AUTOMETIC QUESTION GENERATOR
(Question Bank)

by
Rony Kumar Mondal
ID: CSE 047 06286

Md.Fazley Rabbi
ID: CSE 047 06283

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Computer Science & Engineering

DEPARTMENT OF COMPUTER SCIENCE &ENGINEERING
STAMFORD UNIVERSITY BANGLADESH

September, 2016

DECLARATION

We, hereby, declare that the work presented in this project is the outcome of the investigation performed by us under the supervision of Nazia Hossain, Senior Lecturer, Department of Computer Science & Engineering, Stamford University Bangladesh. We also declare that no part of this project and thereof has been or is being submitted else where for the award of any graduation.

Countersigned

………………………………..
Nazia Hossain

Supervisor

Signature

………………………………..
Rony Kumar Mondal

….………………………
Md. Fazley Rabbi

Candidates

ABSTRACT

Question bank is basically a storage of numerous questions which has been used by the teacher to set the question paper or by the learner to get an idea of question types those are being set on any individual subject examination. This project is faculty member oriented based, where only the teachers can set questions by using this system. Moreover, this question paper format is based on the examination system of Stamford University Bangladesh, so that, the generated question paper is as the format of Mid/Final examination paper of Stamford University Bangladesh CSE Faculty. That is, total four sets of questions, each set carries ten marks. To use this automatic question generator system, one faculty has to register first and get approval by the admin. The admin and faculty members can add more questions in any specific subject/course. The courses are being added by the admin. The questions are generated randomly based on the storage of questionnaires of the respective course. After question generation user can download the pdf format of the question and get a print of it.

ACKNOWLEDGEMENTS

First of all I would like to thank the almighty ALLAH. Today I am successful in completing my work with such ease because He gave me the ability, chance, and cooperating supervisor.

We would like to express our gratitude to our respected Chairman Dr. Kamruddin Md. Nur. We are grateful to our honorable supervisor Nazia Hossain, Senior Lecturer, Department of Computer Science & Engineering for the guidance, inspiration and constructive suggestions that helpful us in the preparation of this thesis. Although she was always loaded with several other activities, she give me more than enough time in the work. She not only gave me time but aslo proper guidance and valuable advice whernever I faced with some difficulties. Her comments and guidance helped me in preparing my thesis report.

Last of all I am grateful to my family; who are, always with me in my every step of life.

TABLE OF CONTENTS

Chapter 5: System Tools Definition 20

LIST OF FIGURES

Chapter 1
Introduction

1.1 Introduction

Question Bank is a free tool which allows you to create question papers from thousands of past paper questions. In just a few clicks, faculty members can create their own customized paper from our bank of past examination questions. Find the questions they need and add them to their paper and export the paper as a PDF ready to use in the examination hall. They can even produce a version of your paper with accompanying mark scheme and examiner's comments as a PDF.

1.2 What is Question Bank

Question bank is to manage groups of questions in a Storyline project. Question bank makes it easy to assemble, use, and even reuse questions. Although question banks are generally used for storing question slides, they aren't limited to question slides. User can add content slides as well.

1.3 Overview of Project

Question Bank System project is aimed to develop an automatic question making system. Using Questing Bank system teacher are being able to make a question set automatically. But to do that every user needs to have an account which will help them to generate new questions Teachers have to registrar this system and admin approves the teacher application. User can also add another user but admin should approve or verify the request to add a new member. Besides admin, teachers can also generate new question papers but only admin can add new courses. In one word the application is user friendly and hopes everyone will like it. We also include the codes in the appendix section. Print the question in PDF file. Moreover if any User wants to make request Question Bank he can also take the help of this site.

Our site provided most significant way to make question paper easily and this system is based on the Mid/Final examination question format of Stamford University Bangladesh.

Chapter 2
Background of Question Bank

2.1 Question Bank

This feature allows a teacher to create, preview, and edit questions in a database of question categories. The categories can be limited to being used on the site, course or quiz level. The questions in a category can be added to a Quiz or to a lesson ac-tivity via an export process. The teacher enters the question bank by creating or editing a quiz activity or via Settings > Course administration > Question bank [1]

2.2 Select Category

Questions are organized into categories. Initially each course has only one category called "Default". It is good practice to create more categories to organize your questions. This not only makes it easier to find questions, but makes the use of random questions and matching question easier. You can create a hierarchy of categories because you can create subcategories inside parent categories. To add or edit categories click on the "Categories" tab.

The question editing screen shows the questions from the currently selected category. You choose this category from the **Category:** drop-down menu. Using the tick box below that menu you determine whether to also show the questions from all subcategories. [1]

2.2.1 Categories are shared in contexts

There are separate question category trees in each different 'context' in which questions are shared. The contexts available to you depend on whether you access the question bank from an activity or from a course and depend on the permissions assigned you for access to questions. See Question contexts for more information on these contexts. [1]

2.2.2 Question Contexts

We can share questions in different contexts. Context may be an unfamiliar word. Think of a context as an area within your Model site. Each different context has a separate question category hierarchy. So, now you can choose the 'context' within which your questions will be shared. And also an administrator can give their users different permissions within different context areas.

See the diagram illustrating the relationship of the different contexts below. Different colors denote different context types (though there are two hues of green, both denote the same type). Notice how question contexts contain each other. Notice also that Course category contexts can be nested. In this example we have two contexts nested within each other 'Humanities' and 'Languages'. We could have a third course category 'English' and fourth and so on. Course categories can be nested infinitely and so you can set up a very flexible system of question contexts for sharing questions and assigning question sharing permissions.

As described at Question categories, questions are stored in a hierarchical structure similar to folders at the file system. You can imagine question context as a place where the tree of question categories is actually placed. By default, every teacher can create new questions in the course context (the blue one in the diagram) and the activity module context (the white one). We will discuss how to allow a teacher to create questions in higher contexts below.

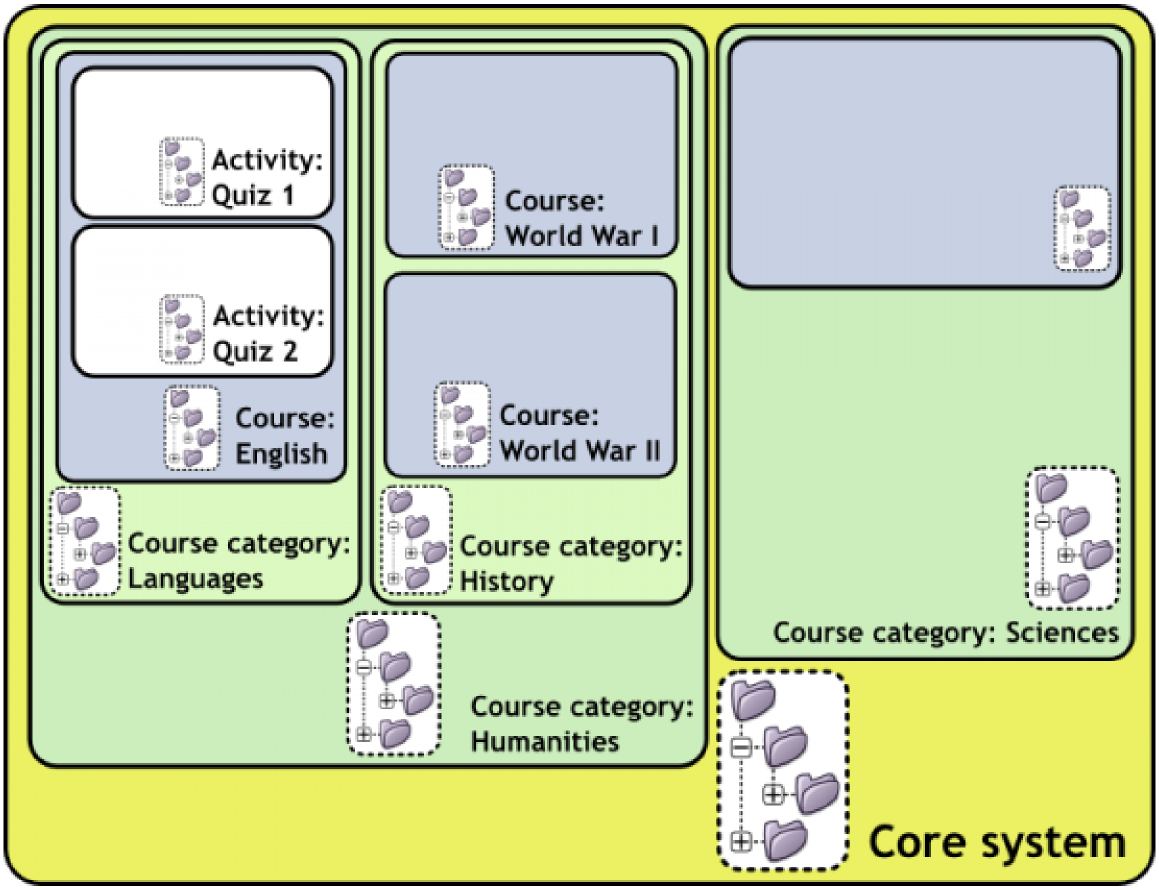


Figure: 2.2.2 Question Contexts

Question contexts are just as a subset of the Roles and Permissions contexts. A context contained by another context inherits the permissions of the containing context. You can access question categories in any containing context if you have permission to do so. Remember, you can access the question bank in two ways: from within an activity (now only the quiz activity), or from the link in the Course admin menu to 'Questions'. When you access the question bank from the :

- Activity. If you have permissions you can access question categories in :
    - that activity's context and containing contexts.
    - the course in which the activity module is contained.
    - the course categories in which the course that contains the activity module is contained.
    - the core system context
- Course. If you have permissions you can access question categories in :
    - that course.

    - the course categories in which the course is contained.
    - the core system context.

So questions contexts are accessible as follows :

- **Activity context** : questions only available to one activity module.
- **Course context** : questions available to all activity modules in a course and within the course from the 'Questions' link in the course administration block.
- **Course category contexts** : questions available to all activity modules and courses in the course category (remember one course category can contain other course categories, you can share your questions in any parent course category).
- **Core System context** : questions available in all courses and activities on your site.

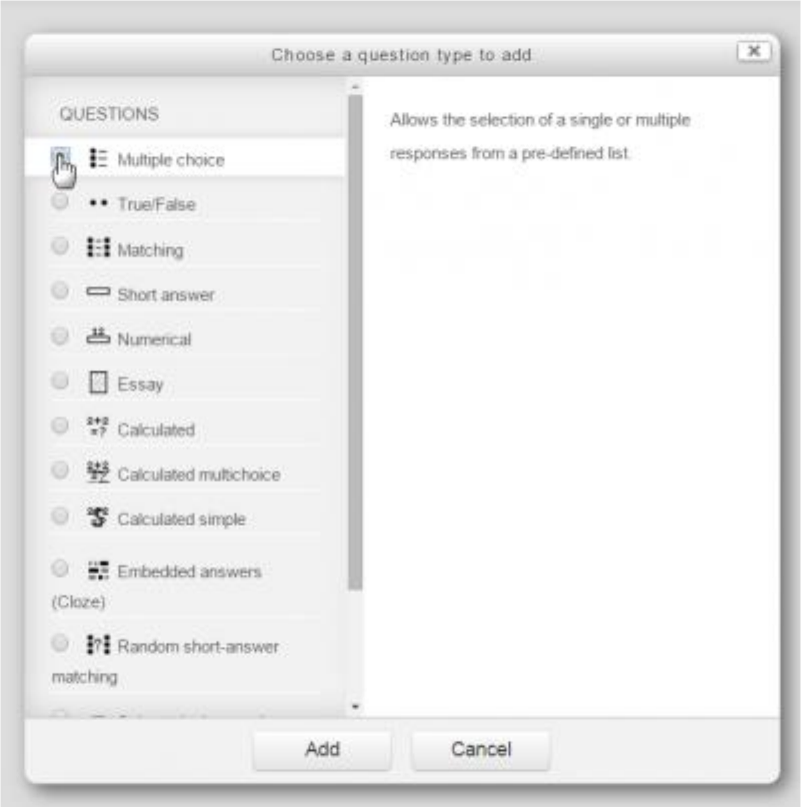
All Question capabilities are tested in the relevant context. So, for example, to add a question to a category you must have the add capability for that context. See roles and capability help for more info. If you don't have permission to do anything in a context, then the categories in that context will not be visible to you. [1]

2.3 Question Types

You may add a variety of different types of questions in the Quiz and Lesson modules. This page is about Quiz module question types; some will be similar to Lesson questions types, which are fewer in number and function differently. The standard quiz question types are listed below with brief descriptions. Please use the links on the right side of this page to find more detailed information about standard questions types. [1]

2.3.1 Changing the order of the question types

The administrator can specify the order the question types appear in the question chooser so that teachers are presented with the most common ones at the top. This is done from *Site administration* > *Plugins* > *Question types* > *Manage question types.* The administrator uses the arrows to reorder the question types and the teacher will then see their favourites first when creating a new question: [1]



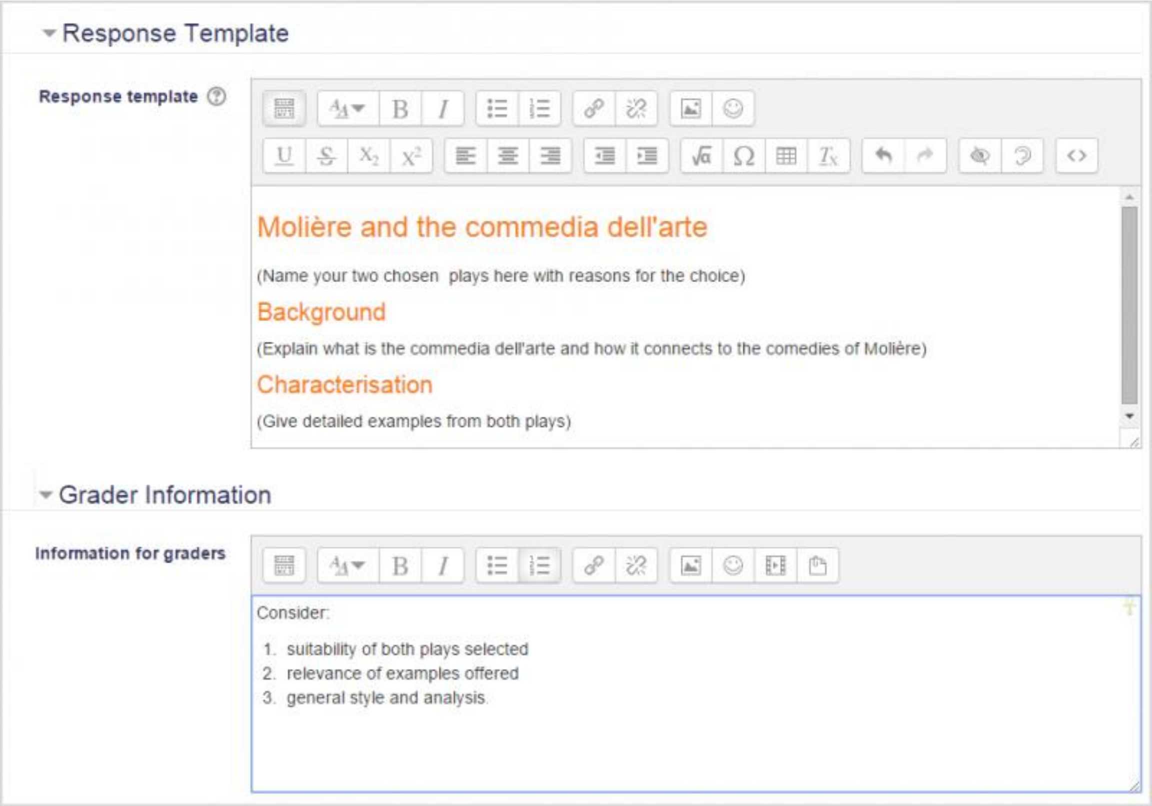
Allows the selection of a single or multiple responses from a pre-defined list.

Figure: 2.3.1 Question Types

2.3.2 Standard Question Types

- **Calculated**
  Calculated questions offer a way to create individual numerical questions by the use of wildcards that are substituted with individual values when the quiz is taken. More on the Calculated question type
- **Calculated multi-choice**
  Calculated multi-choice questions are like multi-choice questions with the additional property that the elements to select can include formula results from numeric values that are selected randomly from a set when the quiz is taken. They use the same wildcards than Calculated questions and their wildcards can be shared with other Calculated multi-choice or regular Calculated questions.
  The main difference is that the formula is included in the answer choice as {=...} i.e if you calculate the surface of a rectangle {={l}*{w}}.
- **Calculated simple**
  Simple calculated questions offer a way to create individual numerical questions whose response is the result of a numerical formula which contain variable numerical values by the use of wildcards (i.e {x} , {y}) that are substituted with random values when the quiz is taken.
  The simple calculated questions offers the most used features of the calculated question with a much simpler creation interface.

- **Description**
  This question type is not actually a question. It just prints some text (and possibly graphics) without requiring an answer. This can be used to provide some information to be used by a subsequent group of questions, for example.
- **Essay**
  This allows students to write at length on a particular subject and must be manually graded.
  It is possible for a teacher to create a template to scaffold the student's answer in order to give them extra support. The template is then reproduced in the text editor when the student starts to answer the question. It is also possible to include grading information for teachers marking the essay to refer to as they assess the essays,



Response Template

Response template

Molière and the commedia dell'arte

(Name your two chosen plays here with reasons for the choice)

Background

(Explain what is the commedia dell'arte and how it connects to the comedies of Molière)

Characterisation

(Give detailed examples from both plays)

Grader Information

Information for graders

Consider:

1. suitability of both plays selected
2. relevance of examples offered
3. general style and analysis

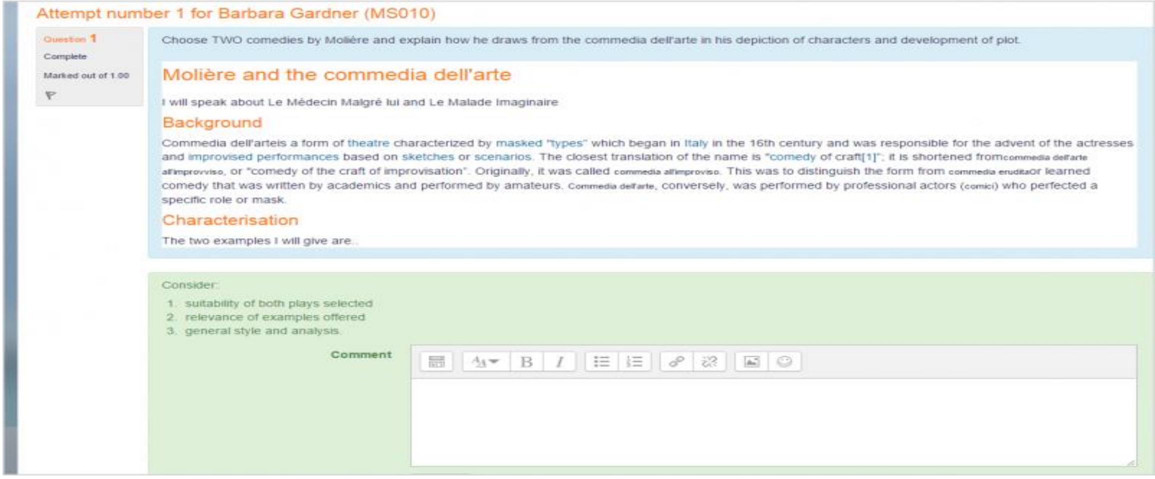
Figure: 2.3.2 Standard Question Types

- **Matching**
  A list of sub-questions is provided, along with a list of answers. The respondent must "match" the correct answers with each question.
- **Embedded Answers (Cloze Test / Gap Fill)**
  These very flexible questions consist of a passage of text (in Moodle format) that has various answers embedded within it, including multiple choice, short answers and numerical answers.

- **Multiple choice**
  With the Multiple Choice question type you can create single-answer and multiple-answer questions, include pictures, sound or other media in the question and/or answer options (by inserting HTML) and weight individual answers.
- **Short Answer**
  In response to a question (that may include an image), the respondent types a word or phrase. There may several possible correct answers, with different grades. Answers may or may not be sensitive to case.
- **Numerical**
  From the student perspective, a numerical question looks just like a short-answer question. The difference is that numerical answers are allowed to have an accepted error. This allows a continuous range of answers to be set.
- **Random short-answer matching**
  From the student perspective, this looks just like a Matching question. The difference is that the sub-questions are drawn randomly from Short Answer questions in the current category.
- **True/False**
  In response to a question (that may include an image), the respondent selects from two options: True or False.
- **Random**
  A Random question type is not a question type as such, but is a way of inserting a randomly-chosen question from a specified category into a quiz. [1]

2.4 Add a new question

1. Click the *Questions* tab to access the Question Bank page, if not there already.
2. From the **Category** drop-down menu, select a category you want to add a question to.
3. The page will change to show the questions already in that category
4. Select the question type you want to create from the **Create new question** drop-down menu.
5. Fill in the form for the question type you are creating. Each question type has its own form and has its own options.
6. Click *Save Changes* at the bottom of the form. [1]



Attempt number 1 for Barbara Gardner (MS010)

Question 1

Complete

Marked out of 1.00

Choose TWO comedies by Molière and explain how he draws from the commedia dell'arte in his depiction of characters and development of plot.

Molière and the commedia dell'arte

I will speak about Le Médecin Malgré lui and Le Malade Imaginaire

Background

Commedia dell'arteis a form of theatre characterized by masked "types" which began in Italy in the 16th century and was responsible for the advent of the actresses and improvised performances based on sketches or scenarios. The closest translation of the name is "comedy of craft[1]"; it is shortened fromcommedia dell'arte all'improvviso, or "comedy of the craft of improvisation". Originally, it was called commedia all'improvviso. This was to distinguish the form from commedia eruditaor learned comedy that was written by academics and performed by amateurs. Commedia dell'arte, conversely, was performed by professional actors (comici) who perfected a specific role or mask.

Characterisation

The two examples I will give are..

Consider:

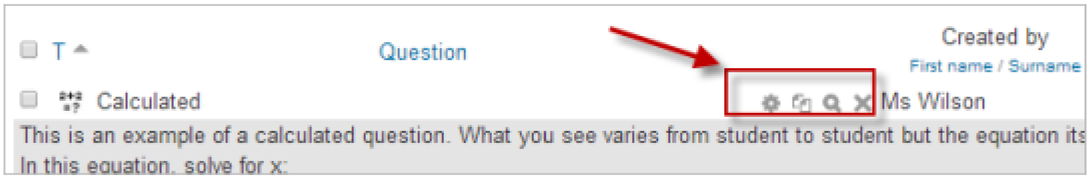
1. suitability of both plays selected
2. relevance of examples offered
3. general style and analysis.

Comment

Figure: 2.4 Add a New Question

2.5 Edit, duplicate, preview, delete and move

Each question in the question bank has four icons which allow you to edit, quickly duplicate, preview and delete the question. (The question may not be deleted if it is already in use elsewhere.) To duplicate a question, click the duplicate icon (second from the left) and a copy of the question editing screen will appear. You can either edit this new copy or simply scroll down and click "Save changes."To move a question into a different category or subcategory, click into the box on its left; scroll down to "With Selected..." and choose "Move to.." [1]



This is an example of a calculated question. What you see varies from student to student but the equation its
In this equation, solve for x:

Figure: 2.5 Edit, duplicate, preview, delete and move

2.6 Tips and tricks

- Put the answer into the question name so you can quickly see the answers when students are asking for answers. This is especially useful if you're dealing with large sets of questions! (No option yet for viewing category or answer of question in the list of questions.)
- Export questions in GIFT or Model XML formats, then import them into a Lesson. (Future versions of Model will make Question bank available to both Lesson and Quiz activities.)
- Remember that while some of the same question types can be found in both the Quiz and Lesson modules, they can be very different. Scoring and grading each student's choice is more robust in a Quiz. On the other hand, each Lesson question answer also has a jump associated it.
- Use GIFT or other export modes to print questions and answers in a category. Hint, clever use of word processor macros, using search and replace, can tidy up a GIFT file for printing.
- The question title is useful in sorting and making notes. For example, "ZZ remove 2010-3 Why did the Moodier cross", will put this question at the bottom of the list. Or where you want a 'the' questions in a category to appear in a specific order, use letters or numbers, knowing that AA will come first, AB will be second in the list.
- Do a copy and paste from a PDF file into the question content area. Reduces "other" hidden code which Word, Open Office and other programs can insert. [1]

2.7 Sharing and Managing Question banks

Be default, teachers can manage only the questions in the context of the courses they are in. You can set up a role to allow teachers to share and manager questions on a larger scale.
You can also use this role to create a special system-wide "Question bank Manager" instead of giving admin level or site-wide Manager access to a person managing the Questions. [1]

Chapter 3
Objective of the Project

3.1 Objective of the Project

In this modern civilization, science is developing day by day. Programmers are inventing new systems which will make life of people easier and comfortable. Our motivation is also like that. The aim of this project is to create new question paper automatically. So we have developed a system called “Automatic Question Generator”. It is an application by which a registered user can develop a new question paper randomly.

But to do that every user need to have an account which will help them to generate new questions. User can also add another user but admin should approve or verify the request to add a new member. Besides admin user can also generate new question papers but only admin can add new courses. In one word the application is user friendly and hopes everyone will like it. We also include the codes in the appendix section.

The structure of the site is not so much large but the features of the site are much more important than the structure. In this section we will discuss about summary what we did to develop the website.

There is total 7 chapters including “Introduction”.

In chapter 2, we will analyze the problems we found from other sites after a research and compare the sites with our website.

In chapter 3, we discuss about Objective of the Project .

In chapter 4, the whole system diagrams of the project will be discussed.

In chapter 5, the project model and project requirements to be followed to complete the project will be discussed.

In chapter 6, we will use system designed

In chapter 7, the whole work of the project will be discussed. The features we will discuss are given below:

1. Details of Home Page
2. Include the required diagrams
3. About Registration & Login system
4. About admin and user panel
5. How to add user(only for admin)
6. How to check the list of all registered users (only for admin)
7. How to approve/decline request (only for admin)
8. How to add subject(only foe admin)
9. All user list
10. All question list

11. How to add question
12. How to search question
13. How to make question
14. How to contact

In chapter 8, in this chapter we will add future work

And many more.

Chapter 4
Software Development Life Cycle

4.1 Software Development Life Cycle

The systems development life cycle (SDLC), also referred to as the application development life-cycle, is a term used in systems engineering, information systems and software engineering to describe a process for planning, creating, testing, and deploying an information system. The systems development life-cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both.[2]

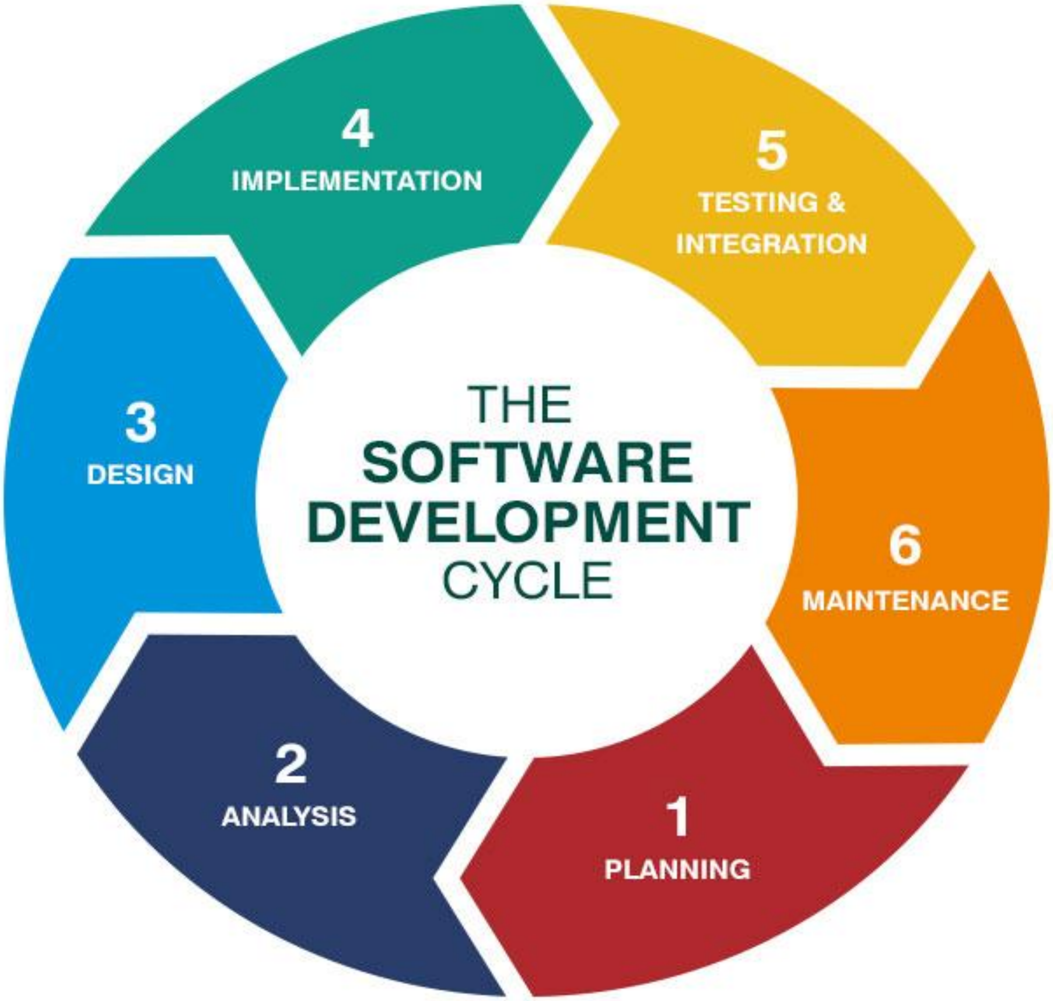


Figure 4.1 SDLC Cycle

4.2 SDLC Models

The software development models are the various processes or methodologies that are being selected for the development of the project depending on the project's aims and goals. There are many development life cycle models that have been developed in order to achieve different required objectives. The models specify the various stages of the process and the order in which they are carried out.

The selection of model has very high impact on the testing that is carried out. It will define the what, where and when of our planned testing, influence regression testing and largely determines which test techniques to use.

There are various Software development models or methodologies. They are as follows:

1. Waterfall model
2. V model
3. Incremental model
4. RAD model
5. Agile model
6. Iterative model
7. Spiral model
8. Prototype model

Choosing right model for developing of the software product or application is very important. Based on the model the development and testing processes are carried out.[2]

4.2.1 Scrutinizing Our Project Model

Selecting accurate model for developing of the software invention or request is very significant. Founded on the model the expansion and testing processes are accepted out. As Agile Model is more traditional and easy to gather requirements and analyzing system, so we choose the modle according to complete the project. This technique works well for big projects that may take numerous to progress. [2]

4.2.2 Agile Model

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.[3]

4.2.3 Agile Model Design

Just as a new approach to VILT helps increase learning and performance, the AGILE instructional design approach helps ensure that learning occurs and is transferred and sustained over time. As instructional designers and learning professionals, we are charged with ensuring people can perform effectively at every changing moment. Once our training courses end, our learners have to transfer knowledge and skills to on-the-job competence and sustain their performance in ever-changing environments. Our job then, according to Conrad Gottfredson, is to build “an entire learning ecosystem,” where all of the elements of learning and gaining competency are taken into consideration.

Traditionally, many instructional designers follow the ADDIE model, which was designed to build discrete learning events (whether eLearning, instructor-led training, or virtual delivery). However, those individual events are just one step on the journey to on-the-job competence, and the ADDIE model itself is not well suited for designing and sustaining an entire learning ecosystem. That's where the AGILE instructional design model can help (Figure 2). As learners will discover in the Guild Academy's upcoming AGILE Instructional Design course, an iterative process of design and development allows learning teams to deploy a core learning solution, with a complementary performance support system, more rapidly. And, according to Gottfredson, although the overall time to develop a solution doesn't change when the AGILE instructional design model is used, the time to initial rollout is much shorter, classroom time is generally more efficient, and intentional performance support ensures that new skills are sustained.[3]

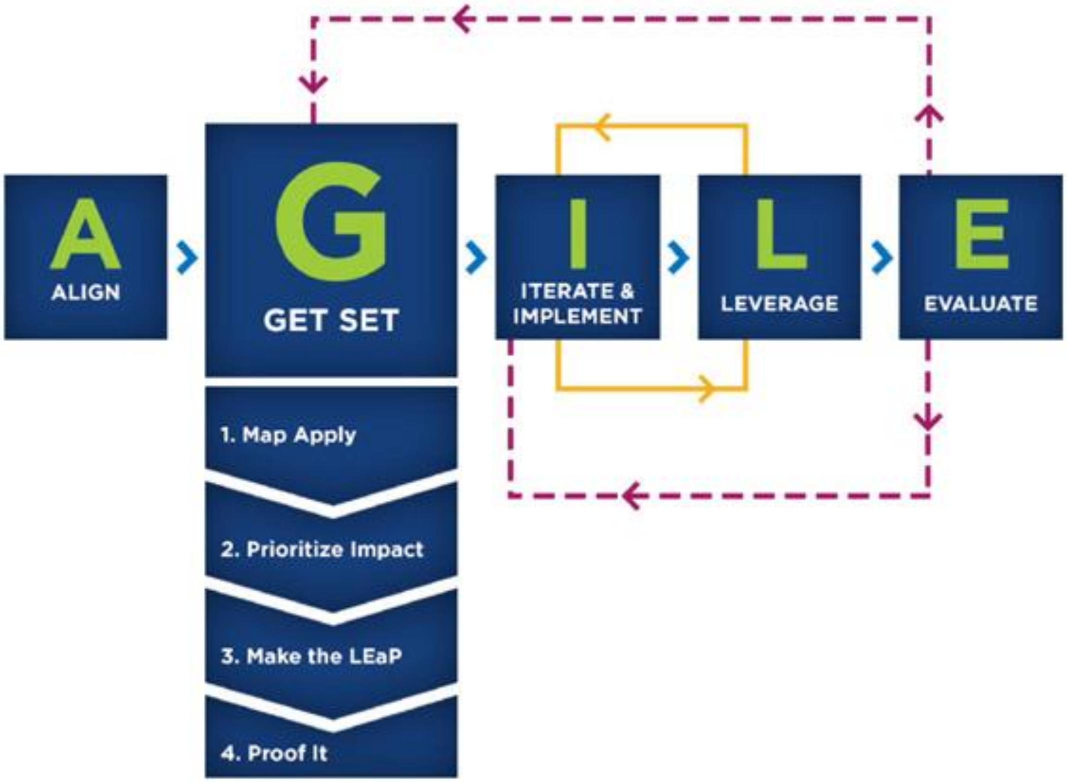


Figure 4.2.3 Agile Model Design

4.2.3 Agile Model Pros and Cons

Agile methods are being widely accepted in the software world recently, however, this method may not always be suitable for all products. Here are some pros and cons of the agile model.

Following table lists out the pros and cons of Agile Model:

Pros	Cons
• **Is a very realistic approach to software development** • **Promotes teamwork and cross training.** • **Functionality can be developed rapidly and demonstrated.** • **Resource requirements are minimum.** • **Suitable for fixed or changing requirements** • **Delivers early partial working solutions.** • **Good model for environments that change steadily.** • **Minimal rules, documentation easily employed.** • **Enables concurrent development and delivery within an overall planned context.** • **Little or no planning required** • **Easy to manage** • **Gives flexibility to developers**	• Not suitable for handling complex dependencies. • More risk of sustainability, maintainability and extensibility. • An overall plan, an agile leader and agile PM practice is a must without which it will not work. • Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines. • Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction. • There is very high individual dependency, since there is minimum documentation generated. • Transfer of technology to new team members may be quite challenging due to lack of documentation.

4.3 Agile Model Advantages and Disadvantages

4.3.1 Advantages of Agile Model

- Customer satisfaction by rapid, continuous delivery of useful software.
- People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.
- Working software is delivered frequently (weeks rather than months).
- Face-to-face conversation is the best form of communication.
- Close, daily cooperation between business people and developers.
- Continuous attention to technical excellence and good design.
- Regular adaptation to changing circumstances.
- Even late changes in requirements are welcomed

4.3.2 Disadvantages of Agile Model

- In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
- There is lack of emphasis on necessary designing and documentation.
- The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
- Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

Chapter 5
System Tools Definition

5.1 System Tools:

In this project we have used the following tools for our system requirement.

Hardware :

Laptop

- Core i3
- Hard Disk 1TD
- RAM 4GB

Software:

1. Laravel 5.2
2. Wamp Server
3. Bootstrap
4. Sublime Text

As a scripting language we have used PHP in backend and for the front end coding we have used HTML/CSS (bootstrap)

5.2 PHP (Hypertext Preprocessor)

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Group. PHP originally stood for *Personal Home Page*, but it now stands for the recursive backronym *PHP: Hypertext Preprocessor*.

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as ado facto standard. Since 2014 work has gone on to create a formal PHP specification.

During the 2010s there have been increased efforts towards standardization and code sharing in PHP applications by projects such as PHP-FIG in the form of PSR-initiatives as well as Composer dependency manager and the Packages repository. [4]

5.3 History of PHP

PHP as it's known today is actually the successor to a product named PHP/FI. Created in 1994 by Rasmus Lerdorf, the very first incarnation of PHP was a simple set of Common Gateway Interface (CGI) binaries written in the C programming language. Originally used for tracking visits to his online resume, he named the suite of scripts "Personal Home Page Tools," more frequently referenced as "PHP Tools." Over time, more functionality was desired, and Rasmus rewrote PHP Tools, producing a much larger and richer implementation. This new model was capable of database interaction and more, providing a framework upon which users could develop simple dynamic web applications such as guestbook’s. In June of 1995, Rasmus » released the source code for PHP Tools to the public, which allowed developers to use it as they saw fit. This also permitted - and encouraged - users to provide fixes for bugs in the code, and to generally improve upon it.

In September of that year, Rasmus expanded upon PHP and - for a short time - actually dropped the PHP name. Now referring to the tools as FI (short for "Forms Interpreter"), the new implementation included some of the basic functionality of PHP as we know it today. It had Perl-like variables, automatic interpretation of form variables, and HTML embedded syntax. The syntax itself was similar to that of Perl, albeit much more limited, simple, and somewhat inconsistent. In fact, to embed the code into an HTML file, developers had to use HTML comments. Though this method was not entirely well-received, FI continued to enjoy growth and acceptance as a CGI tool --- but still not quite as a language. However, this began to change the following month; in October, 1995, Rasmus released a complete rewrite of the code. Bringing back the PHP name, it was now (briefly) named "Personal Home Page Construction Kit," and was the first release to boast what was, at the time, considered an advanced scripting interface. The language was deliberately designed to resemble C in structure, making it an easy adoption for developers familiar with C, Perl, and similar languages. Having been thus far limited to UNIX and POSIX-compliant systems, the potential for a Windows NT implementation was being explored. [5]

5.3.1 History of PHP Version

The code got another complete makeover, and in April of 1996, combining the names of past releases, Rasmus introduced PHP/FI. This second-generation implementation began to truly evolve PHP from a suite of tools into a programming language in its own right. It included built-in support for DBM, myself, and Postgres95 databases, cookies, user-defined function support, and much more. That June, PHP/FI was given a version 2.0 status. An interesting fact about this, however, is that there was only one single full version of PHP 2.0. When it finally graduated from beta status in November, 1997, the underlying parsing engine was already being entirely rewritten.

Though it lived a short development life, it continued to enjoy a growing popularity in still-young world of web development. In 1997 and 1998, PHP/FI had a cult of several thousand users around the world. A Net craft survey as of May, 1998, indicated that nearly 60,000 domains reported having headers containing "PHP", indicating that the host server did indeed have it installed. This number equated to approximately 1% of all domains on the Internet at the time.

Despite these impressive figures, the maturation of PHP/FI was doomed to limitations; while there were several minor contributors, it was still primarily developed by an individual.[5]

Example #1 Example PHP/FI Code

```
<!--include /text/header.html-->

<!--getenv HTTP_USER_AGENT-->
<!--ifsubstr $exec_result Mozilla-->
  Hey, you are using Netscape!<p>
<!--endif-->

<!--sql database select * from table where user='$username'-->
<!--ifless $numentries 1-->
  Sorry, that record does not exist<p>
<!--endif exit-->
  Welcome <!--$user-->!<p>
  You have <!--$index:0--> credits left in your account.<p>

<!--include /text/footer.html-->
```

5.3.2 PHP 3

PHP 3.0 was the first version that closely resembles PHP as it exists today. Finding PHP/FI 2.0 still inefficient and lacking features they needed to power an eCommerce application they were developing for a university project, Andi Gutmans and Zeev Suraski of Tel Aviv, Israel, began yet another complete rewrite of the underlying parser in 1997. Approaching Rasmus online, they discussed various aspects of the current implementation and their redevelopment of PHP. In an effort to improve the engine and start building upon PHP/FI's existing user base, Andi, Rasmus, and Zeev decided to collaborate in the development of a new, independent programming language. This entirely new language was released under a new name, that removed the implication of limited personal use that the PHP/FI 2.0 name held. It was renamed simply 'PHP', with the meaning becoming a recursive acronym - PHP: Hypertext Preprocessor.

One of the biggest strengths of PHP 3.0 was its strong extensibility features. In addition to providing end users with a mature interface for multiple databases, protocols, and APIs, the ease of extending the language itself attracted dozens of developers who submitted a variety of modules. Arguably, this was the key to PHP 3.0's tremendous success. Other key features introduced in PHP 3.0 included object-oriented programming support and a far more powerful and consistent language syntax.

In June, 1998, with many new developers from around the world joining the effort, PHP 3.0 was announced by the new PHP Development Team as the official successor to PHP/FI 2.0. Active development of PHP/FI 2.0, which had all-but ceased as of November of the previous year, was now officially ended. After roughly nine months of open public testing, when the announcement of the official release of PHP 3.0 came, it was already installed on over 70,000 domains around the world, and was no longer limited to POSIX-compliant operating systems. A relatively small share of the domains reporting PHP as installed were hosted on servers running Windows 95, 98, and NT, and Macintosh. At its peak, PHP 3.0 was installed on approximately 10% of the web servers on the Internet.[5]

5.3.3 PHP 4

By the winter of 1998, shortly after PHP 3.0 was officially released, Andi Gutmans and Zeev Suraski had begun working on a rewrite of PHP's core. The design goals were to improve performance of complex applications, and improve the modularity of PHP's code base. Such applications were made possible by PHP 3.0's new features and support for a wide variety of third party databases and APIs, but PHP 3.0 was not designed to handle such complex applications efficiently.

The new engine, dubbed 'Zend Engine' (comprised of their first names, Zeev and Andi), met these design goals successfully, and was first introduced in mid 1999. PHP 4.0, based on this engine, and coupled with a wide range of additional new features, was officially released in May 2000, almost two years after its predecessor. In addition to the highly improved performance of this version, PHP 4.0 included other key features such as support for many more web servers, HTTP sessions, output buffering, more secure ways of handling user input and several new language constructs.

5.3.4 PHP 5

PHP 5 was released in July 2004 after long development and several pre-releases. It is mainly driven by its core, the *Zend Engine 2.0* with a new object model and dozens of other new features.

PHP's development team includes dozens of developers, as well as dozens others working on PHP-related and supporting projects, such as PEAR, PECL, and documentation, and an underlying network infrastructure of well over one-hundred individual web servers on six of the seven continents of the world. Though only an estimate based upon statistics from previous years, it is safe to presume PHP is now installed on tens or even perhaps hundreds of millions of domains around the world. .[5]

5.4 What is Framework?

PHP is a most scripting language from the last few years because of pre-build flexibility, ease-of-use structure code that allows you to spend more time on creating an actual website instead of spending time on repetitive coded .[6]

5.5 Why Use A PHP Framework

But first, let’s take a look at the top reasons why many developers like to use PHP frameworks and how these frameworks can level up your development process. Here’s what PHP frameworks do:

- Make speed development possible

- Provide well-organized, reusable and maintainable code
- Let you grow over time as web apps running on frameworks are scalable
- Spare you from the worries about low-level security of a site
- Follow the MVC (Model-View-Controller) pattern that ensures the separation of presentation and logic
- Promote modern web development practices such as object-oriented programming tools[7]

5.6 Below is Some Advantages To Use PHP Framework:

- Organize file and code structure
- Pre Build Libraries & Tools that can help you with
  - Database Abstraction
  - From Validation
  - Input/Output Filtering
  - Session & Cookie handling
- Faster Application Development using Less Code (Rapid application development)
- Model view control (MVC) Architecture
- Community support
- Suitable for teamwork
- Pretested Framework with
  - Any value passed to database object gets filtered against SQL injection attacks
  - central authentication service architecture
  - All HTML generating functions, such as form helpers and URL helpers filter the output automatically
  - Cross Site Request Forgery (CSRF) Protection
  - session management security integration
  - client-side cross-site scripting protection (xss)
  - Encrypting cookies automatically is only a matter of changing a config option and lot more[7]

5.7 Best PHP Framework For Development

PHP, known as the most popular server-side scripting language in the world, has evolved a lot since the first inline code snippets appeared in static HTML files.

These days developers need to build complex websites and web apps, and above a certain complexity level it can take too much time and hassle to always start from scratch, hence came the need for a more structured natural way of development. PHP frameworks provide developers with an adequate solution for that.

In this post we carefully handpicked 10 popular PHP frameworks that can best facilitate and streamline the process of backend web development.[7]

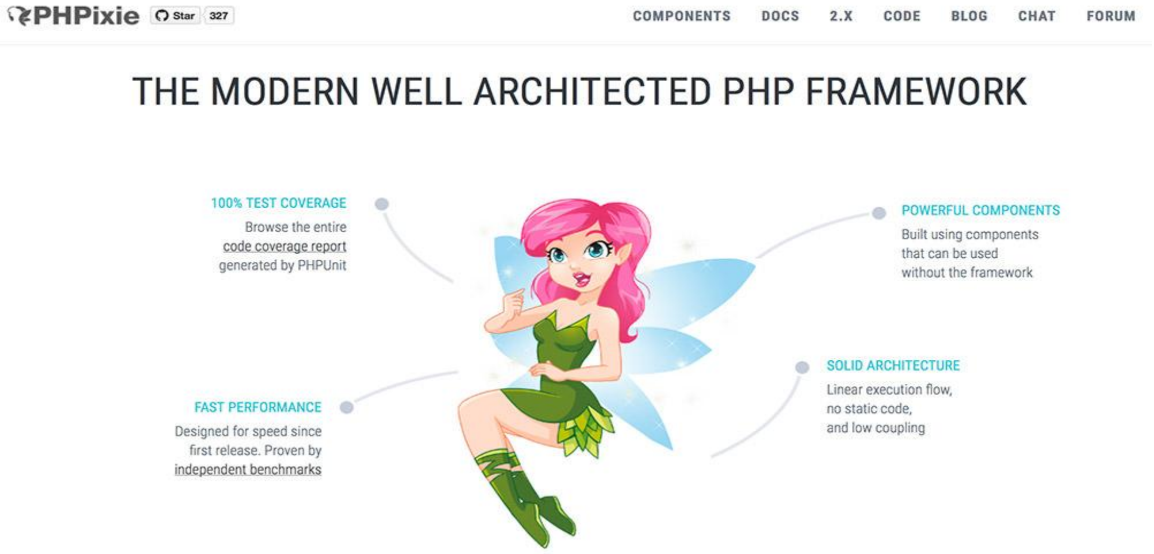
List of Framework :-

1. PHPixie
2. FuelPHP
3. Slim
4. Zend Fremework
5. CakePHP
6. Phalcon
7. Yii 2
8. CodeIgniter
9. Symfony
10. Laravel

5.7.1 PHPixie

PHPixie is a quite new framework, it started in 2012 with the goal of creating a high-performance framework for read-only websites. PHPixie also implements the HMVC design pattern just like FuelPHP, and is built by using independent components that can be used as well without the framework itself. The PHPixie components are 100% unit tested, and require minimum dependencies.

The official website has a tutorial that claims you can learn the framework in 30 minutes, and their blog also details many practical use cases. Among the features you can find the standard ORM (object-relational mapping), caching, input validation, authentication and authorization capabilities. PHPixie also allows you to use the HAML markup language, enables schema migration, and has a sophisticated routing system. [7]



THE MODERN WELL ARCHITECTED PHP FRAMEWORK

100% TEST COVERAGE
Browse the entire code coverage report generated by PHPUnit

POWERFUL COMPONENTS
Built using components that can be used without the framework

SOLID ARCHITECTURE
Linear execution flow, no static code, and low coupling

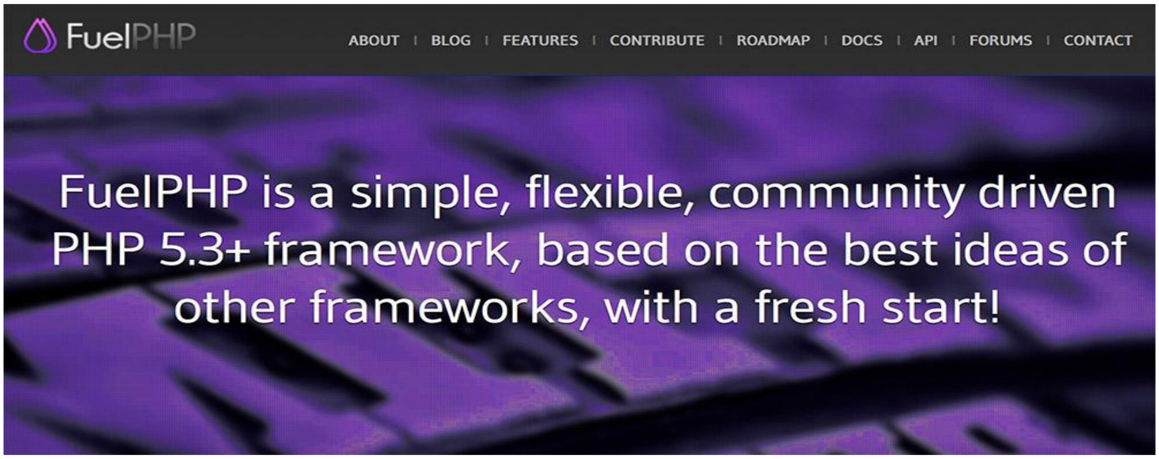
FAST PERFORMANCE
Designed for speed since first release. Proven by independent benchmarks

Figure 5.7.1:- The PHPixie

5.7.2 FuelPHP

FuelPHP is a flexible full-stack PHP framework that doesn't only support the ordinary MVC pattern but also its evolved version, HMVC (Hierarchical Model-View-Controller) at the architecture level. FuelPHP adds an optional class called Presenter (formerly called View Model) between the Controller and View layers to hold the logic needed to generate Views.

FuelPHP is modular and extendable, takes care of security concerns by providing features such as input and URI filtering and output encoding, and it comes with its own authentication framework, with many other sophisticated features and a detailed documentation. [7]



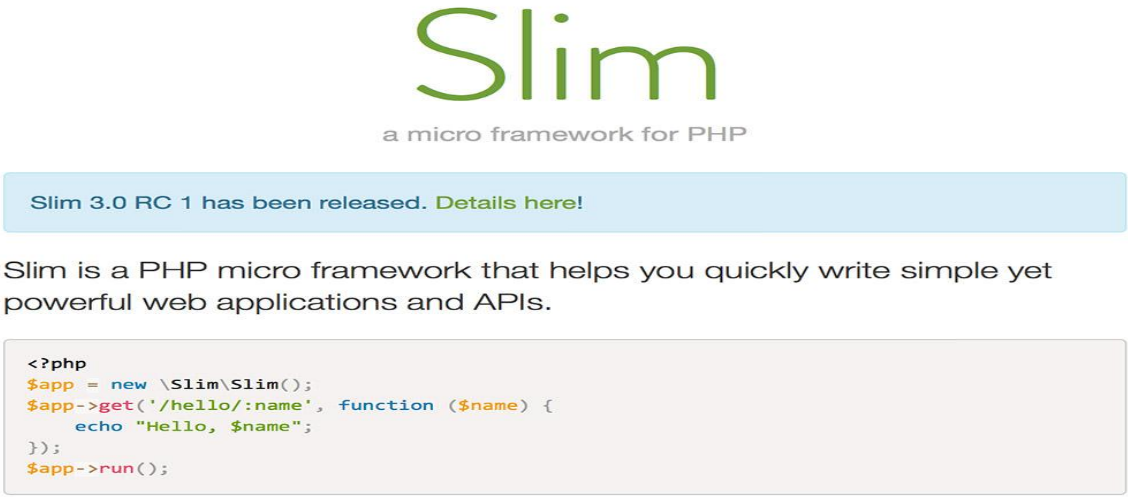
FuelPHP is a simple, flexible, community driven PHP 5.3+ framework, based on the best ideas of other frameworks, with a fresh start!

Figure 2.7.2 The FulePHP Framework

5.7.3 Slim PHP Framework

Slim is a PHP micro framework that provides you with everything you need and nothing you don’t. Micro frameworks are minimalistic in design, they are excellent for smaller apps where a full-stack framework would be an exaggeration. Slim’s creator was inspired by a Ruby micro framework called Sinatra.

Slim is used by many PHP developers for developing RESTful APIs and services. Slim comes with features such as URL routing, client-side HTTP caching, session- and cookie encryption, and it supports “flash” messages across HTTP requests as well. Its User Guide is an easy read, and if you are interested in the new features of the upcoming major release (already in beta), you can watch this primer video about Slim 3. [7]



Slim 3.0 RC 1 has been released. Details here!

Slim is a PHP micro framework that helps you quickly write simple yet powerful web applications and APIs.

```
<?php
$app = new \Slim\Slim();
$app->get('/hello/:name', function ($name) {
    echo "Hello, $name";
});
$app->run();
```

Figure 2.7.3 The Slim PHP

5.7.4 Zend Framework

Zend is a robust and stable PHP framework packed with a lot of configuration options therefore it's usually not recommended for smaller projects but excellent for more complex ones. Zend has partners such as IBM, Microsoft, Google and Adobe. The coming major release, Zend Framework 3 will be optimized for PHP 7 , but will still support PHP 5.5 onwards.

The current release, Zend Framework 2 also has many cool features such as cryptographic coding tools, an easy-to-use drag and drop editor with support for front-end technologies (HTML, CSS, JavaScript), instant online debugging and PHP Unit testing tools, and a connected Database Wizard. Zend Framework was created with the Agile methodology that facilitates delivering high-quality apps to enterprise clients. [7]

The most popular framework for modern, high-performing PHP applications

Why use Zend Framework?

Modular

Building blocks that can be used piece by piece with other applications or frameworks

Secure

All the cryptographic and secure coding tools you need to do things right

ANNOUNCING ZEND FRAMEWORK 3

The Future of PHP Frameworks

Focused on Simplicity, Reusability and Performance

Figure 2.7.4 Zend Framework

5.7.5 CakePHP

CakePHP is already a decade old (the first version was released in 2005), but it's still among the most popular PHP frameworks, as it has always managed to keep up with time. The latest version, CakePHP 3.0 enhanced session management, improved modularity by decoupling several components, and increased the ability of creating more standalone libraries.

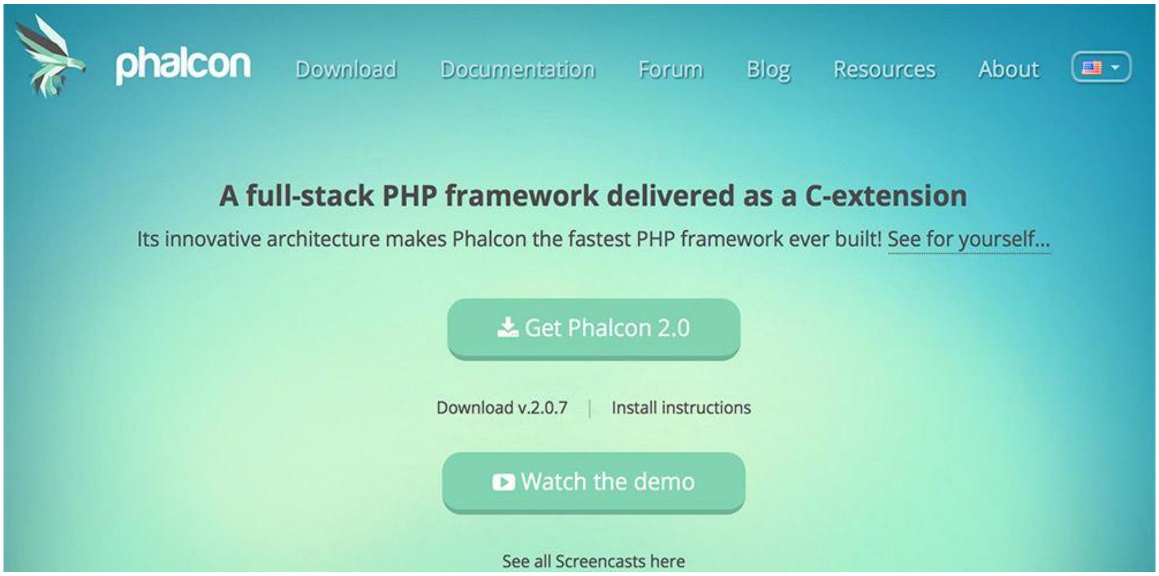
CakePHP has a really remarkable showcase, it powers the websites of big brands such as BMW, Hyundai, and Express. It is an excellent tool for creating web apps that need high-level of security, as it has many built-in security features such as input validation, SQL injection prevention, XSS (cross-site scripting) prevention, CSRF (cross-site request forgery) protection, and many others. [7]

Figure 2.7.5 Cake PHP

5.7.6 Phalcon

The Phalcon framework was released in 2012, and it quickly gained popularity among PHP developers. Phalcon is said to be fast as a falcon, because it was written in C and C++ to reach the highest level of performance optimization possible. Good news is that you don't have to learn the C language, as the functionality is exposed as PHP classes that are ready to use for any application.

As Phalcon is delivered as a C-extension, its architecture is optimized at low levels which significantly reduce the overhead typical of MVC-based apps. Phalcon not only boosts execution speeds, but also decreases resource usage. Phalcon is also packed with many cool features such as a universal auto-loader, asset management, security, translation, caching, and many others. As it's a well-documented and easy-to-use framework, it's definitely worth a try. [7]



A full-stack PHP framework delivered as a C-extension

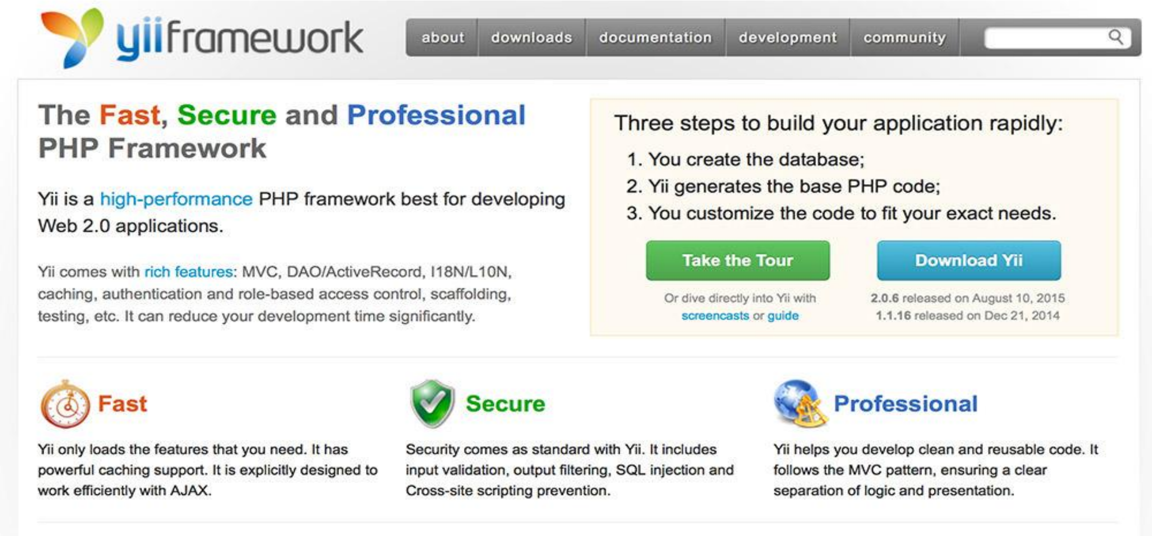
Its innovative architecture makes Phalcon the fastest PHP framework ever built! See for yourself...

Figure 2.7.6 Phalcon Framework

5.7.7 Yii Framework

If you choose the Yii framework you give a boost to the performance of your site as it’s faster than other PHP frameworks, because it extensively uses the lazy loading technique. Yii 2 is purely object-oriented, and it’s based on the DRY(Don’t Repeat Yourself) coding concept, so it provides you with a pretty clean and logical code base.

Yii 2 is integrated with jQuery, and it comes with a set of AJAX-enabled features, and it implements an easy-to-use skinning and theming mechanism, so it can be a great choice for someone who comes from a frontend background. It has also a powerful class code generator called Gii that facilitates object-oriented programming and rapid prototyping, and provides a web-based interface that allows you to interactively generate the code you need. [7]



The Fast, Secure and Professional PHP Framework

Yii is a high-performance PHP framework best for developing Web 2.0 applications.

Yii comes with rich features: MVC, DAO/ActiveRecord, I18N/L10N, caching, authentication and role-based access control, scaffolding, testing, etc. It can reduce your development time significantly.

Three steps to build your application rapidly:

1. You create the database;
2. Yii generates the base PHP code;
3. You customize the code to fit your exact needs.

Or dive directly into Yii with screencasts or guide

2.0.6 released on August 10, 2015
1.1.16 released on Dec 21, 2014

Fast

Yii only loads the features that you need. It has powerful caching support. It is explicitly designed to work efficiently with AJAX.

Secure

Security comes as standard with Yii. It includes input validation, output filtering, SQL injection and Cross-site scripting prevention.

Professional

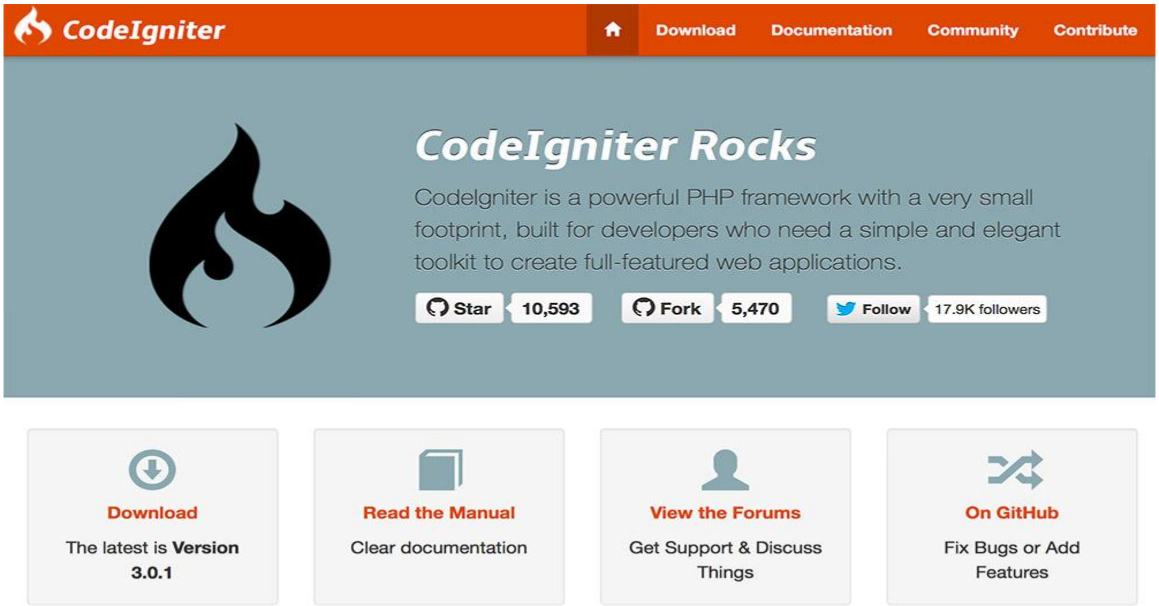
Yii helps you develop clean and reusable code. It follows the MVC pattern, ensuring a clear separation of logic and presentation.

Figure 2.7.7 Yii Framework

5.7.8 CodeIgniter Framework

CodeIgniter is a lightweight PHP framework that is almost 10 years old (initially released in 2006). CodeIgniter has a very straightforward installation process that requires only a minimal configuration, so it can save you a lot of hassle. It’s also an ideal choice if you want to avoid PHP version conflict, as it works nicely on almost all shared and dedicated hosting platforms (currently requires only PHP 5.2.4).

CodeIgniter is not strictly based on the MVC development pattern. Using Controller classes is a must, but Models and Views are optional, and you can use your own coding and naming conventions, evidence that CodeIgniter gives great freedom to developers. If you download it, you’ll see it’s only about 2MB, so it’s a lean framework, but it allows you to add third-party plugins if you need more complicated functionalities. [7]



CodeIgniter Rocks

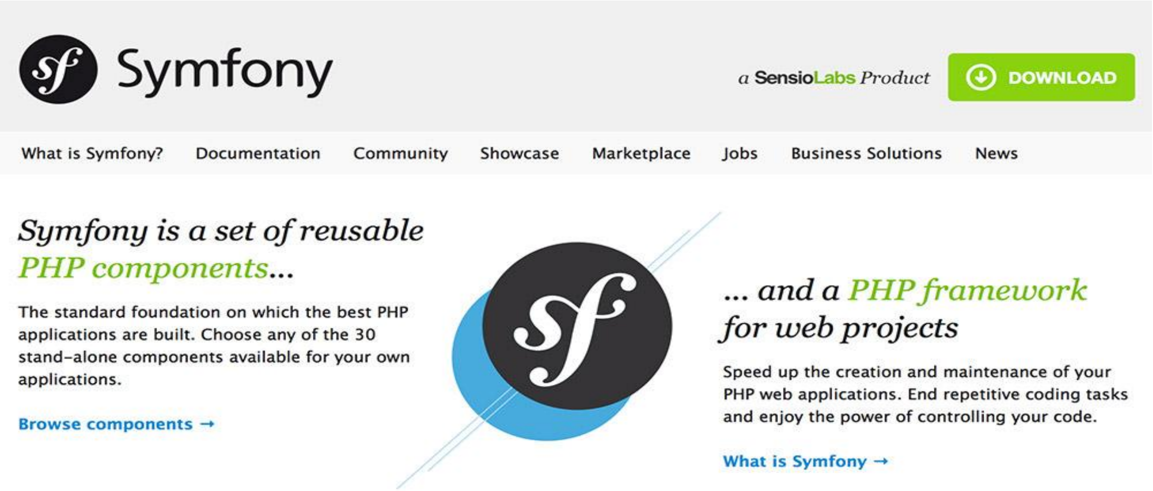
CodeIgniter is a powerful PHP framework with a very small footprint, built for developers who need a simple and elegant toolkit to create full-featured web applications.

Figure 2.7.8 CodeIgniter Framework

5.7.9 Symfony Framework

The components of the Symfony 2 framework are used by many impressive projects such as the Drupal content management system, or the phpBB forum software, but Laravel – the framework listed above – also relies on it. Symfony has a wide developer community and many ardent fans.

Symfony Components are reusable PHP libraries that you can complete different tasks with, such as form creation, object configuration, routing, authentication, templating, and many others. You can install any of the Components with the Composer PHP dependency manager. The website of Symfony has a cool showcase section where you can take a peek at the projects developers accomplished with the help of this handy framework. [7]



Symfony is a set of reusable PHP components...

The standard foundation on which the best PHP applications are built. Choose any of the 30 stand-alone components available for your own applications.

... and a PHP framework for web projects

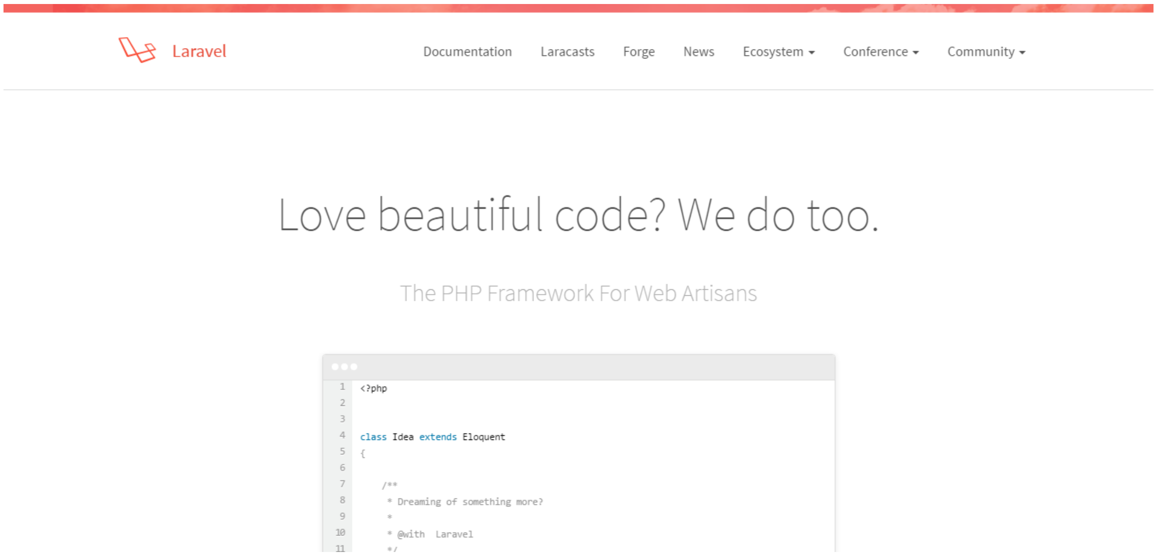
Speed up the creation and maintenance of your PHP web applications. End repetitive coding tasks and enjoy the power of controlling your code.

Figure 2.7.9 Symfony Framework

5.7.10 Laravel Framework

Although Laravel is a relatively new PHP framework (it was released in 2011), according to Site point's recent online survey it is the most popular framework among developers. Laravel has a huge ecosystem with an instant hosting and deployment platform, and its official website offers many screencast tutorials called Laracasts.

Laravel has many features that make rapid application development possible. Laravel has its own light-weight templating engine called "Blade", elegant syntax that facilitates tasks you frequently need to do, such as authentication, sessions, queuing, caching and RESTful routing. Laravel also includes a local development environment called Homestead that is a packaged Vagrant box.[8]



Love beautiful code? We do too.

The PHP Framework For Web Artisans

```
<?php


class Idea extends Eloquent
{

    /**
     * Dreaming of something more?
     *
     * @with  Laravel
     */
```

Figure 2.7.10 Laravel Framework

5.8 Why Use Laravel Framework

Over the last year, I took part in three large projects. My task was to move away from the old architecture based on PHP and server-side HTML generation, and transition to REST API.

With the old approach, back-end developers were expected to know much more about the UI and visual aspects of the application. Because of this, they had to pay attention to different segments of the application, instead of focusing on their primary objective. Having the back-end API strictly separated from the UI allowed our developers to focus on the quality of their code.

Also, testing API services is much easier as REST API can be verified by automated unit testing.

I've had some experience in writing my own framework, as well as working with Yii, CakePHP, CodeIgniter, Slim Framework, Symfony and few other open source frameworks. Each time, I've experienced a lack of functionality or awkward approach to some problems.

I used Laravel for four months before deciding to choose it as the platform for our next project. The project itself was a great success and this article is a product of this experience. Now I am able to call myself a Laravel developer.[9]

Figure 5.8 Use Laravel Framework

5.9 Why I Chose Laravel

I’ve already outlined some of my reasons for using Laravel and my experience, so let’s take a closer look at what made Laravel a better choice for my latest project:

- Quick and functional core that can be extended
- Clean and simple routing
- Effective ORM and database layer
- Easy integration with third-party libraries (AWS, export libs, etc.). You can use Composer or Packagist to include libraries in your project
- Active and growing community that can provide quick support and answers
- Supporting unit tests out of the box
- Async queue and background jobs for the long running tasks[9]

5.10 Laravel installation

- Installation
- Configuration
  - Basic Configuration
  - Environment Configuration
  - Configuration Caching
  - Accessing Configuration Values

  - Naming Your Application
- Maintenance Mode[8]

5.10.1 Installation

Server Requirements

The Laravel framework has a few system requirements. Of course, all of these requirements are satisfied by the Laravel Homestead virtual machine:

- PHP >= 5.5.9
- OpenSSL PHP Extension
- PDO PHP Extension
- Mbstring PHP Extension
- Tokenizer PHP Extension

Installing Laravel

Laravel utilizes Composer to manage its dependencies. So, before using Laravel, make sure you have Composer installed on your machine.

Via Laravel Installer

First, download the Laravel installer using Composer:

```
composer global require "laravel/installer"
```

Make sure to place the ~/.composer/vendor/bin directory in your PATH so the laravel executable can be located by your system.

Once installed, the simple laravel new command will create a fresh Laravel installation in the directory you specify. For instance, laravel new blog will create a directory named blog containing a fresh Laravel installation with all of Laravel's dependencies already installed. This method of installation is much faster than installing via Composer:

```
laravel new blog
```

Via Composer Create-Project

Alternatively, you may also install Laravel by issuing the Composer create-project command in your terminal:

```
composer create-project laravel/laravel blog "5.2.1"
```

5.10.2 Configuration

The detail of basic configurations are listed below:

5.10.2.1 Basic Configuration

All of the configuration files for the Laravel framework are stored in the config directory. Each option is documented, so feel free to look through the files and get familiar with the options available to you.

Directory Permissions
After installing Laravel, you may need to configure some permissions. Directories within thestorage and the bootstrap/cache directories should be writable by your web server. If you are using the Homestead virtual machine, these permissions should already be set.

Application Key
The next thing you should do after installing Laravel is set your application key to a random string. If you installed Laravel via Composer or the Laravel installer, this key has already been set for you by the key:generate command. Typically, this string should be 32 characters long. The key can be set in the .env environment file. If you have not renamed the .env.example file to .env, you should do that now. If the application key is not set, your user sessions and other encrypted data will not be secure!

Additional Configuration
Laravel needs almost no other configuration out of the box. You are free to get started developing! However, you may wish to review the config/app.php file and its documentation. It contains several options such as timezone and locale that you may wish to change according to your application.
You may also want to configure a few additional components of Laravel, such as:

- Cache
- Database
- Session

Once Laravel is installed, you should also configure your local environment.

Pretty URLs

Apache
The framework ships with a public/.htaccess file that is used to allow URLs without index.php. If you use Apache to serve your Laravel application, be sure to enable the mod_rewrite module.
If the .htaccess file that ships with Laravel does not work with your Apache installation, try this one:

```
Options +FollowSymLinks
RewriteEngine On

RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^ index.php [L]
```

Nginx

On Nginx, the following directive in your site configuration will allow "pretty" URLs:

```
location / {
    try_files $uri $uri/ /index.php?$query_string;
}
```

Of course, when using Homestead, pretty URLs will be configured automatically.[8]

5.10.2.2 Environment Configuration

It is often helpful to have different configuration values based on the environment the application is running in. For example, you may wish to use a different cache driver locally than you do on your production server. It's easy using environment based configuration.

To make this a cinch, Laravel utilizes the DotEnv PHP library by Vance Lucas. In a fresh Laravel installation, the root directory of your application will contain a .env.example file. If you install Laravel via Composer, this file will automatically be renamed to .env. Otherwise, you should rename the file manually.

All of the variables listed in this file will be loaded into the $_ENV PHP super-global when your application receives a request. You may use the env helper to retrieve values from these variables. In fact, if you review the Laravel configuration files, you will notice several of the options already using this helper!

Feel free to modify your environment variables as needed for your own local server, as well as your production environment. However, your .env file should not be committed to your application's source control, since each developer / server using your application could require a different environment configuration.

If you are developing with a team, you may wish to continue including a .env.example file with your application. By putting place-holder values in the example configuration file, other developers on your team can clearly see which environment variables are needed to run your application.[8]

Accessing The Current Application Environment

The current application environment is determined via the APP_ENV variable from your .env file. You may access this value via the environment method on the App facade:

```
$environment = App::environment();

You may also pass arguments to the environment method to check
if the environment matches a given value. You may even pass
multiple values if necessary:

if (App::environment('local')) {
    // The environment is local
}

if (App::environment('local', 'staging')) {
    // The environment is either local OR staging...
}
```

An application instance may also be accessed via the app helper method:

$environment = app()->environment();

5.10.2.3 Configuration Caching

To give your application a speed boost, you should cache all of your configuration files into a single file using the config:cache Artisan command. This will combine all of the configuration options for your application into a single file which can be loaded quickly by the framework.

You should typically run the php artisan config:cache command as part of your production deployment routine. The command should not be run during local development as configuration options will frequently need to be changed during the course of your application's development.[8]

5.10.2.4 Accessing Configuration Values

You may easily access your configuration values using the global config helper function. The configuration values may be accessed using "dot" syntax, which includes the name of the file and option you wish to access. A default value may also be specified and will be returned if the configuration option does not exist:

```
$value = config('app.timezone');
```

To set configuration values at runtime, pass an array to the config helper:

```
config(['app.timezone' => 'America/Chicago']);
```

5.10.2.5 Naming Your Application

After installing Laravel, you may wish to "name" your application. By default, the app directory is namespaced under App, and autoloaded by Composer using the PSR-4 auto loading standard.

However, you may change the namespace to match the name of your application, which you can easily do via the app:name Artisan command.

For example, if your application is named "Horsefly", you could run the following command from the root of your installation:

```
php artisan app:name Horsefly
```

Renaming your application is entirely optional, and you are free to keep the App namespace if you wish.[8]

5.10.3 Maintenance Mode

When your application is in maintenance mode, a custom view will be displayed for all requests into your application. This makes it easy to "disable" your application while it is updating or when you are performing maintenance. A maintenance mode check is included in the default middleware stack for your application. If the application is in maintenance mode, an HttpException will be thrown with a status code of 503.

To enable maintenance mode, simply execute the down Artisan command:

```
php artisan down
```

To disable maintenance mode, use the up command:

```
php artisan up
```

Maintenance Mode Response Template
The default template for maintenance mode responses is located

inresources/views/errors/503.blade.php.

Maintenance Mode & Queues

While your application is in maintenance mode, no queued jobs will be handled. The jobs will continue to be handled as normal once the application is out of maintenance mode.[8]

5.11 Laravel Core And Routing

The Laravel kernel is hosted on GitHub. The kernel implements an IoC pattern allowing customization and rewriting of any part of the framework (request, logging, authentication, etc.).
Laravel designers didn’t spend too much time reinventing the wheel. A lot of solutions and practices are transferred from other frameworks. A good example of this approach is the extended Symfony console called Artisan, which is a command-line interface included with Laravel.[7]

5.12 Laravel uses Eloquent ORM

Laravel is based on Eloquent ORM. I’ve used it with PostgreSQL and MySQL, and in both cases it performed flawlessly.
The official documentation is comprehensive, so there is no reason to repeat things in this article:
Query Scope – query logic is created as a function with a special scope prefix.
The following example shows the standard SQL Query

Transformed to a query function in Eloquent ORM:

```
SELECT * WHERE hub_id = 100 AND (name LIKE `~%searchkey%` OR
surname LIKE `~%searchkey%`):
function scopeByFieldListLike($query, $fields, $value){
    $query->where(function($query) use ($fields, $value){
        foreach($fields as $field){
            $query->orWhere($field , 'like', "%".$value."%");
        }
    });
    return $query;
}
```

Using a function in your code is straightforward.

```
$model = new User;
$model->byFieldListLike(['name', 'surname'], 'searchkey');
```

A lot Eloquent methods return an instance of the QueryBuilder. You can use almost all of these methods on models.[7]

5.13 Unit Testing

Writing unit tests is usually very time consuming, however, it’s definitely worth the time, so please do it.
Laravel relies on a TestCase base class for this task. It creates a new instance of the application, resolves the route and runs the controller method in its own sandbox. However, it does not run application filters (App::before and App::after) or more complex routing scenarios. To enable these filters in a sandbox environment, I had to manually enable them by using Route::enableFilters().
Obviously, if you know how to use Laravel, you know it could use some more work in the unit testing segment. I’ve set up a few functions that helped me create more advanced unit tests. Feel free to take these and use them in your projects.
To perform more advanced real-life testing and implement “curl-like” requests, I used the kriswallsmith/buzz library. This provided me with a much needed feature set for testing, including custom headers and uploading files.[7]

5.14 Queuing

Long running tasks are a common bottleneck in web applications. A simple example would be the creation of a PDF report and its distribution inside an organisation, which can take a lot of time.

Blocking user requests to perform this action is not a desirable way of addressing the issue. Using Queue in Laravel is a great way of handling and queuing long running tasks in the background.

One of the last issues was PDF generation and sending reports via email. I used a default queue driver for Beanstalkd, supported with the package pda/pheanstalk. It is very easy to add tasks to the queue

Queues are great for tasks like saving data. They can be used to repeat failed jobs, add sleep timeouts before executing jobs, etc.[7]

5.15 Putting It All Together

I've mentioned few key points related to Laravel development in this Laravel review. How you put it all together and build your application will depend on your specific setup and project. However, here is a brief checklist of steps you might want to consider:

- Setup homestead Vagrant box prepared for development using Laravel and configured local environment.
- Add endpoints (configure routes).
- Add authentication layer.
- Add filters before/after application execution, giving you an option to run anything before and after application is executed. Additionally, you can define filters for any route.

My default "before" filter creates a global scope as a singleton and starts the timer for monitoring performance:

```
class AppBefore {
     function filter($request) {
         \App::singleton('scope', function() {
               $scope = new AppScope();//My scope implementation
          $scope->startTimer();
          return $scope;
         });
     }
}
```

The "after" filter stops the timer, logs it to the performance log, and sends CORS headers.

```
class AppAfter{
      function filter($request, $response){
        $response->header('Access-Control-Allow-Origin', '*');
```

```
       $response->header('Access-Control-Allow-Methods', 'GET,
POST, OPTIONS');
      $response->header('Access-Control-Allow-Headers',
'Content-Type');
       $response->header('Access-Control-Max-Age', '86400');
        \App::make('scope')->endTimer();
       \App::make('scope')->logTotalTime();
         return $response;
       }
}
```

Developing each of your modules will likely require these steps:

Setup data migration, and seed default data. Implement models and relations, transformers, validators and sanitizers (if needed). Write unit tests. Implement controllers and execute tests.[7]

5.16 Why Laravel is user Friendly Framework

Laravel is a free and Open Source PHP Web Application Framework. It is designed for the development of different web application according to the client requirements. It represents a simple, speed and beautiful Syntax so that you can express yourself in coding very well and become creative every time with more ideas. The unique thing about the Laravel PHP framework is that it is very expressive and beautiful coding platform and that’s why i am very sure about that you will get the good experience when you will do coding with this platform.

As we all quite aware about the Laravel is a PHP Framework and very demanding in the market today. For that Here I would like to mentioned the list of Laravel based CMS’s types that include: OctoberCMS, PongoCMS, PyroCMS and Indira CMS. Here i shared the Laravel based CMS so you can get multiple options on it. And I just like to add that Laravel PHP Framework is the platform where you must love to do coding as it is provide a better web environment to us. Now its new updated version is also released frequently so it gives the healthy competition to other technologies also. As per my knowledge conferences was already held in different location to make sense that people will aware about new technology and at least you stay with this industry for a longtime.

Laravel is a modern Framework for PHP Programming language to develop the Web applications. It is built by professional Web developers for the people like you who is very passionate about coding and want to grow with the technology with great sense of understanding on it.

Here I want to share that It was also declared the most popular PHP Framework in the year of 2013-2014 that is very important to know for the web developers who is very much interested to do a powerful coding in their own way. The only reason to create this user friendly web application by developers to provide you a better platform as they can. You can also compare the features of web application framework with others so you can get more idea about why you have to use a Laravel PHP Framework. I hope that you all will keep doing quality work on the user friendly framework like Laravel Web application and give the best work as you can using advance technology for your precious client.[10]

5.17 Difference Between Laravel VS Codeigniter

With more than four years of experience with Codeigniter PHP framework, we believe that learning, setting up and working on it are all pretty easy. Even our junior developers are able to inculcate the skills necessary for scheduled delivery, from a business standpoint, which is lucrative.

Overtime, though we realised that perhaps a PHP framework that is modern with a lot more flexibility in terms of a structured coding pattern with scope for applications that we deliver to perform better is required. In a nutshell, this is the reason behind the switchover to Laravel PHP framework.

Laravel is an open-source PHP web application framework based on MVC architecture. The framework was created and maintained by Taylor Otwell since June 2011. The security feature of Laravel is prompt in taking appropriate action as and when there is a breach in security. The syntax patterns of Laravel are expressive and elegant. The reason behind the rapid emergence of this framework are:

- Modular packaging with composer based dependency manager.
- Class autoloader.
- Eloquent ORM, the advanced active-record-pattern implementation
- Query builder as an alternative to Eloquent ORM, for direct access to the database.
- Support of database platforms including MySQL, PostgreSQL, and SQLServer.
- RESTFul state control over controller methods.
- Blade templating engine, simple yet flexible.
- Rapid automation with the inbuilt authentication mechanism, caching mechanism, powerful routing features, and session control.
- IO component to handle requests and responses.
- Artisan command-d along with sample codes line interface.
- Unit testing support.
- Great documentation.

The documentation of Laravel is detailed and along with sample codes, it’s easier to learn the technology. There are quite a few other successful PHP frameworks that the developers have been using over the years, like Codeigniter, CakePHP, Zend, Yii, Symfony and many more. Each of these frameworks has a reputation and have established themselves in the industry. The question is, why should we then even consider Laravel?
Well, the intention is not to compare frameworks and find out which one is better. Rather, the focus is on the circumstances that would cause a PHP developer or a web development firm to move to a different PHP framework platform like Laravel.[7]

Chapter 6
System Design (UML)

6.1 Use Case

In software and systems engineering, a use case is a list of actions or event steps, typically defining the interactions between a role (known in the Unified Modeling Language as an *actor*) and a system, to achieve a goal. The actor can be a human or other external system. In systems engineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals. The detailed requirements may then be captured in the Systems Modeling Language (SysML) or as contractual statements.

Use case analysis is an important and valuable requirement analysis technique that has been widely used in modern software engineering since its formal introduction by Ivar Jacobson in 1992. Use case driven development is a key characteristic of many process models and frameworks such as ICONIX, the Unified Process (UP), the IBM Rational Unified Process (RUP), and the Oracle Unified Method (OUM). With its inherent iterative, incremental and evolutionary nature, use case also fits well for agile development.[12]

Figure: 6.1 Use Case

6.1.1 Elements of a Use Case Diagram

A use case diagram captures the business processes carried out in the system. Normally, domain experts and business analysts should be involved in writing use cases. Use cases are created when the requirements of a system need to be captured. A use case diagram is quite simple in nature and depicts two types of elements: one representing the business roles and the other representing the business processes. Let us take a closer look at use at what elements constitute a use case diagram.

- **Use cases**. A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.
- **Actors**. An actor is a person, organization, or external system that plays a role in one or more interactions with your system. Actors are drawn as stick figures.
- **Associations**. Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case. Associations are modeled as lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line. The arrowhead is often used to indicating the direction of the initial invocation of the relationship or to indicate the primary actor within the use case. The arrowheads are typically confused with data flow and as a result I avoid their use.
- **System boundary boxes (optional)**. You can draw a rectangle around the use cases, called the system boundary box, to indicates the scope of your system. Anything within the box represents functionality that is in scope and anything outside the box is not.

System boundary boxes are rarely used, although on occasion I have used them to identify which use cases will be delivered in each major release of a system.

- **Packages (optional)**. Packages are UML constructs that enable you to organize model elements (such as use cases) into groups. Packages are depicted as file folders and can be used on any of the UML diagrams, including both use case diagrams and class diagrams. I use packages only when my diagrams become unwieldy, which generally implies they cannot be printed on a single page, to organize a large diagram into smaller ones.[12]

6.1.2 Use Case Diagram for Administrator

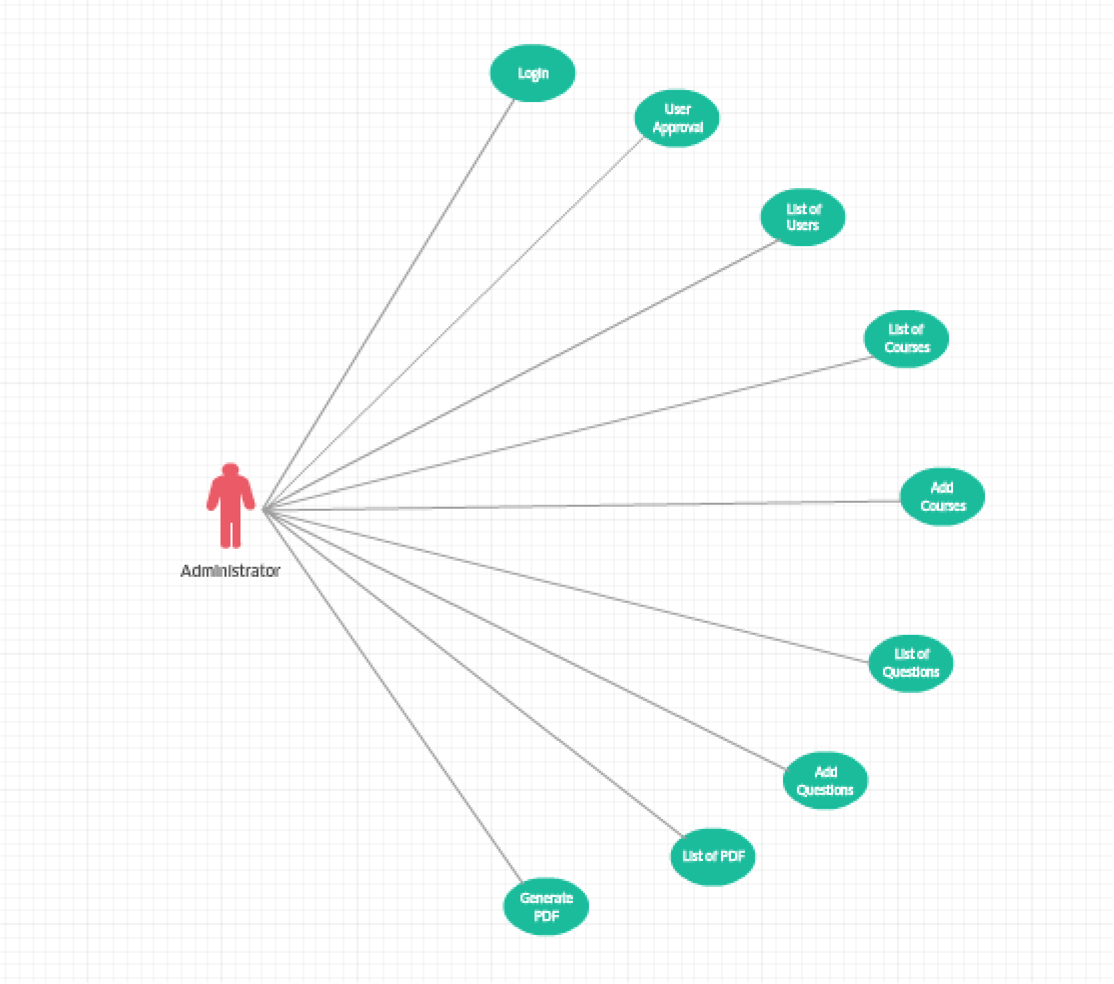


Figure: 6.1.2 Use Case Diagram for Administrator

6.1.3 Use Case Diagram for Teacher

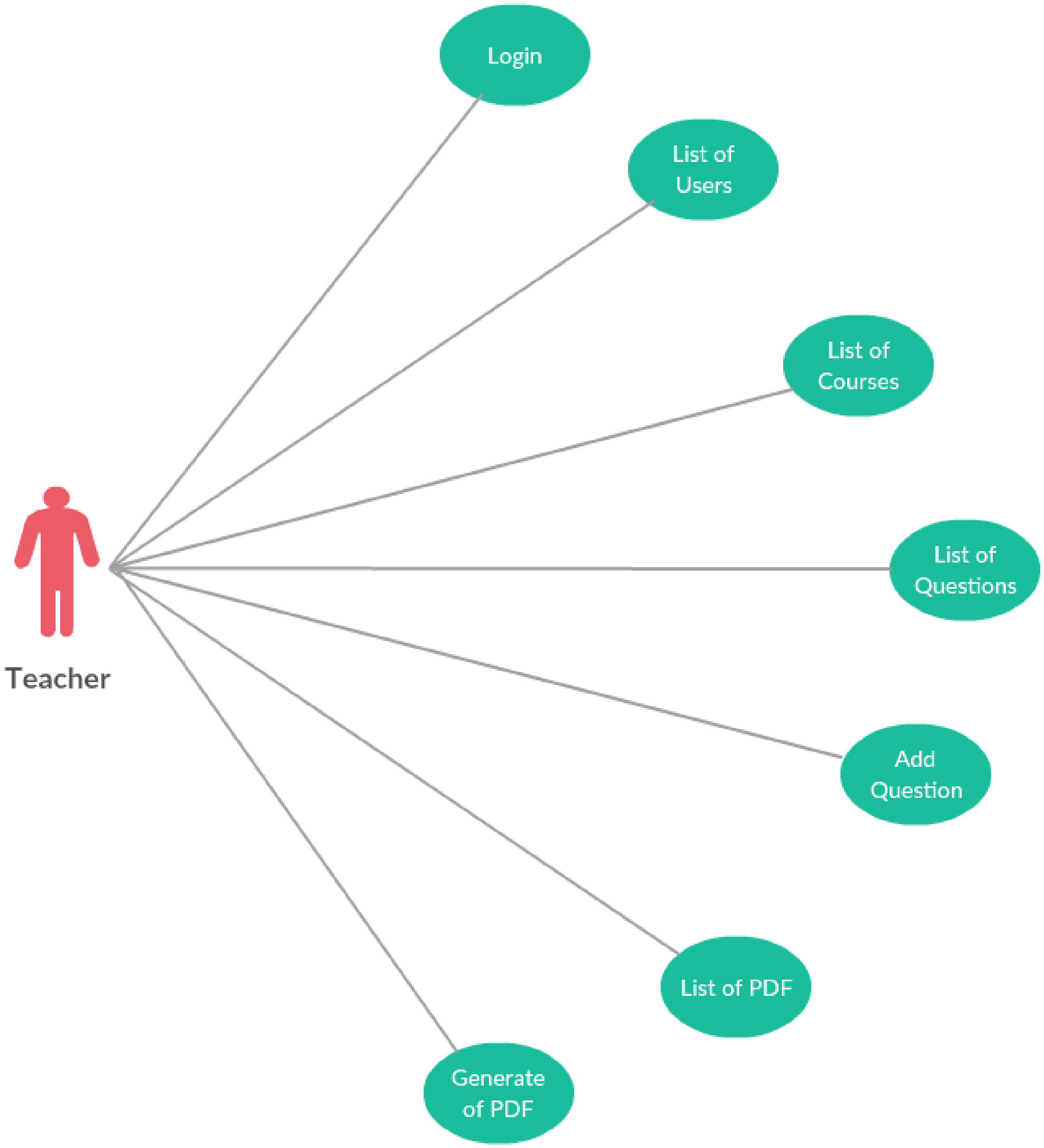
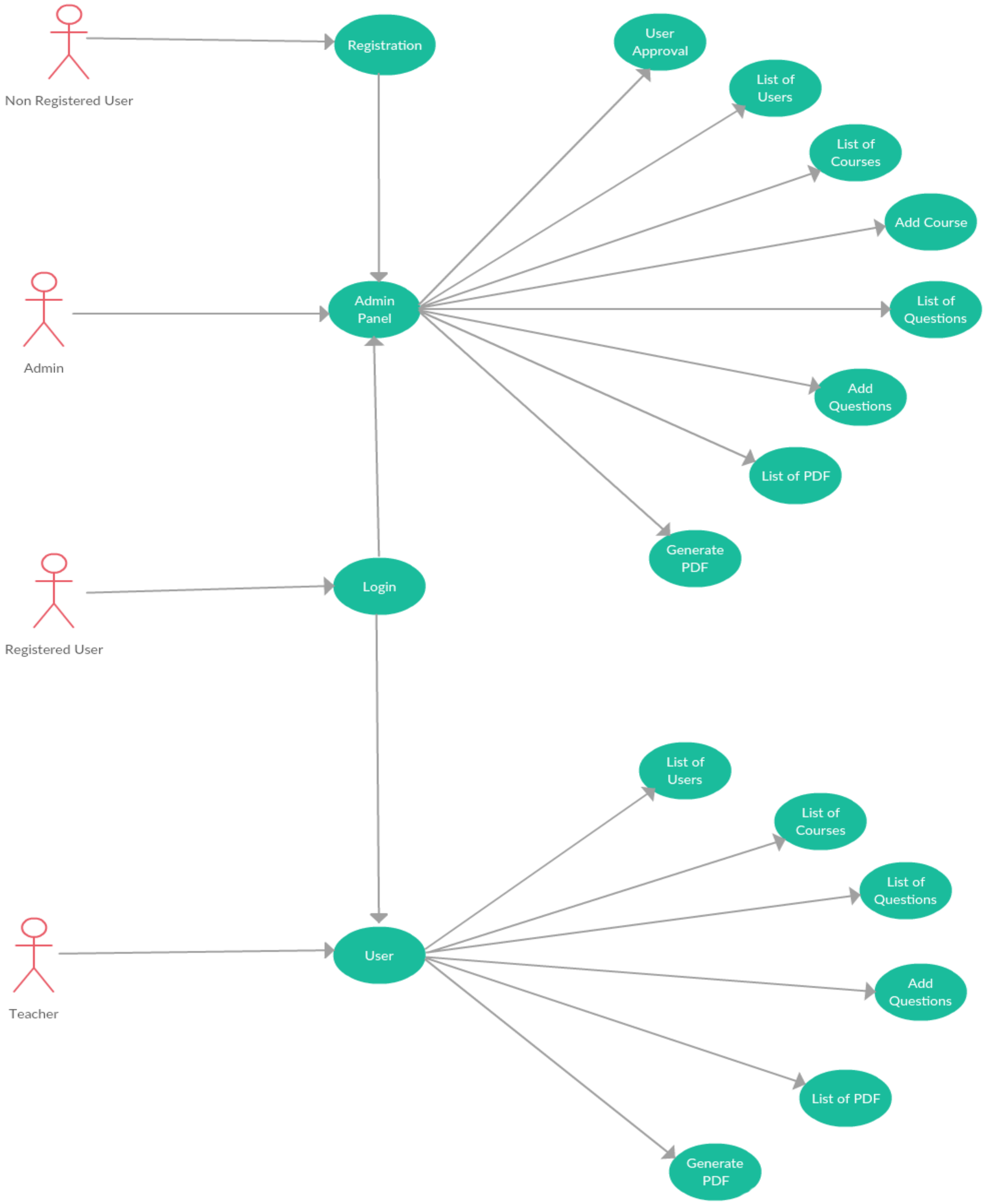


Figure: 6.1.3 Use Case Diagram for Teacher

6.1.4 Use Case Diagram for Question Bank



Figures: 6.1.4 Use Case Diagram for Question Bank

6.2 ER Diagram for Question Bank

An entity-relationship diagram (ERD) is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and can be used as the foundation for a relational database.

While useful for organizing data that can be represented by a relational structure, an entity-relationship diagram can't sufficiently represent semi-structured or unstructured data, and an ERD is unlikely to be helpful on its own in integrating data into a pre-existing information system.[13]

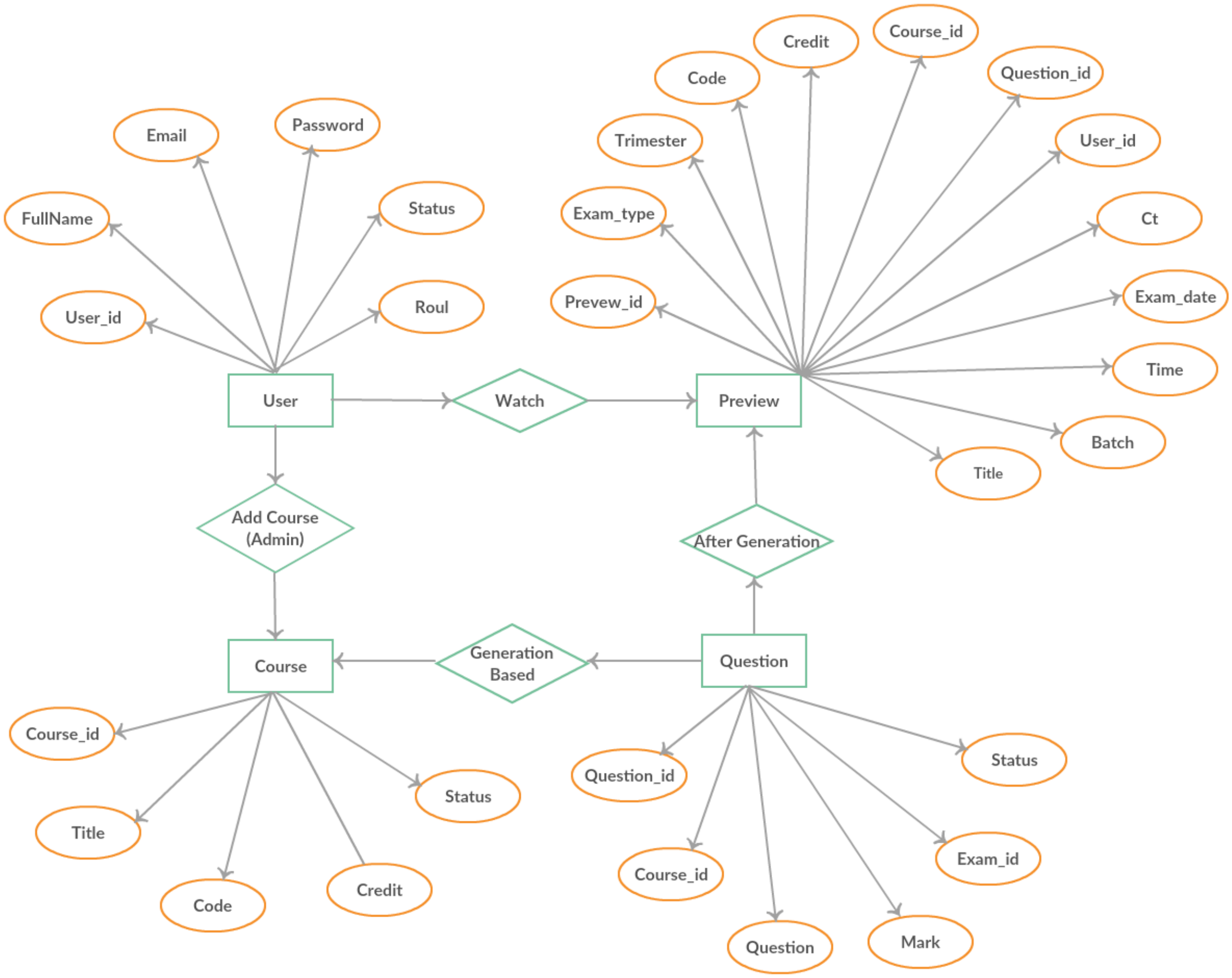
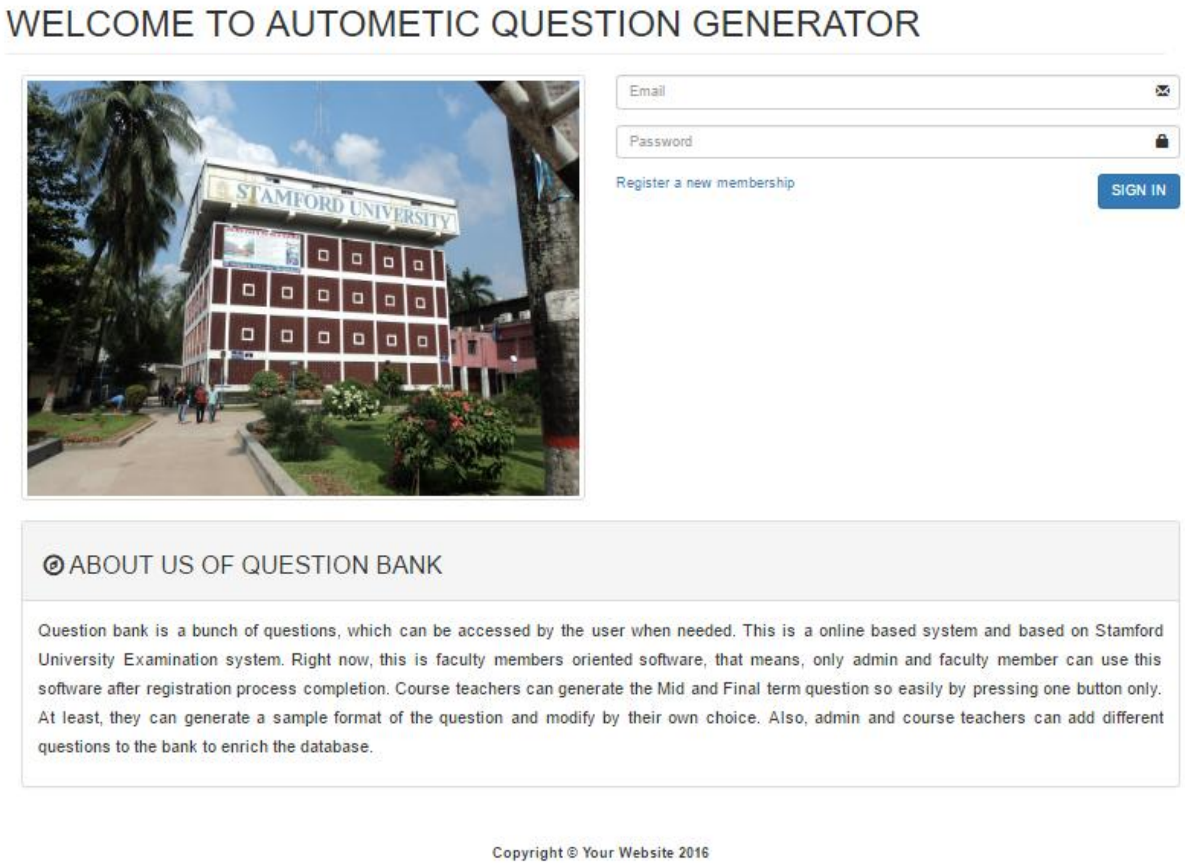


Figure: 6.2 ER Diagram

Chapter 7
Software Overview

7.1 Home page

A home page is generally the main page where a visitor navigating to a website from a search engine will see, and may also serve as a landing page to attract the attention of visitors. The home page is used to facilitate navigation to other pages on the site, by providing links to important and recent articles and pages. A home page can also refer to the first page that appears upon opening a web browser, sometimes called the start page, although the home page of a website can be used as a start page.



WELCOME TO AUTOMETIC QUESTION GENERATOR

ABOUT US OF QUESTION BANK

Question bank is a bunch of questions, which can be accessed by the user when needed. This is a online based system and based on Stamford University Examination system. Right now, this is faculty members oriented software, that means, only admin and faculty member can use this software after registration process completion. Course teachers can generate the Mid and Final term question so easily by pressing one button only. At least, they can generate a sample format of the question and modify by their own choice. Also, admin and course teachers can add different questions to the bank to enrich the database.

Copyright © Your Website 2016

Figure: 7.1 Home Page

7.2 Registration system:

A registration system contains some fields such as Name, Email, Password, Contact number, Age, Occupation, Blood Group Address etc.
where we need to provide members details. So as usual we have also created a registration system which is required to become a member of this application. Every user should fill all the fields given in registration page. Just click "register" and complete the process. The fields of our registration system given below:

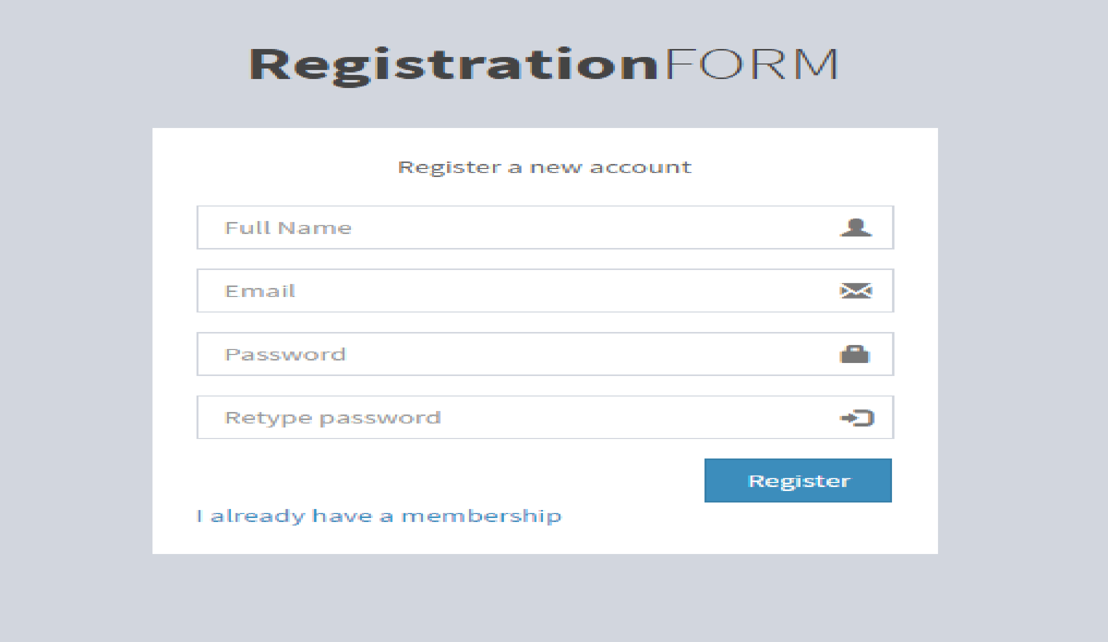


Figure 7.2 Registration Form

7.3 Login /Sign In process:

When a user activates the account successfully, he/she can log in by using the email and password. It will redirect the user to the home page of the site. All the registered users will get all the facilities but non registered users can view the Home page, Events, Information section to get a crystal clear idea about our website, unregistered users also can access into contact page to contact. Our login system has two fields such as Email Address, Password. The system also contains two options of "Forgot password" which will be used to recover forgotten password.

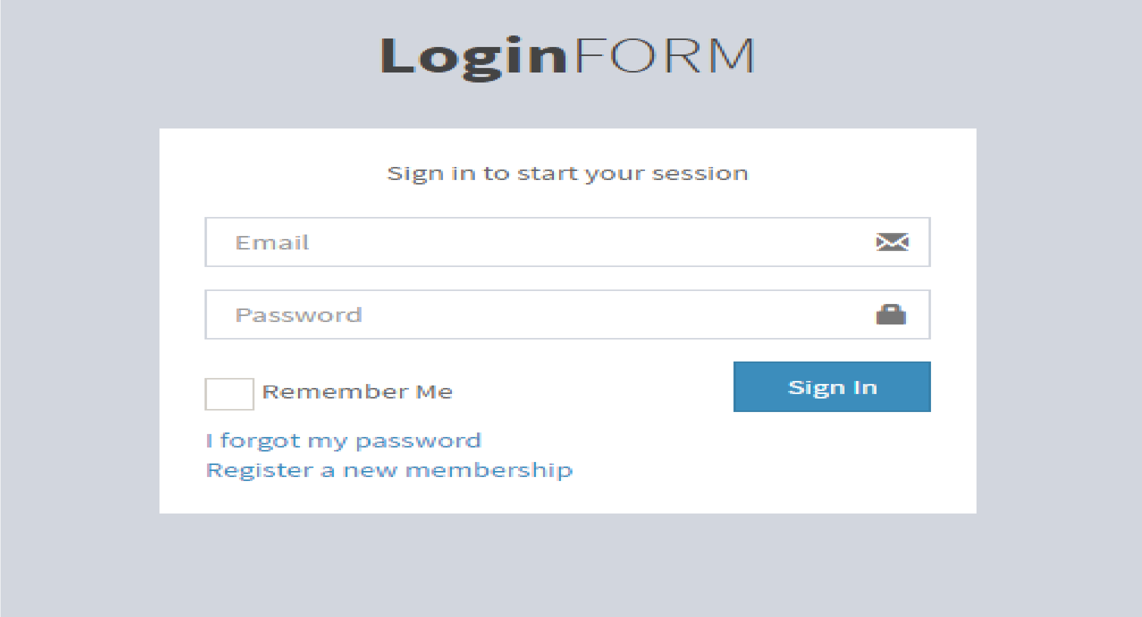
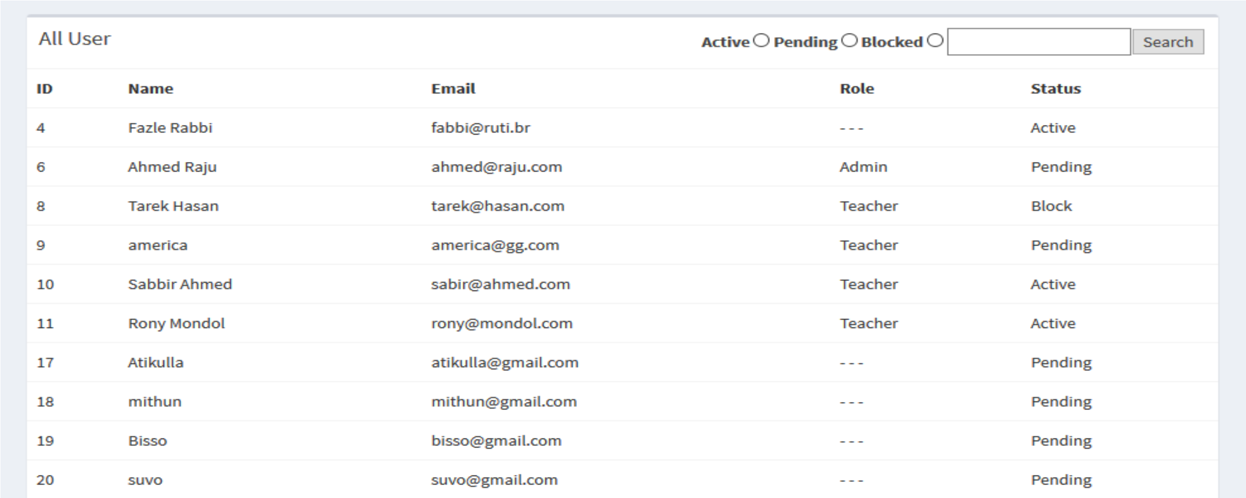


Figure 7.2 Login Form

7.4 Functions of user (As Teacher):

7.4.1 List of users:

Every registered user also can see the list of registered users name, email, role, status. User can search another registered user by their name or email address



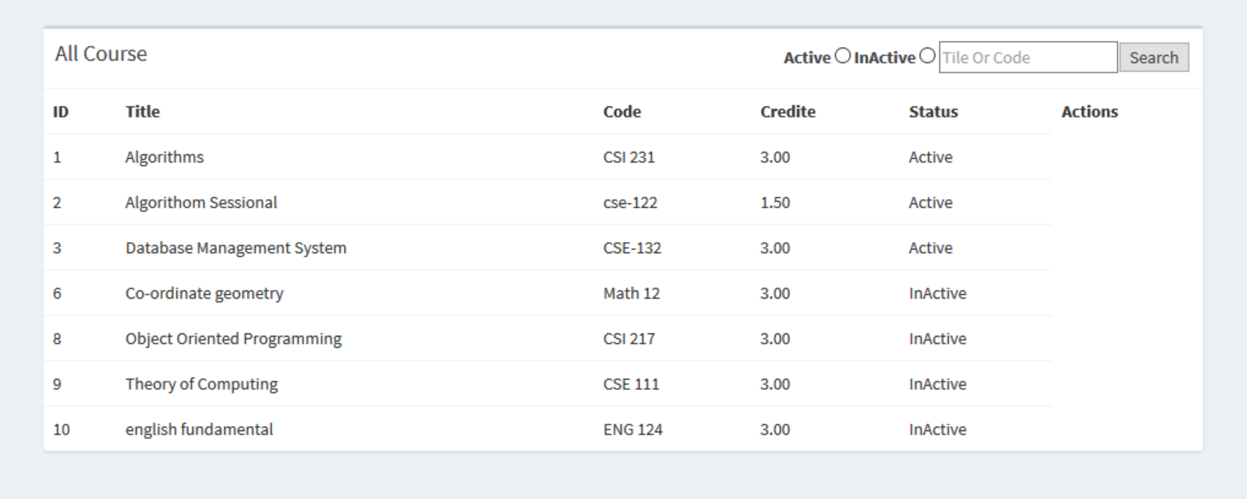
All User

ID	Name	Email	Role	Status
4	Fazle Rabbi	fabbi@ruti.br	- - -	Active
6	Ahmed Raju	ahmed@raju.com	Admin	Pending
8	Tarek Hasan	tarek@hasan.com	Teacher	Block
9	america	america@gg.com	Teacher	Pending
10	Sabbir Ahmed	sabir@ahmed.com	Teacher	Active
11	Rony Mondol	rony@mondol.com	Teacher	Active
17	Atikulla	atikulla@gmail.com	- - -	Pending
18	mithun	mithun@gmail.com	- - -	Pending
19	Bisso	bisso@gmail.com	- - -	Pending
20	suvo	suvo@gmail.com	- - -	Pending

Figure: 7.4.1 List of user

7.4.2 List of Courses:

Every registered user also can see the list of courses and also the courses Title, Code, Credit, Status etc. User can search Course by their Title, Code.



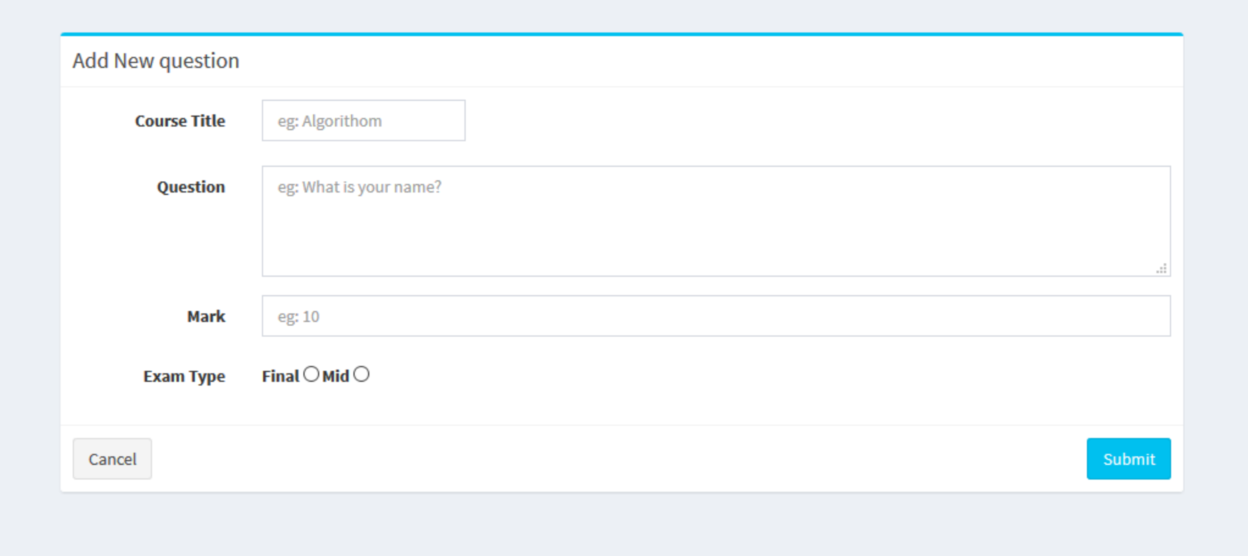
All Course

ID	Title	Code	Credite	Status	Actions
1	Algorithms	CSI 231	3.00	Active	
2	Algorithom Sessional	cse-122	1.50	Active	
3	Database Management System	CSE-132	3.00	Active	
6	Co-ordinate geometry	Math 12	3.00	InActive	
8	Object Oriented Programming	CSI 217	3.00	InActive	
9	Theory of Computing	CSE 111	3.00	InActive	
10	english fundamental	ENG 124	3.00	InActive	

Figure: 7.4.2 List of Courses

7.4.3 Add Question:

Every registered user can add question (Course Title, Question, Mark and Exam Type).



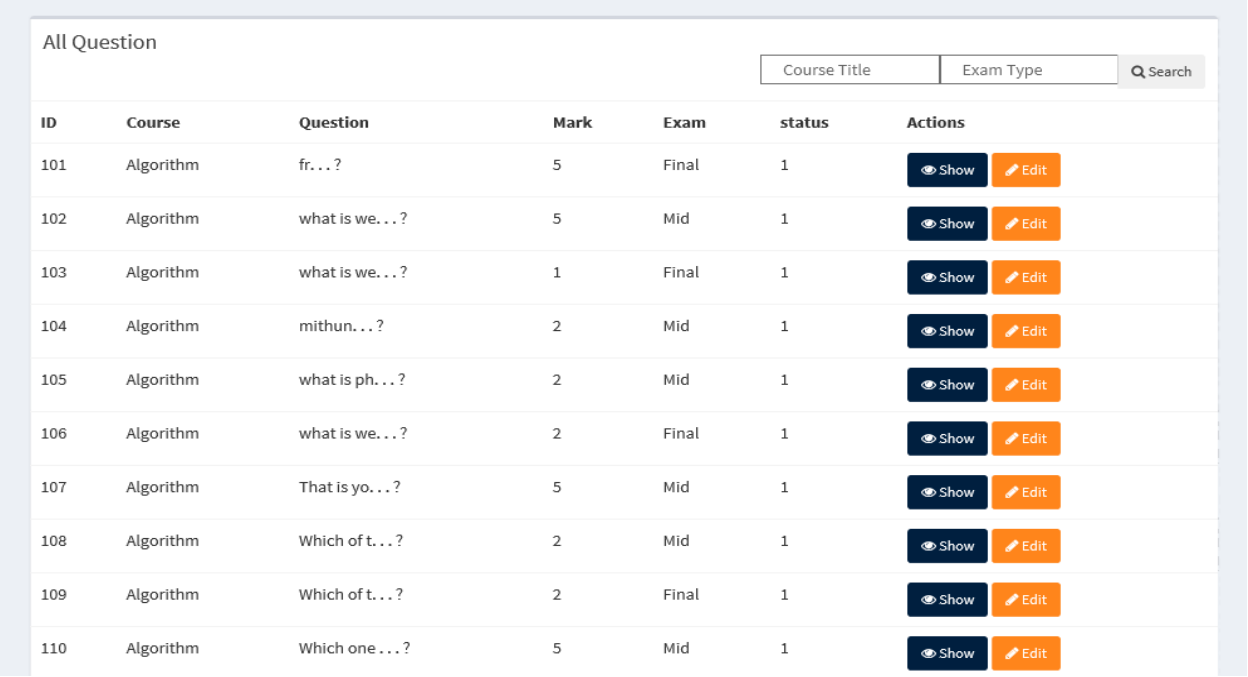
Add New question

Course Title	eg: Algorithom
Question	eg: What is your name?
Mark	eg: 10
Exam Type	Final ○ Mid ○

Figure: 7.4.3 Add Question

7.4.4 List of Questions:

Every registered user also can see the list of questions and also the Course, Question, Mark, Exam, Status etc. User can search questions by their course title, exam type. User also able to see and edit the questions but unable to delete questions.



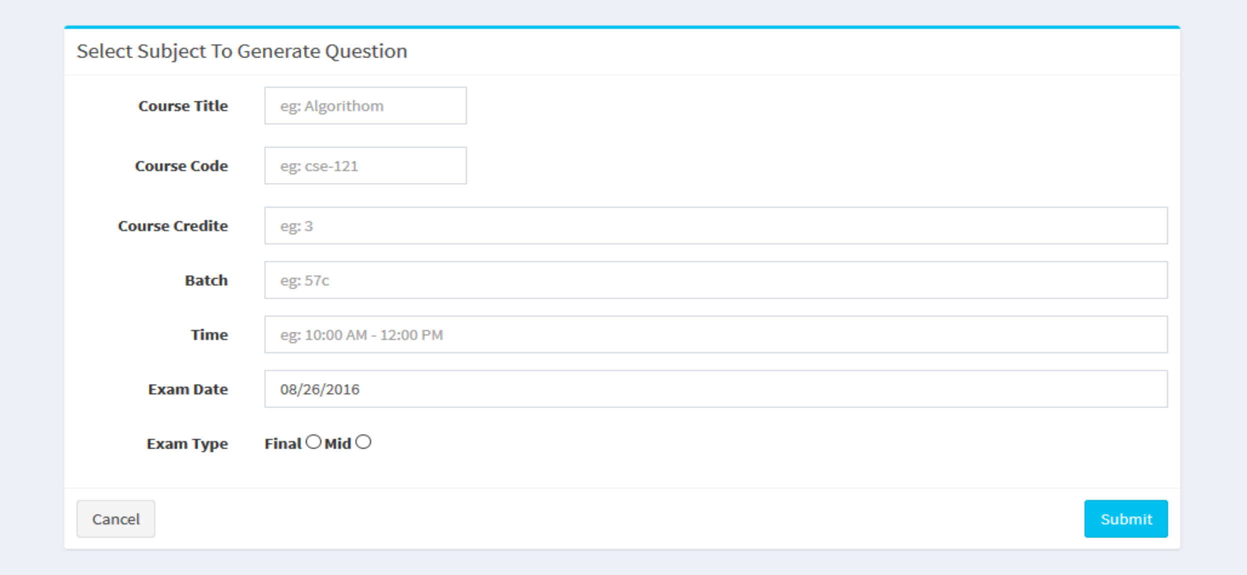
All Question

ID	Course	Question	Mark	Exam	status	Actions
101	Algorithm	fr. . . ?	5	Final	1	Show Edit
102	Algorithm	what is we. . . ?	5	Mid	1	Show Edit
103	Algorithm	what is we. . . ?	1	Final	1	Show Edit
104	Algorithm	mithun. . . ?	2	Mid	1	Show Edit
105	Algorithm	what is ph. . . ?	2	Mid	1	Show Edit
106	Algorithm	what is we. . . ?	2	Final	1	Show Edit
107	Algorithm	That is yo. . . ?	5	Mid	1	Show Edit
108	Algorithm	Which of t. . . ?	2	Mid	1	Show Edit
109	Algorithm	Which of t. . . ?	2	Final	1	Show Edit
110	Algorithm	Which one . . . ?	5	Mid	1	Show Edit

Figure: 7.4.4 List of Question

7.4.5 Generate PDF:

Registered user can generate new questions randomly in PDF (Portable Document Format) version. This PDF contains Course Title, Code, Credit, Batch, Time, Exam Date, Exam Type, Marks etc.



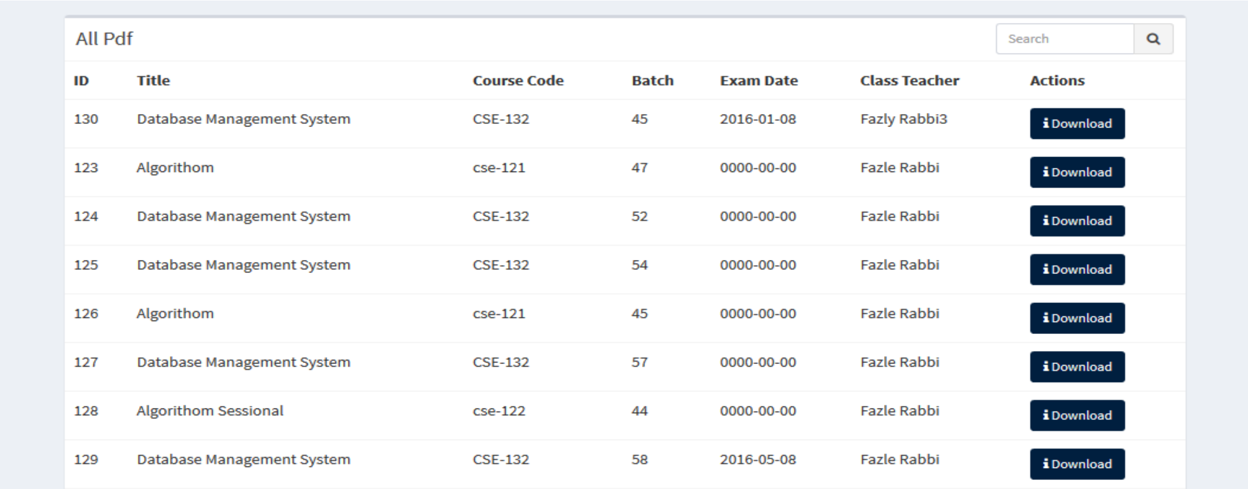
Select Subject To Generate Question

Course Title	eg: Algorithom
Course Code	eg: cse-121
Course Credite	eg: 3
Batch	eg: 57c
Time	eg: 10:00 AM - 12:00 PM
Exam Date	08/26/2016
Exam Type	Final ○ Mid ○

Figure: 6.4.5 Generate PDF

7.4.6 List of PDF:

Every registered user also can see the list of Courses, Title, Course Code, Batch, Exam Date, and Course Teacher etc. User can search Course by their Title, Code, and Batch. User can download Questions from this section.



All Pdf

ID	Title	Course Code	Batch	Exam Date	Class Teacher	Actions
130	Database Management System	CSE-132	45	2016-01-08	Fazly Rabbi3	Download
123	Algorithom	cse-121	47	0000-00-00	Fazle Rabbi	Download
124	Database Management System	CSE-132	52	0000-00-00	Fazle Rabbi	Download
125	Database Management System	CSE-132	54	0000-00-00	Fazle Rabbi	Download
126	Algorithom	cse-121	45	0000-00-00	Fazle Rabbi	Download
127	Database Management System	CSE-132	57	0000-00-00	Fazle Rabbi	Download
128	Algorithom Sessional	cse-122	44	0000-00-00	Fazle Rabbi	Download
129	Database Management System	CSE-132	58	2016-05-08	Fazle Rabbi	Download

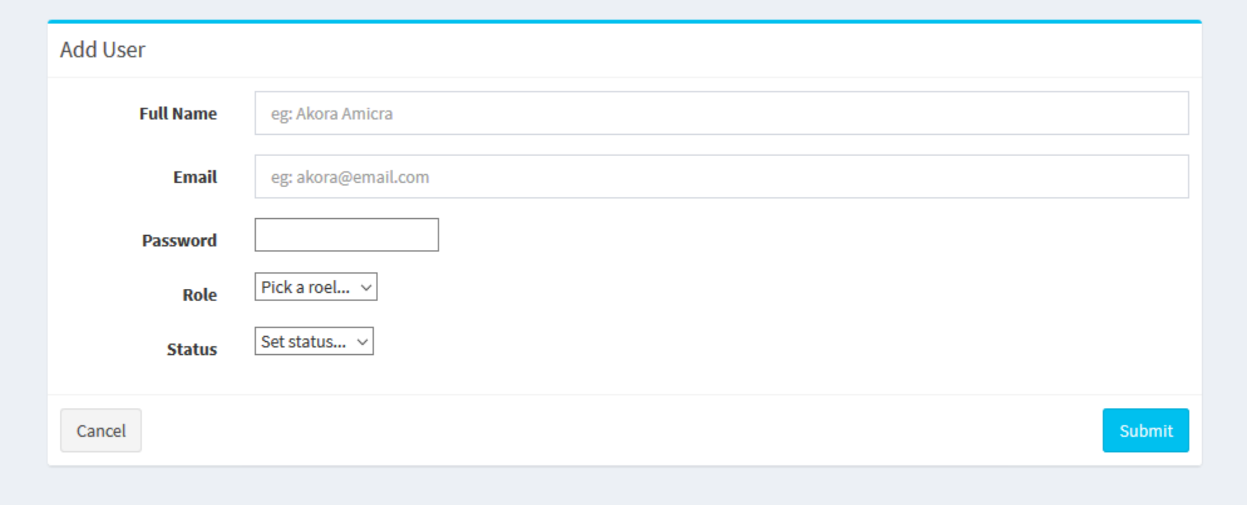
Figure: 7.4.6 List of PDF

7.5 Functions of Admin:

The details of the functions of the admin are given below:

7.5.1 Add user:

Every registered user can add another user but admin should approve this request before login of another user.



Add User

Full Name: eg: Akora Amicra

Email: eg: akora@email.com

Password:

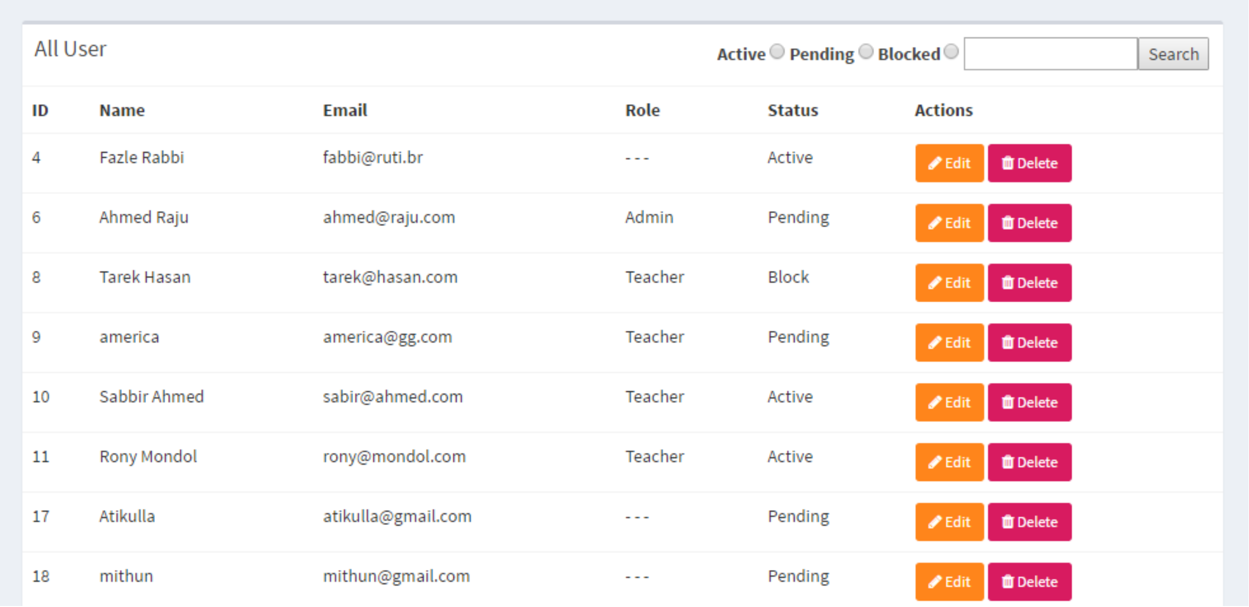
Role: Pick a roel...

Status: Set status...

Figure: 7.5.1 Add user

7.5.2 List of users:

Admin can see the list of registered users and their details. Admin also able to edit, delete user besides these admin can search registered users by their name or email address.



All User

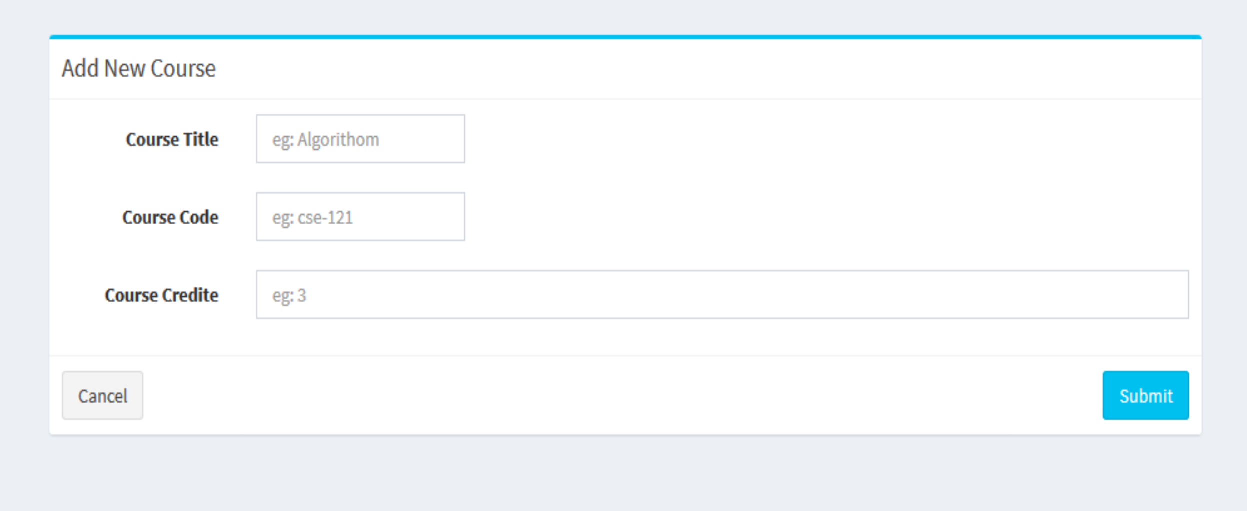
Active ○ Pending ○ Blocked ○

ID	Name	Email	Role	Status	Actions
4	Fazle Rabbi	fabbi@ruti.br	- - -	Active	Edit Delete
6	Ahmed Raju	ahmed@raju.com	Admin	Pending	Edit Delete
8	Tarek Hasan	tarek@hasan.com	Teacher	Block	Edit Delete
9	america	america@gg.com	Teacher	Pending	Edit Delete
10	Sabbir Ahmed	sabir@ahmed.com	Teacher	Active	Edit Delete
11	Rony Mondol	rony@mondol.com	Teacher	Active	Edit Delete
17	Atikulla	atikulla@gmail.com	- - -	Pending	Edit Delete
18	mithun	mithun@gmail.com	- - -	Pending	Edit Delete

Figure: 6.5.2 List of Users

7.5.3 Add Course:

Every registered user can add course (title, code, credit) but admin should approve this request to add new course.



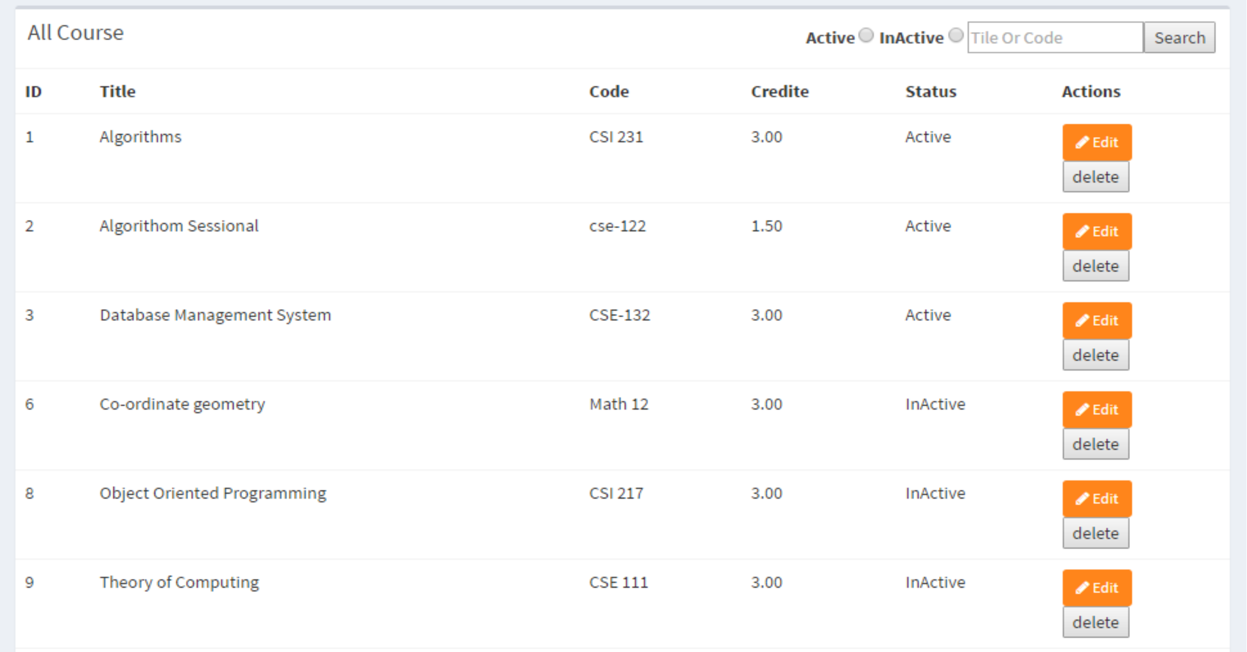
Add New Course

Course Title	eg: Algorithom
Course Code	eg: cse-121
Course Credite	eg: 3

Figure: 7.5.3 Add Course

7.5.4 List of Courses:

Admin can edit or delete courses from this part of application besides this admin able to search courses by their Title or code.



All Course

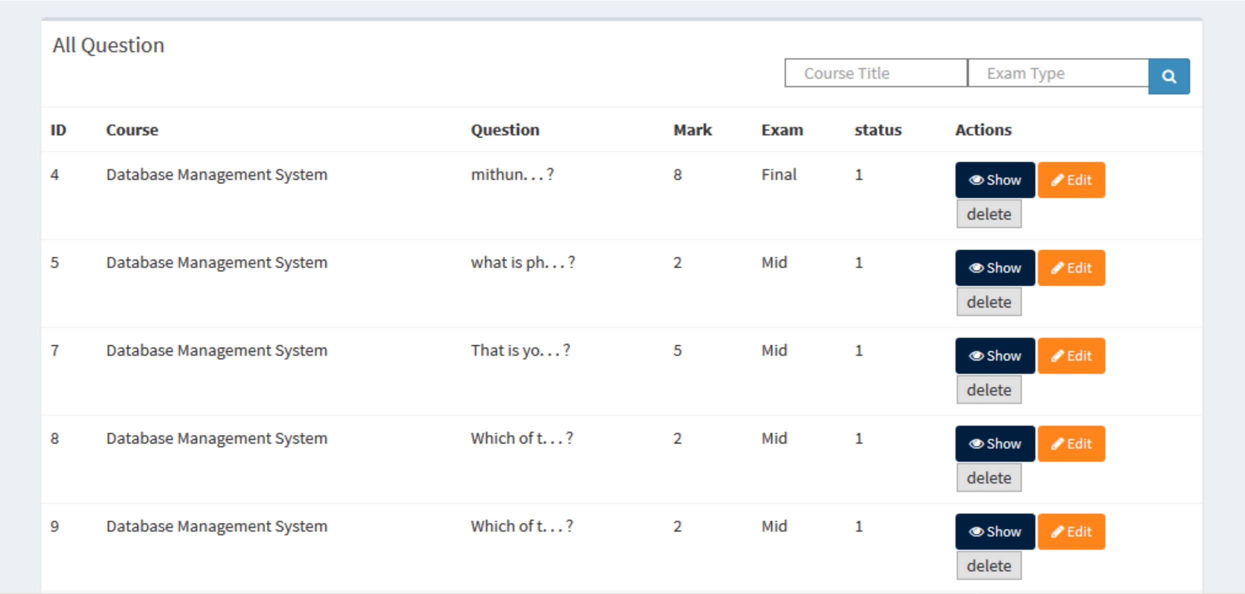
Active ○ InActive ○ Tile Or Code

ID	Title	Code	Credite	Status	Actions
1	Algorithms	CSI 231	3.00	Active	Edit delete
2	Algorithom Sessional	cse-122	1.50	Active	Edit delete
3	Database Management System	CSE-132	3.00	Active	Edit delete
6	Co-ordinate geometry	Math 12	3.00	InActive	Edit delete
8	Object Oriented Programming	CSI 217	3.00	InActive	Edit delete
9	Theory of Computing	CSE 111	3.00	InActive	Edit delete

Figure: 7.5.4 List of Courses

7.5.5 List of Questions:

Admin can see the list of questions as a registered user. Admin also able delete questions from this option.



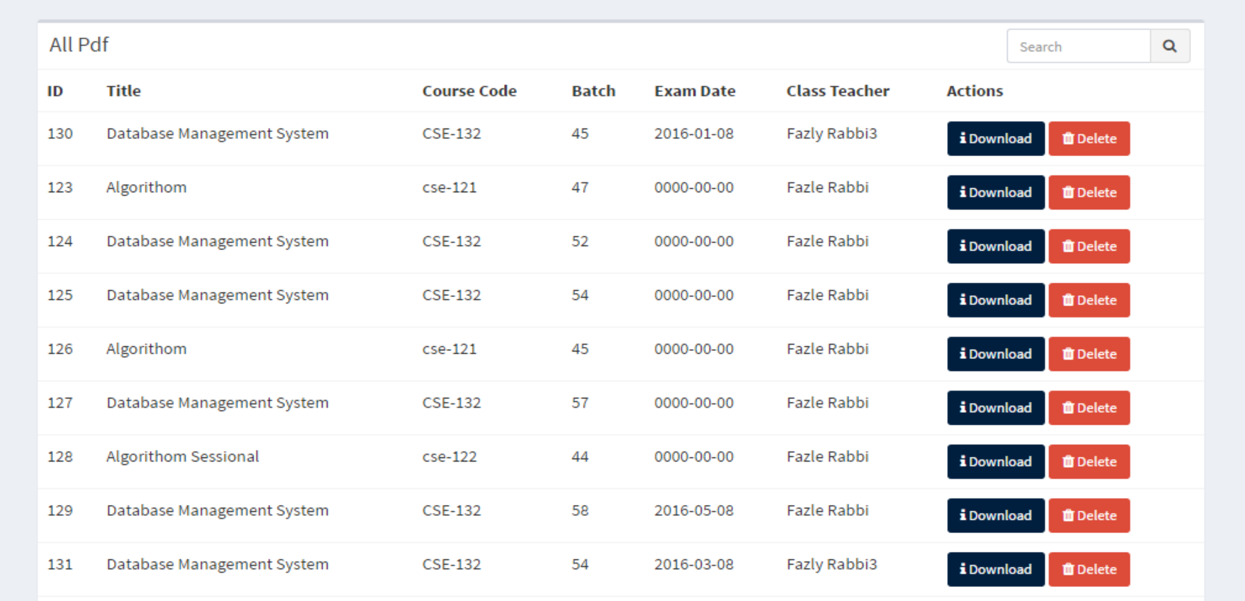
All Question

ID	Course	Question	Mark	Exam	status	Actions
4	Database Management System	mithun. . . ?	8	Final	1	Show Edit delete
5	Database Management System	what is ph. . . ?	2	Mid	1	Show Edit delete
7	Database Management System	That is yo. . . ?	5	Mid	1	Show Edit delete
8	Database Management System	Which of t. . . ?	2	Mid	1	Show Edit delete
9	Database Management System	Which of t. . . ?	2	Mid	1	Show Edit delete

Figure: 6.5.5 List of Question

7.5.6 List of PDF:

Admin can generate new questions as a teacher besides this admin also able to delete any PDF file from this option.



All Pdf

ID	Title	Course Code	Batch	Exam Date	Class Teacher	Actions
130	Database Management System	CSE-132	45	2016-01-08	Fazly Rabbi3	Download Delete
123	Algorithom	cse-121	47	0000-00-00	Fazle Rabbi	Download Delete
124	Database Management System	CSE-132	52	0000-00-00	Fazle Rabbi	Download Delete
125	Database Management System	CSE-132	54	0000-00-00	Fazle Rabbi	Download Delete
126	Algorithom	cse-121	45	0000-00-00	Fazle Rabbi	Download Delete
127	Database Management System	CSE-132	57	0000-00-00	Fazle Rabbi	Download Delete
128	Algorithom Sessional	cse-122	44	0000-00-00	Fazle Rabbi	Download Delete
129	Database Management System	CSE-132	58	2016-05-08	Fazle Rabbi	Download Delete
131	Database Management System	CSE-132	54	2016-03-08	Fazly Rabbi3	Download Delete

Figure: 7.5.6 List of PDF

Chapter 8
Conclusion & Future Plan

8.1 Conclusion & Future Plan

- No repetition of similar questions in consecutive two trimesters.
- Enrich the database of courses and respective questionnaires.
- Make a strong collection of questions.
- Manually editing of questions after generation.

APPENDIX

APPENDIX A

Laravel Framework PHP Codes

I. Login User And Admin

```
public function postLogin(Request $request)
    {
        if (Auth::attempt(['email' => $request->email, 'password' =>$request->password])) {
           if(Auth::user()->status != 1){
               Auth::logout();
               flash('You need to be approved by admin before login','warning');
               return redirect()->back();
           }
        return redirect('dashboard');
       }else{
        flash('Invalid username or password','warning');
       return redirect()->back();
       }
    }
```

II. Registration Users

```
public function store(Request $request){

        if(Auth::user()){
                    if(Auth::user()->role !=1){
                    flash('You have no permission for this section','danger');
                    return redirect()->back();
                    }
        }
                    $this->validate($request, [
        'email' => 'required|unique:users|max:55',
        'fullname' => 'required',
         'password' => 'min:6|required',
        'repassword' => 'min:6|same:password'
        ]);
                               $user = new User();
                    $data['password'] = $request->input('password');
                    $data['repassword'] = $request->input('repassword');
                    $user->password = bcrypt($data['password']);
                    $user->fullname = $request->input('fullname');
                    $user->email  = $request->input('email');
                    if(empty($request->input('status'))){
                            $user->status  = 0;
                    }else{
                            $user->status  = $request->input('status');
                    }

                    if(empty($request->input('role'))){
                            $user->role  = 0;
                    }else{
                            $user->role  = $request->input('role');
                    }
                            $done = $user->save();
                            if($done){
                                    flash('Operation successful','success');
                            }else{
                                    flash('Oh Allah! Something went wrong','danger');
                            }
                    $users = User::all();
                            return redirect('/user');
                }
```

III. Approve Of User Login

```
public function updateUser(Request $request)
        {
                if(!empty($request->password)){
                        $input['password'] = bcrypt($request->password);
                }
                        $input['fullname'] = $request->fullname;
                        $input['email'] = $request->email;
                        $input['role'] = $request->role;
                        $input['status'] = $request->status;
                        $user = User::find($request->user_id);
                $done = $user->update($input);
                if($done){
                                flash('Operation successful','success');
                        }else{
                                flash('Oh Allah! Something went wrong','danger');
                        }
                        return redirect('user');
}
```

IV. List of all Users

```
public function allUser(Request $request)
        {
                $key = $request->input('key');
        $status = $request->input('status');
        $input = $request->only(['key','starus']);
        $users = User::orderBy('user_id');
        if(!empty($key)){
            $users->where('email','like','%'.$key.'%');
            $users->orWhere('fullname','like','%'.$key.'%');
        }
        if(!empty($status)){
            if($status =="Active" ){
                $status = 1;
            }elseif($status == "Pending"){
                $status = 0;
            }else{
                $status = -1;
            }
            $users->where('status','like','%'.$status.'%');
        }
        $users = $users->paginate(12);
        $users->appends($input);
        return view('admin.user.users',compact('users'));
    }
```

V. List if all Courses

```
public function index(Request $request)
{
     $key = $request->input('key');
      $status = $request->input('status');
     $input = $request->only(['key','status']);
      $courses = Course::orderBy('course_id');
     if(!empty($key)){
    $courses->where('title','like','%'.$key.'%');
    $courses->orWhere('code','like','%'.$key.'%');
     }
     if(!empty($status)){
         if($status =="Active" ){
          $status = 1;
         }else{
              $status = 0;
        }
     $courses->where('status','like','%'.$status.'%');
       }
     $courses = $courses->paginate(12);
        $courses->appends($input);
        return view('admin.course.courses',compact('courses'));
}
```

VI. Add Course

```
public function store(Request $request)
{
  if(Auth::user()->role != 1){
      flash('You have no permission to delete.','warning');
      return redirect()->back();
     }
      $input=$request->all();
      Course::create($input);
       return redirect ('course');
   }
```

VII. List of All Question

```
public function index(Request $request)
{
     $input = $request->only(['type','title']);
      $type = $request->type;
       $title = $request->title;
      $questions=Question::join('courses','questions.course_id','=','courses.course_id')
      ->where('exam_type', 'LIKE', '%'.$type.'%')
     ->Where('title', 'LIKE', '%'.$title.'%');
      $questions = $questions->paginate(20);
     $questions->appends($input);
     return view('admin.question.questions',compact('questions'));
   }
```

VIII. Add Question

```
public function store(Request $request)
{
    $input=$request->all();
    Question::create($input);
     return redirect('question');
   }
```

IX. List of PDF

```
public function index(Request $request)
 {
   $key = $request->input('key');
   $allpdf = Preview::orderBy('preview_id','desc');
   if(!empty($key)){
       $allpdf->where('code','like','%'.$key.'%');
       $allpdf->orWhere('ct','like','%'.$key.'%');
     }

    $allpdf = Preview::paginate(12);
    return view('admin.generator.all',compact('allpdf'));
  }
```

X. Random question Generator

```
public function prepareQuestion($allQuestion)
{
        $total = count($allQuestion) - 1;
   $checkId = array();
  $questions = array();
  $ins = '';
 for ($i = 0; $i < 4; $i++) {
        $ind = 0;
        $marks = 0;
            $max = 10;
            $set = array();
         while (true) {
             $ind = rand(0, $total);
              if (!in_array($ind, $checkId)) {
                  if ($allQuestion[$ind]['mark'] <= $max) {
                      $checkId[] = $ind;
                      $set[] = $allQuestion[$ind];
                      $marks += $allQuestion[$ind]['mark'];
                      $ins .= $allQuestion[$ind]['question_id'] . ",";
                      $max = 10 - $marks;
                   }
         }
           if ($marks >= 10) {
                   $questions[] = $set;
                    break;
              }
           }
        }
        session(['ins' => $ins]);
        return $questions;
    }

    public function preview(Request $request)
    {
        $exam_type = $request->input('exam_type');
        $course_id = $request->input('course_id');
        $batch = $request->input('batch');
    $date = $request->input('date');
        $date = date("Y-d-m", strtotime($date));
        $time = $request->input('time');
        if (empty($exam_type) || empty($course_id) || empty($batch) || empty($date) ||
empty($time)) {
            flash('Please check all field','warning');
            return back()->withInput();
        }
   $allQuestion = Question::where('exam_type', '=', $exam_type)->where('course_id', '=',
$course_id)->get()->toArray();
       $questions = $this->prepareQuestion($allQuestion);

        if ($exam_type == "Final") {
            $exam = "Final Examination";
        } else {
            $exam = "Mid Term Examination";
        }
        $m = date('M');
 if (in_array($m, ['Jan', 'Feb', 'Mar', 'Apr'])) {
        $trimister = "Spring " . date('Y');
     } elseif (in_array($m, ['May', 'un', 'Jul', 'Aug'])) {
          $trimister = "Summer " . date('Y');
        } else {
            $trimister = "Fall " . date('Y');
        }
            $course = Course::where('course_id', '=', $course_id)->take(1)->get();
             $title = $course[0]['title'];
              $credite = $course[0]['credite'];
               $code = $course[0]['code'];
                $question_id = session('ins');
                $head = [
             'exam_type' => $exam
             , 'title' => $title
            , 'credite' => $credite
           , 'code' => $code
            , 'trimister' => $trimister
            , 'batch' => $batch
             , 'exam_date' => $date
                , 'time' => $time
            , 'question_id' => $question_id
           , 'course_id' => $course[0]['course_id']
            , 'user_id' => Auth::user()->user_id
            ,'ct'=> Auth::user()->fullname
  ];
  $request->session()->forget('ins');
    session(['data' => $head]);
if ($request->has('download')) {
            Pdfq::create($head);
            $pdf = PDF::loadView('admin.generator.preview',compact('questions', 'head'));
             return $pdf->download('pdfview.pdf');

             }
             return view('admin.generator.preview', compact('questions', 'head'));
            }
```

XI. Download PDF

```
public function download(Request $request,$id=null)
{
 $head=NULL;
 if(session('data')){
            $head = session('data');
            $request->session()->forget('data');
}

$mark = 0;
$questions = array();
 $set = array();
 if($head){
            $pdf=Preview::create($head);
            $id =$pdf->preview_id;
}
$data = Preview::find($id);

 if(empty($head)) {
            $head['exam_type'] = $data->exam_type;
            $head['title'] = $data->title;
            $head['credite'] = $data->credite;
            $head['code'] = $data->code;
            $head['trimister'] = $data->trimister;
             $head['batch'] = $data->batch;
            $head['exam_date'] = $data->exam_date;
             $head['time'] = $data->time;
            $head['ct'] = $data->ct;
            $head['course_id'] = $data->course_id;
 }
            $dd = explode(',',$data->question_id);
             $total = count($dd)-1;

            for($i=0;$i<$total;$i++){
```

```
        $set[$i] = Question::find($dd[$i]);
        $mark += $set[$i]['mark'];
        if($mark == 10){
        $mark = 0;
        $questions[]= $set;
        $set = null;
        }
        }
                $pdf                                                        =
PDF::loadView('admin.generator.preview_download',compact( 'head','questions'));
        $name = $head['title'].$head['batch'];
        if($head)
        unset($head);
        return $pdf->download($name.'.pdf');
}
```

APPENDIX B

Database Connection in Laravel Framework

I. User table

```
Schema::create('users', function (Blueprint $table) {
        $table->increments('user_id',10);
        $table->string('fullname',255);
        $table->string('email')->unique();
        $table->string('password',255);
        $table->tinyInteger('status',4);
        $table->tinyInteger('role',4);
        $table->rememberToken();
        $table->timestamps();
    });
```

II. Question:

```
Schema::create('questions', function (Blueprint $table) {
        $table->increments('question_id',10);
        $table->intger('course_id',11);
        $table->text('question');
        $table->tinyInteger('mark',3);
        $table->tinyInteger('exam_type',10);
        $table->tinyInteger('status',4);
        $table->timestamps();
    });
```

III. Course:

```
Schema::create('courses', function (Blueprint $table) {
        $table->increments('course_id',3);
        $table->string('code',7);
        $table->string('title',80);
        $table->tinyInteger('credite',1);
        $table->timestamps();
    });
```

IV. Preview:

```
Schema::create('courses', function (Blueprint $table) {
        $table->increments(' preview_id');
        $table->string('exam_type',10);
        $table->string('trimister',20);
```

```
   $table->string('code',8);
   $table->decimal('credite', 3, 2);
   $table->intger('course_id',5);
   $table->intger('question_id',50);
   $table->intger('user_id',7);
   $table->string('ct',50);
   $table->string('exam_date',1);
   $table->string('time',20);
   $table->string('batch',10);
   $table->string('title',50);
   $table->timestamps();
});
```

REFERENCES

[1] “Mordent Question Bank Website Page” Available: (Access Date: 05 June 2016) https://docs.moodle.org/29/en/Question_bank#Add_a_new_question

[2] “systems development life cycle (SDLC) ” Available: (Access Date: 11 June 2016) https://en.wikipedia.org/wiki/Systems_development_life_cycle

[3] “Agile Model Design, Agile Model Pros and Cons” Available: (Access Date: 15 June 2016) http://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm

[4] “Wikipedia official page in description PHP (Hypertext Preprocessor) ” Available: (Access Date: 27 June 2016) https://en.wikipedia.org/wiki/PHP

[5] “PHP official page History of PHP, History of PHP Version, PHP 3, PHP4” Available: (Access Date: 01 July 2016) http://php.net/manual/en/history.php.php

[6] “What is Framework” Available: (Access Date: 07 July 2016) https://www.linkedin.com/pulse/why-laravel-best-php-framework-2016-ramoliya-hitesh

[7] “Why Use A PHP Framework, Best PHP Framework For Development” Available: (Access Date: 20 July 2016) http://www.hongkiat.com/blog/best-php-frameworks/

[8] “Laravel Framework, Laravel installation” Available: (Access Date: 30 July 2016) https://laravel.com/

[9] “Why Use Laravel Framework, Why I Chose Laravel” Available: (Access Date: 05 August 2016) https://www.toptal.com/laravel/why-i-decided-to-embrace-laravel

[10] “Why Laravel is user Friendly Framework” Available: (Access Date: 09 August 2016) http://iflair.com/why-you-should-choose-a-laravel-php-framework/

[11] “Difference Between Laravel VS Codeigniter” Available: (Access Date: 15 August 2016) https://www.codeclouds.com/blog/laravel-vs-codeigniter-a-difficult-choice/

[12] “Use Case, Elements of a Use Case Diagram,” Available: (Access Date: 23 August 2016) https://www.e-education.psu.edu/geog468/l8_p4.html

[13] “ER Diagram for Question Bank” Available: (Access Date: 28 August 2016) http://searchcrm.techtarget.com/definition/entity-relationship-diagram