '', '0000-00-00 00:00:00', '2013-04-02 22:17:03',
'2013-02-14 04:14:38');

```

```

-----

```

```

--
-- Table structure for table `user_groups`
--

```

```

CREATE TABLE IF NOT EXISTS `user_groups` (
  `ugrp_id` smallint(5) NOT NULL AUTO_INCREMENT,
  `ugrp_name` varchar(20) NOT NULL,

```

```

`ugrp_desc` varchar(100) NOT NULL,

`ugrp_admin` tinyint(1) NOT NULL,

PRIMARY KEY (`ugrp_id`),

UNIQUE KEY `ugrp_id` (`ugrp_id`) USING BTREE

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ;

--

-- Dumping data for table `user_groups`

--

INSERT INTO `user_groups` (`ugrp_id`, `ugrp_name`, `ugrp_desc`, `ugrp_admin`) VALUES

(1, 'sysadmin', 'System Administrators', 1),

(2, 'contentadmin', 'Content Administrators', 0),

(3, 'staff', 'SWF Staff', 0),

(4, 'contributors', 'Outside Contributors', 0),

(5, 'users', 'Registered Users', 0);

-----

--

-- Table structure for table `user_login_sessions`

--

CREATE TABLE IF NOT EXISTS `user_login_sessions` (

  `usess_uacc_fk` int(11) NOT NULL,

  `usess_series` varchar(40) NOT NULL,

  `usess_token` varchar(40) NOT NULL,

  `usess_login_date` datetime NOT NULL,

```

```

PRIMARY KEY (`usess_token`),

UNIQUE KEY `usess_token` (`usess_token`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-----

--

-- Table structure for table `user_privilege_groups`

--

CREATE TABLE IF NOT EXISTS `user_privilege_groups` (

  `upriv_groups_id` smallint(5) unsigned NOT NULL AUTO_INCREMENT,

  `upriv_groups_ugrp_fk` smallint(5) unsigned NOT NULL,

  `upriv_groups_upriv_fk` smallint(5) unsigned NOT NULL,

  PRIMARY KEY (`upriv_groups_id`),

  UNIQUE KEY `upriv_groups_id` (`upriv_groups_id`) USING BTREE,

  KEY `upriv_groups_ugrp_fk` (`upriv_groups_ugrp_fk`),

  KEY `upriv_groups_upriv_fk` (`upriv_groups_upriv_fk`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-----

--

-- Table structure for table `user_privilege_users`

--

CREATE TABLE IF NOT EXISTS `user_privilege_users` (

  `upriv_users_id` smallint(5) NOT NULL AUTO_INCREMENT,

```

```

`upriv_users_uacc_fk` int(11) NOT NULL,

`upriv_users_upriv_fk` smallint(5) NOT NULL,

PRIMARY KEY (`upriv_users_id`),

UNIQUE KEY `upriv_users_id` (`upriv_users_id`) USING BTREE,

KEY `upriv_users_uacc_fk` (`upriv_users_uacc_fk`),

KEY `upriv_users_upriv_fk` (`upriv_users_upriv_fk`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-----

--

-- Table structure for table `user_privileges`

--

CREATE TABLE IF NOT EXISTS `user_privileges` (

  `upriv_id` smallint(5) NOT NULL AUTO_INCREMENT,

  `upriv_name` varchar(20) NOT NULL,

  `upriv_desc` varchar(100) NOT NULL,

  PRIMARY KEY (`upriv_id`),

  UNIQUE KEY `upriv_id` (`upriv_id`) USING BTREE

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

ACKNOWLEDGEMENTS

I dedicate this Special Problem to the following:

-To my adviser, **Prof. Geoffrey A. Solano**, who along with his kindness to me, helped me inspire to work on my SP all the way, even though he had PhD studies himself at the same time I was having this SP. He never left my side whenever we were around. Also to my client, SWF-UPM chairperson **Prof. Odessa N. Josen**, who realized that there is a real need for such a solution and who helped me appreciate the beauty of the national language.

-To my family, especially to my mom, **Prof. Bethel Buena P. Villarta**, although she was away in Daejeon, South Korea during the proposal stage, and my dad, **Dr. Romeo L. Villarta, Jr.**, who along with his employees at IMS UPM, all helped me motivate to work on this SP. Also my sister and brother-in-law, **Dr. Namnama and Louie de Dios**, my brother, **Atty. Romeo P. Villarta III**, and my dad's fellow consultants and staff at UP-PGH Department of ORL all supported me in finishing my SP. Towards the end, my niece **Vida** arrived, enhancing my appreciation for this family more.

-To a devoted family friend, **Dr. Regie Lyn P. Santos-Cortez**, and her son **Craig**. Ever since Craig was born, his mother and I have been endowing him with our knowledge of Filipino culture, and through this, he became my inspiration for this SP. It is envisioned that Craig should appreciate our efforts in letting him understand why he is a Filipino-American.

-To **my cousins** who grew up in the United States, their significant others from other cultures, and their children, it is envisioned that your stigma of visiting the Philippines should be diminished (or better yet, eliminated) with the help of this SP. It will hopefully motivate you to

return home and/or visit our beautiful archipelago. Your plight from my point of view, which I generalized it as a Filipino-American problem in general, helped me spark this SP.

-To my godmother, **Carmen Auste**, who steered me into the right direction at a time when there was so much uncertainty in my SP's progress. It is envisioned that your two grandchildren will learn Filipino thoroughly with this SP.

-To my friends from **Pacita UMC and Central UMC UMYF chapters**, thank you for keeping me on the right spiritual track while I was studying all these years, while retaining my liberal identity politically.

-**Jesus Christ** swayed me into graduating from Computer Science after several flags along the way. Six years have passed since my mom swayed me into entering UP Manila, forcing me to ditch my original plans of attending a Catholic university nearby and it's only now that I've realized that UP is far freer in many ways than any Catholic university in the country. Even though my high school friends, whom all but one went to other schools and those who went to other schools have since graduated, and most of my original blockmates have long since graduated and left me behind, still, Jesus never left me behind.