

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
“Jnana Sangama”, Belagavi-590018



A  
**Mini project Report**  
on  
**“ONLINE EXAMINATION MANAGEMENT SYSTEM”**

SUBMITTED IN PARTIAL FULFILLMENT FOR 5<sup>TH</sup> SEMESTER  
**BACHELOR OF ENGINEERING**  
IN  
**COMPUTER SCIENCE AND ENGINEERING**

SUBMITTED BY

**SAJEEVAN**  
[1JB21CS127]

**SHREYAS K**  
[1JB21CS142]

**Under the Guidance of:**

**Dr. Bindiya M K**  
Associate Professor,  
Dept of CSE, SJBIT



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**SJB INSTITUTE OF TECHNOLOGY**

No.67, BGS Health & Education City, Dr. Vishnuvardhan Rd, Kengeri,  
Bengaluru, Karnataka 560060.

**2023-24**

|| Jai Sri Gurudev ||  
Sri Adichunchanagiri Shikshana Trust ®

## SJB INSTITUTE OF TECHNOLOGY

No.67, BGS Health & Education City, Dr. Vishnuvardhan Rd, Kengeri, Bengaluru, Karnataka 560060

### Department of Computer Science and Engineering



### CERTIFICATE

Certified that the DBMS Mini Project entitled "**ONLINE EXAMINATION MANAGEMENT SYSTEM**" carried out by **SAJEEVAN, SHREYAS K** bearing USN **1JB21CS127** and **1JB21CS142** are bonafide students of **SJB Institute of Technology** in partial fulfilment for 5<sup>th</sup> Semester of **BACHELOR OF ENGINEERING** in **COMPUTER SCIENCE AND ENGINEERING** of the Visvesvaraya Technological University, Belagavi during the academic year **2023-24**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the Departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project prescribed for the said Degree.

---

**Signature of Guide**

**Dr.Bindiya M K**  
Associate Professor,  
Dept. of CSE, SJBIT

---

**Signature of HOD**

**Dr. Krishna A N**  
Professor & Head  
Dept. of CSE, SJBIT

**Name of the examiner**

**Signature**

1.

---

2.

---



## ACKNOWLEDGEMENT



We would like to express our profound grateful to His Divine Soul **Jagadguru Padmabhushan Sri Sri Sri Dr. Balagangadharanatha Mahaswamiji** and His Holiness **Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Mahaswamiji** for providing us an opportunity to complete our academics in this esteemed institution.

We would also like to express our profound thanks to **Revered Sri Sri Dr. Prakashnath Swamiji**, Managing Director, SJB Institute of Technology, for his continuous support in providing amenities to carry out this Mini Project in this admired institution.

We express our gratitude to **Dr. K. V. Mahendra Prashanth**, Principal, SJB Institute of Technology, for providing us an excellent facility and academic ambience; which have helped us in satisfactory completion of Mini Project work.

We extend our sincere thanks to **Dr. Krishan A N**, Head of the Department, Computer Science and Engineering for providing us an invaluable support throughout the period of our Mini project work.

We wish to express our heartfelt gratitude to our guide **Dr. Bindiya M K, Associate Professor, Dept of CSE**, for her valuable guidance, suggestions and cheerful encouragement during the entire period of our Mini Project work.

Finally, we take this opportunity to extend our earnest gratitude and respect to our parents, Teaching & Non teaching staffs of the department, the library staff and all our friends, who have directly or indirectly supported us during the period of our Mini Project work.

Regards,  
**SAJEEVAN [1JB20CS127]**  
**SHREYAS K [1JB20CS142]**

## **ABSTRACT**

The online examination system is a web-based examination system where examinations are given online. either through the internet or intranet using computer system. The main goal of this online examination system is to effectively evaluate the student thoroughly through a totally automated system that not only reduce the required time but also obtain fast and accurate results.

Online examination system is an online test simulator is to take online examination, test in an efficient manner and no time wasting for manually checking of the test paper. The main objective of this web based online examination system is to efficiently evaluate the student thoroughly through a fully automated system that not only saves lot of time but also gives fast and accurate results. For students they give papers according to their convenience from any location by using internet and time and there is no need of using extra thing like paper, pen etc.

# TABLE OF CONTENTS

<b>Acknowledgement</b>	i
<b>Abstract</b>	ii
<b>Table of contents</b>	iii
<b>List of figures</b>	v
<b>List of tables</b>	vi
<b>Chapter 1</b>	
<b>Introduction</b>	
1.1 Objectives.....	1
1.2 Dbms.....	2
1.3 Design and modeling.....	3
<b>Chapter 2</b>	
<b>Literature survey</b>	
2.1 About the project .....	4
2.2 Purpose.....	4
2.3 Scope.....	5
<b>Chapter 3</b>	
<b>System requirement specification</b>	
3.1 Software requirements.....	6
3.2 Hardware requirements.....	6
<b>Chapter 4</b>	
<b>System design</b>	
4.1 Entity relationship diagram.....	7
4.2 Schema diagram.....	8
4.3 Block diagram .....	10
4.4 Technology used.....	11

## **Chapter 5**

### **Implementation**

5.1 Database structure .....	12
5.2 Front end.....	15
5.2.1 Html.....	16
5.2.2 Css.....	16
5.2.3 Php.....	17
5.3 Back end .....	18
5.3.1 Mysql.....	18
5.3.2 Xammp application.....	19

## **Chapter 6**

### **Snapshots**

6.1 Home page .....	21
6.2 Sign up pages.....	21
6.3 Login pages.....	22
6.4 Quiz List pages .....	22
6.5 Questions page.....	23
6.6 Score page.....	23
6.7 Quiz adding page.....	23
6.8 Profile pages.....	24
6.9 Contact page.....	24

<b>Conclusion.....</b>	<b>25</b>
------------------------	-----------

<b>References .....</b>	<b>26</b>
-------------------------	-----------

## **LIST OF FIGURES**

<b>SL NO</b>	<b>DESCRIPTION</b>	<b>PAGE NO</b>
Fig 1.1	Simplified database system environment	3
Fig 4.1	Entity relationship diagram	9
Fig 4.2	Schema diagram	9
Fig 4.3	Block diagram	11
Fig 6.1	Home page	21
Fig 6.2	Sign up pages	21
Fig 6.3	Login page	22
Fig 6.4	Quiz list page	22
Fig 6.5	Questions page	23
Fig 6.6	Score page	23
Fig 6.7	Quiz adding page	23
Fig 6.8	Profile pages	24
Fig 6.9	Contact page	24

## **LIST OF TABLES**

<b>TABLE NO</b>	<b>DESCRIPTION</b>	<b>PAGE NO</b>
Table 5.1	Department table	12
Table 5.2	Student table	13
Table 5.3	Staff table	14
Table 5.4	Quiz table	14
Table 5.5	Question table	15
Table 5.6	Score table	15

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 OBJECTIVES**

The objective of the Online Examination System is to create a streamlined and efficient method for conducting exams online. Here is a breakdown of the objectives and functional specifications:

**Efficiency:** The system aims to streamline the examination process by eliminating manual tasks such as distributing paper exams and manually grading them. This saves time and resources for both students and staff.

**Accuracy:** By automating the examination process, the system ensures accurate evaluation of students' performance. This reduces errors that may occur during manual grading and ensures fair assessment.

**Convenience:** Students can take exams from any location with internet access, providing flexibility and convenience. They no longer need to travel to a physical location to take exams, saving time and effort.

**Cost-effectiveness:** The system eliminates the need for paper, printing, and other materials traditionally used in exams, reducing costs for educational institutions.

#### **Functional Specifications:**

**Registration:** The system allows new students to register for exams easily, providing a user-friendly interface for the registration process.

**Data Management:** It collects and stores student and staff information securely in databases, ensuring data integrity and confidentiality.

**Question Paper Management:** The system manages the types and numbers of question papers available for exams, ensuring that the right papers are assigned to students.

**Result Generation:** After the exam, the system automatically generates results based on students' responses, providing instant feedback to both students and staff.

Overall, this aims to revolutionize the examination process by leveraging technology to make it more efficient, accurate, and convenient for all stakeholders involved.

## 1.2 DBMS

A Database is a collection of data and Management System is a set of programs to store and retrieve those data. Based on this one can define DBMS as a collection of inter-related data and set of programs to store & access those data in an easy and effective manner.

Database systems are basically developed for large amount of data. When dealing with huge amount of data, there are two things that require optimization: Storage of data and retrieval of data.

**Storage:** According to the principles of database systems, the data is stored in such a way that it acquires lot less space as the redundant data (duplicate data) has been removed before storage. Let's take a layman example to understand this. In a banking system, suppose a customer is having two accounts, one is saving account and another is salary account. Let's say bank stores saving account data at one place (these places are called tables we will learn them later) and salary account data at another place, in that case if the customer information such as customer name, address etc. are stored at both places then this is just a wastage of storage (redundancy/ duplication of data), to organize the data in a better way the information should be stored at one place and both the accounts should be linked to that information somehow. The same thing we achieve in DBMS.

**Fast Retrieval of data:** Along with storing the data in an optimized and systematic manner, it is also important that we retrieve the data quickly when needed. Database systems ensure that the data is retrieved as quickly as possible.

The choice of a database product is often influenced by factors such as:

- The computing platform (i.e., hardware, operating system)the volume of data to be managed
- The number of transactions required per second
- Existing applications or interfaces that an organization may have
- Support for heterogeneous and/or distributed computing
- Cost

### 1.3 DESIGN AND MODELING

The first task of a database designer is to produce a conceptual data-model that reflects the structure of the information to be held in the database. A common approach to this is to develop an entity-relationship model, often with the aid of drawing tools. Another popular approach is the Unified Modeling Language. A successful data model will accurately reflect the possible state of the external world being modeled: for example, if people can have more than one phone number, it will allow this information to be captured.

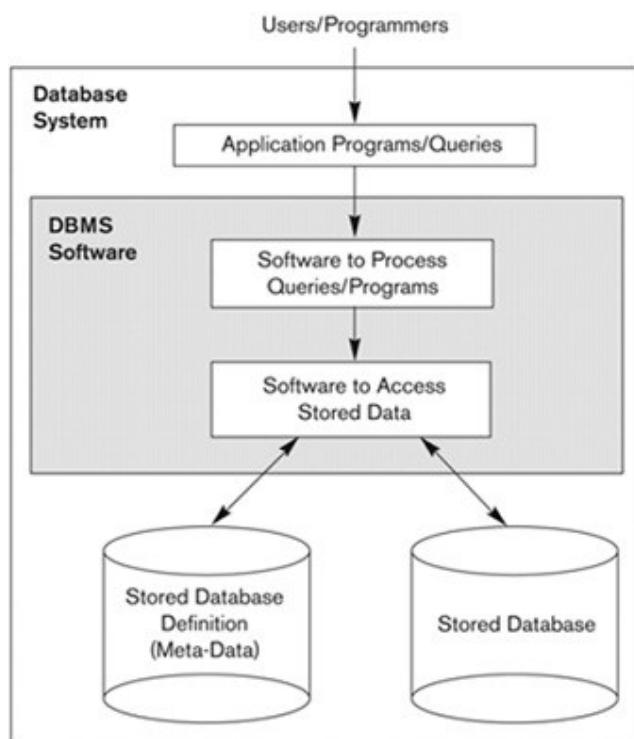


Figure 1.1: Simplified database system environment

## **CHAPTER 2**

# **LITERATURE SURVEY**

### **2.1 ABOUT THE PROJECT**

Online Exam System is a full-fledged system which is beneficial for both teachers and students. With this site, Administrator can allow authorized entry of users which includes faculties and students. Faculties are the one who register on the site and conduct timed exams in their respective courses. They can add questions in the test by selecting the available settings. The tests added by faculties are available to students who can attempt the test and get score then and there, Thus the purpose of the site is to provide a system that saves the efforts and time of both the teachers and the students. Online Exam System is a web application that establishes a network between the faculties and the students. Faculties enter on the site the questions they want in the exam. These questions are displayed as a test to the eligible students. The answers enter by the students are then evaluated and their score is calculated and saved. This score then can be accessed by the faculties and administrator to evaluate their performance.

Online Exam System provides the platform but does not directly participate in, nor is it involved in any tests conducted. Questions are posted not by the site, but users of the site. The administrator keeps an eye on the overall functioning of the system which includes how many users are using the site, how many tests are added by which faculty, how many students have given those tests and view the score of any student.

### **2.2 PURPOSE**

The purpose of on-line test system is to take online test in an efficient manner and no timewasting for checking the paper. It provides online facility to Institutes to conduct online exams and to Students to give online exams. Institutes (i.e the faculties in institute) can enter and edit the questions in test. Students can login and give their respective exams and view their score then and there.

The main objective of on-line test simulator is to efficiently evaluate the candidate thoroughly through a fully automated system that not only saves lot of time but also gives fast results.

## **2.3 SCOPE**

This website provides facility to institutes to conduct online exams and maintaining the records of users by providing authorized entry. Users can register as Faculty / Student and wait for the admin to send a notification email. User can then login with the valid username and password and can carry out their respective tasks.

- Maintaining records of users (Faculties and Students). Admin can add/ delete a particular user who registers on site and can view the activities of users i.e. how many tests are added by a particular faculty, how many tests are given by particular student and view their respective scores.
- Add test in particular course (Faculty). Faculty can add time limited tests in a particular course by adding questions in the same, can view which student has attempted the tests and view their respective scores.
- Attempt test (Student). Student can attempt test added by faculty and get their score then and there. They can also get the test review i.e. answer explanation of each question in test.

## **CHAPTER 3**

# **SYSTEM REQUIREMENT SPECIFICATION**

A computerized way of handling information about property and user's details is efficient organized and time saving, compared to a manual way of doing so, this is done through a database driven web application whose requirements are mentioned in this section.

### **3.1 SOFTWARE REQUIREMENTS**

- Programming language: PHP and MYSQL
- Operating system: ANY OS (Recommended: Windows8, Windows Vista, Windows XP)
- Application required: Standalone desktop application & Xampp application
- Coding language: PHP, HTML, CSS, Java-script

### **3.2 HARDWARE REQUIREMENTS**

- CPU: Pentium IV 2.4 GHz or above
- Memory (Primary): 512 MB, 1 GB or above
- Hard Disk: 40 GB, 80GB, 160GB or above
- Monitor: 15 VGA color

## **CHAPTER 4**

## **SYSTEM DESIGN**

This chapter of the report describes the structure of the project, followed by Entity Relationship Diagram and Schema-Diagram and the table structures.

The purpose of the design phase is to develop a clear understanding of what the developer wants people to gain from his/her project. As the developer works on the project. the test for every design decision should be efficient.

A purpose statement affects the design process by explaining what the developer wants the project to do, rather than describing the project itself. The Design Document will verify that the current design meets all of the explicit requirements contained in the system model as well as the implicit requirements desired by the customer.

System architecture section has:

System Architecture Design-The detailed diagram of the system, server and client.

Data Design-The data design includes an ER as well as Database design.

### **4.1 ENTITY RELATIONSHIP DIAGRAM**

An entity relationship model, also called an entity-relationship (ER) diagram, is a graphical representation of entities and their relationships to each other, typically used in computing in regards to the organization of data within databases or information systems. An entity is a piece of data-an object or concept about which data is stored.

The cardinality or fundamental principle of one data aspect with respect to another is a critical feature. The relationship of one to the other must be precise and exact between each other in-order to explain how each aspect links together. In simple words Cardinality is a way to define the relationship between two entities.

The ER diagram below shows the relationship between the many tables that exist in the database for the functioning of online examination system.

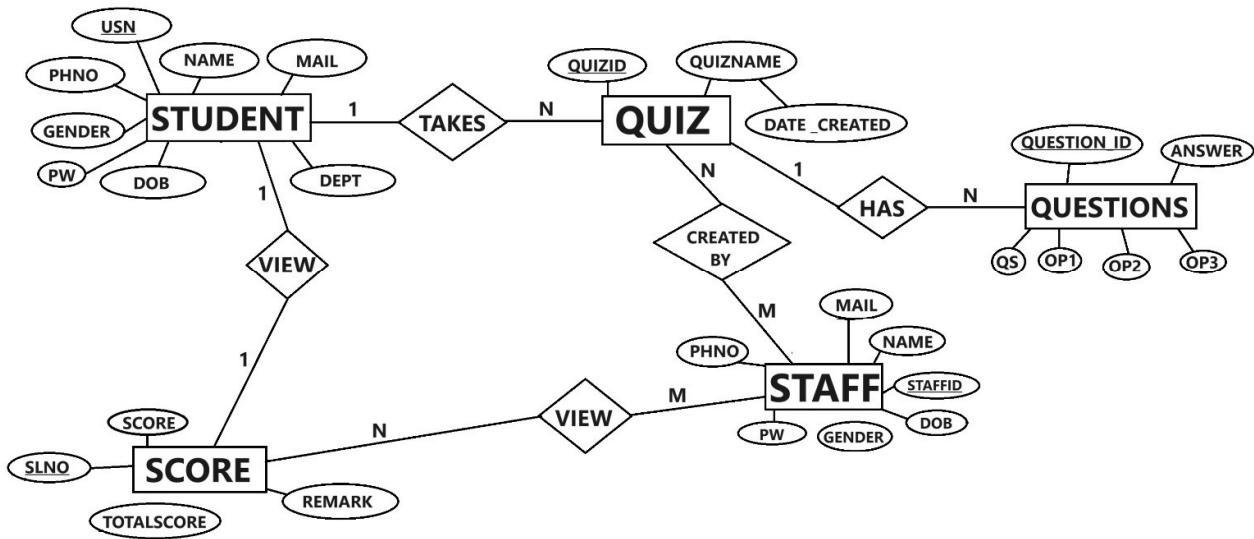


Fig 4.1: Entity-Relationship diagram

## 4.2 SCHEMA DIAGRAM

The Schema Diagram gives us the information about the attributes in the table of the database and how the given tables are related to each other.

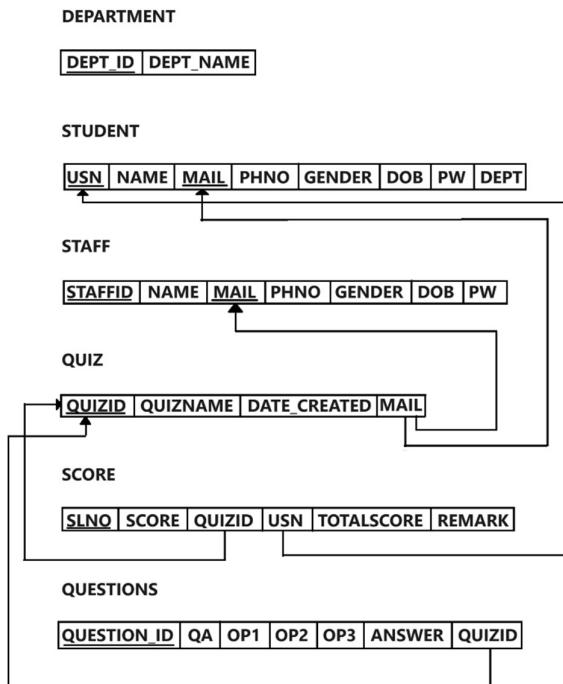


Fig 4.2: Schema diagram

The above diagrams explain:

## ENTITIES

- **Student:** This entity represents students in the system. It includes attributes such as "usn" (Unique Student Number), "name," "mail," "phno" (phone number), "gender," "DOB" (Date of Birth), "pw" (password), and "dept" (department). The "usn" attribute serves as the primary key (PK) for this entity.
- **Staff:** This entity represents staff members (teachers, administrators, etc.). It includes attributes such as "staffid" (Staff ID), "name," "mail," "phno," "gender," "DOB," and "pw." Similar to the student entity, the "staffid" attribute serves as the primary key.
- **Dept:** This entity represents departments within the organization or institution. It includes attributes such as "dept\_id" (Department ID) and "dept\_name." The "dept\_id" attribute serves as the primary key.
- **Quiz:** This entity represents quizzes or examinations conducted within the system. It includes attributes such as "quizid" (Quiz ID), "quizname" (Quiz Name), "date\_created," and "staffid" (the staff member responsible for creating the quiz). The "quizid" attribute serves as the primary key.
- **Score:** This entity represents scores or results achieved by students in quizzes. It includes attributes such as "slno" (Serial Number), "score," "quizid" (Quiz ID), "usn" (Student USN), "totalscore" (Total Score), and "remark" (e.g., 'good' or 'bad'). The "slno" attribute serves as the primary key.
- **questions:** This entity represents the questions that appear in quizzes. It includes attributes such as "question\_id" (Question ID), "qs" (Question), "op1" (Option 1), "op2" (Option 2), "op3" (Option 3), "answer" (Correct Answer), and "quizid" (Quiz ID). The "question\_id" attribute serves as the primary key.

## RELATIONSHIPS

- **Score (FK: quizid, usn):** The "score" entity has a foreign key relationship with both the "quiz" and "student" entities. This relationship indicates that each score entry is associated with a specific quiz (identified by "quizid") and a specific student (identified by "usn").
- **Questions (FK: quizid):** The "questions" entity has a foreign key relationship with the "quiz" entity. This relationship indicates that each question entry is associated with a specific quiz (identified by "quizid"), serves as the primary key.

### 4.3 BLOCK DIAGRAM

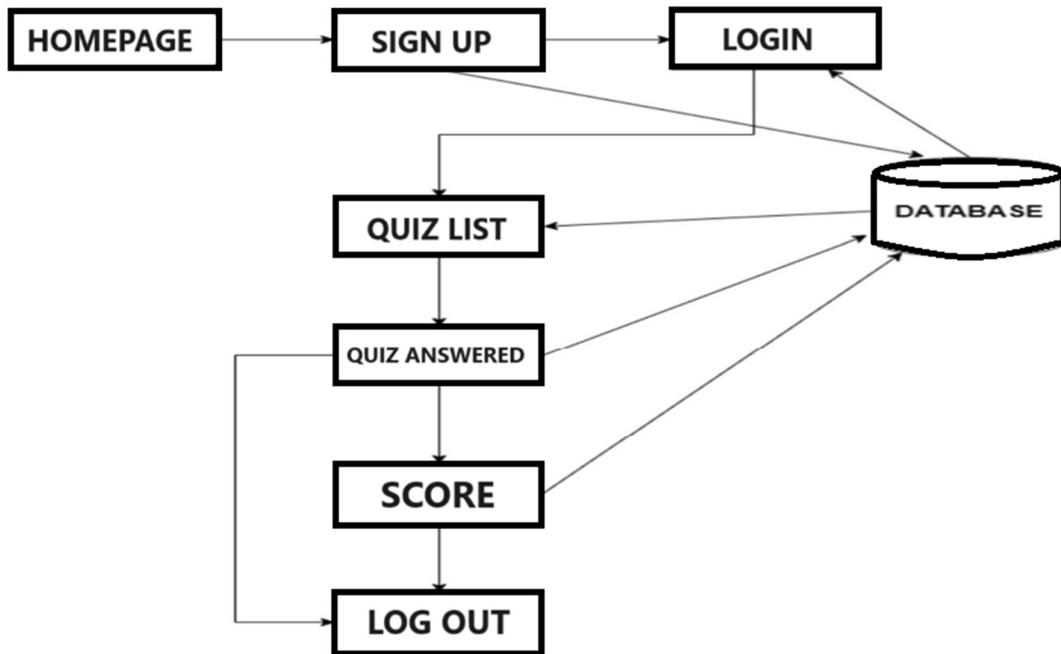


Fig 4.3: Block diagram

The below figure shows the block diagram of the Online exam application. The home page gives the options for registering students and staffs. Then the login page displays, where we the students can login and can take the quiz, they can view their scores and then logout. The teachers can login and create new quiz which can be answered by students

#### **4.4 TECHNOLOGY USED**

**HTML** - 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.2! Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages.

**CSS** - Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a mark-up language.!! Although most often used to set the visual style of web pages and user interfaces written in HIME and XHTML. The language can be applied to any XML document, including plain XML. SVG and XUL. and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript. CSS is a cornerstone technology used by most websites to create engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

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

**JAVASCRIPT** - JavaScript, often abbreviated as JS, is a high-level, dynamic, weakly typed, prototype-based, multi-paradigm, and interpreted programming language. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content production. It is used to make webpages interactive and provide online programs, including video games.

**SQL** - (Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). In comparison to older read/write APIs like ISAM or VSAM, SQL offers two main advantages: first, it introduced the concept of accessing many records with one single command; and second, it eliminates the need to specify how to reach a record, e.g. with or without an index.

**XAMPP** - XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and a web server -server application deployment purpose. Everything needed to set up is included in an (Apache), database (MariaDB), and scripting language extractable file.

# CHAPTER 5

## IMPLEMENTATION

### 5.1 DATABASE STRUCTURE

Mysql is a domain specific language used in programming and designed for managing data held in a relational database system (RDBMS). A website programmed with PHP can have pages that are password protected, a website with no programming cannot do this with other complex things. Standard PHP file extensions are: php, php3 or phtml, but a web server can be used to set up too many extensions. The PHP is a scripting-language the first step is to run and test the PhpMyAdmin. Now that we have run and tested PhpMyAdmin, the next step is running MySQL and creating a database and table which will hold information to be used by our database. Inorder to start MySQL, navigate to thexampp directory and run the mysql\_start.bat batch file. The XAMPP package contains an application called phpMyAdmin which allows developers to administer and maintain MySQL databases. We will be using phpMyAdmin to create a database and table, and enter test data. Before testing phpMyAdmin, make sure that both Apache and MySQL are running by opening their respective batch files. We first open Xampp then we run apache and then mysql .then we need to create database for our project to run in the admin of mysql. Then we need to type <http://localhost/phpmyadmin/index.php> into our web browser.

```
CREATE TABLE `dept`(  
  `dept_id` int(11) NOT NULL,  
  `dept_name` varchar(3) DEFAULT NULL  
)ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Table 5.1: Department Table

The screenshot shows the MySQL Workbench interface with the 'Table structure' tab selected. The table 'dept' has two columns: 'dept\_id' (int(11), primary key, not null) and 'dept\_name' (varchar(3), default NULL). The 'dept\_id' column has a key icon and a 'Change' button. The 'dept\_name' column has a 'Change' button. Action buttons for 'Drop' and 'More' are also visible.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	dept_id	int(11)			No	None			Change  Drop More
2	dept_name	varchar(3)	latin1_swedish_ci		Yes	NULL			Change  Drop More

```
CREATE TABLE `student` (
  `usn` varchar(10) NOT NULL,
  `name` varchar(20) NOT NULL,
  `mail` varchar(30) NOT NULL,
  `phno` varchar(10) NOT NULL,
  `gender` varchar(1) NOT NULL,
  `DOB` varchar(10) NOT NULL,
  `pw` varchar(200) NOT NULL,
  `dept` varchar(3) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Table 5.2: Student Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	usn	varchar(10)	latin1_swedish_ci		No	None		Change	Drop More
2	name	varchar(20)	latin1_swedish_ci		No	None		Change	Drop More
3	mail	varchar(30)	latin1_swedish_ci		No	None		Change	Drop More
4	phno	varchar(10)	latin1_swedish_ci		No	None		Change	Drop More
5	gender	varchar(1)	latin1_swedish_ci		No	None		Change	Drop More
6	DOB	varchar(10)	latin1_swedish_ci		No	None		Change	Drop More
7	pw	varchar(200)	latin1_swedish_ci		No	None		Change	Drop More
8	dept	varchar(3)	latin1_swedish_ci		Yes	NULL		Change	Drop More

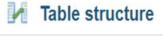
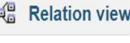
```
CREATE TABLE `staff` (
  `staffid` varchar(10) NOT NULL,
  `name` varchar(20) NOT NULL,
  `mail` varchar(30) NOT NULL,
  `phno` varchar(10) NOT NULL,
  `gender` varchar(1) NOT NULL,
  `DOB` varchar(10) NOT NULL,
  `pw` varchar(200) NOT NULL,
  `dept` varchar(3) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Table 5.3: Staff Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 <b>staffid</b> 	varchar(10)	latin1_swedish_ci		No	None		 Change	 Drop More
<input type="checkbox"/>	2 <b>name</b>	varchar(20)	latin1_swedish_ci		No	None		 Change	 Drop More
<input type="checkbox"/>	3 <b>mail</b> 	varchar(30)	latin1_swedish_ci		No	None		 Change	 Drop More
<input type="checkbox"/>	4 <b>phno</b> 	varchar(10)	latin1_swedish_ci		No	None		 Change	 Drop More
<input type="checkbox"/>	5 <b>gender</b>	varchar(1)	latin1_swedish_ci		No	None		 Change	 Drop More
<input type="checkbox"/>	6 <b>DOB</b>	varchar(10)	latin1_swedish_ci		No	None		 Change	 Drop More
<input type="checkbox"/>	7 <b>pw</b>	varchar(200)	latin1_swedish_ci		No	None		 Change	 Drop More
<input type="checkbox"/>	8 <b>dept</b>	varchar(3)	latin1_swedish_ci		Yes	NULL		 Change	 Drop More

```
CREATE TABLE `quiz` (
  `quizid` int(11) NOT NULL,
  `quizname` varchar(20) NOT NULL,
  `date_created` timestamp NOT NULL DEFAULT current_timestamp(),
  `staffid` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Table 5.4: Quiz Table

 Table structure     Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	
<input type="checkbox"/>	1 <b>quizid</b> 	int(11)			No	None		AUTO_INCREMENT	 Change	 Drop More
<input type="checkbox"/>	2 <b>quizname</b>	varchar(20)	latin1_swedish_ci		No	None			 Change	 Drop More
<input type="checkbox"/>	3 <b>date_created</b>	timestamp			No	current_timestamp()			 Change	 Drop More
<input type="checkbox"/>	4 <b>staffid</b>	varchar(30)	latin1_swedish_ci		No	None			 Change	 Drop More

```
CREATE TABLE `questions` (
  `qs` varchar(200) NOT NULL,
  `op1` varchar(30) NOT NULL,
  `op2` varchar(30) NOT NULL,
  `op3` varchar(30) NOT NULL,
  `answer` varchar(30) NOT NULL,
  `quizid` int(11) NOT NULL,
  `question_id` int(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Table 5.5: Questions Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	qs	varchar(200)	latin1_swedish_ci		No	None			Change  Drop  More
2	op1	varchar(30)	latin1_swedish_ci		No	None			Change  Drop  More
3	op2	varchar(30)	latin1_swedish_ci		No	None			Change  Drop  More
4	op3	varchar(30)	latin1_swedish_ci		No	None			Change  Drop  More
5	answer	varchar(30)	latin1_swedish_ci		No	None			Change  Drop  More
6	quizid	int(11)			No	None			Change  Drop  More
7	question_id	int(20)			No	None	AUTO_INCREMENT		Change  Drop  More

```
CREATE TABLE `score` (
  `slno` int(11) NOT NULL,
  `score` int(11) NOT NULL,
  `quizid` int(11) NOT NULL,
  `usn` varchar(30) NOT NULL,
  `totalscore` int(11) NOT NULL,
  `remark` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Table 5.6: Score Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	slno	int(11)			No	None	AUTO_INCREMENT		Change  Drop  More
2	score	int(11)			No	None			Change  Drop  More
3	quizid	int(11)			No	None			Change  Drop  More
4	usn	varchar(30)	latin1_swedish_ci		No	None			Change  Drop  More
5	totalscore	int(11)			No	None			Change  Drop  More
6	remark	varchar(20)	latin1_swedish_ci		No	None			Change  Drop  More

## 5.2 FRONT END

The front end of this project is designed using

- HTML
- CSS
- PHP

### 5.2.1 HTML

In a database management system (DBMS) project, HTML (Hypertext Markup Language) is often used for building the user interface or front end of the application. Here's a brief explanation of how HTML is used in a DBMS project:

- **User Interface Design:** HTML is used to design the layout and structure of the web pages that interact with the database. These web pages typically include forms, tables, buttons, and other elements that allow users to interact with the data stored in the database.
- **Form Creation:** HTML '`<form>`' elements are used to create input forms where users can submit data to be stored or updated in the database. Each form field corresponds to a database column, and when the form is submitted, the data is sent to the server-side application for processing.
- **Displaying Data:** HTML is used to display data retrieved from the database to the user. This can include displaying data in tables, lists, or other formats, depending on the requirements of the application.
- **Dynamic Content:** HTML can be combined with server-side scripting languages like PHP, Python, or Node.js to create dynamic web pages that interact with the database in real-time. For example, when a user submits a form, the server-side script can process the form data, perform database operations, and then dynamically generate an HTML response to display the results to the user.
- **Navigation:** HTML is used to create navigation menus, links, and buttons that allow users to move between different pages or sections of the application. These navigation elements can be used to perform actions such as viewing, editing, or deleting data from the database.

Overall, HTML plays a crucial role in a DBMS project by providing the structure and layout for the user interface, allowing users to interact with the database and view or manipulate data stored within it.

### 5.2.2 CSS

In a database management system (DBMS) project, CSS (Cascading Style Sheets) is used to enhance the visual presentation and layout of the HTML elements that make up the user interface. Here's a brief explanation of how CSS is used in a DBMS project:

- **Styling HTML Elements:** CSS is used to style the HTML elements generated by the DBMS project. This includes setting properties such as colors, fonts, margins, padding, borders, and backgrounds to create visually appealing and consistent designs across different web pages.

- **Layout Design:** CSS is used to define the layout of the web pages, including the positioning and alignment of elements. By using techniques such as Flexbox or Grid layout, developers can create responsive and flexible designs that adapt to different screen sizes and devices.
- **Typography:** CSS is used to style text elements within the DBMS project, including setting font styles, sizes, weights, line heights, and text alignments. This helps to improve readability and enhance the overall aesthetic appeal of the user interface.
- **Responsive Design:** CSS media queries are used to create responsive designs that adapt to different viewport sizes, such as those of desktops, tablets, and smartphones. This ensures that the DBMS project is accessible and usable across a wide range of devices and screen resolutions.
- **Customization:** CSS allows developers to customize the appearance of various UI components, such as buttons, forms, tables, and navigation menus. This customization can help to create a unique and branded look for the DBMS project that reflects the organization's identity and design preferences.
- **Theme Management:** CSS can be used to implement theming functionality, allowing users to choose from different visual themes or stylesheets to personalize their experience within the DBMS application.

Overall, CSS plays a crucial role in a DBMS project by providing the means to style and customize the user interface, thereby enhancing its visual appeal, usability, and responsiveness.

### 5.2.2 PHP

In a database management system (DBMS) project, PHP (Hypertext Preprocessor) is commonly used as a server-side scripting language to interact with the database and dynamically generate HTML content. Here's a brief explanation of how PHP is used in a DBMS project:

- **Database Connectivity:** PHP is used to establish connections to the database management system, such as MySQL, PostgreSQL, or SQLite. It provides functions and APIs (Application Programming Interfaces) to connect to the database server, authenticate users, and select a specific database for operations.
- **Query Execution:** PHP is used to execute SQL (Structured Query Language) queries against the database. These queries can include operations such as SELECT, INSERT, UPDATE, DELETE, and JOIN to retrieve, insert, modify, or delete data from the database tables based on user interactions.
- **Data Manipulation:** PHP is used to process the data retrieved from the database and manipulate it according to the application's requirements. This may involve formatting the data, performing

calculations, applying business logic, or transforming it into different data structures before rendering it in the user interface.

- **Dynamic Content Generation:** PHP is used to dynamically generate HTML content based on the data retrieved from the database. This includes populating web pages with dynamic data, generating lists or tables of records, displaying user-specific information, and rendering interactive forms for data input.
- **User Authentication and Authorization:** PHP is used to implement user authentication and authorization mechanisms within the DBMS project. This involves verifying user credentials, managing user sessions, enforcing access control policies, and restricting access to certain parts of the application based on user roles or permissions stored in the database.
- **Form Handling:** PHP is used to handle form submissions from the user interface. It processes form data submitted via HTTP POST or GET requests, validates the input against predefined rules or constraints, sanitizes it to prevent security vulnerabilities such as SQL injection or cross-site scripting (XSS), and inserts or updates the data in the database accordingly.
- **Error Handling and Logging:** PHP is used to handle errors and exceptions that occur during the execution of database operations. It provides mechanisms for error handling, logging, and debugging to help developers identify and troubleshoot issues in the application code, database queries, or server configuration.

Overall, PHP serves as the backbone of many DBMS projects, enabling developers to build dynamic and interactive web applications that interact seamlessly with the underlying database system, process user requests, and deliver personalized content to users based on their interactions and data stored in the database.

## 5.2 BACK END

The front end of this project is designed using

- MYSQL
- XAMPP APPLICATION

### 5.3.1 MYSQL

In MySQL is a powerful and widely used open-source relational database management system (RDBMS), ranking as the second most popular globally. Its name comes from "My" (the daughter of one of its co-founders) and "SQL" for Structured Query Language, which is the standard language used to manage and manipulate databases.

Originally developed by MySQL AB, a Swedish company, MySQL's source code became available under the GNU General Public License (GPL) and also under various proprietary agreements. In 2008, MySQL AB was acquired by Sun Microsystems, which was later acquired by Oracle Corporation in 2010. MySQL remains under active development, with contributions from the open-source community and Oracle.

MySQL is particularly favored for web applications and is a key component of the LAMP stack (Linux, Apache, MySQL, PHP/Python/Perl), a popular open-source software bundle for web development. It's known for its reliability, scalability, and performance, making it suitable for a wide range of applications, from small websites to large-scale enterprise systems.

For commercial use, MySQL offers additional functionality and support through various licensing options provided by Oracle Corporation.

Overall, MySQL's versatility, ease of use, and robust feature set have made it a top choice for developers and businesses worldwide, powering countless web applications and other projects that require a reliable and full-featured database management system.

### 5.3.2 XAMPP APPLICATION

XAMPP is a popular software package that provides a convenient and comprehensive environment for web development. It combines several key technologies necessary for web development into a single, easy-to-install package. The main components included in XAMPP are:

- **Apache HTTP Server:** Apache is one of the most widely used web servers in the world. It serves as the foundation for hosting websites and web applications.
- **MySQL:** MySQL is a powerful relational database management system (RDBMS) that allows for the storage and retrieval of data in a structured manner. It's commonly used in conjunction with web applications to manage backend databases.
- **PHP:** PHP is a server-side scripting language that is widely used for web development. It allows developers to create dynamic and interactive web pages by embedding PHP code within HTML.
- **phpMyAdmin:** phpMyAdmin is a web-based application written in PHP that provides a graphical user interface (GUI) for administering MySQL databases. It allows users to perform tasks such as database creation, deletion, querying, and management through a web browser.
- **SQLite:** SQLite is a lightweight, serverless database engine that allows for local storage of data within applications. It's often used for development and testing purposes, particularly in situations where a full-fledged RDBMS like MySQL is not required.

XAMPP is available in two main packages: full and lite. The full package includes a wide array of development tools, making it suitable for a broad range of web development projects. On the other hand, XAMPP Lite is a smaller package that contains only the essential technologies required for web development, meeting the standards set by organizations like the Ontario Skills Competition. Overall, XAMPP is an excellent choice for students, developers, and anyone else looking to set up a local web development environment quickly and easily. Its simplicity, portability, and inclusion of essential web technologies make it a valuable tool for building and testing PHP and MySQL-based applications.

# CHAPTER 6

## SNAPSHOTS

### 6.1 HOME PAGE

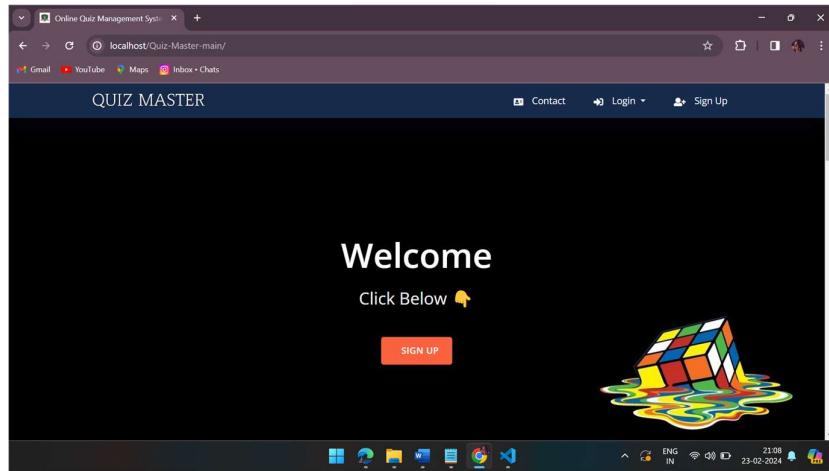
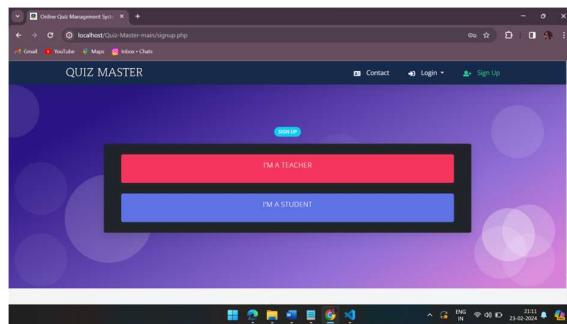


Fig 6.1: Home page

Fig 6.1 Shows the Home page contains navigation bar, through which can navigate to other pages. It also contains some details about the website at the home page.

### 6.2 SIGN UP PAGES

Fig 6.2 Shows the Sign-up page contains two options, through which students and staffs can sign up. It asks the user to enter the details like full name, usn, email, phone number, password, department, dob and gender.



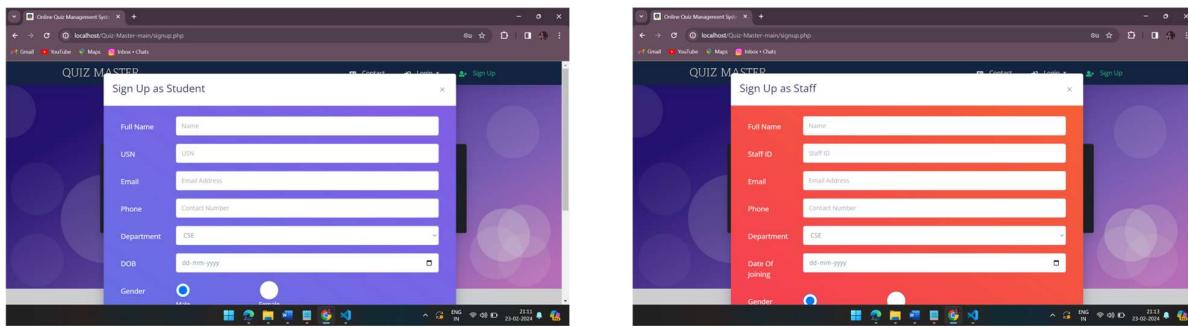


Fig 6.2: Sign up pages

### 6.3 LOGIN PAGES

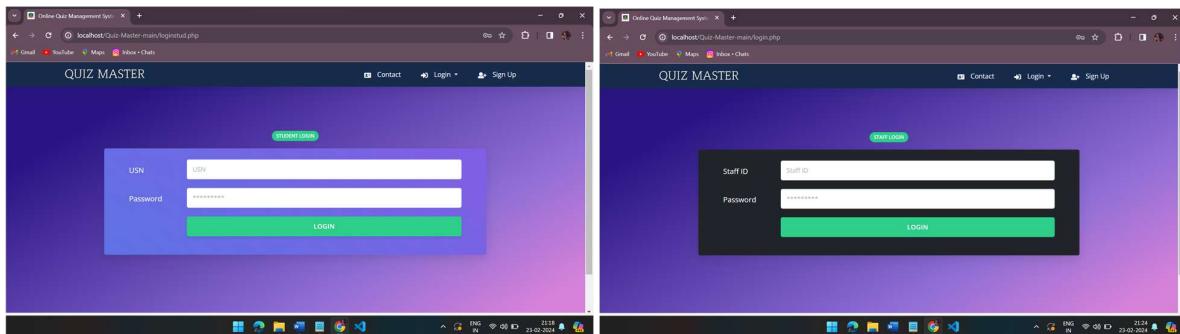


Fig 6.3: Login page

Fig 6.3 Shows the Login page contains two options, through which students and staffs can login to quiz website. It asks the user to enter the email and password in order to enter the website.

### 6.4 QUIZ LIST PAGE

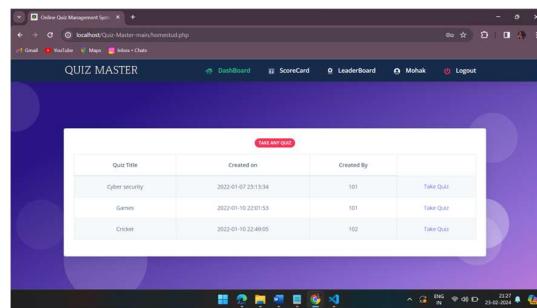


Fig 6.4: Quiz List page

Fig 6.4 Shows the Quiz list page displays the list of quiz and their titles. It helps the students to take the respective quiz.

## 6.5 QUESTIONS PAGE

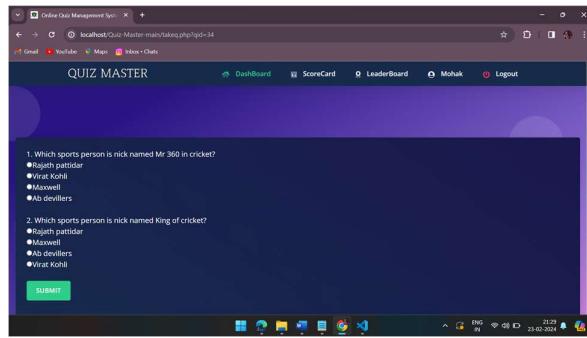


Fig 6.5: Questions page

Fig 6.5 Shows the questions page displays the list of questions which can be answered by the students.

## 6.6 SCORE PAGE

SCORE CARD			
Quiz Title	Score Obtained	Total Questions	Remarks
Cyber security	1	5	good
Games	1	1	good

Fig 6.6: Score page

Fig 6.6 Shows the Score page displays the score of the quiz questions answered.

## 6.7 QUIZ ADDING PAGE

The screenshots show the 'Add Quiz' section of the application. The first screenshot shows the initial form with a 'Quiz Name' input field containing 'Enter Quiz Name' and a 'SUBMIT' button. The second screenshot shows the form after a question has been added. The 'Question' field contains 'What is the capital of India?'. Below it are four 'Option' fields, each containing 'Delhi'. At the bottom right is a 'SUBMIT' button.

Fig 6.7: Quiz adding page

Fig 6.7 Shows the Quiz adding page gives an options for the staffs to add the quiz and questions which can be answered by students.

## 6.8 PROFILE PAGES

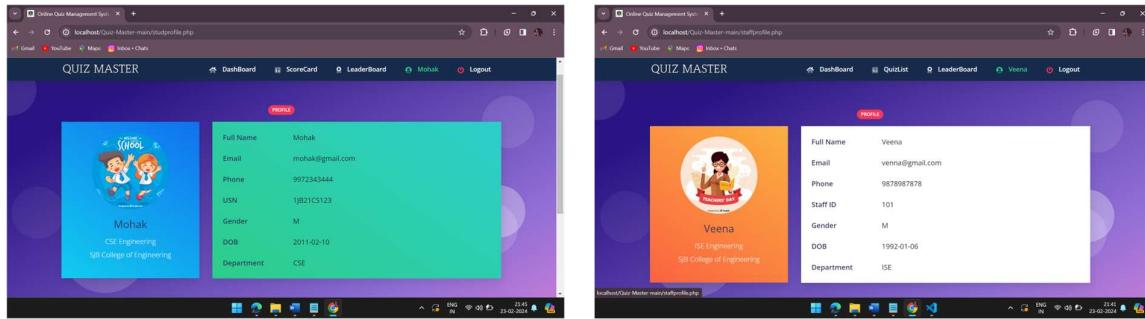


Fig 6.8: Profile Pages

Fig 6.8 Shows the profile page displays the profile of students and staffs. Its contains details like name, email, phone, staffed, usn, gender, dob, department.

## 6.9 CONTACT PAGE

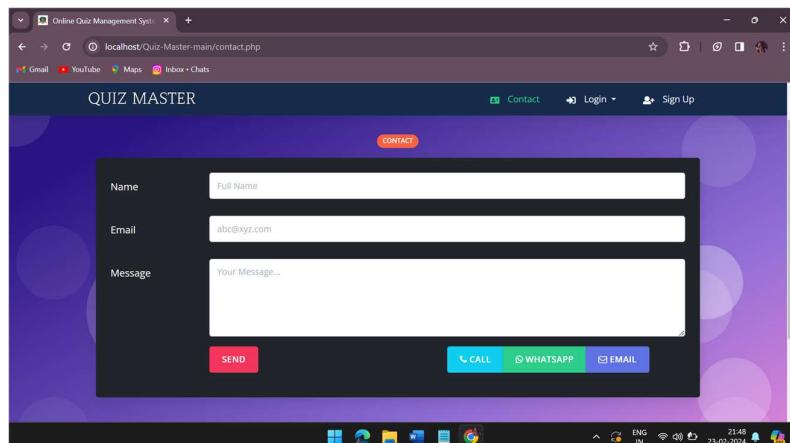


Fig 6.9: Contact page

Fig 6.9 Shows the contact page helps the users to contact the administrator through call, whatsapp or email.

## **CONCLUSION**

The development of the online examination system within the scope of this DBMS project marks a pivotal step towards revolutionizing the traditional examination process. By harnessing the power of database management systems alongside PHP, HTML, CSS, JavaScript, and MySQL, the system offers a comprehensive solution for conducting examinations in a digital environment.

The system's core functionalities, which include quiz management, automated grading, and the ability to add/delete quizzes and questions, significantly streamline the examination workflow while minimizing manual intervention and paperwork. However, there remains ample room for future enhancements to further elevate the system's functionality and user experience.

Potential avenues for improvement encompass the implementation of a timed quiz feature to simulate real exam conditions, the integration of email notifications for user registrations and exam submissions to enhance communication, support for a wider variety of question types including multiple-choice questions (MCQs), and the incorporation of programming questions that enable users to execute and evaluate code directly within the system.

The online examination system developed in this project not only modernizes the examination process but also lays the groundwork for future innovations in the realm of digital assessment. With ongoing refinements and updates, the system holds the potential to become a cornerstone tool for educational institutions seeking to streamline and enhance their examination procedures in the digital age.

## **REFERENCES**

- [1]. Date, C.J. (2003). An Introduction to Database Systems (8th ed.). Pearson Education.
- [2]. Silberschatz, A., Korth, H.F., & Sudarshan, S. (2010). Database System Concepts (6th ed.). McGraw-Hill.
- [3]. W3Schools. (n.d.). HTML Tutorial. <https://www.w3schools.com/html/>
- [4]. W3Schools. (n.d.). CSS Tutorial. <https://www.w3schools.com/css/>
- [5]. W3Schools. (n.d.). JavaScript Tutorial. <https://www.w3schools.com/js/>
- [6]. PHP Manual. (n.d.). PHP: Hypertext Preprocessor. <https://www.php.net/manual/en/>
- [7]. Kumar, A. (2015). Development of Online Examination System. International Journal of Computer Applications, 118(4), 1-4.
- [8]. Singh, S., & Arora, P. (2016). Design and Implementation of Online Examination System Using Web Services. International Journal of Computer Applications, 139(14), 1-5.
- [9]. DuBois, P. (2008). MySQL (5th ed.). Addison-Wesley.
- [10]. <https://dev.mysql.com/doc/refman/8.0/en/>