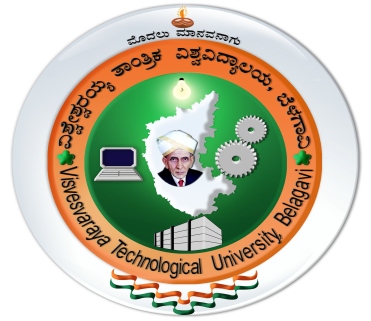
VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi - 590018



Mini Project Report

on

ONLINE QUIZ MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

by

|  |  |
| --- | --- |
| SHAMIR L C | 4MT20CS144 |
| SHIVARAJ SHETTY | 4MT20CS153 |

*Under the Guidance of*

|  |  |
| --- | --- |
| Mr. Pradeep Nazareth | Dr. Dinesha L |
| Senior Assistant Professor | Associate Professor |



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

*(Accredited by NBA)*

MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

*Accredited by NAAC with A+ Grade, An ISO 9001: 2015 Certified Institution*

(*A Unit of Rajalaxmi Education Trust®, Mangalore - 575001*)

Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi

Badaga Mijar, Moodabidri-574225, Karnataka

2022-23

MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

*Accredited by NAAC with A+ Grade, An ISO 9001: 2015 Certified Institution*

(*A Unit of Rajalaxmi Education Trust®, Mangalore - 575001*)

Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi

Badaga Mijar, Moodabidri-574225, Karnataka

**Department of Computer Science & Engineering**

(Accredited by NBA)

**CERTIFICATE**

This is to certify that the mini project entitled “**Online Quiz Management**” is a bonafide work carried out by SHAMIR L C(4MT20CS144) & SHIVARAJ SHETTY (4MT20CS153) in partial fulfillment for the requirement of 5th semester DBMS Laboratory with mini project (18CSL58). It is certified that all the corrections / suggestions indicated for the Internal Assessment have been incorporated in the report. The mini project has been approved as it satisfies the academic requirement in respect of the 18CSL58 prescribed for the 5th Semester B.E in Computer Science & Engineering Program by the **Visvesvaraya Technological University, Belagavi**, for the academic year 2022 – 2023.

|  |  |  |
| --- | --- | --- |
| ………………………….. | ………………………….. | ………………………….. |
| Mr. Pradeep Nazareth  Senior Assistant Professor  MITE, Moodabidri | Dr. Dinesha L  Associate Professor  MITE, Moodabidri | **Prof. Ravinarayana B**  Associate Professor & HOD  MITE, Moodabidri |

|  |  |
| --- | --- |
| **Name of the Examiners** | **Signature with Date** |
| 1. **………………………** | **………………………** |
| 1. **………………………** | **………………………** |

**ABSTRACT**

Online Quiz Management System is a software solution, which allows a particular institute to arrange, conduct and manage quiz via an online environment. This can be done through internet or Local Area Network environments. Some of the problems faced by manual quiz systems are delays in result processing, filtering of results is not easy. The chance of loss of records is high and also record searching is difficult. Maintenance of the system is also very difficult and takes lot of time and effort. In this a candidate can give online quiz on a particular topic and get the results instantly through which the user can know his/her potentials and how much more effort he/she needs to put in to get better marks.

**ACKNOWLEDGEMENT**

The successful completion of any significant task is the outcome of invaluable aggregate combination of different people in radial direction explicitly and implicitly. We would therefore take opportunity to thank and express our gratitude to all those without whom the completion of project would not be possible.

We express our thanks to **Mr. Pradeep Nazarath, Senior Assistant** and Dr. Dinesha L, Associate Professor, Department of Computer Science and Engineering for having provided all the facilities that helped us in timely completion of this report.

We express our sincere gratitude to **Prof. Ravinarayana B, Associate Professor, Head of the Department, Computer Science and Engineering** for his support and guidance.

We would like to thank **Dr. M S Ganesha Prasad, Principal, Mangalore Institute of Technology and Engineering, Moodabidri** for his support and encouragement.

I express my sincere gratitude to our institution and management for providing us with good infrastructure, laboratory facilities, qualified and inspiring staffs, and whose guidance was of immense help in completion of this seminar successfully.

**SHAMIR L C (4MT20CS144)**

**SHIVARAJ SHETTY (4MT20CS153)**

**TABLE OF CONTENTS**

Abstract ................................................................................... i

Acknowledgement ....................................................................ii

|  |  |  |
| --- | --- | --- |
| CHAPTER | CONTENT | PAGE NO |
| 1 | Introduction | 1 |
| 2 | Requirement Analysis and Specification | 4 |
| 3 | System Design | 6 |
| 4 | Implementation | 8 |
| 5 | Testing | 15 |
| 6 | Screenshots | 17 |
| 7 | Conclusion | 19 |
|  | References | 20 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| SL NO | TITLE | PAGE NO |
| Figure 3.1 | ER Diagram for Online Quiz Management | 06 |
| Figure 3.2 | Schema Diagram for Online Quiz Management | 07 |
| Figure 4.1 | Code Snippet of SQL Module | 08 |
| Figure 4.2 | Code Snippet of Database Connection Module | 08 |
| Figure 4.3 | Code Snippet of Staff Login | 09 |
| Figure 4.4 | Code Snippet of Student Signup module | 10 |
| Figure 4.5 | Code Snippet of Delete module | 11 |
| Figure 4.6 | Code Snippet of Update module | 12 |
| Figure 4.7 | Code Snippet of Stored Procedure of Leaderboard module | 13 |
| Figure 4.8 | Code Snippet of Remarks module | 13 |
| Figure 4.9 | Code Snippet of View module | 13 |
| Figure 6.1 | Screenshot of Home Page | 17 |
| Figure 6.2 | Screenshot of Login Form | 17 |
| Figure 6.3 | Screenshot of Signup Form | 17 |
| Figure 6.4 | Screenshot of Student’s Homepage | 18 |
| Figure 6.5 | Screenshot of Quiz Page | 18 |
| Figure 6.6 | Screenshot of Update Page | 18 |

Chapter 1

**INTRODUCTION**

A Database Management System (DBMS) refers to the technology for creating and managing databases. DBMS is a software tool to organize (create, retrieve, update or manage) data in a database. The main aim of a DBMS is to supply a way to store up and retrieve database information that is both convenient and efficient. Database contains meaningful data combined to form information. DBMS is used for data integrity and security. Users may be of any kind such as database administrator, system developer or database users.

* 1. **Problem Statement**

Offline Quiz usually requires large administrative and operational setup. Arrangement of question papers and answer sheet takes heavy cost and wastage of papers. Chances of cheating or use of unfair means is more in offline Quizzes. It usually takes much time in checking the answer copy and in the result preparation. There is a chance of question paper leak in offline Quizzes. Online Quiz System is very helpful to users. The aim of this project is to provide quick, immediate and easy way to appear the Quiz. It can provide special advantage to the candidates. The online Quiz system can automatically add the marks allocated in each question to determine the total marks for the questions. The online Quiz system limits the number of times the student can write a question. Login module helps the user to login to the site. For that he/she has to type the email-id and password correctly. The login provision in this page helps the already registered user to directly access the site and there is a link for registration to a user who is new to this site. Candidate module is mainly for the students. This helps the students to register for the Quiz and answer the Quiz.

* 1. **Objectives**

General objective of our project is to change the current manual system into computerized one. This project would be very useful for educational institutes where regular evaluation of students is required. Specific objectives are:

• Online Quiz system assesses student by conducting online objective quizzes.

• Responses by the candidate will be checked automatically.

• It reduces time consumption.

• Being an integrated online Quiz system reduce paper work.

• The result will be shown immediately after the participating students.

• This project will enable educational institutes to conduct quiz and have automated checking of answers based on the responses by the candidates.

• It will enable educational institutes to perform quiz and get the leaderboard.

* 1. **SQL**

SQL (Structured Query Language) is a computer-based structured, formatted database language designed for managing data in relational database managing systems (RDBMS). SQL is a standardized computer language which was initially developed by IBM for querying, altering and defining relational databases, using declarative statements. SQL is Structured Query Language, which was initially developed by IBM. SQL is pronounced as "sequel". SQL is a computer language for storing, manipulating, and retrieving data in a relational database. SQL is the standard language for Relation Database System.

* 1. **PHP**

PHP is an open-source server-side language which is used for creating dynamic web pages. It can be embedded into HTML. PHP is usually used in conjunction with a MySQL database on Linux/UNIX web servers. It is probably the most popular scripting language. And it is a widely- used general purpose scripting language and interpreter that is freely available.

* 1. **HTML5**

HTML is the standard markup language used to create web pages. Web browsers can read HTML files and render them into visible or audible web pages. HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

* 1. **CSS3**

CSS is abbreviated as Cascading Style Sheets and describes how HTML elements need to be displayed when represented in a web page format or other media. It also helps save a lot of work because controlling the layout of multiple web pages can be done all at a time. It helps in representing how markup-based documents can be presented in conjunction with HTML. CSS is said to as the cornerstone design tool of the World Wide Web along with HTML and JavaScript. CSS is intended for enabling the separation of appearance with content, which includes layout, coloring and font styles.

* 1. **XAMPP**

XAMPP is an abbreviation where X stands for Cross-Platform, A stands for Apache, M stands for MySQL, and the Ps stand for PHP and Perl, respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command-line executables along with modules such as Apache server, MariaDB, PHP, and Perl. XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database,

and PHP through the system of the host itself. Among these technologies, Perl is a programming language used for web development, PHP is a backend scripting language, and MariaDB is the most vividly used database developed by MySQL.

* 1. **JavaScript**

JavaScript is a very powerful client-side scripting language. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage livelier and more interactive, with the help of JavaScript. JavaScript is also being used widely in game development and Mobile application development.

* 1. **Bootstrap**

Bootstrap is a free and open-source web development framework. It’s designed to ease the web development process of responsive, mobile-first websites by providing a collection of syntax for template designs. Bootstrap helps web developers build websites faster as they don’t need to worry about basic commands and functions. It consists of HTML, CSS, and JS-based scripts for various web design-related functions and components.

As a framework, Bootstrap includes the basics for responsive web development, so developers only need to insert the code into a pre-defined grid system. The Bootstrap framework is built on Hypertext Markup Language (HTML), cascading style sheets (CSS) and JavaScript. Web developers using Bootstrap can build websites much faster without spending time worrying about basic commands and functions

Chapter 2

**REQUIREMENT ANALYSIS AND SPECIFICATION**

**2.1 Functional Requirements**

These are statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations. In some cases, the functional requirements may also explicitly state what the system should not do. The functional requirements for a system describe what the system should do. These requirements depend on the type of software being developed, the expected users of the software and the general approach taken by the organization when writing requirements. When expressed as user requirements, the requirements are usually described in an abstract way. However, functional system requirements describe the system function in detail, its inputs and outputs, exceptions, and so on. Functional requirements of Online Quiz are as follows:

Register module:

* Staff and Student can register on the platform with their details by setting up the password for later use to log in.
* Once registered user details will be stored in the database.

Login module:

* Staff and Student can log in to the website with their email address and password which is provided by them on registration.
* The user can only log in if the email has been verified.

Functional requirements for a software system may be expressed in a number of ways. The functional Requirements are broadly classified into 2 categories, they are:

Hardware Requirements

Software Requirements

**Hardware requirements**

• Processor: Intel i3/i5,1.8GHz machine or above

• Main memory: 4GB RAM or more.

• Hard disk drive: 100GB

**Software requirements**

• Operating System: Windows 7 and higher

• Front end: HTML5, CSS3, JavaScript

• Back end: PHP, SQL

• Software: Brackets, XAMPP

• Framework: Bootstrap

**2.2 Non-Functional Requirements**

Non-functional requirements are requirements that are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability, response time and store occupancy. Alternatively, they may define constraints on the system such as the capabilities of I/O devices and the data representations used in system interfaces. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture. Non- functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals", "quality of service requirements" and "nonbehavioral requirements". Qualities, that are non-functional requirements, can be divided into two main categories: Execution qualities, such as security and usability, which are observable at run time.

The Non-Functional requirements of Online Quiz is as follows:

Reliability: Online Quiz is a reliable interface as it provides data security and data safety. Data provided by the user is confidential and safe. User cannot use other user account without password and email id verification.

Consistency: Online Quiz provide consistency in data. Student cannot add or delete Quizzes. The Quizzes added by the staffs can only be viewed by the student. Student cannot attend quiz which does not exist.

Performance: Online Quiz interface performs smoothly for user to have a good and easy experience to attend Quizzes. It is easy to understand and can accessed anywhere through the internet.

Security: The registration process has been made very secure with the help of email validation and password; the user has to verify his email before logging in to the website. The login can be done by email and password.

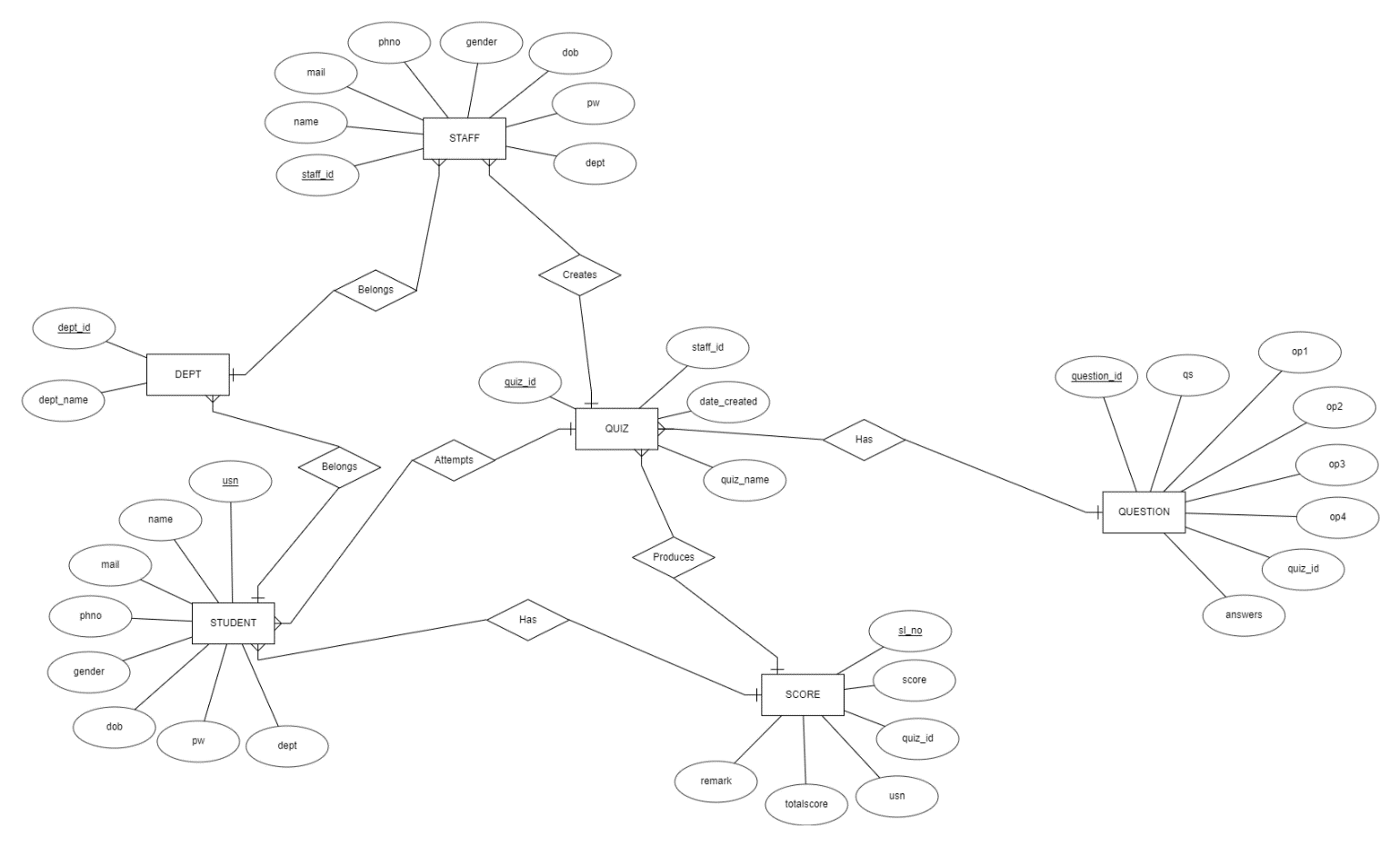
Chapter 3

**SYSTEM DESIGN**

System Design process partitions the system into subsystems based on the requirements. It establishes overall system architecture and is concerned with identifying various components, specifying relationships among components, specifying software structure, maintaining a record of design decisions and providing a blue print for the implementation phase. Design consists of architecture design and detailed design is concerned with the details of how to package processing modules and how to implement the processing algorithms, data structures and interconnections among modules and data structures.

**3.1 ER Diagram**

The Entity-Relationship Data Model (ERD) perceives the real world as consisting of basic objects, called entity & relationship among these objects. It was developed to facilitate database design by allowing specification of an enterprise schema, which represents overall logical structure of a database. The ERD model is very useful in mapping the meaning & interactions of the outside world enterprises onto a conceptual schema.

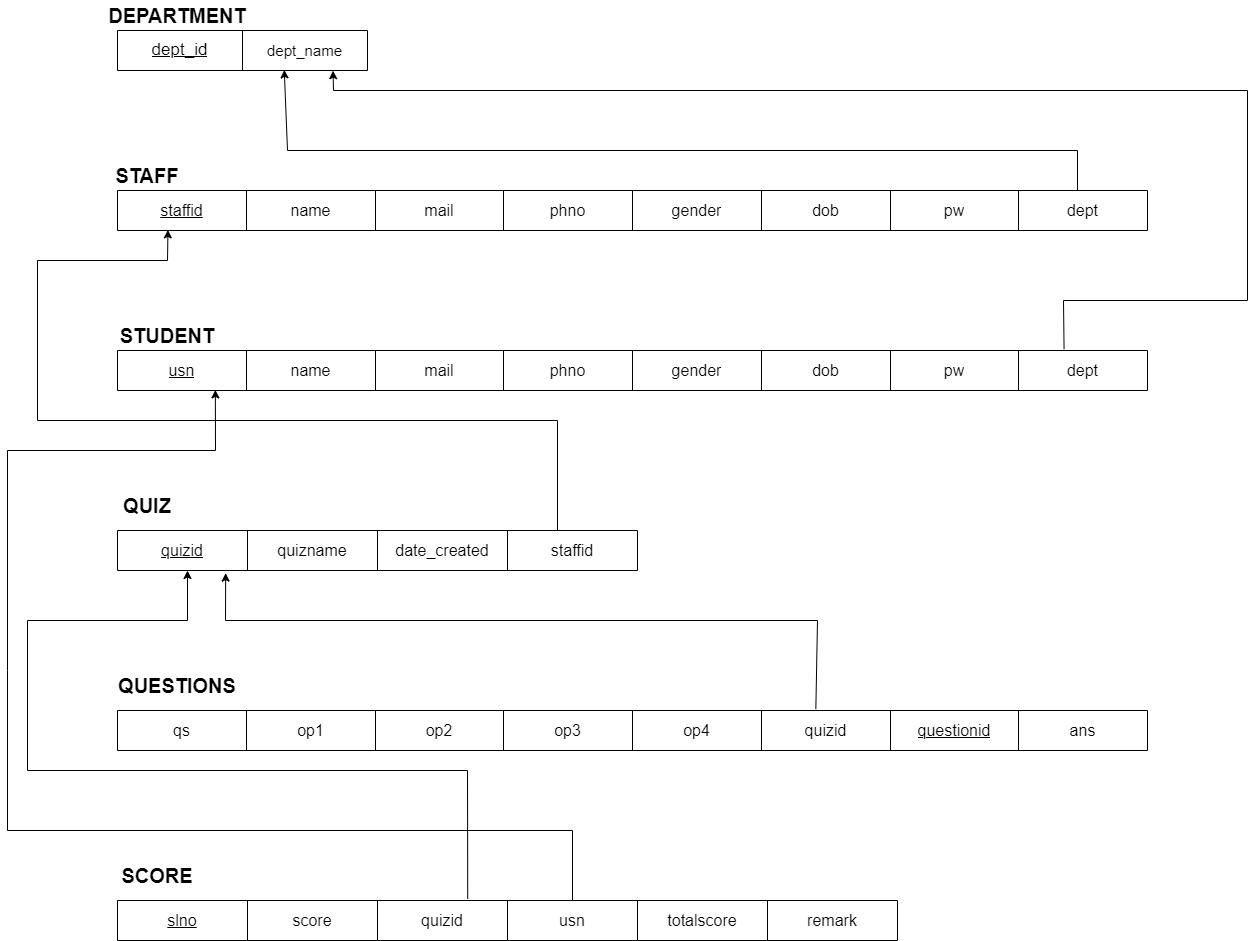


**Figure 3.1 ER Diagram for Online Quiz Management**

The ER diagram consists of six entities: Staff, Student, Dept, Quiz, Score and Question. Each having its own attributes which defines its properties. Here Student has 8 attributes usn, name, mail, phno, gender, dob, pw, dept. Candidate has to make use of the mail and the password registered during registration in the login process. In this mail is the primary key.

**3.2 Schema Diagram**

A schema diagram is a diagram which contains entities and the attributes that will define that schema. A schema diagram only shows us the database design. It does not show the actual data of the database. Schema can be a single table or it can have more than one table which is related. Our database schema contains six tables each defining the relationship between entities.



**Figure 3.2 Schema Diagram for Online Quiz Management**

In the figure 3.2, dept relation has 2 entities that is dept\_id and dept\_name. Dept\_id the primary key. Staff relation has 8 entities that is staff\_id, name, mail, phno, gender, dob, pw, dept; Staff\_id is the primary key. Student relation has 8 entities that is usn, name, mail, phno, gender, dob, pw, dept; usn is the primary key. Quiz relation has 4 entities that is quizid, quizname, date\_created and staff\_id; quizid is the primary key.

Chapter 4

**IMPLEMENTATION**

PHP: Hypertext Pre-processor (or simply PHP) is a server-side scripting language designed for web development, and also used as a general-purpose programming language. 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.

This project uses HTML as front-end tool. Hypertext Mark-up Language (HTML) is the standard mark-up language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the world wide web. Web browser receive HTML documents from a web server or from local storage and render the documents into multimedia web pages.HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.HTML elements are the building blocks of HTML pages. With HTML constructs, images and other pojects such as interactive forms may be embedded into the rendered page.HTML provides a means to create structured documents by structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.HTML elements are delineated by tags, written using angle brackets. Browsers do not display the HTML tags, but use them to interrupt the content of the page.

**4.1 Code Snippet**

Figure 4.1 Code Snippet of SQL Module

<?php

$host='localhost:3306';

$user='root';

$project='quiz';

$ps='';

?>

Figure 4.2 Code Snippet of Database Connection Module

<?php

session\_start();

require\_once 'sql.php';

$conn = mysqli\_connect($host, $user, $ps, $project);

if (!$conn)

{

echo "<script>alert(\"Database error retry after some time !\")</script>";

}

else{

echo "<script>alert(\"Database connected !\")</script>";

}

?>

Figure 4.3 Code Snippet of Staff Login

<?php

if (isset($\_POST['login'])) {

if ( isset($\_POST['staffid']) && isset($\_POST['pass'])) {

require\_once 'sql.php';

$conn = mysqli\_connect($host, $user, $ps, $project);

if (!$conn) {

echo "<script>alert(\"Database error retry after some time !\")</script>";

}

$staffid = mysqli\_real\_escape\_string($conn, $\_POST['staffid']);

$password = mysqli\_real\_escape\_string($conn, $\_POST['pass']);

$sql = "select \* from staff where staffid='{$staffid}'";

$res = mysqli\_query($conn, $sql);

if ($res == true) {

global $dbstaffid, $dbpw;

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

$dbpw = $row['pw'];

$dbstaffid = $row['staffid'];

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

$\_SESSION["staffid"] = $dbstaffid;

}

if ($dbpw === $password) {

header("Location: staffprofile.php");

}

else {

echo "<script>alert('username or password is wrong');</script>";

} } } }

?>

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

$\_SESSION["staffid"] = $dbstaffid;

}

if ($dbpw === $password) {

header("Location: staffprofile.php");

}

else {

echo "<script>alert('username or password is wrong');</script>";

} } } }

?>

}

else{

echo "<script>alert(\"Database connected !\")</script>";

}

?>

Figure 4.4 Code Snippet of Student Signup module

<?php

if (isset($\_POST['studsu'])) {

session\_start();

if (isset($\_POST['name1']) && isset($\_POST['usn1']) && isset($\_POST['mail1']) && isset($\_POST['phno1']) && isset($\_POST['dept1']) && isset($\_POST['dob1']) && isset($\_POST['gender1']) && isset($\_POST['password1']) && isset($\_POST['cpassword1'])) {

require\_once 'sql.php';

$conn = mysqli\_connect($host, $user, $ps, $project); if (!$conn) {

echo "<script>alert(\"Database error retry after some time !\")</script>";

}

$name1 = mysqli\_real\_escape\_string($conn, $\_POST['name1']);

$usn1 = mysqli\_real\_escape\_string($conn, $\_POST['usn1']);

$mail1 = mysqli\_real\_escape\_string($conn, $\_POST['mail1']);

$phno1 = mysqli\_real\_escape\_string($conn, $\_POST['phno1']);

$dept1 = mysqli\_real\_escape\_string($conn, $\_POST['dept1']);

$dob1 = mysqli\_real\_escape\_string($conn, $\_POST['dob1']);

$gender1 = mysqli\_real\_escape\_string($conn, $\_POST['gender1']);

$password1 = mysqli\_real\_escape\_string($conn, $\_POST['password1']);

$cpassword1 = mysqli\_real\_escape\_string($conn, $\_POST['cpassword1']);

if ($password1 == $cpassword1) {

$sql = "insert ignore into student (usn,name,mail,phno,dept,gender,DOB,pw) values('$usn1','$name1','$mail1','$phno1','$dept1','$gender1','$dob1','$password1')";

if (mysqli\_query($conn, $sql)) {

echo "<script>

alert('Done!');

window.location.replace(\"index.php\");</script>";

session\_destroy();

} else {

echo "<script>

alert('Data enter by you alreay exist in Database please Sign In');

window.location.replace(\"index.php\");</script>";

session\_destroy();

}

} else {

echo "<script>

alert(' Password should be same');

window.location.replace(\"singup.php\");</script>";

session\_destroy();

}

}

}

?>

Figure 4.5 Code Snippet of Delete module

<?php

session\_start();

$servername = "localhost:3306";

$username = "root";

$password = "";

$database = "quiz";

$con = mysqli\_connect($servername, $username, $password,$database);

if (!$con) {

die(" Connection Error ");

}

$id = $\_GET['id'];

$sql = "DELETE FROM quiz WHERE quizid = $id";

if (mysqli\_query($con, $sql)) {

echo "<script type='text/javascript'>alert('Quiz Deleted Successfully');

window.location='quizlist.php';</script>";

die;

}

mysqli\_close($con);

?>

Figure 4.6 Code Snippet of Update module

<?php

if (isset($\_POST['submit1'])) {

$op2 = $\_POST["name"];

$op3 = $\_POST["mail"];

$ans = $\_POST["phone"];

$sql = "update student set name='$name',mail='$mail',phno='$phone' where usn='{$stid}'";

$res = mysqli\_query($conn, $sql);

if ($res == true) {

echo '<script>alert("Successfully Updated");</script>';

echo "<script>window.location.replace(\"studprofile.php?usn=".$stid."\")</script>";

}

elseif ($res != true) {

echo '<script>alert("Updation Failed");</script>';

}

}

?>

Figure 4.7 Code Snippet of Stored Procedure of Leaderboard module

CREATE DEFINER=`root`@`localhost` PROCEDURE `leaderboard` () NO SQL SELECT DISTINCT q.quizname, s.score,s.totalscore,st.usn,st.name,s.usn

FROM score s, student st, quiz q

WHERE s.usn=st.usn AND q.quizid=s.quizid

ORDER BY score DESC$$

DELIMITER ;

Figure 4.8 Code Snippet of Remarks module

DELIMITER $$

CREATE TRIGGER `remarks` BEFORE INSERT ON `score` FOR EACH ROW set NEW.remark = IF(NEW.score < NEW.totalscore / 2, 'Bad', 'Good')

$$

DELIMITER ;

Figure 4.9 Code Snippet of View module

<?php

if(isset($\_GET["qid"])){

$qid=$\_GET["qid"];

$sql ="select \* from questions where quizid='{$qid}'";

$res=mysqli\_query($conn,$sql);

if($res)

{

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

if(mysqli\_num\_rows($res)==0)

{

echo "No questions found under this quiz please come later";

echo "<form method=\"POST\">";

echo "<input id=\"btn\" type=\"submit\" name=\"submit\" value=\"Add Questions\" class=\" btn btn-success \"><br><br><br>";

}

else{

$i=1;

$j=0;

echo "<form method=\"POST\">";

echo "<input id=\"btn\" type=\"submit\" name=\"submit\" value=\"Add Questions\" class=\" btn btn-success \"><br><br><br>";

echo "</form><br><br>";

while ($row = mysqli\_fetch\_assoc($res)) {

echo $i.". ".$row["qs"]."<br>";

echo " A:".$row["op1"]."<br>";

echo " B: ".$row["op2"]."<br>";

echo "C: ".$row["op3"]."<br>";

echo " D: ".$row["op4"]."<br><br>";

echo "<label>Correct answer: </label>".$row["answers"]."<br><br><br>";

$i++;

$j = 0;

}

echo "</form><br><br>";

}

}

else

{

echo "error".mysqli\_error($conn).".";

}

if(isset($\_POST["submit"])){

echo "<script>window.location.replace(\"addq.php?qid=".$qid."\")</script>";

}

}

?>

Chapter 5

**TESTING**

Software testing is the process of used to identify the correctness, security, completeness and quality of developed computer software. This includes the process of executing the program or applications with the intent of finding errors. An individual unit, functions or procedures of developed project is verified and validated and these units are fit for use.

**5.1 Testing process**

Best testing process is to test each subsystem separately, as we have done in project. Best done during implementation. Best done after small sub-steps of the implementation rather than large chunks. Once each lowest level unit has been tested, units are combined with related units and retested in combination. This proceeds hierarchically bottom-up until the entire system is tested as a whole. Typical levels of testing:

• Module- package, abstract data type, class

• Sub-system- collection of related modules, cluster of classes, method-message paths

• Acceptance testing- whole system with real data (involve customer, user, etc.)

Alpha testing is acceptance testing with a single client (common for bespoke systems). Beta testing involves distributing system to potential customers to use and provide feedback. In this project, beta testing has been followed. This exposes system to situations and errors that might not be anticipated by us.

**5.1.1 Unit testing**

Unit testing is the process of testing individual software components unit or modules. Since it needs the detailed knowledge of the internal program design and code this task is done by the programmer and not by testers.

**5.1.2 Integration Testing**

Integration testing is another aspect of testing that is generally done in order to uncover errors associated with the flow of data across interfaces. The unit-tested modules are grouped together and tested in small segment, which makes it easier to isolate and correct errors. This approach is continued until we have integrated all modules to form the system as a whole. After the completion of each module, it has been combined with the remaining module to ensure that the project is working properly as expected.

**5.1.3 System Testing**

System testing tests a completely integrated system to verify that it meets its requirements. After the completion of the entire module, they are combined together to test whether the entire project is working properly.

**5.2 Test Cases**

A Test Case is a software testing document, which consists of events, action, input, output, expected result and actual result. Technically a test case includes test description, procedure, expected result and remarks. Test cases should be based primarily on the software requirements and developed to verify correct functionality and to establish conditions that reveal potential errors.

|  |  |  |  |
| --- | --- | --- | --- |
| Test cases no | Test Case | Expected results | Status |
| 1 | Logging into website | Email and password provided correct | Successful |
| 2 | Logging into website | Email incorrect | Unsuccessful |
| 3 | Logging into website | Password Incorrect | Unsuccessful |
| 4 | Logging into website | Any field left empty | Unsuccessful |

Table 5.1 Test Case for Login

Table 5.1 represents the test case for login module. It shows both successful and unsuccessful results for the test cases.

|  |  |  |  |
| --- | --- | --- | --- |
| Test cases no | Test Case | Expected results | Status |
| 1 | Registration for new user | All details provided correctly | Successful |
| 2 | Registration for new user | Any one field is incorrect | Unsuccessful |
| 3 | Registration for new user | Any field left empty | Unsuccessful |

Table 5.2 Test Case for Signup

Table 5.2 represents the test case for sign up module. It shows both successful and unsuccessful results for the test cases.

|  |  |  |  |
| --- | --- | --- | --- |
| Test cases no | Test Case | Expected results | Status |
| 1 | Add Quiz for students | All details provided correctly | Successful |
| 2 | Add Quiz for students | Any field left empty | Unsuccessful |

Table 5.3 Test Case for Add Quiz

Table 5.3 represents the test case for add quiz module. It shows both successful and unsuccessful results for the test cases.

Chapter 6

**SCREENSHOT**

Figure 6.1 Screenshot of Homepage

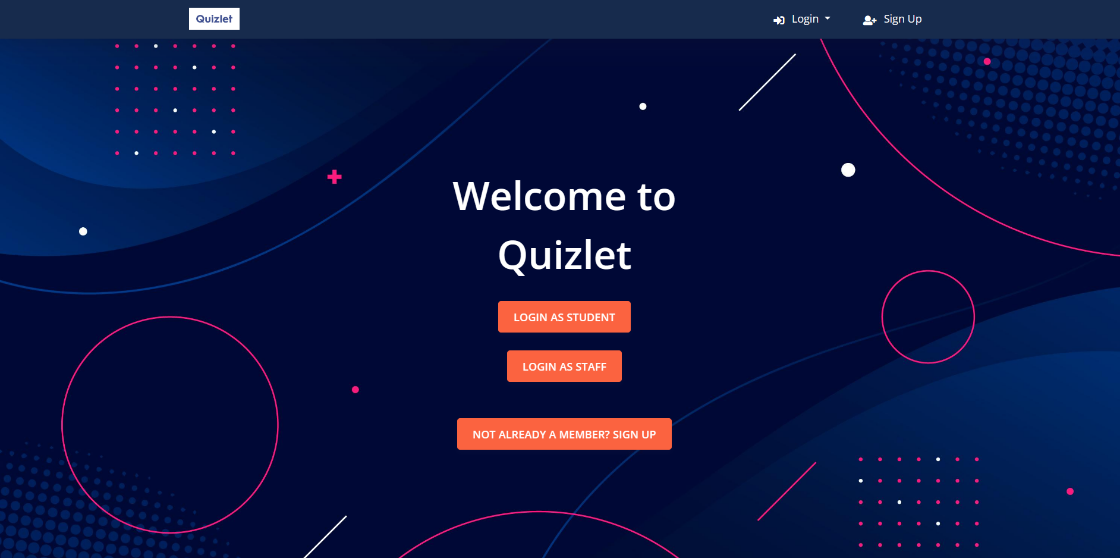


Figure 6.1 indicates the home page. This contains navigation bar, through which you can navigate to other pages. It also contains some details about the website at the home page.

Figure 6.2 Screenshot of Login Form

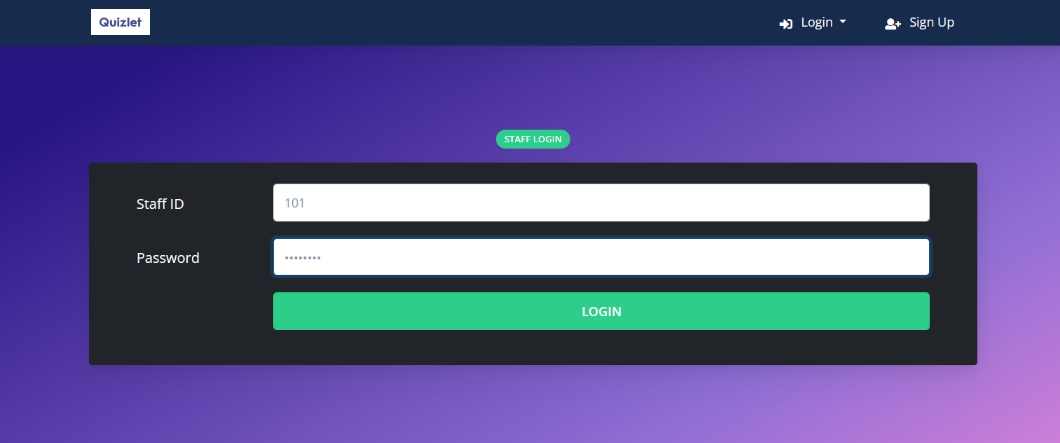


Figure 6.2 indicates the login page. If the staff has already registered to the website, then he/she can get access to the website by filling the details.

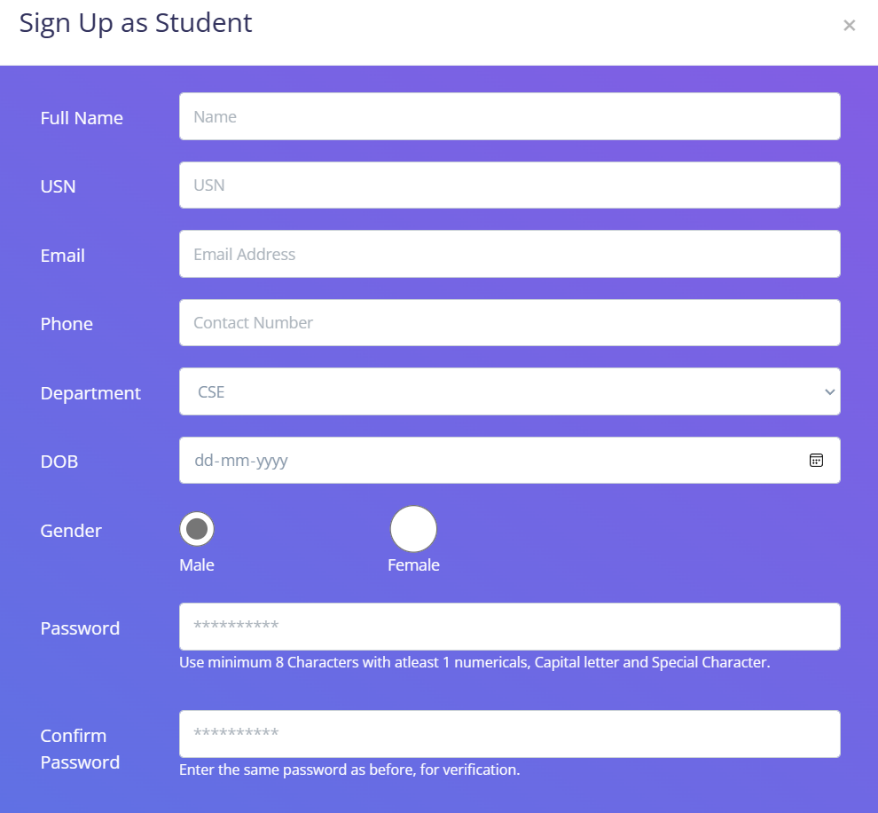


Figure 6.3 Screenshot of Signup Form

Figure 6.3 indicates the Signup page. The student can register into the website by filling the above details.

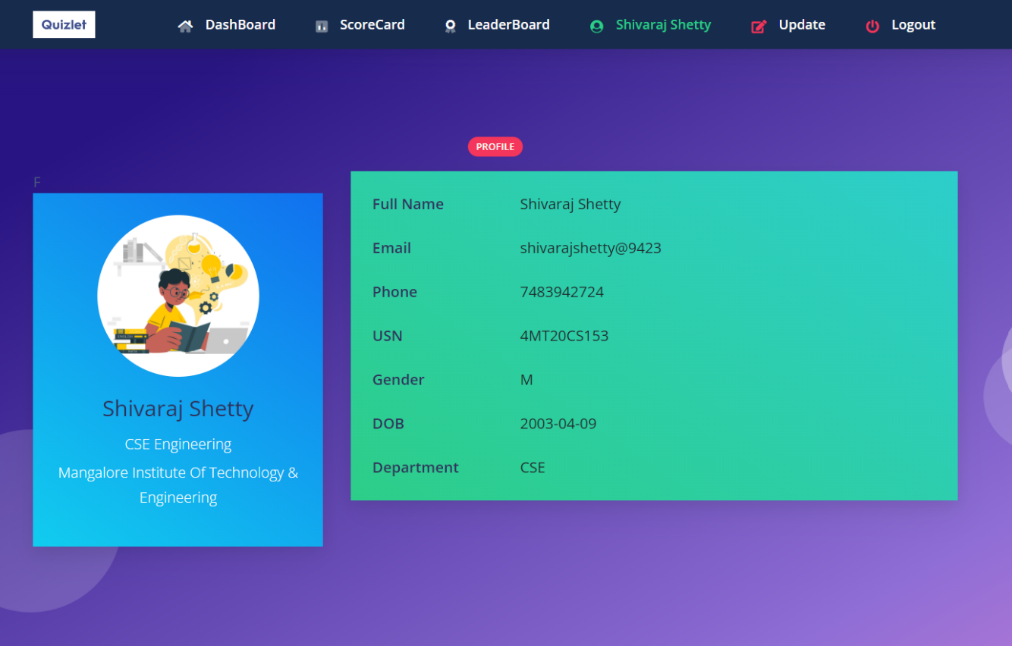


Figure 6.4 Screenshot of Student’s Homepage

Figure 6.4 indicates the home page of student after a successful login. Here it contains navigation, through which the student can navigate.



Figure 6.5 Screenshot of Quiz page

Figure 6.5 indicates the quiz page. Here the question is displayed with four options, the student can select any option and submit.

Figure 6.6 Screenshot of Update page

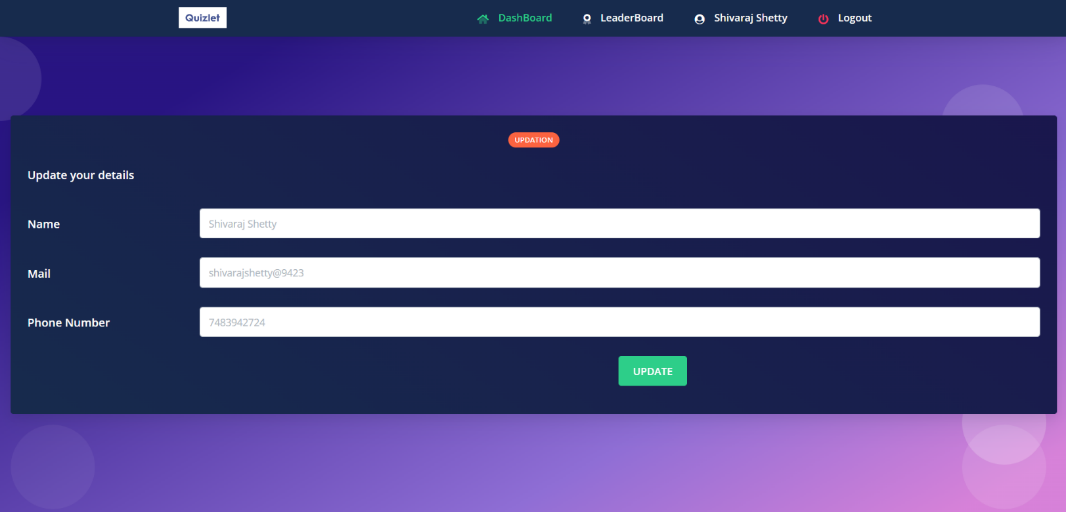


Figure 6.6 indicates the update page. Here the candidate can update some of the details they want to change.

Chapter 7

**CONCLUSION AND FUTURE ENHANCEMENTS**

Online Quiz Management System is actually a software which handles the essential data. This software helps in effectively management of the online quizzes. The main purpose is effective and easy handling on online quizzes. From a proper analysis of positive points and constraints on the component, it can be safely concluded that the product is a highly efficient GUI based component. This application is working properly and meeting to all user requirements. This component can be easily plugged in many other systems.

The future enhancement for the application is by further making the quizzes in a timed quiz format and by notifying the students when a new quiz has been made by the staff user.

**REFERENCES**

1. Fundamentals of Database System, Ramez Elmarsri Shamkanth B, 7th edition, 2017, Pearson.
2. <https://www.w3schools.com/>
3. <https://www.chat.openai.com/>