**CHAPTER 1**

**INTRODUCTION:**

On-line examinations contents providers to focus on creating effective assessment questions and focusing on exam’s feedback delivery to students. In the paper we present techniques that are pertinent to the elements of assessment process: answers submission, computerized grading, and feedback after submission.

As the modern organizations are automated and computers are working as per the instructions, it becomes essential for the coordination of human beings, commodity and computers in a modern organization.

The administrators ,instructor,Students who are attending for online examination can communicate with the system through this projects, thus facilitating effective implementation and monitoring of various activities of Online Examinations like conducting Exams as per scheduled basis and delivering result to that particular use or student.And the details of students who attempted Online Examination are maintained at administrator.

**CHAPTER 2**

**ANALYSIS:**

**2.1 SYSTEM ANALYSIS:**

1. Existing System

Existing system is a manual one in which users are maintaining books to store the information like Student Details,Instructor Details,Schedule Details and feedbacks about students who attempted exam as per schedule.. It is very difficult to maintain historical data.

DISADVANTAGES:

The following drawbacks of existing system emphasize the need for computerization:

1. A lot of copies of question papers have to be made

2. A lot of correction work hence delay in giving the results

3. A lot of tabulation work for each subject results

2. Proposed System

This application is used to conduct online examination. The students can sit at individual terminals and login to write the exam in the given duration. . The questions have to be given to the students.This application will perform correction, display the result immediately and also store it in database. This application provides the administrator with a facility to add new exams.This application provides the Instructor add questions to the exam, modify questions in the exam in a particular exam. This application takes care of authentication of the administrator,Instructor as well as the student.

1. Objective of the System

The objective of the Online Examination Tool is to provide better information for the users of this system for better results for their maintainence in student examination schedule details and grading details.

# 

**2.2 System Specifications**

Hardware Requirements:-

* Pentium-IV(Processor).
* 256 MB Ram
* 512 KB Cache Memory
* Hard disk 10 GB
* Microsoft Compatible 101 or more Key Board

**Software Requirements: -**

* **Operating System :** Windows
* **Web-Technology: PHP**
* **Front-End: HTML,CSS,JAVASCRIPT**
* **Back-End:** MySQL
* **Web Server:** Apache SERVER.

**CHAPTER 3**

**DESIGN APPROACH**

3.1 INTRODUCTION:

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer’s requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

3.2 UML Diagrams:

Actor:A coherent set of roles that users of use cases play when interacting with the use `cases.

Use case:

A description of sequence of actions, including variants, that a system performs that yields anobservable result of value of an actor.

UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

There are various kinds of methods in software design:

They are as follows:

* Use case Diagram
* Sequence Diagram
* Collaboration Diagram
* Activity Diagram
* State chat Diagram

**USECASE DIAGRAMS:**Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what’s called an actor.

Use case diagram can be useful for getting an overall view of the system and clarifying who can do and more importantly what they can’t do.

Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

* The purpose is to show the interactions between the use case and actor.
* To represent the system requirements from user’s perspective.
* An actor could be the end-user of the system or an external system.

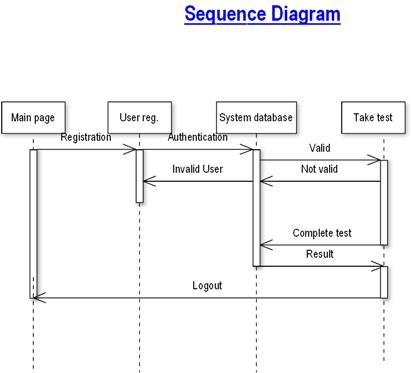
**USECASE DIAGRAM:**

A Use case is a description of set of sequence of actions. Graphically it is rendered as an ellipse with solid line including only its name. Use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationship. It is an association between the use cases and actors. An actor represents a real-world object. Primary Actor – Sender, Secondary ActorReceiver.

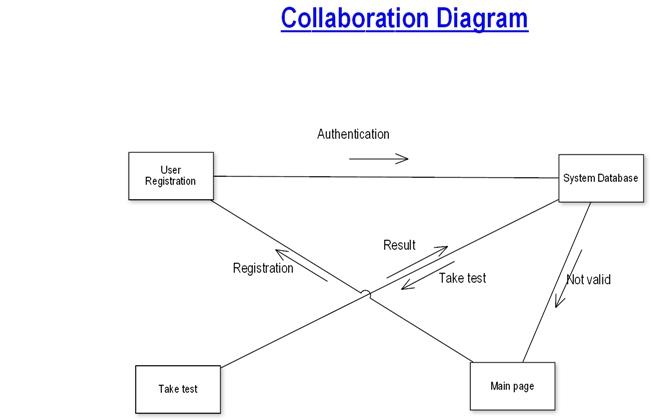


**SEQUENCE DIAGRAM:**

Sequence diagram and collaboration diagram are called INTERACTION DIAGRAMS. An interaction diagram shows an interaction, consisting of set of objects and their relationship including the messages that may be dispatched among themA sequence diagram is an introduction that empathizes the time ordering of messages. Graphically a sequence diagram is a table that shows objects arranged along the X-axis andmessages ordered in increasing time along the Y-axis



**COLLABORATION DIAGRAM:**

A collaboration diagram is an introduction diagram that emphasizes the structural organization of the objects that send and receive messages. Graphically a collaboration diagram is a collection of vertices and arcs.

**CLASS DIAGRAM:**

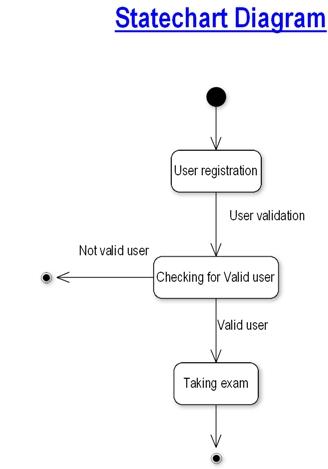
Class is nothing but a structure that contains both variables and methods. The Class Diagram shows a set of classes, interfaces, and collaborations and their relating ships. There is most common diagram in modeling the object oriented systems and are used to give the static view of a system. It shows the dependency between the classes that can be used in our system.

The interactions between the modules or classes of our projects are shown below. Each block contains Class Name, Variables and Methods.

**CLASS:**

A description of set of objects that share the same attributes, operations, relationships, and semantics.

**State Chart Diagram**



**3.3 DATA FLOW DIAGRAMS**:

The DFD takes an input-process-output view of a system i.e. data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software. Data objects represented by labeled arrows and transformation are represented by circles also called as bubbles. DFD is presented in a hierarchical fashion i.e. the first data flow model represents the system as a whole. Subsequent DFD refine the context diagram (level 0 DFD), providing increasing details with each subsequent level. The DFD enables the software engineer to develop models of the information domain & functional domain at the same time. As the DFD is refined into greater levels of details, the analyst perform an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of the data as it moves through the process that embody the applications.A context-level DFD for the system the primary external entities produce information for use by the system and consume information generated by the system. The labeled arrow represents data objects or object hierarchy.

Database

**RULES FOR DFD:**

* Fix the scope of the system by means of context diagrams.
* Organize the DFD so that the main sequence of the actions
* Reads left to right and top to bottom.
* Identify all inputs and outputs.
* Identify and label each process internal to the system with Rounded circles.
* A process is required for all the data transformation and Transfers.
* Do not indicate hardware and ignore control information.
* Make sure the names of the processes accurately convey everything the process is done.
* There must not be unnamed process.
* Indicate external sources and destinations of the data, with Squares.
* Number each occurrence of repeated external entities.
* Identify all data flows for each process step, except simple Record retrievals.
* Label data flow on each arrow.
* Use details flow on each arrow.
* Use the details flow arrow to indicate data movements.

**DATAFLOW DIAGRAMS:**

**User registration**

User registration

Take Test

**User registration**

Register user

Register user

Search for user details

Update user details

view user details

**Taking Test**

Taking test

Start Exam

View Result

End Exam

**3.4** **E-R Diagrams:**

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represents data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design For the database designer, the utility of the ER model is:

* it maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
* it is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
* In addition, the model can be used as a design plan by the database developer to implement a data model in a specific database management software.

## Connectivity and Cardinality

The basic types of connectivity for relations are: one-to-one, one-to-many, and many-to-many. A one-to-one (1:1) relationship is when at most one instance of a entity A is associated with one instance of entity B. For example, "employees in the company are each assigned their own office. For each employee there exists a unique office and for each office there exists a unique employee.

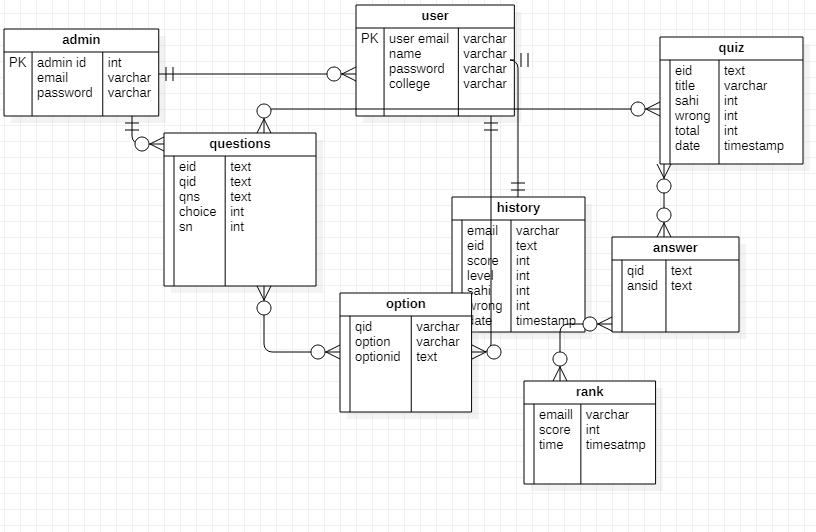
A one-to-many (1:N) relationships is when for one instance of entity A, there are zero, one, or many instances of entity B, but for one instance of entity B, there is only one instance of entity A. An example of a 1:N relationships is a department has many employees each employee is assigned to one department A many-to-many (M:N) relationship, sometimes called non-specific, is when for one instance of entity A, there are zero, one, or many instances of entity B and for one instance of entity B there are zero, one, or many instances of entity A. The connectivity of a relationship describes the mapping of associated

Figure: ER diagram

#### CHAPTER 4

#### PROJECT MODULES:

#### 4.1.Admin module:

#### register:to be authenticated firest have to be registered.

#### login:the registered user can be allowed to view inner details for which he permitted

#### change password&forgotpassword:user has rights to modify his login details& also be informed through mails if he is unable to login.

#### student -modifing details:user can be modified to change status of each user.

#### departments-entering/modifying details:new departments adding and old departmentd deletions are spend by this user.

#### instructor details-modifying details:according to staff he can add or delete instructors for specific platforms.

### 4.2 Instructor module:

### Register:to be authenticated firest have to be registered.

### Login:the registered user can be allowed to view inner details for which he permitted

### Change password&forgotpassword:user has rights to modify his loging details& also be informed through mails if he is unable to login

### Add questions-departments verifing:according to flow of questions & technology he can add questions into the database.

### Update questions -departments verifing:if any corrections in data of questions he can modify them

### Create exams:he will be prepared schedule for exams periodically.

### Update exams:he has rights to modify exam schedule.

### View exam details- view no of registered students, view no of attended students:can view at attended students who has registered.

### Evaluate question:multiple choice tue/false:evaluation of marks based on his initiations when adding questions

### 4.3. Student details:

### Register:to be authenticated firest have to be registered

### Login:the registered user can be allowed to view inner details for which he permitted

### Take exam- multiple choice, true/false:the registred student allowed to start the ex

### See exam results:after completion of exam he can view at his result.

### Logout:after the process of examination he turned to logout page.

**CHAPTER 5**

# **CONCEPTS AND TECHNIQUES**

5.1 OVERVIEW OF TECHNOLOGIES USED

PHP

PHP: Hypertext Preprocessor, is a widely used, general-purpose [scripting language](http://en.wikipedia.org/wiki/Scripting_language) that was originally designed for [web development](http://en.wikipedia.org/wiki/Web_development), to produce [dynamic web pages](http://en.wikipedia.org/wiki/Dynamic_web_page). It can be embedded into [HTML](http://en.wikipedia.org/wiki/HTML) and generally runs on a [web server](http://en.wikipedia.org/wiki/Web_server), which needs to be configured to process PHP code and create [web page](http://en.wikipedia.org/wiki/Web_page) content from it. It can be deployed on most web servers and on almost every [operating system](http://en.wikipedia.org/wiki/Operating_system) and [platform](http://en.wikipedia.org/wiki/Platform_(computing)) free of charge.

PHP was originally created by [Rasmus Lerdorf](http://en.wikipedia.org/wiki/Rasmus_Lerdorf) in [1995](http://en.wikipedia.org/wiki/1995) and has been in continuous development ever since. The main implementation of PHP is now produced by The PHP Group and serves as the [de facto standard](http://en.wikipedia.org/wiki/De_facto_standard) for PHP as there is no [formal specification](http://en.wikipedia.org/wiki/Formal_specification).PHP is [free software](http://en.wikipedia.org/wiki/Free_software) released under the [PHP License](http://en.wikipedia.org/wiki/PHP_License), which is incompatible with the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License) (GPL) because of restrictions on the use of the term PHP

PHP has evolved to include a [command line interface](http://en.wikipedia.org/wiki/Command_line_interface) capability and can also be used in [standalone](http://en.wikipedia.org/wiki/Standalone_software) [graphical applications](http://en.wikipedia.org/wiki/Graphical_user_interface).

USAGE

PHP is a general-purpose scripting language that is especially suited for [web development](http://en.wikipedia.org/wiki/Web_development). PHP generally runs on a [web server](http://en.wikipedia.org/wiki/Web_server). Any PHP code in a requested file is [executed](http://en.wikipedia.org/wiki/Execution_(computing)) by the PHP runtime, usually to create [dynamic web page](http://en.wikipedia.org/wiki/Dynamic_web_page) content. It can also be used for [command-line](http://en.wikipedia.org/wiki/Command-line) scripting and [client-side](http://en.wikipedia.org/wiki/Client-side) [GUI](http://en.wikipedia.org/wiki/Graphical_user_interface) applications. PHP can be deployed on most [web servers](http://en.wikipedia.org/wiki/Web_server), many [operating systems](http://en.wikipedia.org/wiki/Operating_system) and [platforms](http://en.wikipedia.org/wiki/Platform_(computing)), and can be used with many [relational database management systems](http://en.wikipedia.org/wiki/Relational_database_management_system). It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

PHP primarily acts as a [filter](http://en.wikipedia.org/wiki/Filter_(software)), taking input from a file or stream containing text and/or PHP instructions and outputs another stream of data; most commonly the output will be HTML. Since PHP 4, the PHP [parser](http://en.wikipedia.org/wiki/Parser) [compiles](http://en.wikipedia.org/wiki/Compiler) input to produce [byte code](http://en.wikipedia.org/wiki/Bytecode) for processing by the [Zend Engine](http://en.wikipedia.org/wiki/Zend_Engine), giving improved performance over its [interpreter](http://en.wikipedia.org/wiki/Interpreter_(computing)) predecessor

Originally designed to create dynamic web pages, PHP now focuses mainly on [server-side scripting](http://en.wikipedia.org/wiki/Server-side_scripting), and it is similar to other server-side scripting languages that provide dynamic content from a web server to a [client](http://en.wikipedia.org/wiki/Client_(computing)), such as [Microsoft](http://en.wikipedia.org/wiki/Microsoft)'s [Active Server Pages](http://en.wikipedia.org/wiki/Active_Server_Pages), [Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems)' [JavaServer Pages](http://en.wikipedia.org/wiki/JavaServer_Pages) and [mod\_perl](http://en.wikipedia.org/wiki/Mod_perl). PHP has also attracted the development of many [frameworks](http://en.wikipedia.org/wiki/Software_framework) that provide building blocks and a design structure to promote [rapid application development](http://en.wikipedia.org/wiki/Rapid_application_development) (RAD). Some of these include [CakePHP](http://en.wikipedia.org/wiki/CakePHP), [Symfony](http://en.wikipedia.org/wiki/Symfony), [CodeIgniter](http://en.wikipedia.org/wiki/CodeIgniter), and [Zend Framework](http://en.wikipedia.org/wiki/Zend_Framework), offering features similar to other [web application frameworks](http://en.wikipedia.org/wiki/List_of_web_application_frameworks).

About HTML

HTML, which stands for Hyper Text Markup Language, is the predominant [markup language](http://en.wikipedia.org/wiki/Markup_language) for [web pages](http://en.wikipedia.org/wiki/Web_page). It provides a means to create [structured documents](http://en.wikipedia.org/wiki/Structured_document) by denoting structural [semantics](http://en.wikipedia.org/wiki/Semantic) for text such as headings, paragraphs, lists etc as well as for links, quotes, and other items. It allows [images and objects](http://en.wikipedia.org/wiki/HTML_element#Images_and_objects) to be embedded and can be used to create [interactive forms](http://en.wikipedia.org/wiki/HTML_element#Forms). It is written in the form of [HTML elements](http://en.wikipedia.org/wiki/HTML_element) consisting of "tags" surrounded by [angle brackets](http://en.wikipedia.org/wiki/Brackets#Angle_brackets_or_chevrons_.3C_.3E) within the web page content. It can include or can load [scripts](http://en.wikipedia.org/wiki/Scripting_language) in languages such as [JavaScript](http://en.wikipedia.org/wiki/JavaScript) which affect the behavior of HTML processors like [Web browsers](http://en.wikipedia.org/wiki/Web_browser); and [Cascading Style Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) to define the appearance and layout of text and other material. The [W3C](http://en.wikipedia.org/wiki/W3C), maintainer of both HTML and CSS standards, encourages the use of CSS over explicit presentational markup.

Hyper Text Markup Language(HTML) is the encoding scheme used to create and format a web document. A user need not be an expert programmer to make use of HTML for creating hypertext documents that can be put on the internet.

Most graphical [e-mail](http://en.wikipedia.org/wiki/E-mail) clients allow the use of a subset of HTML (often ill-defined) to provide formatting and [semantic](http://en.wikipedia.org/wiki/Semantic_web) markup not available with [plain text](http://en.wikipedia.org/wiki/Plain_text). This may include typographic information like coloured headings, emphasized and quoted text, inline images and diagrams. Many such clients include both a [GUI](http://en.wikipedia.org/wiki/GUI) editor for composing HTML e-mail messages and a rendering engine for displaying them. Use of HTML in e-mail is controversial because of compatibility issues, because it can help disguise [phishing](http://en.wikipedia.org/wiki/Phishing) attacks, because it can confuse [spam](http://en.wikipedia.org/wiki/E-Mail_spam) filters and because the message size is larger than plain text.

NAMING CONVENTIONS

The most common [filename extension](http://en.wikipedia.org/wiki/Filename_extension) for [files](http://en.wikipedia.org/wiki/Computer_file) containing HTML is .html. A common abbreviation of this is .html, which originated because some early operating systems and file systems, such as [DOS](http://en.wikipedia.org/wiki/DOS) and [FAT](http://en.wikipedia.org/wiki/File_Allocation_Table), limited file extensions to [three letters](http://en.wikipedia.org/wiki/8.3_filename).

HTML APPLICATION

An HTML Application is a [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows) application that uses HTML and Dynamic HTML in a browser to provide the application's graphical interface. A regular HTML file is confined to the

security model of the web browser, communicating only to web servers and manipulating only webpage objects and [site cookies](http://en.wikipedia.org/wiki/HTTP_cookie). An HTA runs as a fully trusted application and therefore has more privileges, like creation/editing/removal of files and [Windows Registry](http://en.wikipedia.org/wiki/Windows_Registry) entries. Because they operate outside the browser's security model, HTAs cannot be executed via HTTP, but must be downloaded (just like an [EXE file](http://en.wikipedia.org/wiki/EXE)) and executed from local file system

ABOUT JAVASCRIPT

JavaScript is an [object-oriented](http://en.wikipedia.org/wiki/Object-oriented) [scripting language](http://en.wikipedia.org/wiki/Scripting_language) used to enable [programmatic](http://en.wikipedia.org/wiki/Computer_programming) access to objects within both the [client application](http://en.wikipedia.org/wiki/Client_(computing)) and other [applications](http://en.wikipedia.org/wiki/Application_software). It is primarily used in the form of [client-side JavaScript](http://en.wikipedia.org/wiki/Client-side_JavaScript), implemented as an integrated component of the [web browser](http://en.wikipedia.org/wiki/Web_browser), allowing the development of enhanced [user interfaces](http://en.wikipedia.org/wiki/User_interface) and dynamic [websites](http://en.wikipedia.org/wiki/Website). JavaScript is a [dialect](http://en.wikipedia.org/wiki/Programming_language_dialect) of the [ECMAScript](http://en.wikipedia.org/wiki/ECMAScript) standard and is characterized as a [dynamic](http://en.wikipedia.org/wiki/Dynamic_language), [weakly typed](http://en.wikipedia.org/wiki/Weak_typing), [prototype-based](http://en.wikipedia.org/wiki/Prototype-based_programming) language with [first-class functions](http://en.wikipedia.org/wiki/First-class_function). JavaScript was influenced by many languages and was designed to look like [Java](http://en.wikipedia.org/wiki/Java_(programming_language)), but to be easier for non-programmers to work with.

PROTOTYPE-BASED

JavaScript uses [prototypes](http://en.wikipedia.org/wiki/Prototype-based_programming) instead of [classes](http://en.wikipedia.org/wiki/Class_(computer_science)) for [inheritance](http://en.wikipedia.org/wiki/Inheritance_(computer_science)). It is possible to simulate many class-based features with prototypes in JavaScript.

Functions double as object constructors along with their typical role. Prefixing a function call with new creates a new object and calls that function with its local this keyword bound to that object for that invocation. The constructor's prototype property determines the object used for the new object's internal prototype. JavaScript's built-in constructors, such as Array, also have prototypes that can be modified.

Unlike many object-oriented languages, there is no distinction between a function definition and a [method](http://en.wikipedia.org/wiki/Method_(computer_science)) definition. Rather, the distinction occurs during function calling; a function can be called as a method. When a function is called as a method of an object, the function's local this keyword is bound to that object for that invocation.

USAGE

The primary use of JavaScript is to write functions that are embedded in or included from [HTML](http://en.wikipedia.org/wiki/HTML) pages and interact with the [Document Object Model](http://en.wikipedia.org/wiki/Document_Object_Model) (DOM) of the page.

Because JavaScript code can run locally in a user's browser (rather than on a remote server) it can respond to user actions quickly, making an application feel more responsive. Furthermore, JavaScript code can detect user actions which HTML alone cannot, such as individual keystrokes. Applications such as [Gmail](http://en.wikipedia.org/wiki/Gmail) take advantage of this: much of the user-interface logic is written in JavaScript, and JavaScript dispatches requests for information (such as the content of an e-mail message) to the server. The wider trend of [Ajax programming](http://en.wikipedia.org/wiki/AJAX) similarly exploits this strength.

A [JavaScript engine](http://en.wikipedia.org/wiki/JavaScript_engine) (also known as JavaScript interpreter or JavaScript implementation) is an [interpreter](http://en.wikipedia.org/wiki/Interpreter_(computing)) that interprets JavaScript [source code](http://en.wikipedia.org/wiki/Source_code) and executes the [script](http://en.wikipedia.org/wiki/Computer_program) accordingly. The first JavaScript engine was created by [Brendan Eich](http://en.wikipedia.org/wiki/Brendan_Eich) at Netscape Communications Corporation, for the [Netscape Navigator](http://en.wikipedia.org/wiki/Netscape_Navigator) [web browser](http://en.wikipedia.org/wiki/Web_browser). A web browser is by far the most common host environment for JavaScript. Web browsers typically use the public [API](http://en.wikipedia.org/wiki/Application_programming_interface) to create "host objects" responsible for reflecting the [DOM](http://en.wikipedia.org/wiki/Document_Object_Model) into JavaScript.

ABOUT MySQL

## MySQL Introduction

There are a large number of database management systems currently available, some commercial and some free.Some of them : Oracle, Microsoft Access, Mysql and PostgreSQL.  
These database systems are powerful, feature-rich software, capable of organizing and searching millions of records at very high speeds.

Understanding Databases, Records, and Primary KeY Every Database is composed of one or more tables.  
These Tables, which structure data into rows and columns, Impose organization on the data The records in a table(below) are not arranged in any particular order.To make it easy to identify a specific record,therefore, it becomes necessarystanding Relationships and Foreign Keys(RDBMS)You already know that a single database can hold multiple tables.In a Relational database management system(RDBMS), these tables can be linked to each other by one or more common fields, called foreign keys.

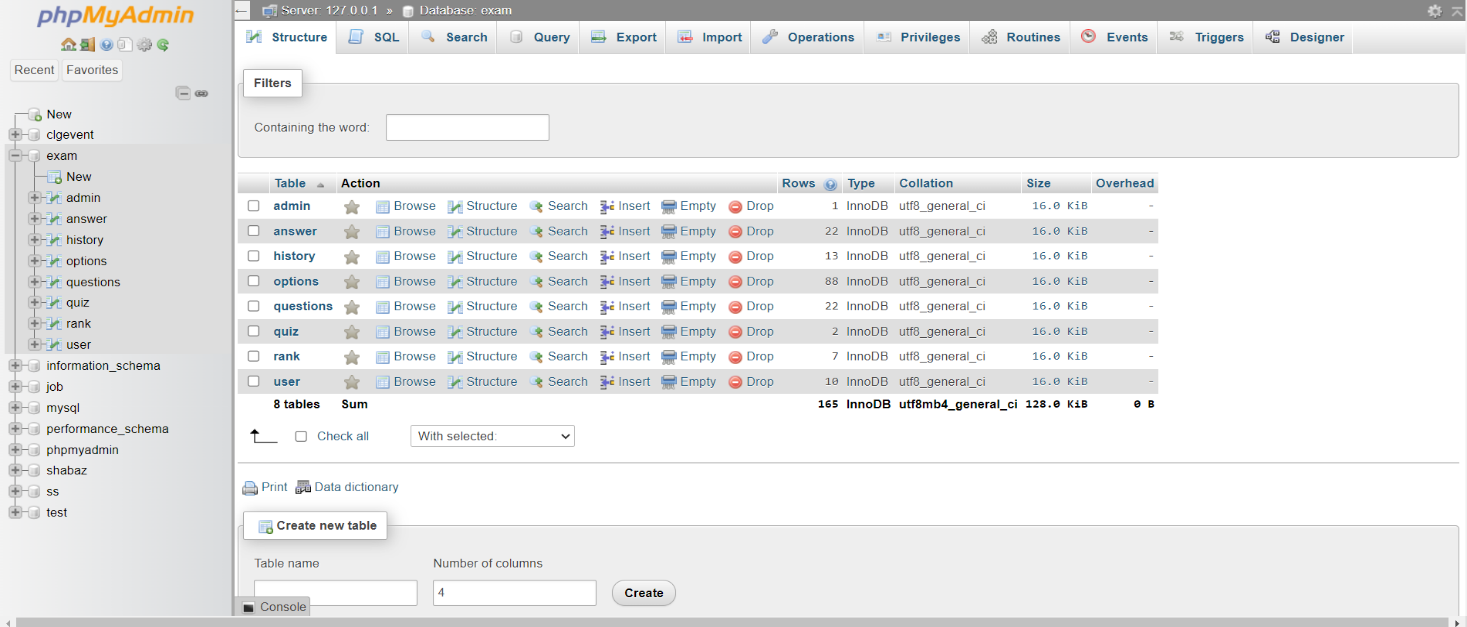
What is Database administrator(DBA) ?

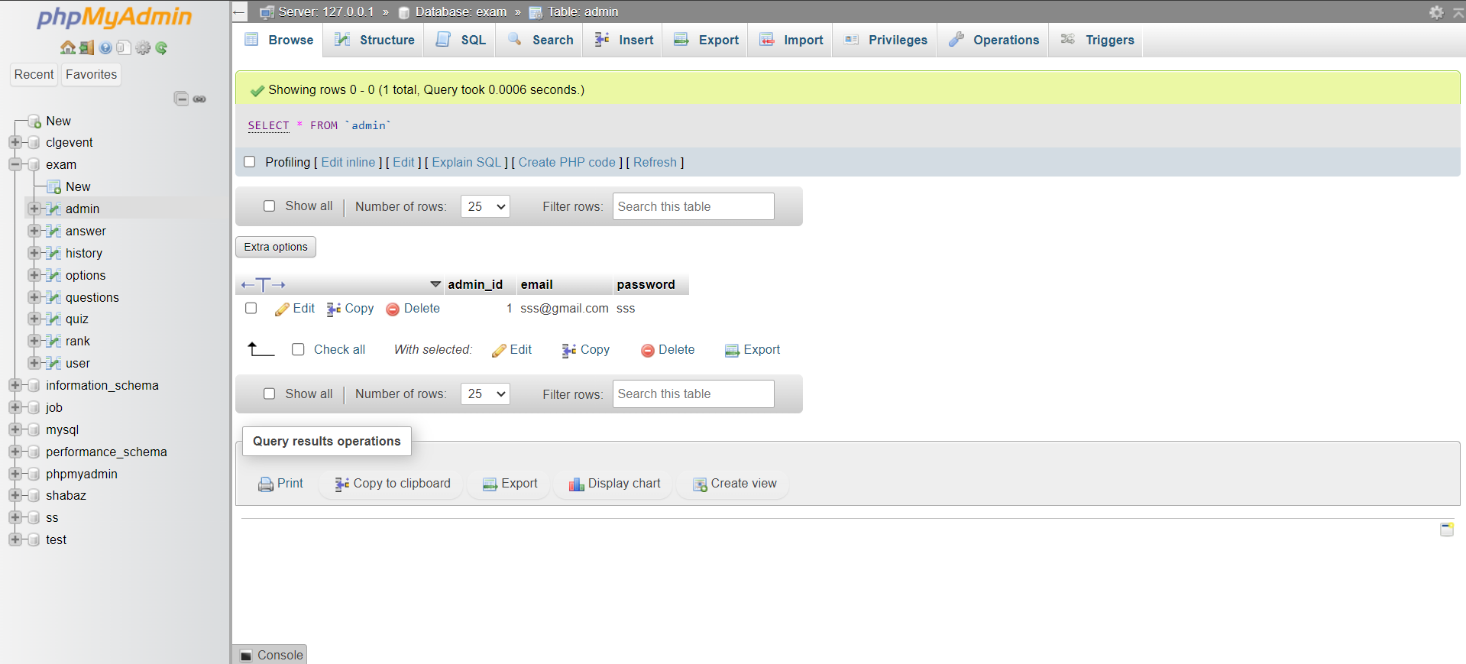
Database administrator is the super user of database, he has unrestricted rights and privileges to access database, grant permission to other database users.

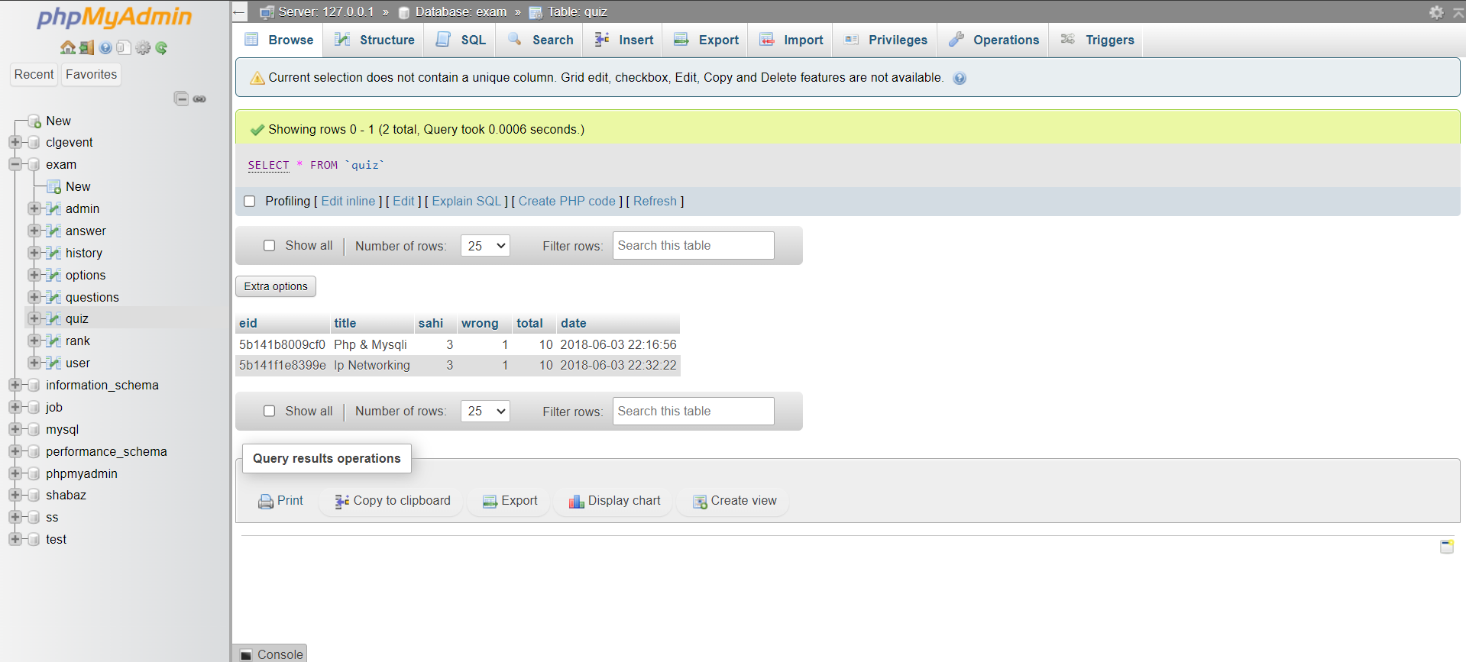
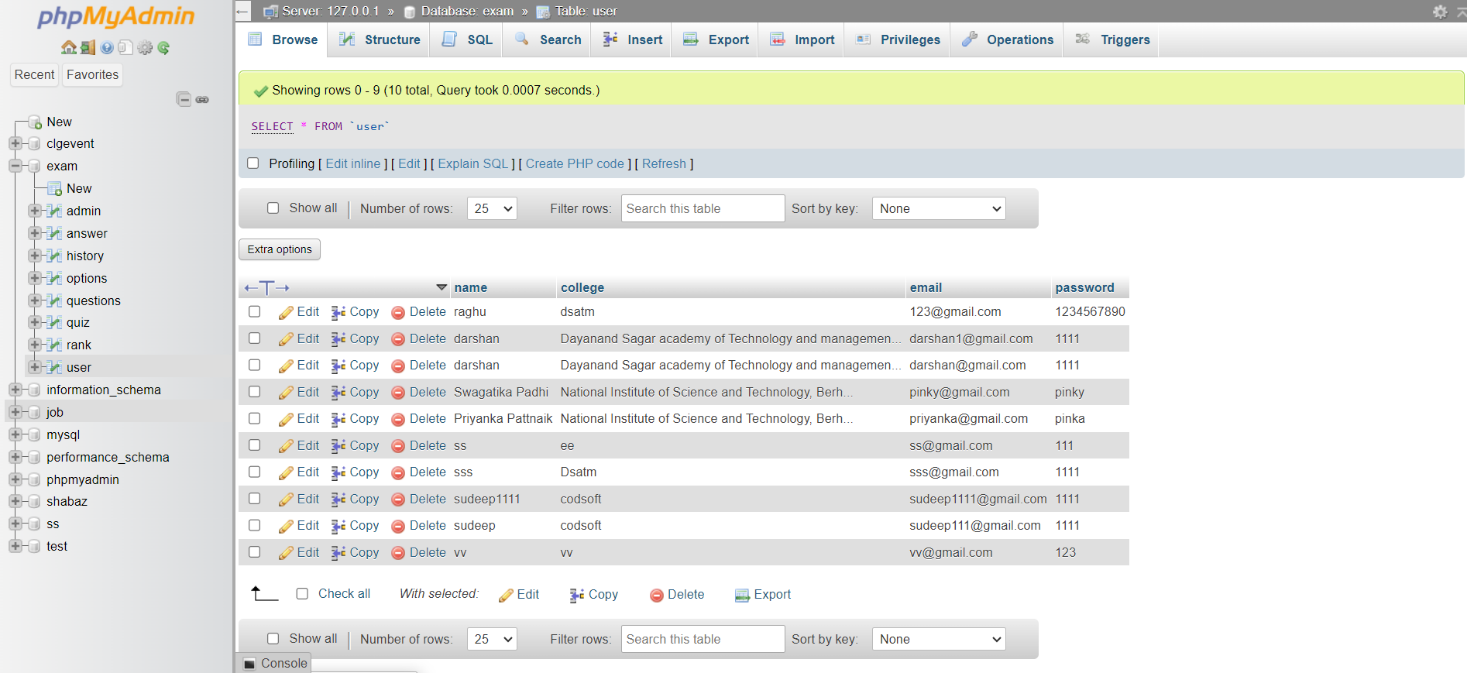
What is Database user(DBU) ?

Database user is the person who uses the database in a restricted privileges, provided by database administrator.

Download MySQL DatabaseIf you have installed PHP’s WAMP or XAMPP server, then mysql database already exists. if you don’t have then download mysql database from here[http://www.mysql.com](http://www.phptpoint.com/mysql/)

5.2 DATABASE TABLES:





5.3 TESTING

Testing is a process of executing a program with the indent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding.System Testing is an important phase. Testing represents an interesting anomaly for the software. Thus a series of testing are performed for the proposed system before the system is ready for user acceptance testing.A good test case is one that has a high probability of finding an as undiscovered error. A successful test is one that uncovers an as undiscovered error.

Testing Objectives:

1.Testing is a process of executing a program with the intent of finding an error

2.A good test case is one that has a probability of finding an as yet undiscovered error

3.A successful test is one that uncovers an undiscovered error

Testing Principles

1. All tests should be traceable to end user requirementS
2. Tests should be planned long before testing begins
3. Testing should begin on a small scale and progress towards testing in large
4. Exhaustive testing is not possible
5. To be most effective testing should be conducted by a independent third party

The primary objective for test case design is to derive a set of tests that has the highest livelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used. They are

White box testing.

Black box testing.

# White-box testing:

White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

# Block-box testing:

Black box testing is designed to validate functional requirements without regard to the internal workings of a program. Black box testing mainly focuses on the information domain of the software, deriving test cases by partitioning input and output in a manner that provides through test coverage. Incorrect and missing functions, interface errors, errors in data structures, error in functional logic are the errors falling in this category.

Testing strategies:

A strategy for software testing must accommodate low-level tests that are necessary to verify that all small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements.

Testing fundamentals:

Testing is a process of executing program with the intent of finding error. A good test case is one that has high probability of finding an undiscovered error. If testing is conducted successfully it uncovers the errors in the software. Testing cannot show the absence of defects, it can only show that software defects present.

Testing Information flow:Information flow for testing flows the pattern. Two class of input provided to test the process. The software configuration includes a software requirements specification, a design specification and source code.

Test configuration includes test plan and test cases and test tools. Tests are conducted and all the results are evaluated. That is test results are compared with expected results. When erroneous data are uncovered, an error is implied and debugging commences.

Unit testing:

Unit testing is essential for the verification of the code produced during the coding phase and hence the goal is to test the internal logic of the modules. Using the detailed design description as a guide, important paths are tested to uncover errors with in the boundary of the modules. These tests were carried out during the programming stage itself. All units of ViennaSQL were successfully tested.

Integration testing :

Integration testing focuses on unit tested modules and build the program structure that is dictated by the design phase.

System testing:

System testing tests the integration of each module in the system. It also tests to find discrepancies between the system and it’s original objective, current specification and system documentation. The primary concern is the compatibility of individual modules. Entire system is working properly or not will be tested here, and specified path ODBC connection will correct or not, and giving output or not are tested here these verifications and validations are done by giving input values to the system and by comparing with expected output. Top-down testing implementing here.

5.3.1Test Cases:

Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

Using White-Box testing methods, the software engineer can drive test cases that

* Guarantee that logical decisions on their true and false sides.
* Exercise all logical decisions on their true and false sides.
* Execute all loops at their boundaries and with in their operational bounds.

SOURCE CODE

**Admin Page**

<?php

include\_once 'database.php';

session\_start();

if(isset($\_SESSION["email"]))

{

session\_destroy();

}

$ref=@$\_GET['q'];

if(isset($\_POST['submit']))

{

$email = $\_POST['email'];

$password = $\_POST['password'];

$email = stripslashes($email);

$email = addslashes($email);

$password = stripslashes($password);

$password = addslashes($password);

$email = mysqli\_real\_escape\_string($con,$email);

$password = mysqli\_real\_escape\_string($con,$password);

$result = mysqli\_query($con,"SELECT email FROM admin WHERE email = '$email' and password = '$password'") or die('Error');

$count=mysqli\_num\_rows($result);

if($count==1)

{

session\_start();

if(isset($\_SESSION['email']))

{

session\_unset();

}

$\_SESSION["name"] = 'Admin';

$\_SESSION["key"] ='admin';

$\_SESSION["email"] = $email;

header("location:dashboard.php?q=0");

}

else

{

echo "<center><h3><script>alert('Sorry.. Wrong Username (or) Password');</script></h3></center>";

header("refresh:0;url=admin.php");

}

}

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Admin Login | Online Quiz System</title>

<link rel="stylesheet" href="scripts/bootstrap/bootstrap.min.css">

<link rel="stylesheet" href="scripts/ionicons/css/ionicons.min.css">

<link rel="stylesheet" href="css/form.css">

<style type="text/css">

body{

width: 100%;

background: url(image/book.png) ;

background-position: center center;

background-repeat: no-repeat;

background-attachment: fixed;

background-size: cover;

}

</style>

</head>

<body>

<section class="login first grey">

<div class="container">

<div class="box-wrapper">

<div class="box box-border">

<div class="box-body">

<center> <h5 style="font-family: Noto Sans;">Login to </h5><h4 style="font-family: Noto Sans;">Admin Page</h4></center><br><form method="post" action="admin.php" enctype="multipart/form-data">

<div class="form-group">

<label>Enter Your Email Id:</label>

<input type="email" name="email" class="form-control">

</div><div class="form-group">

<label class="fw">Enter Your Password:

<a href="javascript:void(0)" class="pull-right">Forgot Password?</a>

</label>

<input type="password" name="password" class="form-control">

</div>

<div class="form-group text-right">

<button class="btn btn-primary btn-block" name="submit">Login</button>

</div>

</form>

</div>

</div>

</div>

</div>

</section>

<script src="js/jquery.js"></script>

<script src="scripts/bootstrap/bootstrap.min.js"></script>

</body>

</html>

**Dashboard**

<?php

include\_once 'database.php';

session\_start();

if(!(isset($\_SESSION['email'])))

{

header("location:login.php");

}

else

{

$name = $\_SESSION['name'];

$email = $\_SESSION['email'];

include\_once 'database.php';

}

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Dashboard | Online Quiz System</title>

<link rel="stylesheet" href="css/bootstrap.min.css"/>

<link rel="stylesheet" href="css/bootstrap-theme.min.css"/>

<link rel="stylesheet" href="css/welcome.css">

<link rel="stylesheet" href="css/font.css">

<script src="js/jquery.js" type="text/javascript"></script>

<script src="js/bootstrap.min.js" type="text/javascript"></script>

</head>

<body>

<nav class="navbar navbar-default title1">

<div class="container-fluid">

<div class="navbar-header">

<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a class="navbar-brand" href="Javascript:void(0)"><b>Online Quiz System</b></a>

</div>

<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

<ul class="nav navbar-nav navbar-left">

<li <?php if(@$\_GET['q']==0) echo'class="active"'; ?>><a href="dashboard.php?q=0">Home<span class="sr-only">(current)</span></a></li>

<li <?php if(@$\_GET['q']==1) echo'class="active"'; ?>><a href="dashboard.php?q=1">User</a></li>

<li <?php if(@$\_GET['q']==2) echo'class="active"'; ?>><a href="dashboard.php?q=2">Ranking</a></li>

<li class="dropdown <?php if(@$\_GET['q']==4 || @$\_GET['q']==5) echo'active"'; ?>">

<li><a href="dashboard.php?q=4">Add Quiz</a></li>

<li><a href="dashboard.php?q=5">Remove Quiz</a></li>

</ul>

<ul class="nav navbar-nav navbar-right">

<li <?php echo''; ?> > <a href="logout1.php?q=dashboard.php"><span class="glyphicon glyphicon-log-out" aria-hidden="true"></span>&nbsp;Log out</a></li>

</ul>

</div>

</div>

</nav>

<div class="container">

<div class="row">

<div class="col-md-12">

<?php if(@$\_GET['q']==0)

{

echo "<h1> WELCOME TO Admin Page!!

</h1>";

}

if(@$\_GET['q']== 2)

{

$q=mysqli\_query($con,"SELECT \* FROM rank ORDER BY score DESC " )or die('Error223');

echo '<div class="panel title"><div class="table-responsive">

<table class="table table-striped title1" >

<tr style="color:red"><td><center><b>Rank</b></center></td><td><center><b>Name</b></center></td><td><center><b>Score</b></center></td></tr>';

$c=0;

while($row=mysqli\_fetch\_array($q) )

{

$e=$row['email'];

$s=$row['score'];

$q12=mysqli\_query($con,"SELECT \* FROM user WHERE email='$e' " )or die('Error231');

while($row=mysqli\_fetch\_array($q12) )

{

$name=$row['name'];

$college=$row['college'];

}

$c++;

echo '<tr><td style="color:#99cc32"><center><b>'.$c.'</b></center></td><td><center>'.$e.'</center></td><td><center>'.$s.'</center></td>';

}

echo '</table></div></div>';

}?>

<?php

if(@$\_GET['q']==1)

{

$result = mysqli\_query($con,"SELECT \* FROM user") or die('Error');

echo '<div class="panel"><div class="table-responsive"><table class="table table-striped title1">

<tr><td><center><b>S.N.</b></center></td><td><center><b>Name</b></center></td><td><center><b>College</b></center></td><td><center><b>Email</b></center></td><td><center><b>Action</b></center></td></tr>';

$c=1;

while($row = mysqli\_fetch\_array($result))

{

$name = $row['name'];

$email = $row['email'];

$college = $row['college'];

echo '<tr><td><center>'.$c++.'</center></td><td><center>'.$name.'</center></td><td><center>'.$college.'</center></td><td><center>'.$email.'</center></td><td><center><a title="Delete User" href="update.php?demail='.$email.'"><b><span class="glyphicon glyphicon-trash" aria-hidden="true"></span></b></a></center></td></tr>';

}

$c=0;

echo '</table></div></div>';

}

?>

<?php

if(@$\_GET['q']==4 && !(@$\_GET['step']) )

{

echo '<div class="row"><span class="title1" style="margin-left:40%;font-size:30px;color:#fff;"><b>Enter Quiz Details</b></span><br /><br />

<div class="col-md-3"></div><div class="col-md-6">

<form class="form-horizontal title1" name="form" action="update.php?q=addquiz" method="POST">

<fieldset>

<div class="form-group">

<label class="col-md-12 control-label" for="name"></label>

<div class="col-md-12">

<input id="name" name="name" placeholder="Enter Quiz title" class="form-control input-md" type="text">

</div>

</div>

<div class="form-group">

<label class="col-md-12 control-label" for="total"></label>

<div class="col-md-12">

<input id="total" name="total" placeholder="Enter total number of questions" class="form-control input-md" type="number">

</div>

</div>

<div class="form-group">

<label class="col-md-12 control-label" for="right"></label>

<div class="col-md-12">

<input id="right" name="right" placeholder="Enter marks on right answer" class="form-control input-md" min="0" type="number">

</div>

</div>

<div class="form-group">

<label class="col-md-12 control-label" for="wrong"></label>

<div class="col-md-12">

<input id="wrong" name="wrong" placeholder="Enter minus marks on wrong answer without sign" class="form-control input-md" min="0" type="number">

</div>

</div>

<div class="form-group">

<label class="col-md-12 control-label" for=""></label>

<div class="col-md-12">

<input type="submit" style="margin-left:45%" class="btn btn-primary" value="Submit" class="btn btn-primary"/>

</div>

</div>

</fieldset>

</form></div>';

}

?>

<?php

if(@$\_GET['q']==4 && (@$\_GET['step'])==2 )

{

echo '

<div class="row">

<span class="title1" style="margin-left:40%;font-size:30px;"><b>Enter Question Details</b></span><br /><br />

<div class="col-md-3"></div><div class="col-md-6"><form class="form-horizontal title1" name="form" action="update.php?q=addqns&n='.@$\_GET['n'].'&eid='.@$\_GET['eid'].'&ch=4 " method="POST">

<fieldset>';

for($i=1;$i<=@$\_GET['n'];$i++)

{

echo '<b>Question number&nbsp;'.$i.'&nbsp;:</><br /><!-- Text input-->

<div class="form-group">

<label class="col-md-12 control-label" for="qns'.$i.' "></label>

<div class="col-md-12">

<textarea rows="3" cols="5" name="qns'.$i.'" class="form-control" placeholder="Write question number '.$i.' here..."></textarea>

</div>

</div>

<div class="form-group">

<label class="col-md-12 control-label" for="'.$i.'1"></label>

<div class="col-md-12">

<input id="'.$i.'1" name="'.$i.'1" placeholder="Enter option a" class="form-control input-md" type="text">

</div>

</div>

<div class="form-group">

<label class="col-md-12 control-label" for="'.$i.'2"></label>

<div class="col-md-12">

<input id="'.$i.'2" name="'.$i.'2" placeholder="Enter option b" class="form-control input-md" type="text">

</div>

</div>

<div class="form-group">

<label class="col-md-12 control-label" for="'.$i.'3"></label>

<div class="col-md-12">

<input id="'.$i.'3" name="'.$i.'3" placeholder="Enter option c" class="form-control input-md" type="text">

</div>

</div>

<div class="form-group">

<label class="col-md-12 control-label" for="'.$i.'4"></label>

<div class="col-md-12">

<input id="'.$i.'4" name="'.$i.'4" placeholder="Enter option d" class="form-control input-md" type="text">

</div>

</div>

<br />

<b>Correct answer</b>:<br />

<select id="ans'.$i.'" name="ans'.$i.'" placeholder="Choose correct answer " class="form-control input-md" >

<option value="a">Select answer for question '.$i.'</option>

<option value="a"> option a</option>

<option value="b"> option b</option>

<option value="c"> option c</option>

<option value="d"> option d</option> </select><br /><br />';

}

echo '<div class="form-group">

<label class="col-md-12 control-label" for=""></label>

<div class="col-md-12">

<input type="submit" style="margin-left:45%" class="btn btn-primary" value="Submit" class="btn btn-primary"/>

</div>

</div>

</fieldset>

</form></div>';

}

?>

<?php

if(@$\_GET['q']==5)

{

$result = mysqli\_query($con,"SELECT \* FROM quiz ORDER BY date DESC") or die('Error');

echo '<div class="panel"><div class="table-responsive"><table class="table table-striped title1">

<tr><td><center><b>S.N.</b></center></td><td><center><b>Topic</b></center></td><td><center><b>Total question</b></center></td><td><center><b>Marks</b></center></td><td><center><b>Action</b></center></td></tr>';

$c=1;

while($row = mysqli\_fetch\_array($result)) {

$title = $row['title'];

$total = $row['total'];

$sahi = $row['sahi'];

$eid = $row['eid'];

echo '<tr><td><center>'.$c++.'</center></td><td><center>'.$title.'</center></td><td><center>'.$total.'</center></td><td><center>'.$sahi\*$total.'</center></td>

<td><center><b><a href="update.php?q=rmquiz&eid='.$eid.'" class="pull-right btn sub1" style="margin:0px;background:red;color:black"><span class="glyphicon glyphicon-trash" aria-hidden="true"></span>&nbsp;<span class="title1"><b>Remove</b></span></a></b></center></td></tr>';

}

$c=0;

echo '</table></div></div>';

}

?>

</div>

</div>

</div>

</body>

</html>

**Index Page**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>| Online Quiz System |</title>

<link rel="stylesheet" type="text/css" href="css/index.css" />

<link rel="shortcut icon" type="image/png" href="image/logo.png" />

<style type="text/css">

body {

width: 100%;

background: url(image/book.png) ;

background-position: center center;

background-repeat: no-repeat;

background-attachment: fixed;

background-size: cover;

}

</style>

</head>

<body>

<center>

<div class="intro">

<h1> online quiz system </h1>

<a href="login.php" class="btn"> login </a> &emsp;

<a href="register.php" class="btn"> register </a>

<h2> Good &nbsp;Luck. </h2>

</div>

</center>

</body>

</html>

**Login Page**

<?php

require('database.php');

session\_start();

if(isset($\_SESSION["email"]))

{

session\_destroy();

}

$ref=@$\_GET['q'];

if(isset($\_POST['submit']))

{

$email = $\_POST['email'];

$pass = $\_POST['password'];

$email = stripslashes($email);

$email = addslashes($email);

$pass = stripslashes($pass);

$pass = addslashes($pass);

$email = mysqli\_real\_escape\_string($con,$email);

$pass = mysqli\_real\_escape\_string($con,$pass);

$str = "SELECT \* FROM user WHERE email='$email' and password='$pass'";

$result = mysqli\_query($con,$str);

if((mysqli\_num\_rows($result))!=1)

{

echo "<center><h3><script>alert('Sorry.. Wrong Username (or) Password');</script></h3></center>";

header("refresh:0;url=login.php");

}

else

{

$\_SESSION['logged']=$email;

$row=mysqli\_fetch\_array($result);

$\_SESSION['name']=$row[1];

$\_SESSION['id']=$row[0];

$\_SESSION['email']=$row[2];

$\_SESSION['password']=$row[3];

header('location: welcome.php?q=1');

}

}

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Login | Online Quiz System</title>

<link rel="stylesheet" href="scripts/bootstrap/bootstrap.min.css">

<link rel="stylesheet" href="scripts/ionicons/css/ionicons.min.css">

<link rel="stylesheet" href="css/form.css">

<style type="text/css">

body{

width: 100%;

background: url(image/book.png) ;

background-position: center center;

background-repeat: no-repeat;

background-attachment: fixed;

background-size: cover;

}

</style>

</head>

<body>

<section class="login first grey">

<div class="container">

<div class="box-wrapper">

<div class="box box-border">

<div class="box-body">

<center> <h5 style="font-family: Noto Sans;">Login to </h5><h4 style="font-family: Noto Sans;">Online Quiz System</h4></center><br>

<form method="post" action="login.php" enctype="multipart/form-data">

<div class="form-group">

<label>Enter Your Email Id:</label>

<input type="email" name="email" class="form-control">

</div>

<div class="form-group">

<label class="fw">Enter Your Password:

<a href="javascript:void(0)" class="pull-right">Forgot Password?</a>

</label>

<input type="password" name="password" class="form-control">

</div>

<div class="form-group text-right">

<button class="btn btn-primary btn-block" name="submit">Login</button>

</div>

<div class="form-group text-center">

<span class="text-muted">Don't have an account?</span> <a href="register.php">Register</a> Here..

</div>

</form>

</div>

</div>

</div>

</div>

</section>

<script src="js/jquery.js"></script>

<script src="scripts/bootstrap/bootstrap.min.js"></script>

</body>

</html>

**Logout**

<?php

session\_start();

if(isset($\_SESSION['email'])){

session\_destroy();}

$ref= @$\_GET['q'];

header("location:$ref");

?>

**Logout 1**

<?php

session\_start();

if(isset($\_SESSION['email'])){

session\_destroy();}

$ref= @$\_GET['q'];

header("location:admin.php");

?>

**Register**

<?php

include("database.php");

session\_start();

if(isset($\_POST['submit']))

{

$name = $\_POST['name'];

$name = stripslashes($name);

$name = addslashes($name);

$email = $\_POST['email'];

$email = stripslashes($email);

$email = addslashes($email);

$password = $\_POST['password'];

$password = stripslashes($password);

$password = addslashes($password);

$college = $\_POST['college'];

$college = stripslashes($college);

$college = addslashes($college);

$str="SELECT email from user WHERE email='$email'";

$result=mysqli\_query($con,$str);

if((mysqli\_num\_rows($result))>0)

{

echo "<center><h3><script>alert('Sorry.. This email is already registered !!');</script></h3></center>";

header("refresh:0;url=login.php");

}

else

{

$str="insert into user set name='$name',email='$email',password='$password',college='$college'";

if((mysqli\_query($con,$str)))

echo "<center><h3><script>alert('Congrats.. You have successfully registered !!');</script></h3></center>";

header('location: welcome.php?q=1');

}

}

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Register | Online Quiz System</title>

<link rel="stylesheet" href="scripts/bootstrap/bootstrap.min.css">

<link rel="stylesheet" href="scripts/ionicons/css/ionicons.min.css">

<link rel="stylesheet" href="css/form.css">

<style type="text/css">

body{

width: 100%;

background: url(image/book.png) ;

background-position: center center;

background-repeat: no-repeat;

background-attachment: fixed;

background-size: cover;

}

</style>

</head>

<body>

<section class="login first grey">

<div class="container">

<div class="box-wrapper">

<div class="box box-border">

<div class="box-body">

<center> <h5 style="font-family: Noto Sans;">Register to </h5><h4 style="font-family: Noto Sans;">Online Quiz System</h4></center><br>

<form method="post" action="register.php" enctype="multipart/form-data">

<div class="form-group">

<label>Enter Your Username:</label>

<input type="text" name="name" class="form-control" required />

</div>

<div class="form-group">

<label>Enter Your Email Id:</label>

<input type="email" name="email" class="form-control" required />

</div>

<div class="form-group">

<label>Enter Your Password:</label>

<input type="password" name="password" class="form-control" required />

</div>

<div class="form-group">

<label>Enter Your College Name:</label>

<input type="text" name="college" class="form-control" required />

</div>

<div class="form-group text-right">

<button class="btn btn-primary btn-block" name="submit">Register</button>

</div>

<div class="form-group text-center">

<span class="text-muted">Already have an account! </span> <a href="login.php">Login </a> Here..

</div>

</form>

</div>

</div>

</div>

</div>

</section>

<script src="js/jquery.js"></script>

<script src="scripts/bootstrap/bootstrap.min.js"></script>

</body>

</html>

**Update**

<?php

include\_once 'database.php';

session\_start();

$email=$\_SESSION['email'];

if(isset($\_SESSION['key']))

{

if(@$\_GET['demail'] && $\_SESSION['key']=='suryapinky')

{

$demail=@$\_GET['demail'];

$r1 = mysqli\_query($con,"DELETE FROM rank WHERE email='$demail' ") or die('Error');

$r2 = mysqli\_query($con,"DELETE FROM history WHERE email='$demail' ") or die('Error');

$result = mysqli\_query($con,"DELETE FROM user WHERE email='$demail' ") or die('Error');

header("location:dashboard.php?q=1");

}

}

if(isset($\_SESSION['key']))

{

if(@$\_GET['q']== 'rmquiz' && $\_SESSION['key']=='suryapinky')

{

$eid=@$\_GET['eid'];

$result = mysqli\_query($con,"SELECT \* FROM questions WHERE eid='$eid' ") or die('Error');

while($row = mysqli\_fetch\_array($result))

{

$qid = $row['qid'];

$r1 = mysqli\_query($con,"DELETE FROM options WHERE qid='$qid'") or die('Error');

$r2 = mysqli\_query($con,"DELETE FROM answer WHERE qid='$qid' ") or die('Error');

}

$r3 = mysqli\_query($con,"DELETE FROM questions WHERE eid='$eid' ") or die('Error');

$r4 = mysqli\_query($con,"DELETE FROM quiz WHERE eid='$eid' ") or die('Error');

$r4 = mysqli\_query($con,"DELETE FROM history WHERE eid='$eid' ") or die('Error');

header("location:dashboard.php?q=5");

}

}

if(isset($\_SESSION['key']))

{

if(@$\_GET['q']== 'addquiz' && $\_SESSION['key']=='suryapinky')

{

$name = $\_POST['name'];

$name= ucwords(strtolower($name));

$total = $\_POST['total'];

$sahi = $\_POST['right'];

$wrong = $\_POST['wrong'];

$id=uniqid();

$q3=mysqli\_query($con,"INSERT INTO quiz VALUES ('$id','$name' , '$sahi' , '$wrong','$total', NOW())");

header("location:dashboard.php?q=4&step=2&eid=$id&n=$total");

}

}

if(isset($\_SESSION['key']))

{

if(@$\_GET['q']== 'addqns' && $\_SESSION['key']=='suryapinky')

{

$n=@$\_GET['n'];

$eid=@$\_GET['eid'];

$ch=@$\_GET['ch'];

for($i=1;$i<=$n;$i++)

{

$qid=uniqid();

$qns=$\_POST['qns'.$i];

$q3=mysqli\_query($con,"INSERT INTO questions VALUES ('$eid','$qid','$qns' , '$ch' , '$i')");

$oaid=uniqid();

$obid=uniqid();

$ocid=uniqid();

$odid=uniqid();

$a=$\_POST[$i.'1'];

$b=$\_POST[$i.'2'];

$c=$\_POST[$i.'3'];

$d=$\_POST[$i.'4'];

$qa=mysqli\_query($con,"INSERT INTO options VALUES ('$qid','$a','$oaid')") or die('Error61');

$qb=mysqli\_query($con,"INSERT INTO options VALUES ('$qid','$b','$obid')") or die('Error62');

$qc=mysqli\_query($con,"INSERT INTO options VALUES ('$qid','$c','$ocid')") or die('Error63');

$qd=mysqli\_query($con,"INSERT INTO options VALUES ('$qid','$d','$odid')") or die('Error64');

$e=$\_POST['ans'.$i];

switch($e)

{

case 'a': $ansid=$oaid; break;

case 'b': $ansid=$obid; break;

case 'c': $ansid=$ocid; break;

case 'd': $ansid=$odid; break;

default: $ansid=$oaid;

}

$qans=mysqli\_query($con,"INSERT INTO answer VALUES ('$qid','$ansid')");

}

header("location:dashboard.php?q=0");

}

}

if(@$\_GET['q']== 'quiz' && @$\_GET['step']== 2)

{

$eid=@$\_GET['eid'];

$sn=@$\_GET['n'];

$total=@$\_GET['t'];

$ans=$\_POST['ans'];

$qid=@$\_GET['qid'];

$q=mysqli\_query($con,"SELECT \* FROM answer WHERE qid='$qid' " );

while($row=mysqli\_fetch\_array($q) )

{ $ansid=$row['ansid']; }

if($ans == $ansid)

{

$q=mysqli\_query($con,"SELECT \* FROM quiz WHERE eid='$eid' " );

while($row=mysqli\_fetch\_array($q) )

{

$sahi=$row['sahi'];

}

if($sn == 1)

{

$q=mysqli\_query($con,"INSERT INTO history VALUES('$email','$eid' ,'0','0','0','0',NOW())")or die('Error');

}

$q=mysqli\_query($con,"SELECT \* FROM history WHERE eid='$eid' AND email='$email' ")or die('Error115');

while($row=mysqli\_fetch\_array($q) )

{

$s=$row['score'];

$r=$row['sahi'];

}

$r++;

$s=$s+$sahi;

$q=mysqli\_query($con,"UPDATE `history` SET `score`=$s,`level`=$sn,`sahi`=$r, date= NOW() WHERE email = '$email' AND eid = '$eid'")or die('Error124');

}

else

{

$q=mysqli\_query($con,"SELECT \* FROM quiz WHERE eid='$eid' " )or die('Error129');

while($row=mysqli\_fetch\_array($q) )

{

$wrong=$row['wrong'];

}

if($sn == 1)

{

$q=mysqli\_query($con,"INSERT INTO history VALUES('$email','$eid' ,'0','0','0','0',NOW() )")or die('Error137');

}

$q=mysqli\_query($con,"SELECT \* FROM history WHERE eid='$eid' AND email='$email' " )or die('Error139');

while($row=mysqli\_fetch\_array($q) )

{

$s=$row['score'];

$w=$row['wrong'];

}

$w++;

$s=$s-$wrong;

$q=mysqli\_query($con,"UPDATE `history` SET `score`=$s,`level`=$sn,`wrong`=$w, date=NOW() WHERE email = '$email' AND eid = '$eid'")or die('Error147');

}

if($sn != $total)

{

$sn++;

header("location:welcome.php?q=quiz&step=2&eid=$eid&n=$sn&t=$total")or die('Error152');

}

else if( $\_SESSION['key']!='suryapinky')

{

$q=mysqli\_query($con,"SELECT score FROM history WHERE eid='$eid' AND email='$email'" )or die('Error156');

while($row=mysqli\_fetch\_array($q) )

{

$s=$row['score'];

}

$q=mysqli\_query($con,"SELECT \* FROM rank WHERE email='$email'" )or die('Error161');

$rowcount=mysqli\_num\_rows($q);

if($rowcount == 0)

{

$q2=mysqli\_query($con,"INSERT INTO rank VALUES('$email','$s',NOW())")or die('Error165');

}

else

{

while($row=mysqli\_fetch\_array($q) )

{

$sun=$row['score'];

}

$sun=$s+$sun;

$q=mysqli\_query($con,"UPDATE `rank` SET `score`=$sun ,time=NOW() WHERE email= '$email'")or die('Error174');

}

header("location:welcome.php?q=result&eid=$eid");

}

else

{

header("location:welcome.php?q=result&eid=$eid");

}

}

if(@$\_GET['q']== 'quizre' && @$\_GET['step']== 25 )

{

$eid=@$\_GET['eid'];

$n=@$\_GET['n'];

$t=@$\_GET['t'];

$q=mysqli\_query($con,"SELECT score FROM history WHERE eid='$eid' AND email='$email'" )or die('Error156');

while($row=mysqli\_fetch\_array($q) )

{

$s=$row['score'];

}

$q=mysqli\_query($con,"DELETE FROM `history` WHERE eid='$eid' AND email='$email' " )or die('Error184');

$q=mysqli\_query($con,"SELECT \* FROM rank WHERE email='$email'" )or die('Error161');

while($row=mysqli\_fetch\_array($q) )

{

$sun=$row['score'];

}

$sun=$sun-$s;

$q=mysqli\_query($con,"UPDATE `rank` SET `score`=$sun ,time=NOW() WHERE email= '$email'")or die('Error174');

header("location:welcome.php?q=quiz&step=2&eid=$eid&n=1&t=$t");

}

?>

**Welcome**

<?php

include\_once 'database.php';

session\_start();

if(!(isset($\_SESSION['email'])))

{

header("location:login.php");

}

else

{

$name = $\_SESSION['name'];

$email = $\_SESSION['email'];

include\_once 'database.php';

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Welcome | Online Quiz System</title>

<link rel="stylesheet" href="css/bootstrap.min.css"/>

<link rel="stylesheet" href="css/bootstrap-theme.min.css"/>

<link rel="stylesheet" href="css/welcome.css">

<link rel="stylesheet" href="css/font.css">

<script src="js/jquery.js" type="text/javascript"></script>

<script src="js/bootstrap.min.js" type="text/javascript"></script>

</head>

<body>

<nav class="navbar navbar-default title1">

<div class="container-fluid">

<div class="navbar-header">

<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a class="navbar-brand" href="#"><b>Online Quiz System</b></a>

</div>

<!-- Collect the nav links, forms, and other content for toggling -->

<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

<ul class="nav navbar-nav navbar-left">

<li <?php if(@$\_GET['q']==1) echo'class="active"'; ?> ><a href="welcome.php?q=1"><span class="glyphicon glyphicon-home" aria-hidden="true"></span>&nbsp;Home<span class="sr-only">(current)</span></a></li>

<li <?php if(@$\_GET['q']==2) echo'class="active"'; ?>> <a href="welcome.php?q=2"><span class="glyphicon glyphicon-list-alt" aria-hidden="true"></span>&nbsp;History</a></li>

<li <?php if(@$\_GET['q']==3) echo'class="active"'; ?>> <a href="welcome.php?q=3"><span class="glyphicon glyphicon-stats" aria-hidden="true"></span>&nbsp;Ranking</a></li>

</ul>

<ul class="nav navbar-nav navbar-right">

<li <?php echo''; ?> > <a href="logout.php?q=welcome.php"><span class="glyphicon glyphicon-log-out" aria-hidden="true"></span>&nbsp;Log out</a></li>

</ul>

</div>

</div>

</nav>

<br><br>

<div class="container">

<div class="row">

<div class="col-md-12">

<?php if(@$\_GET['q']==1)

{

$result = mysqli\_query($con,"SELECT \* FROM quiz ORDER BY date DESC") or die('Error');

echo '<div class="panel"><div class="table-responsive"><table class="table table-striped title1">

<tr><td><center><b>S.N.</b></center></td><td><center><b>Topic</b></center></td><td><center><b>Total question</b></center></td><td><center><b>Marks</center></b></td><td><center><b>Action</b></center></td></tr>';

$c=1;

while($row = mysqli\_fetch\_array($result)) {

$title = $row['title'];

$total = $row['total'];

$sahi = $row['sahi'];

$eid = $row['eid'];

$q12=mysqli\_query($con,"SELECT score FROM history WHERE eid='$eid' AND email='$email'" )or die('Error98');

$rowcount=mysqli\_num\_rows($q12);

if($rowcount == 0){

echo '<tr><td><center>'.$c++.'</center></td><td><center>'.$title.'</center></td><td><center>'.$total.'</center></td><td><center>'.$sahi\*$total.'</center></td><td><center><b><a href="welcome.php?q=quiz&step=2&eid='.$eid.'&n=1&t='.$total.'" class="btn sub1" style="color:black;margin:0px;background:#1de9b6"><span class="glyphicon glyphicon-new-window" aria-hidden="true"></span>&nbsp;<span class="title1"><b>Start</b></span></a></b></center></td></tr>';

}

else

{

echo '<tr style="color:#99cc32"><td><center>'.$c++.'</center></td><td><center>'.$title.'&nbsp;<span title="This quiz is already solve by you" class="glyphicon glyphicon-ok" aria-hidden="true"></span></center></td><td><center>'.$total.'</center></td><td><center>'.$sahi\*$total.'</center></td><td><center><b><a href="update.php?q=quizre&step=25&eid='.$eid.'&n=1&t='.$total.'" class="pull-right btn sub1" style="color:black;margin:0px;background:red"><span class="glyphicon glyphicon-repeat" aria-hidden="true"></span>&nbsp;<span class="title1"><b>Restart</b></span></a></b></center></td></tr>';

}

}

$c=0;

echo '</table></div></div>';

}?>

<?php

if(@$\_GET['q']== 'quiz' && @$\_GET['step']== 2)

{

$eid=@$\_GET['eid'];

$sn=@$\_GET['n'];

$total=@$\_GET['t'];

$q=mysqli\_query($con,"SELECT \* FROM questions WHERE eid='$eid' AND sn='$sn' " );

echo '<div class="panel" style="margin:5%">';

while($row=mysqli\_fetch\_array($q) )

{

$qns=$row['qns'];

$qid=$row['qid'];

echo '<b>Question &nbsp;'.$sn.'&nbsp;::<br /><br />'.$qns.'</b><br /><br />';

}

$q=mysqli\_query($con,"SELECT \* FROM options WHERE qid='$qid' " );

echo '<form action="update.php?q=quiz&step=2&eid='.$eid.'&n='.$sn.'&t='.$total.'&qid='.$qid.'" method="POST" class="form-horizontal">

<br />';

while($row=mysqli\_fetch\_array($q) )

{

$option=$row['option'];

$optionid=$row['optionid'];

echo'<input type="radio" name="ans" value="'.$optionid.'">&nbsp;'.$option.'<br /><br />';

}

echo'<br /><button type="submit" class="btn btn-primary"><span class="glyphicon glyphicon-lock" aria-hidden="true"></span>&nbsp;Submit</button></form></div>';

}

if(@$\_GET['q']== 'result' && @$\_GET['eid'])

{

$eid=@$\_GET['eid'];

$q=mysqli\_query($con,"SELECT \* FROM history WHERE eid='$eid' AND email='$email' " )or die('Error157');

echo '<div class="panel">

<center><h1 class="title" style="color:#660033">Result</h1><center><br /><table class="table table-striped title1" style="font-size:20px;font-weight:1000;">';

while($row=mysqli\_fetch\_array($q) )

{

$s=$row['score'];

$w=$row['wrong'];

$r=$row['sahi'];

$qa=$row['level'];

echo '<tr style="color:#66CCFF"><td>Total Questions</td><td>'.$qa.'</td></tr>

<tr style="color:#99cc32"><td>right Answer&nbsp;<span class="glyphicon glyphicon-ok-circle" aria-hidden="true"></span></td><td>'.$r.'</td></tr>

<tr style="color:red"><td>Wrong Answer&nbsp;<span class="glyphicon glyphicon-remove-circle" aria-hidden="true"></span></td><td>'.$w.'</td></tr>

<tr style="color:#66CCFF"><td>Score&nbsp;<span class="glyphicon glyphicon-star" aria-hidden="true"></span></td><td>'.$s.'</td></tr>';

}

$q=mysqli\_query($con,"SELECT \* FROM rank WHERE email='$email' " )or die('Error157');

while($row=mysqli\_fetch\_array($q) )

{

$s=$row['score'];

echo '<tr style="color:#990000"><td>Overall Score&nbsp;<span class="glyphicon glyphicon-stats" aria-hidden="true"></span></td><td>'.$s.'</td></tr>';

}

echo '</table></div>';

}

?>

<?php

if(@$\_GET['q']== 2)

{

$q=mysqli\_query($con,"SELECT \* FROM history WHERE email='$email' ORDER BY date DESC " )or die('Error197');

echo '<div class="panel title">

<table class="table table-striped title1" >

<tr style="color:black;"><td><center><b>S.N.</b></center></td><td><center><b>Quiz</b></center></td><td><center><b>Question Solved</b></center></td><td><center><b>Right</b></center></td><td><center><b>Wrong<b></center></td><td><center><b>Score</b></center></td>';

$c=0;

while($row=mysqli\_fetch\_array($q) )

{

$eid=$row['eid'];

$s=$row['score'];

$w=$row['wrong'];

$r=$row['sahi'];

$qa=$row['level'];

$q23=mysqli\_query($con,"SELECT title FROM quiz WHERE eid='$eid' " )or die('Error208');

while($row=mysqli\_fetch\_array($q23) )

{ $title=$row['title']; }

$c++;

echo '<tr><td><center>'.$c.'</center></td><td><center>'.$title.'</center></td><td><center>'.$qa.'</center></td><td><center>'.$r.'</center></td><td><center>'.$w.'</center></td><td><center>'.$s.'</center></td></tr>';

}

echo'</table></div>';}

if(@$\_GET['q']== 3)

{

$q=mysqli\_query($con,"SELECT \* FROM rank ORDER BY score DESC " )or die('Error223');

echo '<div class="panel title"><div class="table-responsive">

<table class="table table-striped title1" >

<tr style="color:red"><td><center><b>Rank</b></center></td><td><center><b>Name</b></center></td><td><center><b>Email</b></center></td><td><center><b>Score</b></center></td></tr>';

$c=0;

while($row=mysqli\_fetch\_array($q) )

{

$e=$row['email'];

$s=$row['score'];

$q12=mysqli\_query($con,"SELECT \* FROM user WHERE email='$e' " )or die('Error231');

while($row=mysqli\_fetch\_array($q12) )

{

$name=$row['name'];

}

$c++;

echo '<tr><td style="color:black"><center><b>'.$c.'</b></center></td><td><center>'.$name.'</center></td><td><center>'.$e.'</center></td><td><center>'.$s.'</center></td></tr>';

}

echo '</table></div></div>';

}

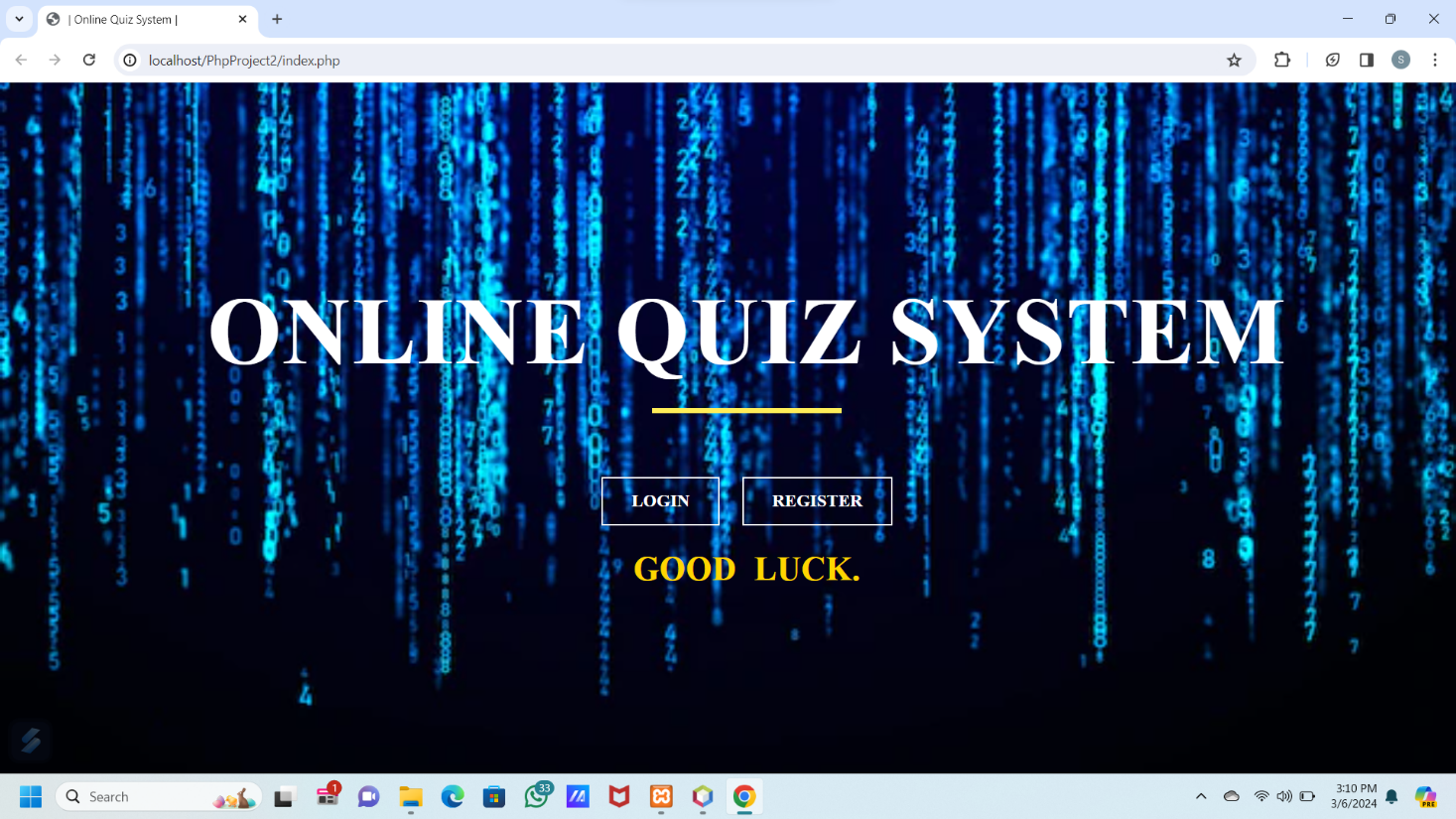
?>

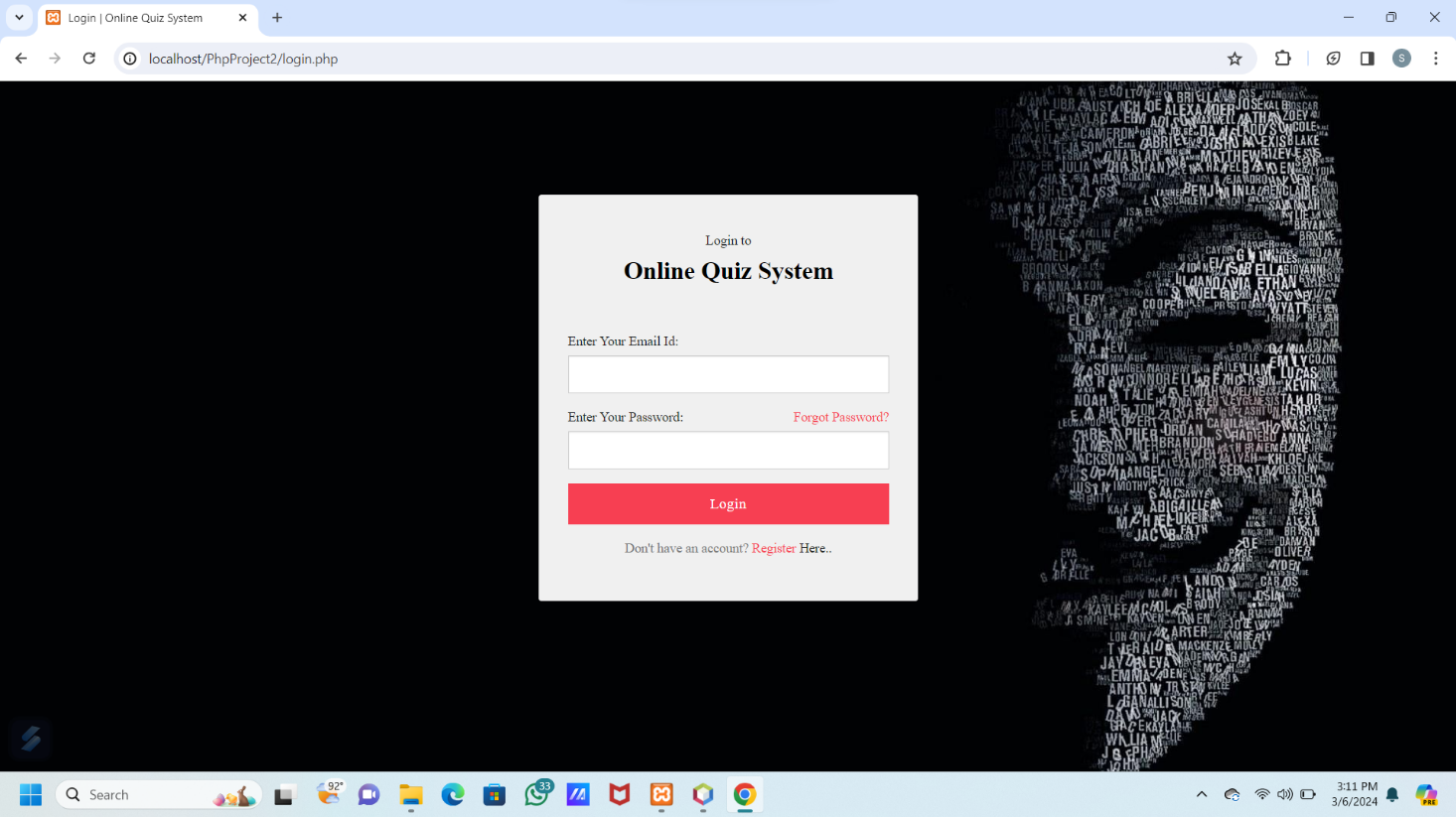
</body>

</html>

**SCREENSHOTS**

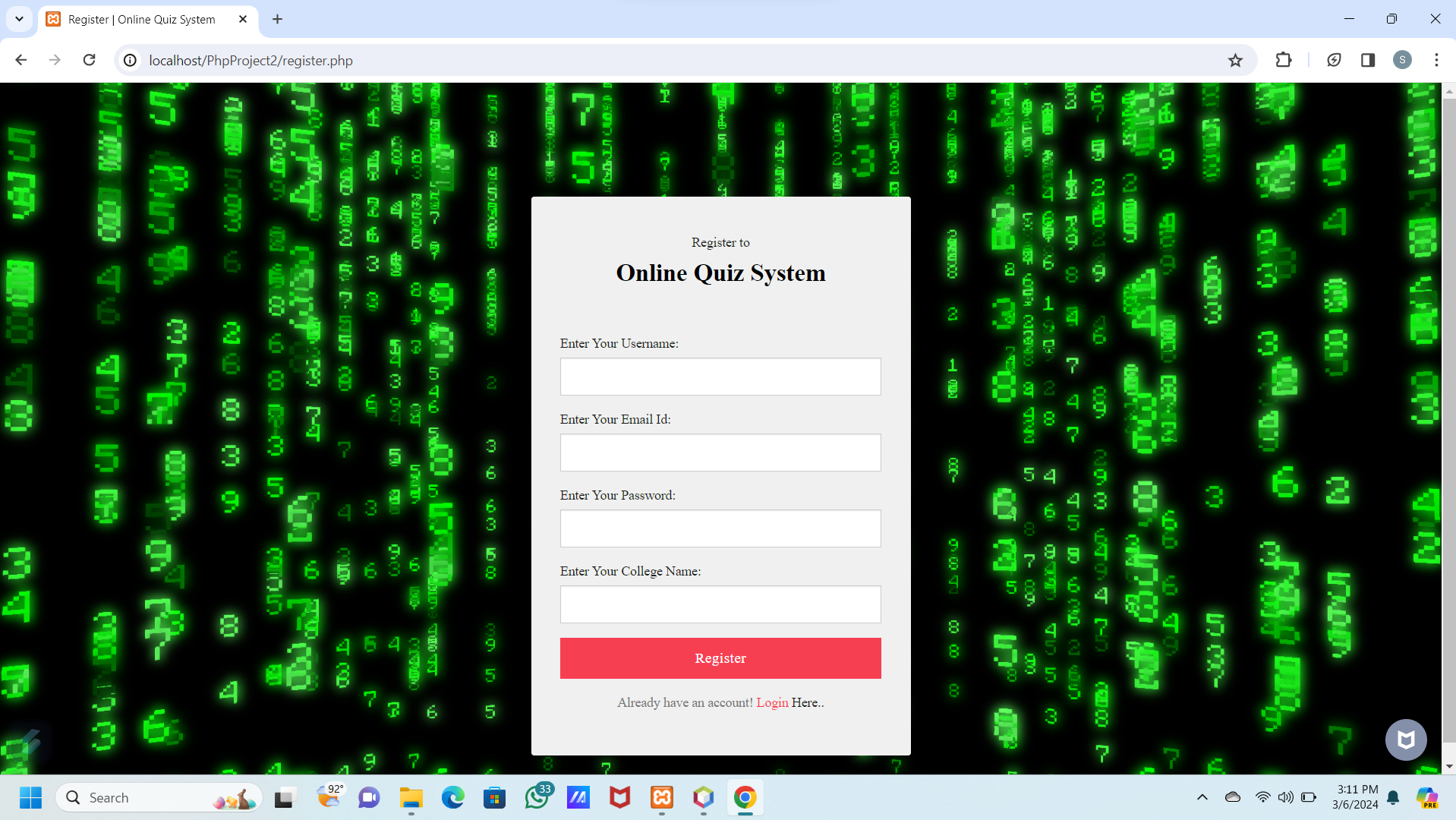
Login Page



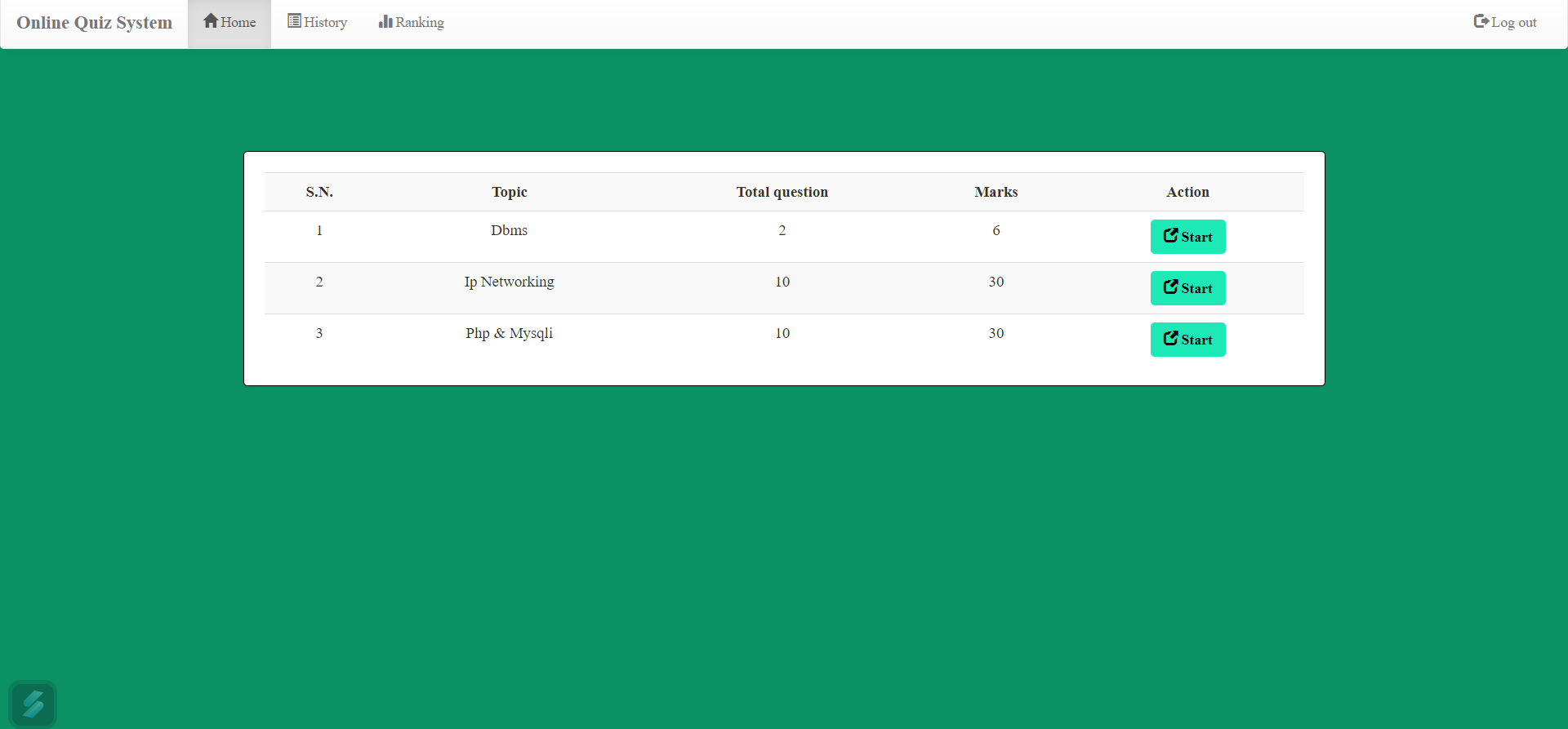


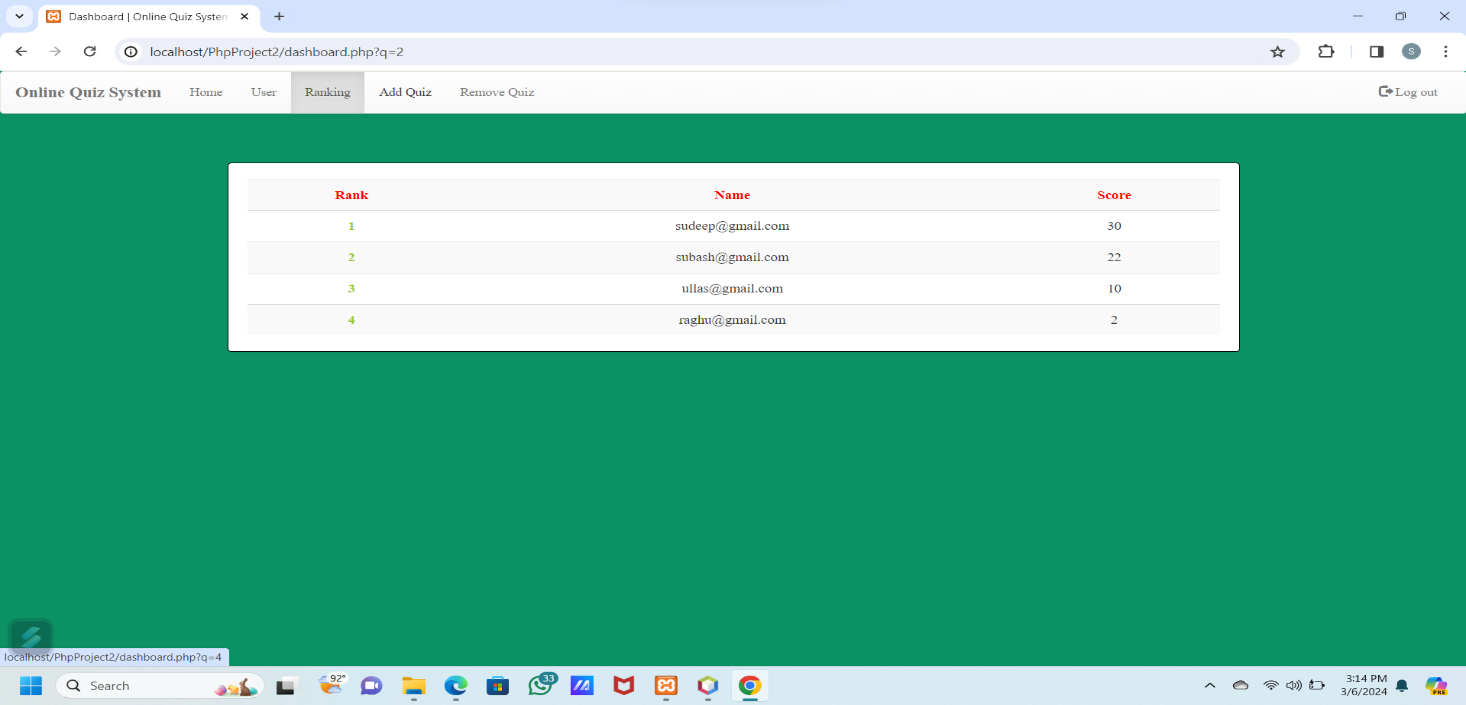
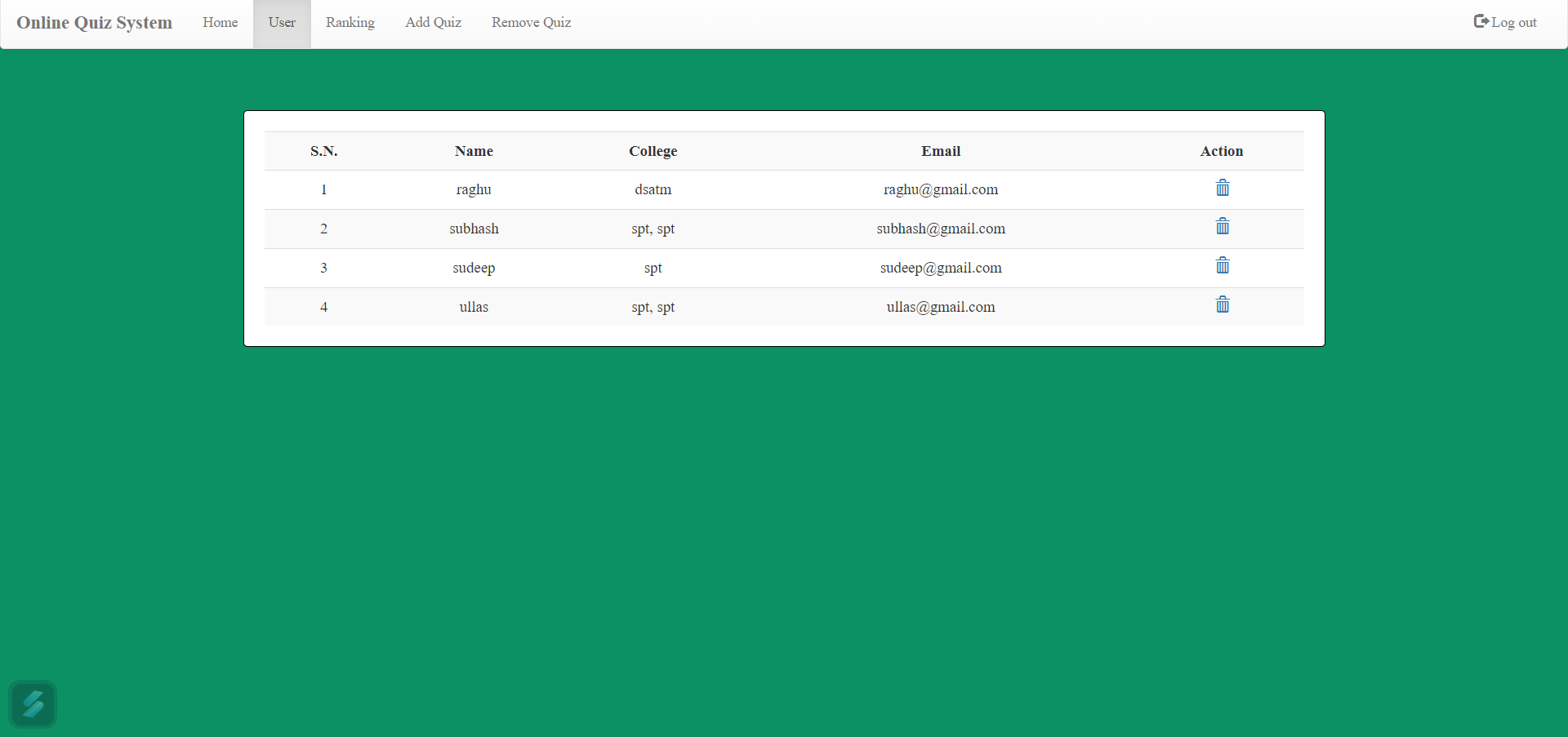
Register Page

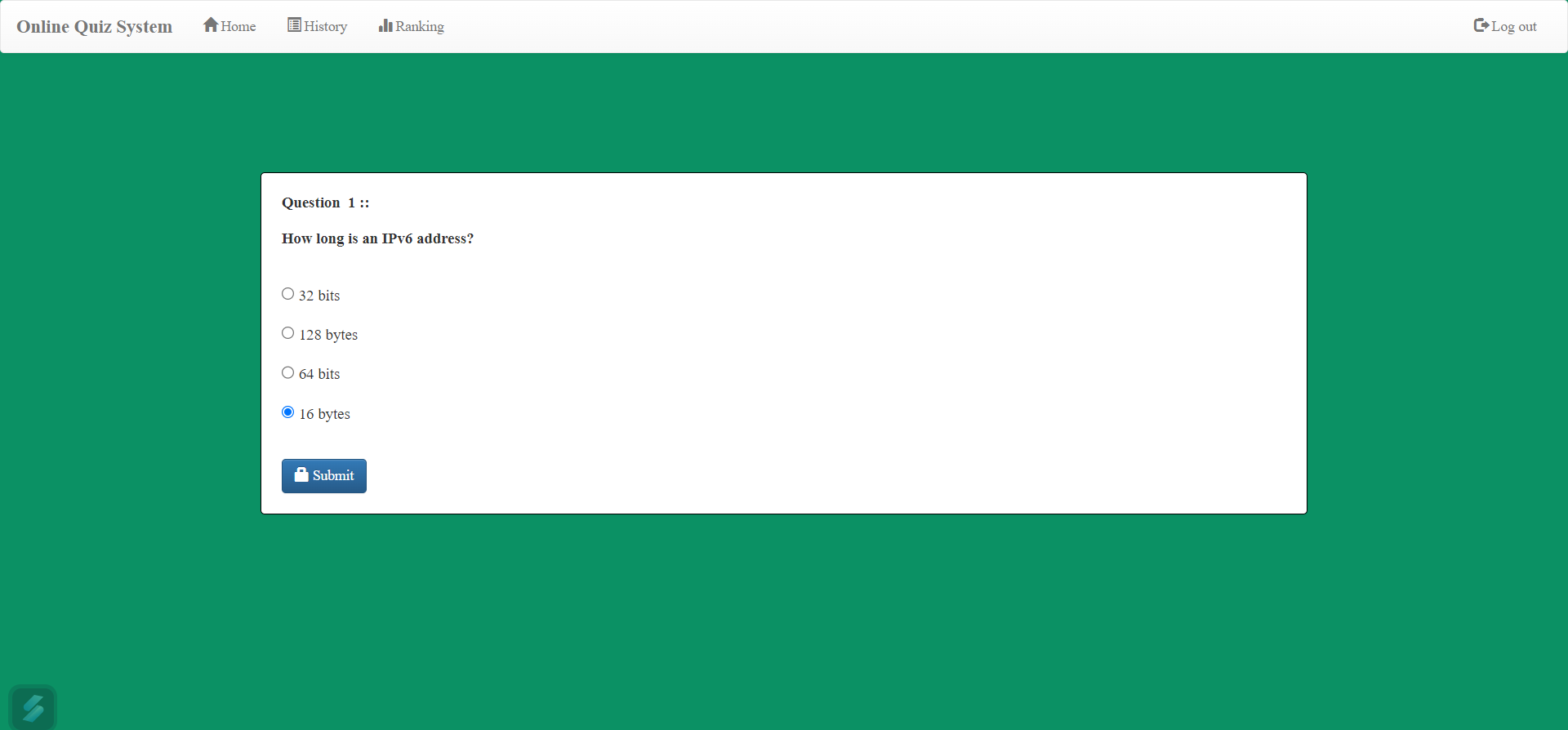
New Register Page

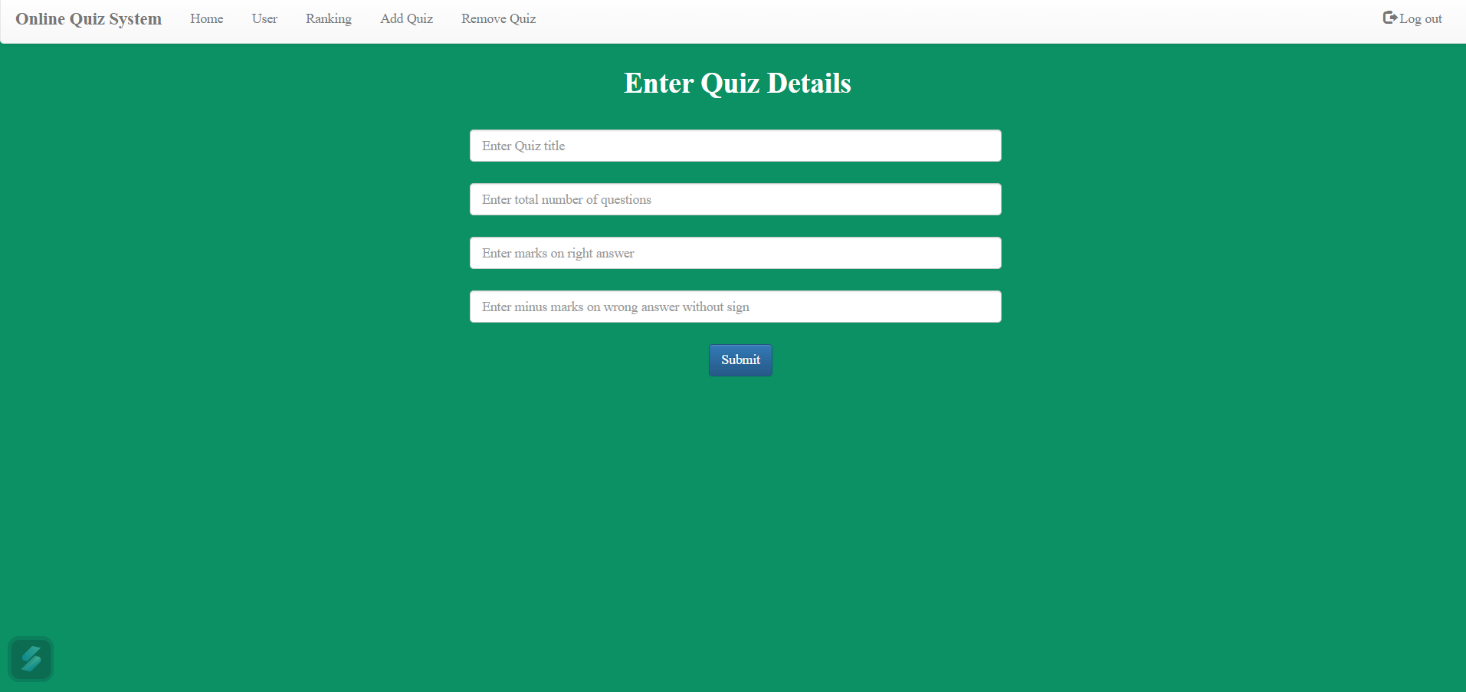


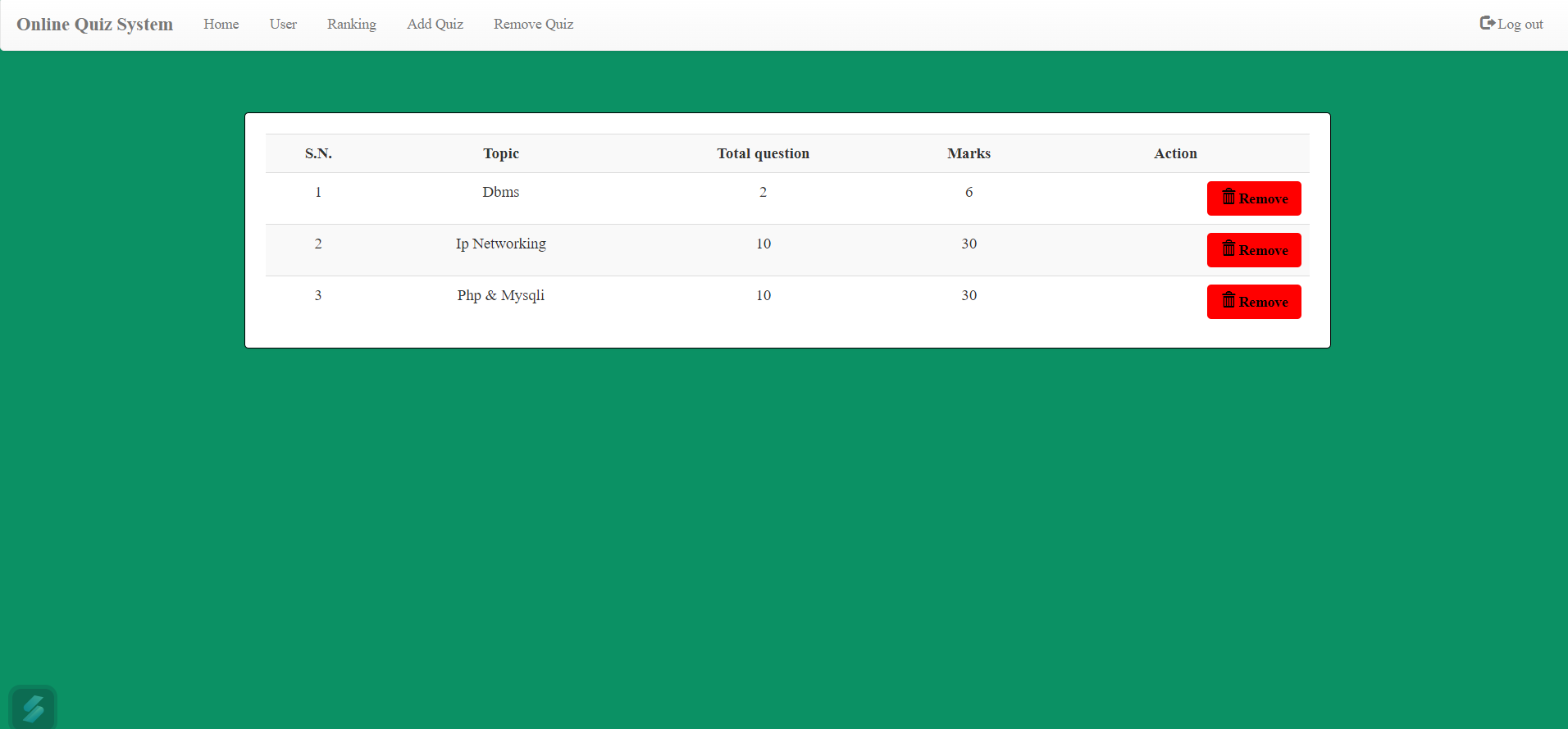
Dashboards











**CONCLUSION:**

The package was designed in such a way that future modifications can bedone easily. The following conclusions can be deduced from the development of the project.

* + Automation of the entire system improves the efficiency
  + It provides a friendly graphical user interface which proves to be better when compared to the existing system.
  + It gives appropriate access to the authorized users depending on their permissions.
  + It effectively overcomes the delay in communications.
  + Updating of information becomes so easier.
  + System security, data security and reliability are the striking features.
  + The System has adequate scope for modification in future if it is necessary.

**FUTURE ENHANCEMENTS:**

This application avoids the manual work and the problems concern with it. It is an easy way to obtain the information regarding the different scheduled examinations information that are

Currently issued. Well I and my team members have worked hard in order to present an improved website better than the existing one’s regarding the information about the various activities. Still ,we found out that the project can be done in a better way. Primarily, when we request information about a particular schedules it just shows the exam date and platform. So, after getting the information we can get access to the onlineexam. The enhancement that we can add the searching option. We can directly search to the particular student details from this site.

**BIBLIOGRAPHY**

The following books were referred during the analysis and execution phase of the project

* Books Referred:
* BEGINNING PHP 5 ---DAVE MERCER
* BLACK BOOK HTML ---WILEY DREAMTECH
* PHP AND MYSQL WEB DEVELOPMENT ---LUKEWELLING,LAURA
* MICROSOFT SQL SERVER-2000 ---RANKIN, PAUL & JENSEN
* SQL SERVER-2000 ---DUSAN PETKOVIC
* PHP IN A NUTSHELL --- PAUL HUDSON
* Websites Referred: http://www.projectworlds.in

#### 