VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590010

DBMS MINI PROJECT REPORT

ON

"**ONLINE EXAMINATION SYSTEM**"

Submitted in partial fulfilment for the requirements for the fifth semester

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

For the Academic Year 2022-2023

Submitted by:

SRIPAVAN B	**1MV20CS135**
NUMAIR SHAIKH	**1MV20CS137**

Under the guidance of:

Dr. Suma Swamy

Professor, Department of CSE

Sir M. Visvesvaraya Institute of Technology, Bengaluru

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY

HUNASAMARANAHALLI, BENGALURU-562157

SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY

Krishnadevaraya Nagar, International Airport Road,
Hunasmaranahalli, Bengaluru – 562157

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

It is certified that the **DBMS Mini Project work** entitled " **ONLINE EXAMINATION SYSTEM** " is carried out by **SRIPAVAN B (1MV20CS135)** and **NUMAIR SHAIKH (1MV20CS137)** bonafide students of **Sir M Visvesvaraya Institute of Technology** in partial fulfilment for the 5th semester for the award of the Degree of Bachelor of Engineering in Computer Science and Engineering of the **Visvesvaraya Technological University, Belagavi** during the academic year **2022-2023**. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the course of Bachelor of Engineering.

Name & Signature of Guide	Name & Signature of HOD	Name & Signature of Principal
Dr. Suma Swamy Prof & Internal Guide Dept. Of CSE, Sir MVIT Bengaluru - 562157	**Dr. G. C. Bhanu Prakash** HOD, Dept. Of CSE, Sir MVIT Bengaluru - 562157	**Dr. Rakesh S G** Principal, Sir MVIT Bengaluru – 562157

External Examination:

Name of Examiner	Signature with Date
1)	
2)	

DECLARATION

We hereby declare that the entire project work embodied in this dissertation has been carried out by us and no part has been submitted for any degree or diploma of any institution previously.

Place: Bengaluru

Date:

Signature of Students:

SRIPAVAN B (1MV20CS135)	NUMAIR SHAIKH (1MV20CS137)

ACKNOWLEDGMENT

It gives us immense pleasure to express our sincere gratitude to the management of **Sir M. Visvesvaraya Institute of Technology,** Bengaluru for providing the opportunity and the resources to accomplish our project work in their premises.

On the path of learning, the presence of an experienced guide is indispensable and we would like to thank our guide **Dr Suma Swamy,** Professor, Dept. of CSE, for her invaluable help and guidance.

Heartfelt and sincere thanks to **Dr. G. C. Bhanu Prakash,** HOD, Dept. of CSE, for his suggestions, constant support and encouragement.

We would also like to convey our regards to **Dr. Rakesh S G,** Principal, Sir MVIT for providing us with the infrastructure and facilities needed to develop our project.

We would also like to thank the staff of Department of Computer Science and Engineering and lab-in-charges for their co-operation and suggestions. Finally, we would like to thank all our friends for their help and suggestions without which completing this project would not have been possible.

- SRIPAVAN B 1MV20CS135
- NUMAIR SHAIKH 1MV20CS137

ABSTRACT

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

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

The project assesses students by conducting online objective tests. The tests would be highly customizable, this project will enable educational institutions to conduct test and have automated checking of answers based on the response by the candidates.

The project allows faculties to create their own tests. It would enable educational institutes to perform tests, quiz and create feedback forms. It asks faculty to create his/her set of questions. Faculty then creates groups and adds related students into the groups. Further the tests are associated with specific groups so that only associated students can appear for the test. The result of the response would be available to the faculty of the question set. This project would be helpful for creating practice tests, say for educational institutes and as a feedback form.

1.2 Limitation of Existing Examination System

We find that existing system is manual entry and keeping of the details of the student who are registered already. And it is very difficult for each student to come to the exam centre. It is very difficult to the students from far distance to reach the exam centre. This system is required to prepare registration\application form, question paper for the students and required to print a lot of number manually. To calculate how many students registered, and verification of details of these students in a month by hand is very difficult. This requires quite a lot of time and wastage of money as it requires quite lot of manpower to do that. Another factor that takes into account that is the possibility of errors. The limitation of existing system is that it is not all personalized. It cannot be used for personal and quick reference. Even the other staff members can make quick entries if the responsible person is not present.

- Time Consuming for creating question paper
- Time to check right and wrong answers
- Calculation of Marks
- Human error
- Limitation of no of student can give examination at a time
- Require teacher to monitor exam centre

- Student needs to come exam centre for giving test

1.3 Proposed System and its Advantages

The modern computerized system is developed with the aim to overcome the drawbacks of existing manual system. We have study manual examination system of ABC college and identify possible automation. The proposed system has got many advantages. People from different parts of the world can register very easily. The new system is more personalized. It is maze in such a manner that all the new users can understand all the options in it very easily. It is made in a quick and easy referential manner. Access to all important matters are not always locked and can be opened easily at the time of urgency. The advantages of proposed system are that security is maintained in the new system. Securities for all important data are maintained confidentially. As it is easily understandable and user friendly, quick entries can be made in this system

- Provides complete online web-based solution, including student registration, giving tests, storing of results.
- Complete web-based administration, administrator can manage examination and question bank from web interface.
- No geographical boundary
- Student can give examination from anywhere of the world by 24X7
- 100% accuracy in result calculation
- Randomization of question set

CHAPTER 2

IMPLEMENTATIONS

2.1 PHP (Back-End)

PHP is Hypertext Pre-processor is a general-purpose programming language originally designed for web development.

PHP is an intuitive, server side scripting language. Like any other scripting language it allows developers to build logic into the creation of web page content and handle data returned from a web browser. PHP also contains a number of extensions that make it easy to interact with databases, extracting data to be displayed on a web page and storing information entered by a web site visitor back into the database.

PHP consists of a scripting language and an interpreter. Like other scripting languages, PHP enables web developers to define the behavior and logic they need in a web page. These scripts are embedded into the HTML documents that are served by the web server. The interpreter takes the form of a module that integrates into the web server, converting the scripts into commands the computer then executes to achieve the results defined in the script by the web developer.

2.2 HTML (Front-End)

Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript.

2.2.1 CSS

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. Functional Modules

2.2.2 JavaScript

JavaScript, often abbreviated as JS, is a high-level, interpreted scripting language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

PHPMailer

PHPMailer is a code library to send emails safely and easily via PHP code from a web server. Sending emails directly by PHP code requires a high-level familiarity to SMTP standard protocol and related issues and vulnerabilities about Email injection for spamming.

CHAPTER 3

SPECIFICATIONS

The program works on Desktop PC and is executed using a PHP 5 interface which interacts with a MySQL database running on localhost.

3.1 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of 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

3.2 SOFTWARE REQUIREMENTS

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

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

3.2.1 FUNCTIONAL REQUIREMENTS

A description of the facility or feature required. Functional requirements deal with what the system should do or provide for users. They include description of the required functions, outlines of associated reports or online queries, and details of data to be held in the system.

Interface Requirements:

- The system shall provide an option to add/delete quizzes with questions.
- The system shall provide an option to see and attend the quizzes.
- The system should give option for login for staffs and students.
- The system shall provide option to see scores.

3.2.2 NON-FUNCTIONAL REQUIREMENTS:

Non-functional requirements define the overall qualities or attributes of the resulting system.

Usability

Usability is the ease with which a user can learn to operate the online examination system and get results.

Security

Security requirements are included in a system to ensure:

- All questions and users are well secured
- SQL injection is prevented

Reliability

Reliability is the ability of a system to perform its required functions under stated conditions for a specific period of time. Constraints on the run-time behavior of the system can be considered under two separate headings:

- Availability: is the system available for service when requested by end-users.
- Failure rate: how often does the system fail to deliver the service as expected by end-users.

CHAPTER 4

SYSTEM DESIGN

4.1 UML DIAGRAM

UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behaviour of artifacts of software systems.

UML can be applied to diverse application domains (e.g., banking, finance, internet, aerospace, healthcare, etc.) It can be used with all major object and component software development methods and for various implementation platforms (e.g., J2EE, .NET).

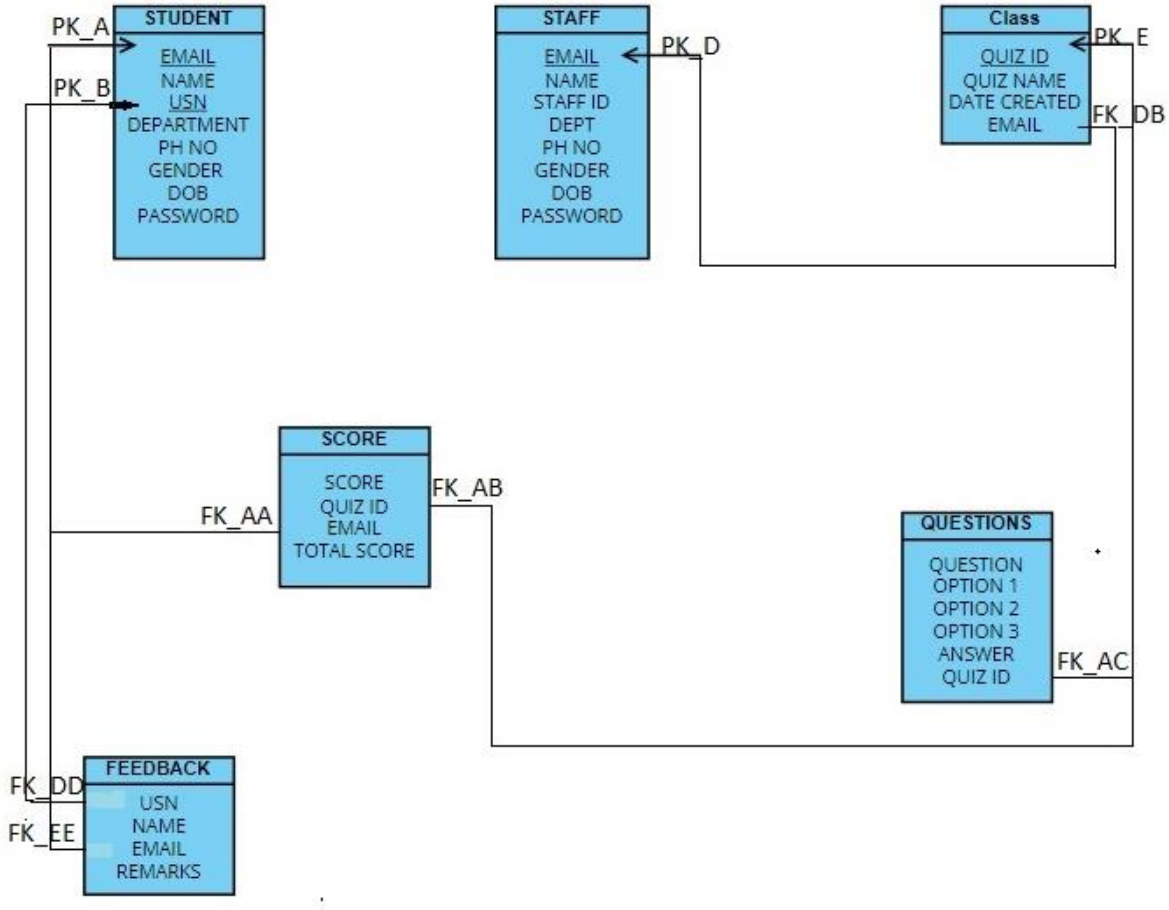


Figure 4.1;UML Diagram

4.2 ENTITY RELATIONSHIP DIAGRAM

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships. At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure.

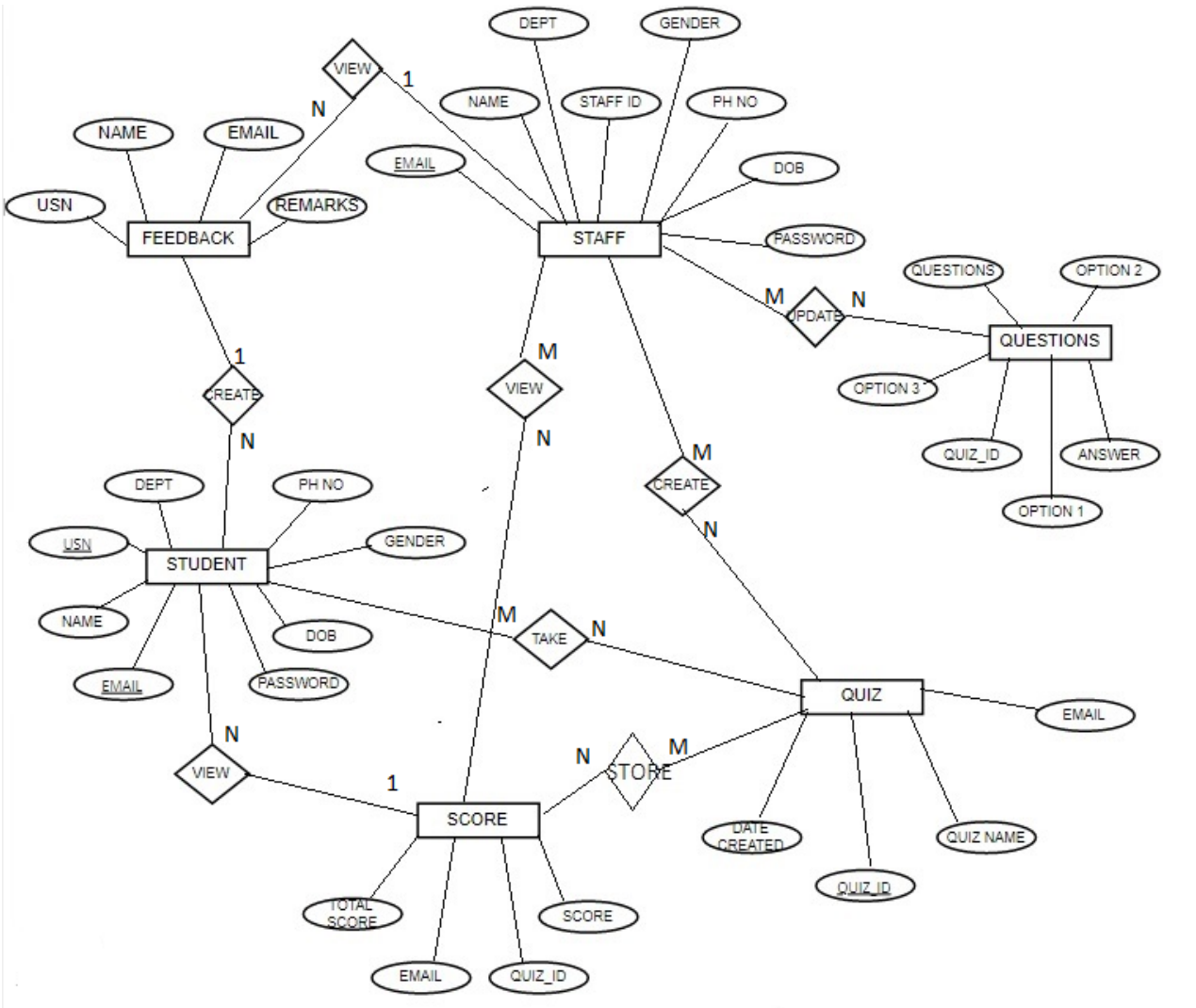


Figure 4.2:ER Diagram

Tool Used – Visual Paradigm

CARDINALITY RATIOS

- Staff and Quiz (Many to Many)
- Staff and Questions (Many to Many)
- Staff and Score (Many to Many)
- Student and Quiz (Many to Many)
- Student and Score (Many to One)
- Student and Feedback (Many to One)
- Quiz and Score (Many to Many)

CHAPTER 5

BACKEND DESIGN AND IMPLEMENTATION

5.1 Backend Used

PHP is the most popular scripting language for web development. It is free, open source and server-side (the code is executed on the server). XAMPP is the most popular PHP development environment.

MySQL is a Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). It is also free and open source. The combination of PHP and MySQL gives unmet options to create just about any kind of website - from small contact form to large corporate portal. MySQL database server offers several advantages:

- MySQL is easy to use, yet extremely powerful, fast, secure, and scalable.
- MySQL runs on a wide range of operating systems, including UNIX or Linux, Microsoft Windows, Apple Mac OS X, and others.
- MySQL supports standard SQL (Structured Query Language).
- MySQL is ideal database solution for both small and large applications.
- MySQL is developed, and distributed by Oracle Corporation.
- MySQL includes data security layers that protect sensitive data from intruders.

MySQL database stores data into tables like other relational database. A table is a collection of related data, and it is divided into rows and columns.

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.

5.2 Description of Tables

The database consists of 6 tables:

1. Student: It stores the student details.
    - EMAIL: Email ID of the student. It takes the datatype varchar with size 10 bytes. It is a Primary key.
    - Name: Name of the student. It takes the datatype varchar with size 20 bytes.
    - USN: University Seat Number of the student. It takes the datatype varchar with size 30 bytes. It is a Unique key.
    - Department: Department to which student belongs to. It takes the datatype varchar with size 3 bytes. It's a foreign key.
    - PhNo: Phone Number of the student. It takes the datatype bigint with size 10 bytes. It is a Unique key.
    - Gender: Gender of the student. It takes the datatype char with size 1 byte.
    - DOB: Date of Birth of the student. It takes the datatype date.
    - Password: Password used by the student. It takes the datatype varchar with size 200 bytes.

2. Staff: It stores the staff details.
    - EMAIL: Email ID of the staff member. It takes the datatype varchar with size 30 bytes.
    - Name: Name of the staff. It takes the datatype varchar with size 20 bytes.
    - STAFF ID: It stores the unique ID of the staff. It takes the datatype varchar with size 10 bytes.
    - Department: Department to which staff belongs to. It takes the datatype varchar with size 3 bytes.
    - PhNo: Phone Number of the staff. It takes the datatype bigint with size 10 bytes.
    - Gender: Gender of the staff. It takes the datatype char with size 1 byte.
    - DOB: Date of Birth of the staff. It takes the datatype date.

- Password: Password used by the staff. It takes the datatype varchar with size 200 bytes.

3. Quiz: Stores the quiz details.
    - QUIZ ID: Unique number given to each quiz. It takes the datatype integer. It's a primary key.
    - Quiz Name: The name of the quiz. It takes the datatype varchar with size 20 bytes.
    - Date Created: The date on which the quiz was created. It takes the datatype date.
    - EMAIL: Mail ID of the staff that created the quiz. It takes the datatype varchar with size 30 bytes. It's a foreign key.

4. Questions: Stores the details of questions and its answers in a quiz.
    - Question: The question present in a quiz. It takes the datatype varchar with size 200 bytes. It's a primary key.
    - Oprion1: First option for the question. It takes the datatype varchar with size 30 bytes.
    - Option2: Second option for the question. It takes the datatype varchar with size 30 bytes.
    - Option3: Third option for the question. It takes the datatype varchar with size 30 bytes.
    - Answer: The answer for the options provided. It takes the datatype varchar with size 30 bytes.
    - Quiz ID: Unique ID of the quiz. It takes the datatype integer. It's a foreign key.

5. Score: Stores the results of the students.
    - Score: Score of the student for a particular quiz. It takes the datatype integer.

- Quiz ID: Id number of the quiz. It takes the datatype integer. It's a foreign key.
- EMAIL: Mail ID of the student. It takes the datatype varchar with size 30 bytes. It's a foreign key.
- Total score: Final score of the student. It takes the datatype integer.

6. Feedback: Stores the feedbacks provided by the student.
    - USN: USN of the student. It takes the datatype varchar with size 10 bytes. It's a foreign key.
    - Name: Name of the student. It takes the datatype varchar with size 20 bytes.
    - Email: Mail ID of the student. It takes the datatype varchar with size 30 bytes. It's a foreign key.
    - Remarks: Feedback given by the student. It takes the datatype varchar with size 100 bytes.

5.2.1 Create Commands

The create commands for each table are as follows

Table Student: -

```
CREATE TABLE IF NOT EXISTS `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,
  PRIMARY KEY (`mail`),
  UNIQUE KEY `mail` (`mail`),
  UNIQUE KEY `phno` (`phno`),
  UNIQUE KEY `usn` (`usn`),
  KEY `dept` (`dept`)
) ;
```

Figure 5.1: Creation of student

Table Staff: -

```
CREATE TABLE IF NOT EXISTS `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,
  PRIMARY KEY (`mail`),
  UNIQUE KEY `mail` (`mail`,`phno`),
  UNIQUE KEY `staffid` (`staffid`)
) ;
```

Figure 5.2:Creation of staff

Table Quiz: -

```
CREATE TABLE IF NOT EXISTS `quiz` (
  `quizid` int(11) NOT NULL AUTO_INCREMENT,
  `quizname` varchar(20) NOT NULL,
  `date_created` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `mail` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`quizid`),
  KEY `mail` (`mail`)
) ;
```

Figure 5.3: Creation of quiz

Table Questions: -

```
CREATE TABLE IF NOT EXISTS `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,
  UNIQUE KEY `qs` (`qs`),
  KEY `quizid` (`quizid`),
  KEY `quizid_2` (`quizid`),
  KEY `quizid_3` (`quizid`)
) ;
```

Figure 5.4: Creation of questions

Table Score: -

```
CREATE TABLE IF NOT EXISTS `score` (
  `slno` int(11) NOT NULL AUTO_INCREMENT,
  `score` int(11) NOT NULL,
  `quizid` int(11) NOT NULL,
  `mail` varchar(30) DEFAULT NULL,
  `totalscore` int(11) DEFAULT NULL,
  `remark` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`slno`),
  KEY `quizid` (`quizid`),
  KEY `mail` (`mail`)
) ;
```

Figure 5.5: Creation of score

Table Feedback: -

```
CREATE TABLE IF NOT EXISTS `feedback` (
  `usn` varchar(10) NOT NULL,
  `name` varchar(15) DEFAULT NULL,
  `email` varchar(20) NOT NULL,
  `remarks` varchar(100) NOT NULL,
  PRIMARY KEY (`usn`)
);
```

Figure 5.6: Creation of feedback

5.2.2 Describe Commands

DESCRIBE STUDENT;

```
mysql> describe student;
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| usn    | varchar(10)  | NO   | UNI | NULL    |       |
| name   | varchar(20)  | NO   |     | NULL    |       |
| mail   | varchar(30)  | NO   | PRI | NULL    |       |
| phno   | varchar(10)  | NO   | UNI | NULL    |       |
| gender | varchar(1)   | NO   |     | NULL    |       |
| DOB    | varchar(10)  | NO   |     | NULL    |       |
| pw     | varchar(200) | NO   |     | NULL    |       |
| dept   | varchar(3)   | YES  | MUL | NULL    |       |
+--------+--------------+------+-----+---------+-------+
8 rows in set (0.04 sec)
```

Figure 5.7:Description of student

DESCRIBE STAFF;

```
mysql> describe staff;
+---------+--------------+------+-----+---------+-------+
| Field   | Type         | Null | Key | Default | Extra |
+---------+--------------+------+-----+---------+-------+
| staffid | varchar(10)  | NO   | UNI | NULL    |       |
| name    | varchar(20)  | NO   |     | NULL    |       |
| mail    | varchar(30)  | NO   | PRI | NULL    |       |
| phno    | varchar(10)  | NO   |     | NULL    |       |
| gender  | varchar(1)   | NO   |     | NULL    |       |
| DOB     | varchar(10)  | NO   |     | NULL    |       |
| pw      | varchar(200) | NO   |     | NULL    |       |
| dept    | varchar(3)   | YES  |     | NULL    |       |
+---------+--------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

Figure 5.8:Description of staff

DESCRIBE QUIZ;

```
mysql> describe quiz;
+--------------+-------------+------+-----+-------------------+-------------------+
| Field        | Type        | Null | Key | Default           | Extra             |
+--------------+-------------+------+-----+-------------------+-------------------+
| quizid       | int         | NO   | PRI | NULL              | auto_increment    |
| quizname     | varchar(20) | NO   |     | NULL              |                   |
| date_created | timestamp   | NO   |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| mail         | varchar(30) | YES  | MUL | NULL              |                   |
+--------------+-------------+------+-----+-------------------+-------------------+
4 rows in set (0.00 sec)
```

Figure 5.9:Description of quiz

DESCRIBE QUESTIONS;

```
mysql> describe questions;
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| qs     | varchar(200) | NO   | PRI | NULL    |       |
| op1    | varchar(30)  | NO   |     | NULL    |       |
| op2    | varchar(30)  | NO   |     | NULL    |       |
| op3    | varchar(30)  | NO   |     | NULL    |       |
| answer | varchar(30)  | NO   |     | NULL    |       |
| quizid | int          | NO   | MUL | NULL    |       |
+--------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

Figure 5.10:Description of questions

DESCRIBE SCORE;

```
mysql> describe score;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| slno       | int         | NO   | PRI | NULL    | auto_increment |
| score      | int         | NO   |     | NULL    |                |
| quizid     | int         | NO   | MUL | NULL    |                |
| mail       | varchar(30) | YES  | MUL | NULL    |                |
| totalscore | int         | YES  |     | NULL    |                |
| remark     | varchar(20) | YES  |     | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

Figure 5.11:Description of score

DESCRIBE FEEDBACK;

```
mysql> describe feedback;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| usn      | varchar(10)  | YES  | MUL | NULL    |       |
| name     | varchar(20)  | YES  |     | NULL    |       |
| mail     | varchar(30)  | YES  | MUL | NULL    |       |
| feedback | varchar(100) | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

Figure 5.12:Description of feedback

5.2.3 Insert Commands

Student:

INSERT INTO `student` (`usn`, `name`, `mail`, `phno`, `gender`, `DOB`, `pw`, `dept`) VALUES ('1MV17CS010', 'B G VINAYAK', 'BHATVINAYAK94@GMAIL.COM', '9740834260', 'M', '1999-09-23', 'ral7gku4rfhLk', 'CSE');

Select * from student;

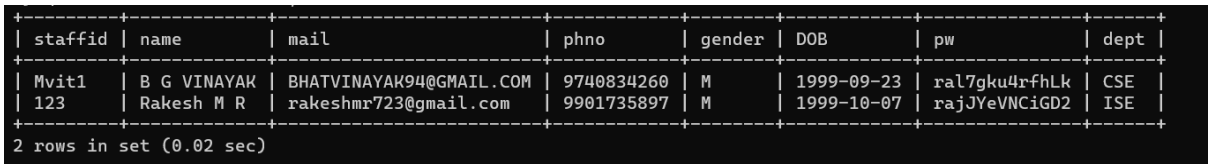
```
+------------+-------------------+---------------------------+------------+--------+------------+---------------+------+
| usn        | name              | mail                      | phno       | gender | DOB        | pw            | dept |
+------------+-------------------+---------------------------+------------+--------+------------+---------------+------+
| 1MV17CS010 | Dattatreya        | dattatreya94@GMAIL.COM    | 9740834260 | M      | 1999-09-23 | ral7gku4rfhLk | CSE  |
| 1MV17CS040 | Dhruv             | dhruv723@gmail.com        | 9901735897 | M      | 2000-10-07 | rajJYeVNCiGD2 | CSE  |
| 1MV17CS140 | Jaishankar        | jaishankarar1@gmail.com   | 6360300095 | M      | 1999-10-07 | rajJYeVNCiGD2 | CSE  |
| 1MV17EC084 | Manu              | klmanu@gmail.com          | 8769043857 | M      | 1999-05-13 | kpLkj09uK     | ECE  |
| 1MV17IS090 | Nachiketh         | nachiketh69@hotmail.com   | 9869742585 | M      | 1998-05-21 | jhuuWGsR45    | ECE  |
| 1MV17EC111 | Pradyuman         | pradece@gmail.com         | 8404975894 | M      | 1999-08-29 | uhui3ryi38wy  | ECE  |
| 1MV17CS051 | Siddhanth Sipoliya | siddhanthsipoliya@mvit.edu. | 7619360459 | M      | 1999-11-15 | ray.whoA8HjCQ | CSE  |
| 1MV17IS010 | Sonia             | sonia345@gmail.com        | 9887547809 | F      | 1998-03-11 | jio76uWGsR45  | ISE  |
| 1MV17IS054 | Rose              | srosewad@gmail.com        | 9754785467 | F      | 1998-02-07 | klLU*HHKhugu  | ISE  |
+------------+-------------------+---------------------------+------------+--------+------------+---------------+------+
9 rows in set (0.06 sec)
```

Figure 5.13: Insertion of student

Staff:

INSERT INTO `staff` (`staffid`, `name`, `mail`, `phno`, `gender`, `DOB`, `pw`, `dept`) VALUES ('mvit1', 'B G VINAYAK', 'BHATVINAYAK94@GMAIL.COM', '9740834260', 'M', '1999-09-23', 'ral7gku4rfhLk', 'CSE');

Select * from staff;



staffid	name	mail	phno	gender	DOB	pw	dept
Mvit1	B G VINAYAK	BHATVINAYAK94@GMAIL.COM	9740834260	M	1999-09-23	ral7gku4rfhLk	CSE
123	Rakesh M R	rakeshmr723@gmail.com	9901735897	M	1999-10-07	rajJYeVNCiGD2	ISE

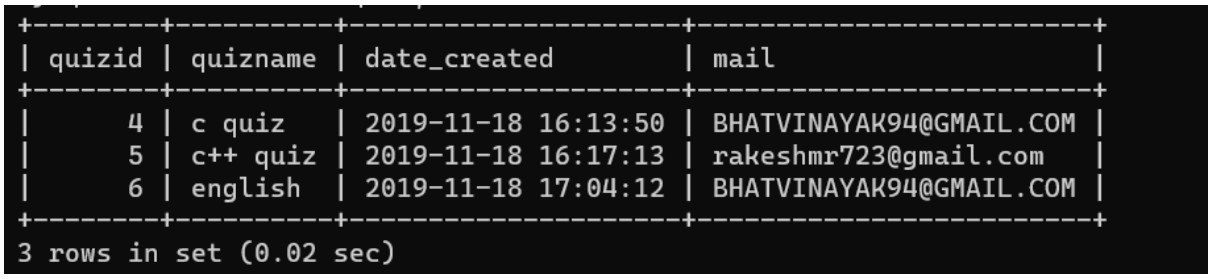
2 rows in set (0.02 sec)

Figure 5.14: Insertion of staff

Quiz:

INSERT INTO `quiz` (`quizid`, `quizname`, `date_created`, `mail`) VALUES
(4, 'c quiz', '2019-11-18 16:13:50', 'BHATVINAYAK94@GMAIL.COM');

Select * from quiz;



quizid	quizname	date_created	mail
4	c quiz	2019-11-18 16:13:50	BHATVINAYAK94@GMAIL.COM
5	c++ quiz	2019-11-18 16:17:13	rakeshmr723@gmail.com
6	english	2019-11-18 17:04:12	BHATVINAYAK94@GMAIL.COM

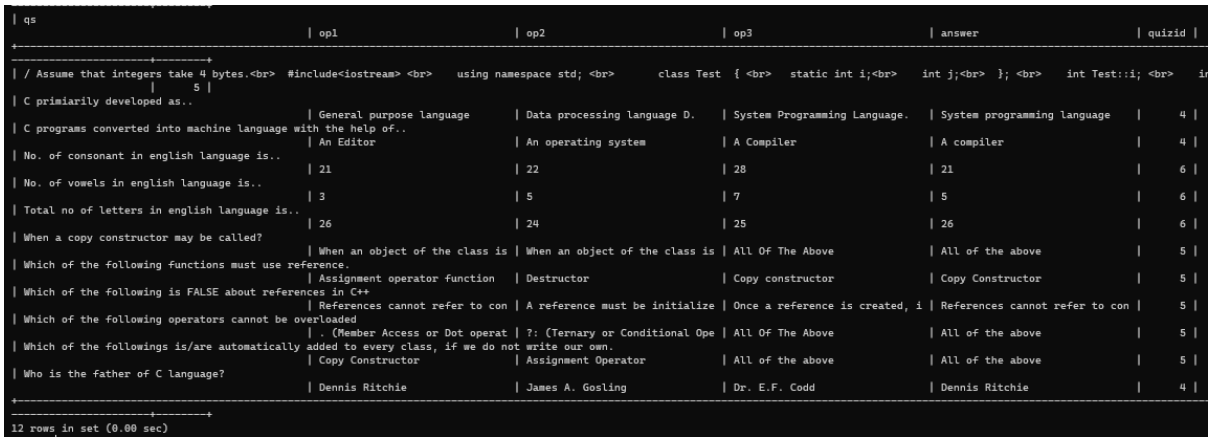
3 rows in set (0.02 sec)

Figure 5.15 Insertion of quiz

Questions:

INSERT INTO `questions` (`qs`, `op1`, `op2`, `op3`, `answer`, `quizid`) VALUES
('C primiarily developed as..', 'General purpose language', 'Data processing language D.', 'None of the above.', 'System programming language ', 4);

Select * from questions;



qs	op1	op2	op3	answer	quizid

/ Assume that integers take 4 bytes.
 #include<iostream>
 using namespace std;
 class Test {
 static int i;
 int j;
 };
 int Test::i;
 i
	5				

C primiarily developed as..	General purpose language	Data processing language D.	System Programming Language.	System programming language	4
C programs converted into machine language with the help of..	An Editor	An operating system	A Compiler	A compiler	4
No. of consonant in english language is..	21	22	28	21	6
No. of vowels in english language is..	3	5	7	5	6
Total no of letters in english language is..	26	24	25	26	6
When a copy constructor may be called?	When an object of the class is	When an object of the class is	All Of The Above	All of the above	5
Which of the following functions must use reference.	Assignment operator function	Destructor	Copy constructor	Copy Constructor	5
Which of the following is FALSE about references in C++	References cannot refer to con	A reference must be initialize	Once a reference is created, i	References cannot refer to con	5
Which of the following operators cannot be overloaded	. (Member Access or Dot operat	?: (Ternary or Conditional Ope	All Of The Above	All of the above	5
Which of the followings is/are automatically added to every class, if we do not write our own.	Copy Constructor	Assignment Operator	All of the above	All of the above	5
Who is the father of C language?	Dennis Ritchie	James A. Gosling	Dr. E.F. Codd	Dennis Ritchie	4

12 rows in set (0.00 sec)

Figure 5.16: Insertion of questions

Score:

INSERT INTO `score` (`slno`, `score`, `quizid`, `mail`, `totalscore`, `remark`) VALUES (13, 6, 5, 'rakeshmariyaplar1@gmail.com', 6, 'good');

Select * from score;

slno	score	quizid	mail	totalscore	remark
14	2	4	jaishankarar1@gmail.com	3	good
15	5	5	dhruv723@gmail.com	6	good
16	7	4	dhruv723@gmail.com	8	good
18	12	6	dhruv723@gmail.com	12	excellent
19	2	4	dattatreya94@gmail.com	2	fail
21	2	6	dattatreya94@gmail.com	4	fail
22	9	4	siddhanthsipoliya@mvit.edu.	9	good
24	4	6	siddhanthsipoliya@mvit.edu.	5	average

8 rows in set (0.02 sec)

Figure 5.17: Insertion of score

Feedback:

INSERT INTO feedback (‘usn’,’name’,’mail’,feedback’) VALUES (‘1MV2o17CS040’,’Dhruv’,’dhruv723@gmail.com’,’Quiz was lenghty’);

Select * from feedback;

usn	name	mail	feedback
1MV17CS040	Dhruv	dhruv723@gmail.com	QUIZ WAS LENGTHY
1MV17CS140	Jaishankar	jaishankarar1@gmail.com	some questions were out of syllabus
1MV17CS010	Dattatreya	dattatreya94@GMAIL.COM	i expect good marks in this quiz
1MV17CS051	Siddhanth Sipoliya	siddhanthsipoliya@mvit.edu.	The quiz was easy to solve and i believe there is more room for improvement in me

4 rows in set (0.02 sec)

Figure 5.18 Insertion of feedback

5.2.4 Queries

1.Create a view all students and their score of the respective quiz

CREATE VIEW ALL_STUDENTS AS SELECT S.USN,S.NAME,S.MAIL,C.SCORE,C.QUIZID,C.TOTALSCORE FROM STUDENT S,SCORE C WHERE S.MAIL=C.MAIL;

USN	NAME	MAIL	SCORE	QUIZID	TOTALSCORE
1MV17CS140	Jaishankar	jaishankarar1@gmail.com	2	4	3
1MV17CS040	Dhruv	dhruv723@gmail.com	5	5	6
1MV17CS040	Dhruv	dhruv723@gmail.com	7	4	8
1MV17CS040	Dhruv	dhruv723@gmail.com	12	6	12
1MV17CS010	Dattatreya	dattatreya94@GMAIL.COM	2	4	2
1MV17CS010	Dattatreya	dattatreya94@GMAIL.COM	2	6	4
1MV17CS051	Siddhanth Sipoliya	siddhanthsipoliya@mvit.edu.	9	4	9
1MV17CS051	Siddhanth Sipoliya	siddhanthsipoliya@mvit.edu.	4	6	5

8 rows in set (0.00 sec)

Figure 5.19: Query 1

2.Categorise students based on their results

If totalscore=0-7 then CAT= 'CAN IMPROVE'

If totalscore=8-10 then CAT= 'GOOD'

IF totalscore=11-12 then CAT= 'EXCELLENT'

SELECT S.USN,S.NAME,S.MAIL,S.PHNO,S.GENDER,S.DEPT,

(CASE WHEN C.totalscore BETWEEN 0 AND 7 THEN 'CAN IMPROVE'

WHEN C.totalscore BETWEEN 8 AND 10 THEN 'GOOD' ELSE 'EXCELLENT'

END) AS CAT FROM STUDENT S,SCORE C WHERE S.MAIL=C.MAIL;

USN	NAME	MAIL	PHNO	GENDER	DEPT	CAT
1MV17CS140	Jaishankar	jaishankarar1@gmail.com	6360300095	M	CSE	CAN IMPROVE
1MV17CS040	Dhruv	dhruv723@gmail.com	9901735897	M	CSE	CAN IMPROVE
1MV17CS040	Dhruv	dhruv723@gmail.com	9901735897	M	CSE	GOOD
1MV17CS040	Dhruv	dhruv723@gmail.com	9901735897	M	CSE	EXCELLENT
1MV17CS010	Dattatreya	dattatreya94@GMAIL.COM	9740834260	M	CSE	CAN IMPROVE
1MV17CS010	Dattatreya	dattatreya94@GMAIL.COM	9740834260	M	CSE	CAN IMPROVE
1MV17CS051	Siddhanth Sipoliya	siddhanthsipoliya@mvit.edu.	7619360459	M	CSE	GOOD
1MV17CS051	Siddhanth Sipoliya	siddhanthsipoliya@mvit.edu.	7619360459	M	CSE	CAN IMPROVE

8 rows in set (0.00 sec)

Figure 5.20: Query 2

3. List the number of students taking up quiz from computer science department

SELECT COUNT(dept) FROM STUDENT where dept='cse';

```
+-------------+
| COUNT(dept) |
+-------------+
|           4 |
+-------------+
1 row in set (0.01 sec)
```

Figure 5.21: Query 3

4. Display the scoresheet of the students according to their decreasing order of marks

SELECT * FROM SCORE ORDER BY TOTALSCORE DESC;

```
mysql> SELECT * FROM SCORE ORDER BY TOTALSCORE DESC;
+------+-------+--------+----------------------------+------------+-----------+
| slno | score | quizid | mail                       | totalscore | remark    |
+------+-------+--------+----------------------------+------------+-----------+
|   18 |    12 |      6 | dhruv723@gmail.com         |         12 | excellent |
|   22 |     9 |      4 | siddhanthsipoliya@mvit.edu. |          9 | good      |
|   16 |     7 |      4 | dhruv723@gmail.com         |          8 | good      |
|   15 |     5 |      5 | dhruv723@gmail.com         |          6 | good      |
|   24 |     4 |      6 | siddhanthsipoliya@mvit.edu. |          5 | average   |
|   21 |     2 |      6 | dattatreya94@gmail.com     |          4 | fail      |
|   14 |     2 |      4 | jaishankarar1@gmail.com    |          3 | good      |
|   19 |     2 |      4 | dattatreya94@gmail.com     |          2 | fail      |
+------+-------+--------+----------------------------+------------+-----------+
8 rows in set (0.00 sec)
```

Figure 5.22: Query 4

5. Using Inner Join list the quiz names created by the respective staff

SELECT STAFF.STAFFID, QUIZ.QUIZNAME, STAFF.NAME

FROM STAFF

INNER JOIN QUIZ ON QUIZ.MAIL=STAFF.MAIL;

```
+---------+----------+-------------+
| STAFFID | QUIZNAME | NAME        |
+---------+----------+-------------+
| Mvit1   | c quiz   | B G VINAYAK |
| 123     | c++ quiz | Rakesh M R  |
| Mvit1   | english  | B G VINAYAK |
+---------+----------+-------------+
3 rows in set (0.00 sec)
```

Figure 5.23: Query 5

6. List all students who were born before 1999 and after 2000(use UNION)

(SELECT DOB, NAME

FROM student

WHERE DOB BETWEEN '1998-01-01' AND '1999-01-01') UNION

(SELECT DOB, NAME

FROM student

WHERE DOB BETWEEN '2000-01-01' AND '2001-01-01');

```
+------------+-----------+
| DOB        | NAME      |
+------------+-----------+
| 1998-05-21 | Nachiketh |
| 1998-03-11 | Sonia     |
| 1998-02-07 | Rose      |
| 2000-10-07 | Dhruv     |
+------------+-----------+
4 rows in set (0.00 sec)
```

Figure 5.24: Query 6

7. Delete the quiz with the ID 5 thus removing all the questions.

DELETE FROM QUIZ WHERE QUIZID=5;

```
+--------+----------+---------------------+-------------------------+
| quizid | quizname | date_created        | mail                    |
+--------+----------+---------------------+-------------------------+
|      4 | c quiz   | 2019-11-18 16:13:50 | BHATVINAYAK94@GMAIL.COM |
|      5 | c++ quiz | 2019-11-18 16:17:13 | rakeshmr723@gmail.com   |
|      6 | english  | 2019-11-18 17:04:12 | BHATVINAYAK94@GMAIL.COM |
+--------+----------+---------------------+-------------------------+
3 rows in set (0.02 sec)
```

Figure 5.25: Query 7

8. UPDATE THE FEEDBACK GIVEN BY DHRUV.

UPDATE FEEDBACK SET FEEDBACK='QUIZ WAS LENGTHY' WHERE USN='1MV17CS040';

```
+-----------+-----------------+--------------------------+-----------------------------------------------------------------------------------------+
| usn       | name            | mail                     | feedback                                                                                |
+-----------+-----------------+--------------------------+-----------------------------------------------------------------------------------------+
| 1MV17CS040 | Dhruv           | dhruv723@gmail.com       | QUIZ WAS LENGTHY                                                                        |
| 1MV17CS140 | Jaishankar      | jaishankarar1@gmail.com  | some questions were out of syllabus                                                     |
| 1MV17CS010 | Dattatreya      | dattatreya94@GMAIL.COM   | i expect good marks in this quiz                                                        |
| 1MV17CS051 | Siddhanth Sipoliya | siddhanthsipoliya@mvit.edu. | The quiz was easy to solve and i believe there is more room for improvement in me |
+-----------+-----------------+--------------------------+-----------------------------------------------------------------------------------------+
4 rows in set (0.02 sec)
```

Figure 5.26: Query 8

9. TRIGGER FOR DELETING QUESTIONS IF ITS QUIZ ID IS DELETED.

```
CREATE TRIGGER `ondeleteqs` AFTER DELETE ON `quiz`
 FOR EACH ROW delete from questions where questions.quizid=old.quizid
```

CHAPTER 6

FRONTEND DESIGN AND IMPLEMENTATION

6.1 Frontend Used

In this project we make use of HTML along with CSS and JAVASCRPT.

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages. Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext. As its name suggests, HTML is a Markup Language which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display. Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

In an HTML page, PHP code is enclosed within special PHP tags. When a visitor opens the page, the server processes the PHP code and then sends the output (not the PHP code itself) to the visitor's browser. Actually it is quite simple to integrate HTML and PHP. A PHP script can be treated as an HTML page, with bits of PHP inserted here and there. Anything in a PHP script that is not contained within <?php ?> tags is ignored by the PHP compiler and passed directly to the web browser.

6.2 Modules Used

Online Examination System is that which enables the Students & Teachers to register for the system. Students are allowed to take the online test and see their progress. Also, it enables the Teachers to add, delete, update the test questions and also to keep track of the student progress.

Student:

Student has to log in to the system and can then view all the quizzes. Student can see the quiz

list and attend the quizzes. After attending the quizzes students will get instant result and the same will be stored in the database.

Staff:

Staff also has to log in to the system first, then they can add/remove quizzes. They can see the progress and they can also update the existing quiz.

Login:

Login is must and should for both staff and student. So that all records will be safely saved to the database. If someone had forgot the password, one can reset that password.

Add/Remove Quiz Questions:

Staff can add, remove the quiz. And also they can add extra questions to an existing quiz.

Score:

Staff can see the scoreboard of the quiz which is added by him/her. And student can see the score of the quiz which he/she is attended.

CHAPTER 7

SOURCE CODE

Adding a question to the quiz:

```
 $qid=$_GET["qid"];
    if (isset($_POST['submit'])) {
        $qs = $_POST["qs"];
        $op1 = $_POST["op1"];
        $op2 = $_POST["op2"];
        $op3 = $_POST["op3"];
        $ans = $_POST["ans"];
        $sql = "insert into questions(qs,op1,op2,op3,answer,quizid)
values('$qs','$op1','$op2','$op3','$ans','$qid');";
        $res =   mysqli_query($conn, $sql);
        if ($res == true) {
            echo '<script>history.pushState({}, "", "");</script>';
        } elseif ($res != true) {
            echo '<script>alert("Question already exsits");</script>';
        }
    }
    if (isset($_POST['submit1'])) {
        $qs = $_POST["qs"];
        $op1 = $_POST["op1"];
        $op2 = $_POST["op2"];
        $op3 = $_POST["op3"];
        $ans = $_POST["ans"];
        $sql = "insert into questions(qs,op1,op2,op3,answer,quizid)
values('$qs','$op1','$op2','$op3','$ans','$qid');";
        $res =   mysqli_query($conn, $sql);
        if ($res == true) {
            header("Location: homestaff.php");
```

```
        } elseif ($res != true) {
            echo '<script>alert("Question already exsits");</script>';
        }
    }
```

Staff login into homepage:

```
<?php
session_start();
$conn = mysqli_connect('127.0.0.1:3307', 'root', '', 'project');
if (!$conn) {
    echo "<script>alert(\"Database error retry after some time !\")</script>";
} else {
    $type1 = $_SESSION["type"];
    $username1 = $_SESSION["username"];
    $sql = "select * from " . $type1 . " where mail='{$username1}'";
    $res =   mysqli_query($conn, $sql);
    if ($res == true) {
        global $dbmail, $dbpw, $dbusn;
        while ($row = mysqli_fetch_array($res)) {
            $dbmail = $row['mail'];
            $dbname = $row['name'];
            $dbusn = $row['staffid'];
            $dbphno = $row['phno'];
            $dbgender = $row['gender'];
            $dbdob = $row['DOB'];
            $dbdept = $row['dept'];
        }
    }
    if (isset($_POST['submit'])) {
        $qname = strtolower($_POST['quizname']);
        $_SESSION["qname"] = $qname;
        $sql1 = "insert into quiz(quizname,mail) values('$qname','$username1')";
        $res1 = mysqli_query($conn, $sql1);
```

```
        if ($res1 == true) {
            $sql = "select quizid from quiz where quizname='" . $qname . "';";
            $res = mysqli_query($conn, $sql);
            if ($res == true) {
                header("location: addqs.php");
            } else {
                echo "<script>alert(\"some error occured\");</script>";
            }
        } else {
            echo "<script>alert(\"Already name exists\");</script>";
        }  }
    if (isset($_POST['submit1'])) {
        $qid1 = strtolower($_POST['quizid']);
        $sql1 = "delete from quiz where quizid='{$qid1}'";
        $res1 = mysqli_query($conn, $sql1);
        if ($res1 == true) {
            echo "<script>alert(\"Quiz successfully deleted\");</script>";
        } else {
            echo "<script>alert(\"Unknown error occured during deletion of quiz\");</script>";
        }    }
    if (isset($_POST['submit2'])) {
        $qid1 = $_POST['quizid'];
        $sql1 = "select quizid from quiz where quizid='{$qid1}'";
        $res1 = mysqli_query($conn, $sql1);
        if ($res1 == true) {
            echo "<script>window.location.replace(\"viewq.php?qid=" . $qid1 . "\");</script>";
        } else {
            echo "<script>alert(\"Unknown error occured during viweing of quiz\");</script>";
        }
    }
}

?>
```

Student login into homepage:

```
<?php
        $sql ="select * from quiz";
        $res=mysqli_query($conn,$sql);
        if($res)
        {

            echo "<center><table><thead><tr><td>Quiz Title</td><td>Created
on</td><td>Created By</td><td>  </td></tr></thead>";
            while ($row = mysqli_fetch_assoc($res)) {
               echo
"<tr><td>".$row["quizname"]."</td><td>".$row["date_created"]."</td><td>".$row["mail"]."
</td><td><a id=\"tq\" href='takeq.php?qid=".$row['quizid']."'>Take Quiz</button></tr>";
            }
            echo "</table></center>";
        }
        ?>
```

Signing up into the Online Examination System:

```
<?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']))
{
      $conn = mysqli_connect('127.0.0.1:3307', 'root', '', 'project');
      if (!$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']);
        $password1 = crypt($password1,'rakeshmariyaplarrakesh');
        $cpassword1 = crypt($cpassword1,'rakeshmariyaplarrakesh');
        if ($password1 == $cpassword1) {
            $sql = "insert into student (usn,name,mail,phno,dept,gender,DOB,pw)
values('$usn1','$name1','$mail1','$phno1','$dept1','$gender1','$dob1','$password1')";
            if (mysqli_query($conn, $sql)) {
                echo "<script>
                alert('successful!');
                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();
        }
    }
}

if (isset($_POST['staffsu'])) {
    session_start();
```

```
   if (isset($_POST['name2']) && isset($_POST['staffid']) && isset($_POST['mail2']) &&
isset($_POST['phno2']) && isset($_POST['dept2']) && isset($_POST['dob2']) &&
isset($_POST['gender2']) && isset($_POST['password2']) && isset($_POST['cpassword2']))
{
      $conn = mysqli_connect('127.0.0.1:3307', 'root', '', 'project');
      if (!$conn) {
         echo "<script>alert(\"Database error retry after some time !\")</script>";
      }
      $name2 = mysqli_real_escape_string($conn, $_POST['name2']);
      $usn2 = mysqli_real_escape_string($conn, $_POST['staffid']);
      $mail2 = mysqli_real_escape_string($conn, $_POST['mail2']);
      $phno2 = mysqli_real_escape_string($conn, $_POST['phno2']);
      $dept2 = mysqli_real_escape_string($conn, $_POST['dept2']);
      $dob2 = mysqli_real_escape_string($conn, $_POST['dob2']);
      $gender2 = mysqli_real_escape_string($conn, $_POST['gender2']);
      $password2 = mysqli_real_escape_string($conn, $_POST['password2']);
      $cpassword2 = mysqli_real_escape_string($conn, $_POST['cpassword2']);
      $password2 = crypt($password2,'rakeshmariyaplarrakesh');
      $cpassword2 = crypt( $cpassword2,'rakeshmariyaplarrakesh');
      if ($password2 == $cpassword2) {
         $sql = "insert into staff (staffid,name,mail,phno,dept,gender,DOB,pw)
values('$usn2','$name2','$mail2','$phno2','$dept2','$gender2','$dob2','$password2')";
         if (mysqli_query($conn, $sql)) {
            echo "<script>
            alert('successful!');
            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(\"signup.php\");</script>";
            session_destroy();
        }
    }
}
?>
```

Taking up the Quiz:

```
<?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";
                }else{
                $i=1;
                $j=0;
                echo "<form method=\"POST\">";
                while ($row = mysqli_fetch_assoc($res)) {
                    echo $i.". ".$row["qs"]."<br>";
                    echo "<input type=\"radio\" value=\"".$j."\"
name=\"ans".$i.$j."\">".$row["op1"]."<br>";
                    echo "<input type=\"radio\" value=\"".($j+1)."\"
name=\"ans".$i.$j."\">".$row["op2"]."<br>";
                    echo "<input type=\"radio\"
value=\"".($j+2)."\"name=\"ans".$i.$j."\">".$row["op3"]."<br>";
```

```
                echo "<input type=\"radio\"value=\"".($j+3)."\"
name=\"ans".$i.$j."\">".$row["answer"]."<br><br>";
                $i++;
            }
            echo "<input id=\"btn\" type=\"submit\" name=\"submit\"
value=\"submit\"><br><br><br>";
            echo "</form><br><br>";
        }
        }
        else
        {
            echo "error".mysqli_error($conn).".";
        }
        if(isset($_POST["submit"])){
            global $score;
            for($i=1;$i<=$count;$i++)
            {
                if(isset($_POST["ans".$i.$j]) && $_POST["ans".$i.$j]==3){
                    $score++;
                }
            }
            echo "<script>alert(\"You scored ".$score." out of ".$count."\");</script>";
            $sql ="insert into score(score,mail,quizid,totalscore)
values('$score','$dbmail','$qid','$count');";
            $res=mysqli_query($conn,$sql);
            if($res)
            {
                echo '<script>history.pushState({}, "", "");</script>';
                echo "<script>window.location.replace(\"homestud.php\");</script>";
            }else{
                echo "<script>alert(\"error occured updating score in
database".mysqli_error($conn)."\");</script>";
            }
```

```
    }
  } ?>
```

Viewing the Quiz:

```
<?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";
        }else{
        $i=1;
        $j=0;
        echo "<form method=\"POST\">";
        echo "<input id=\"btn\" type=\"submit\" name=\"submit\" value=\"Add
Questions\"><br><br><br>";
        while ($row = mysqli_fetch_assoc($res)) {
          echo $i.". ".$row["qs"]."<br>";
          echo "<input type=\"radio\" value=\"".$j."\"
name=\"ans".$i.$j."\">".$row["op1"]."<br>";
          echo "<input type=\"radio\" value=\"".($j+1)."\"
name=\"ans".$i.$j."\">".$row["op2"]."<br>";
          echo "<input type=\"radio\"
value=\"".($j+2)."\"name=\"ans".$i.$j."\">".$row["op3"]."<br>";
          echo "<input type=\"radio\"value=\"".($j+3)."\"
name=\"ans".$i.$j."\">".$row["answer"]."<br><br>";
```

```
            $i++;
        }
        echo "</form><br><br>";
    }
    }
    else
    {
        echo "error".mysqli_error($conn).".";
    }
    if(isset($_POST["submit"])){
        echo "<script>window.location.replace(\"addq.php?qid=".$qid."\")</script>";
    }
} ?>
```

CHAPTER 8

RESULTS & SNAPSHOTS

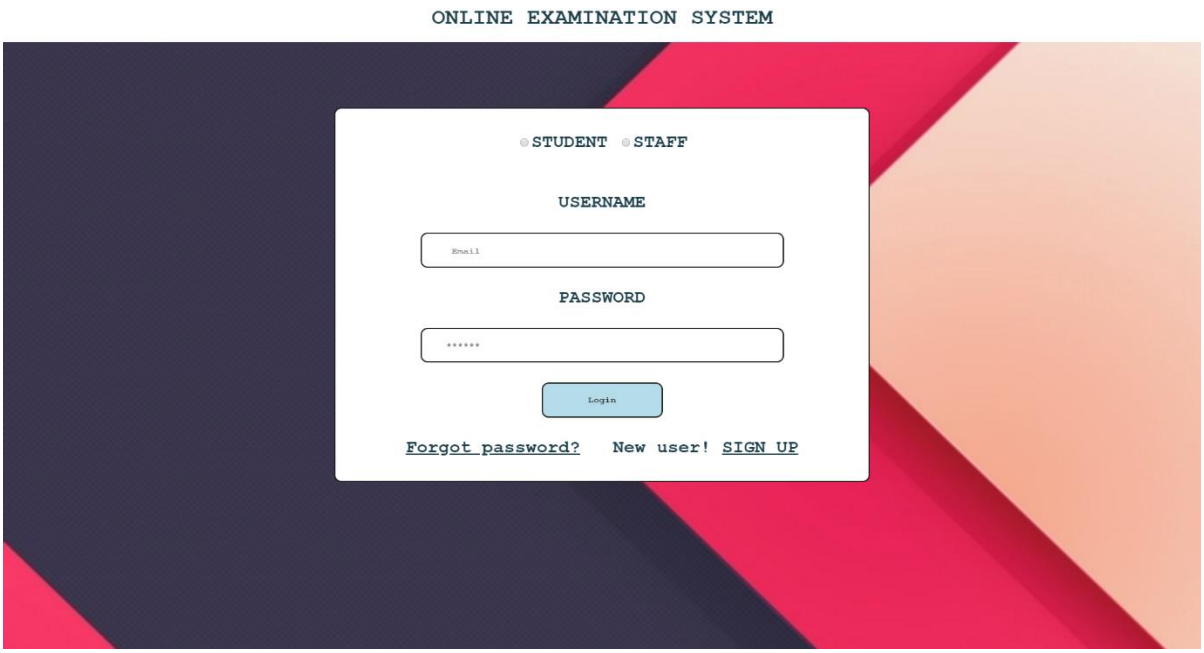


Figure 8.1: Start page of the project

The home window showing the login page for both student and staff along with the password retrieval option

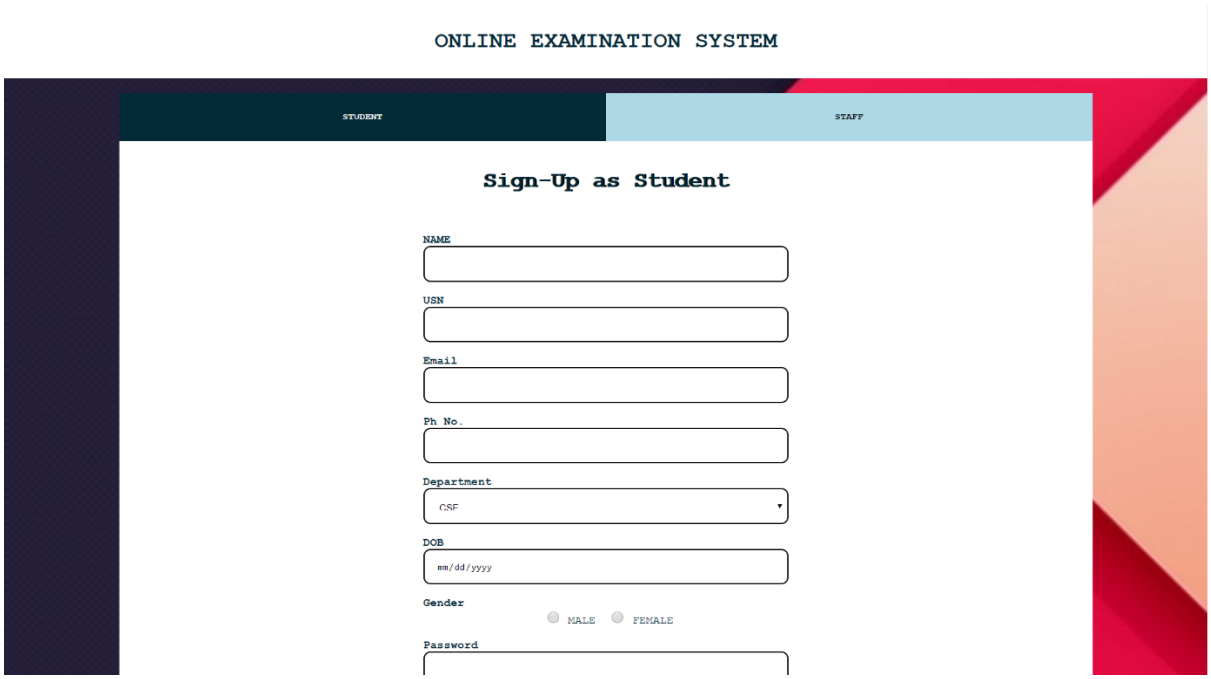


Figure 8.2: Sign Up Page

Registration page for student and staff for first time users

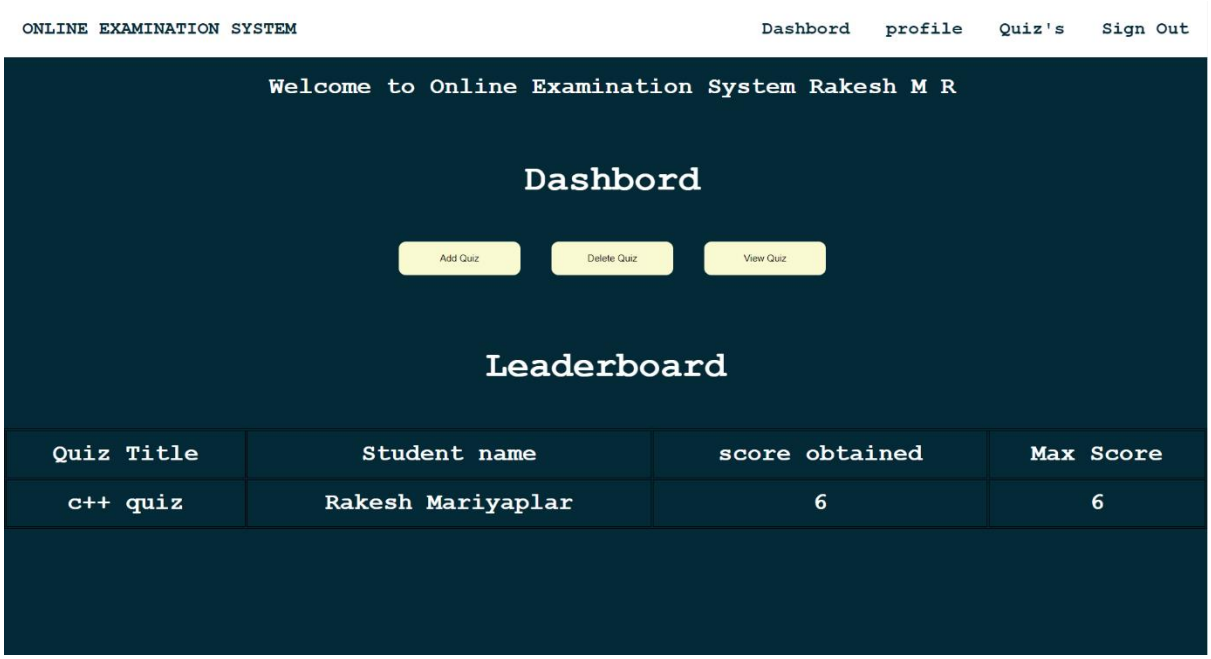


Figure 8.3: Dashboard for Staff

Dashboard for staff to either add, delete and view the quiz along with the leaderboard

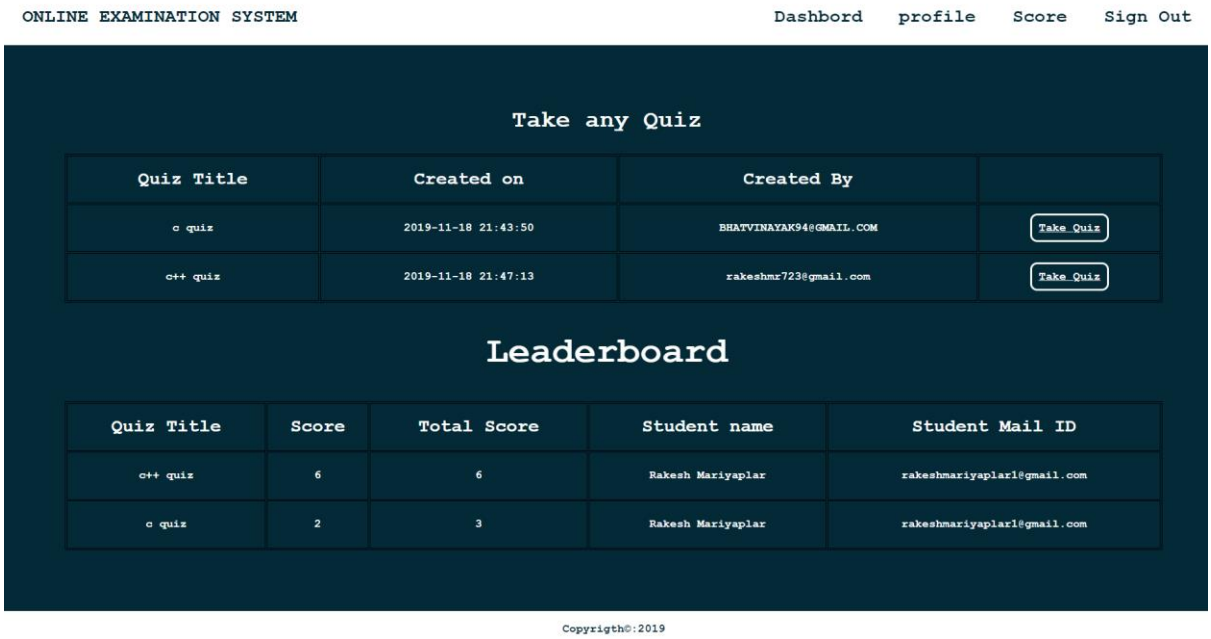


Figure 8.4: Dashboard for Students

Dashboard for students showing the various quizzes available along with the leaderboard in that particular quiz

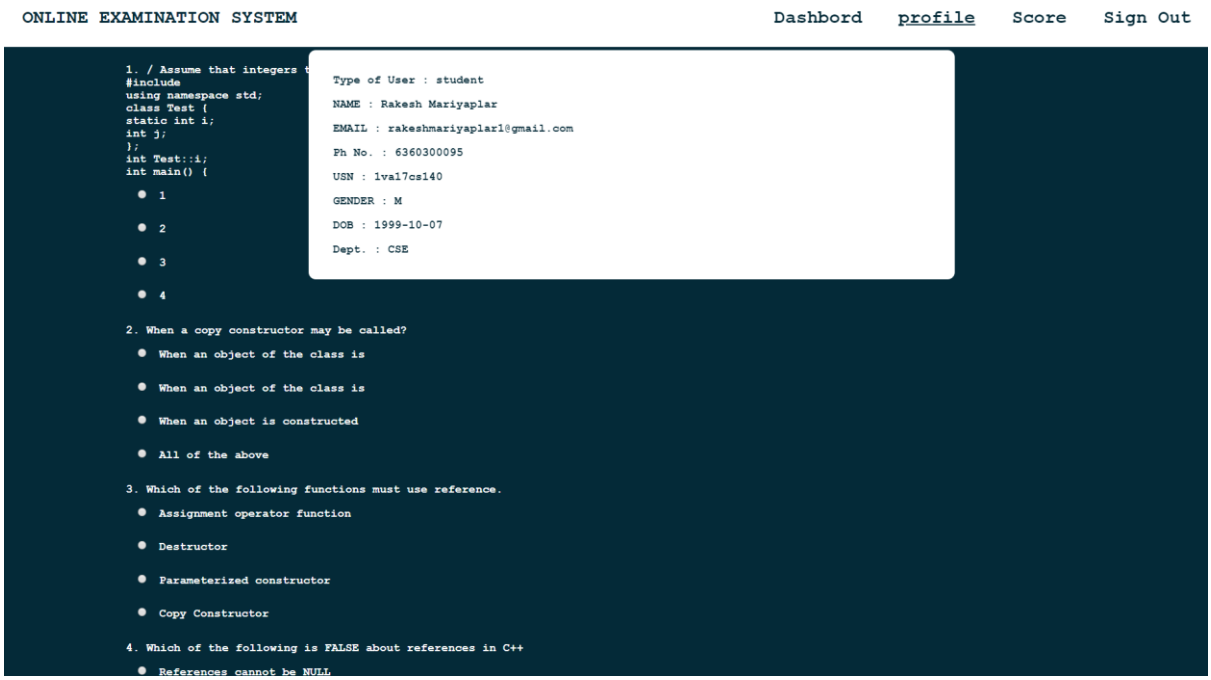


Figure 8.5: Profile view of Student

Dialogue box showing the details of the respective student

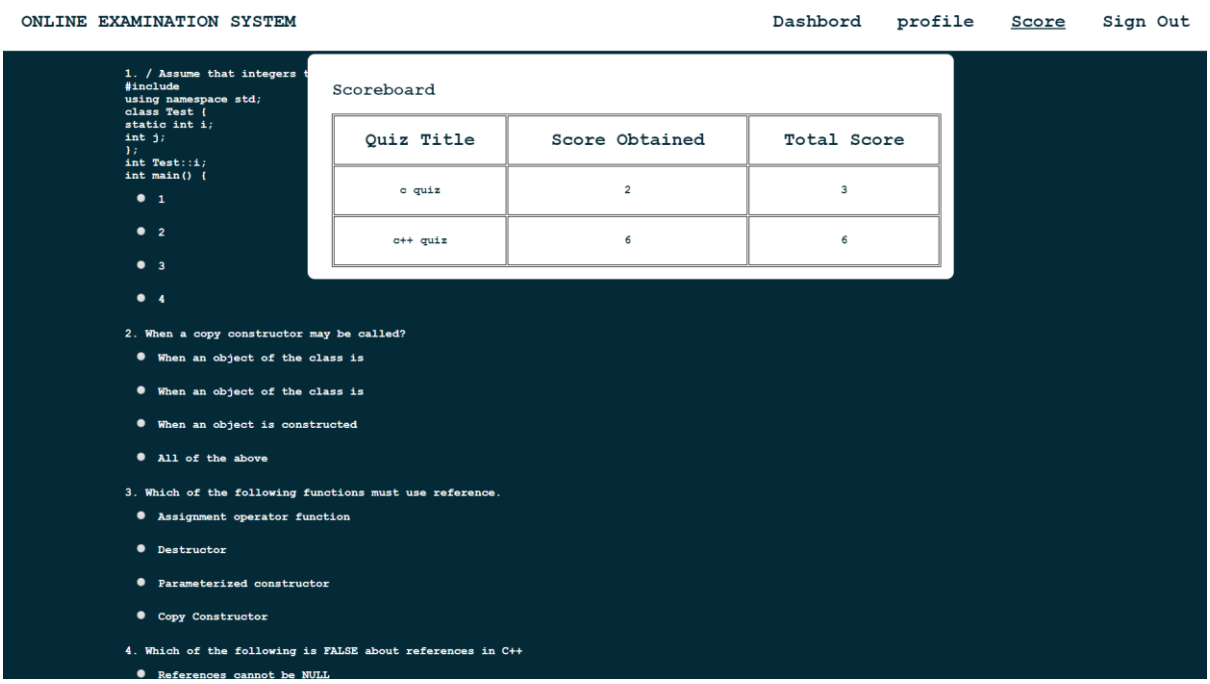


Figure 8.6: Scoreboard of user

Shows the total score by a student in all the quizzes taken up

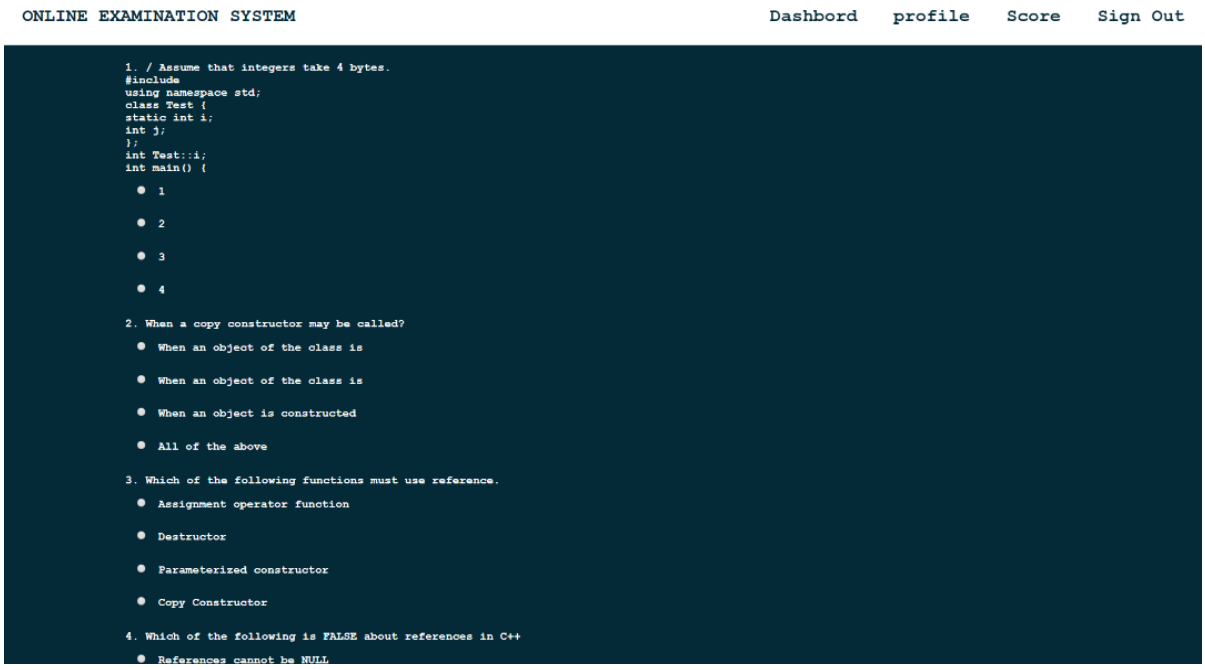


Figure 8.7: Taking Quiz by Student

Displays questions along with options belonging to a particular quiz id

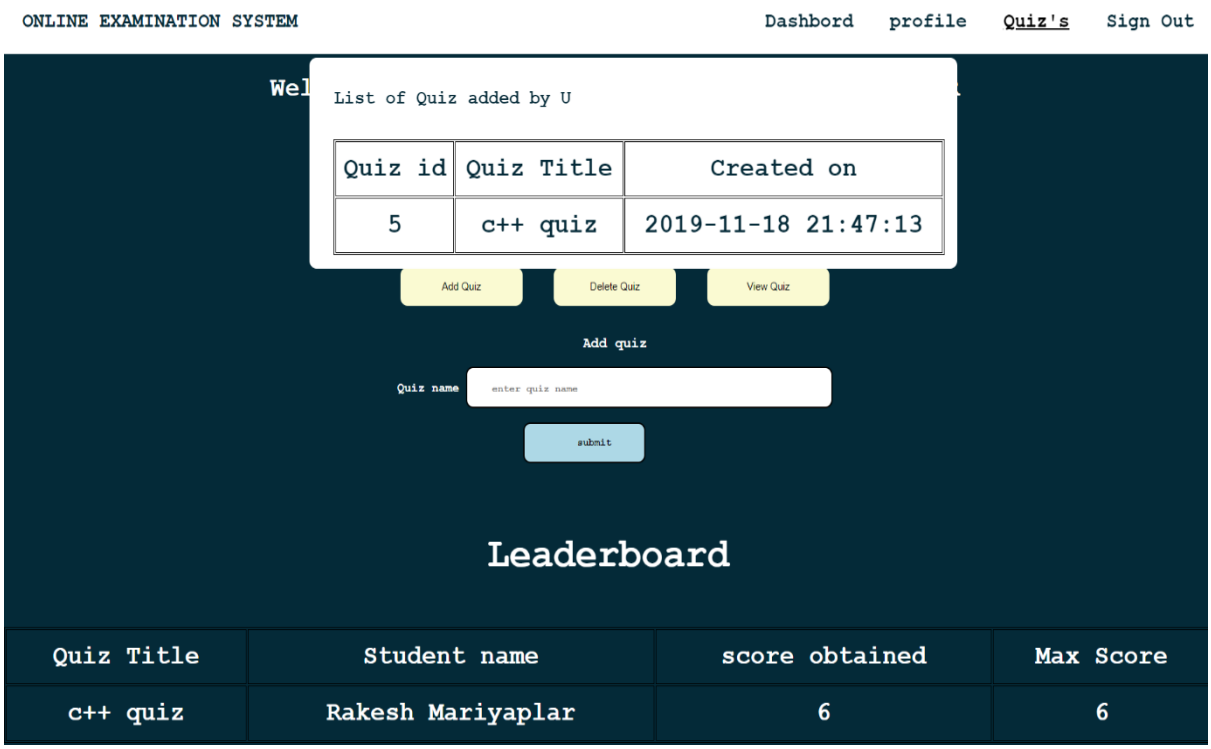


Figure 8.8: Quiz Added by The Staff

Displays total number of quizzes added by the staff.

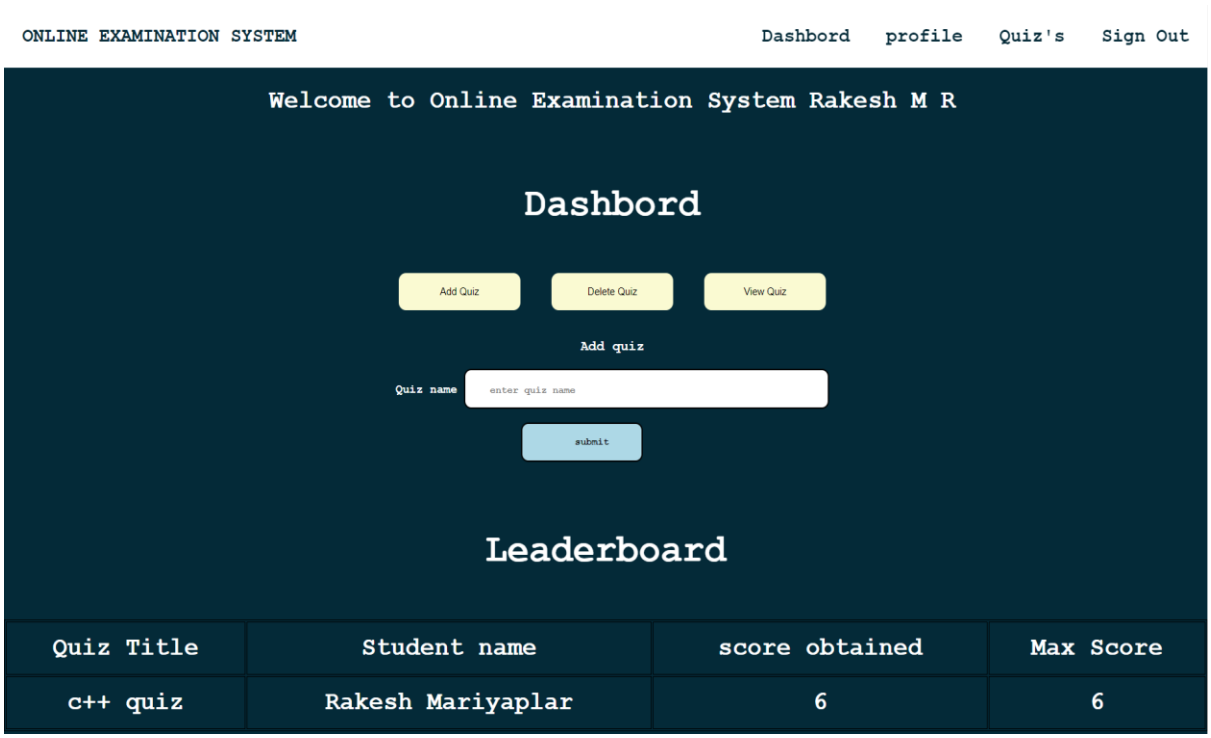


Figure 8.9: Adding quiz page by staff

Shows the page where the staff adds the quiz

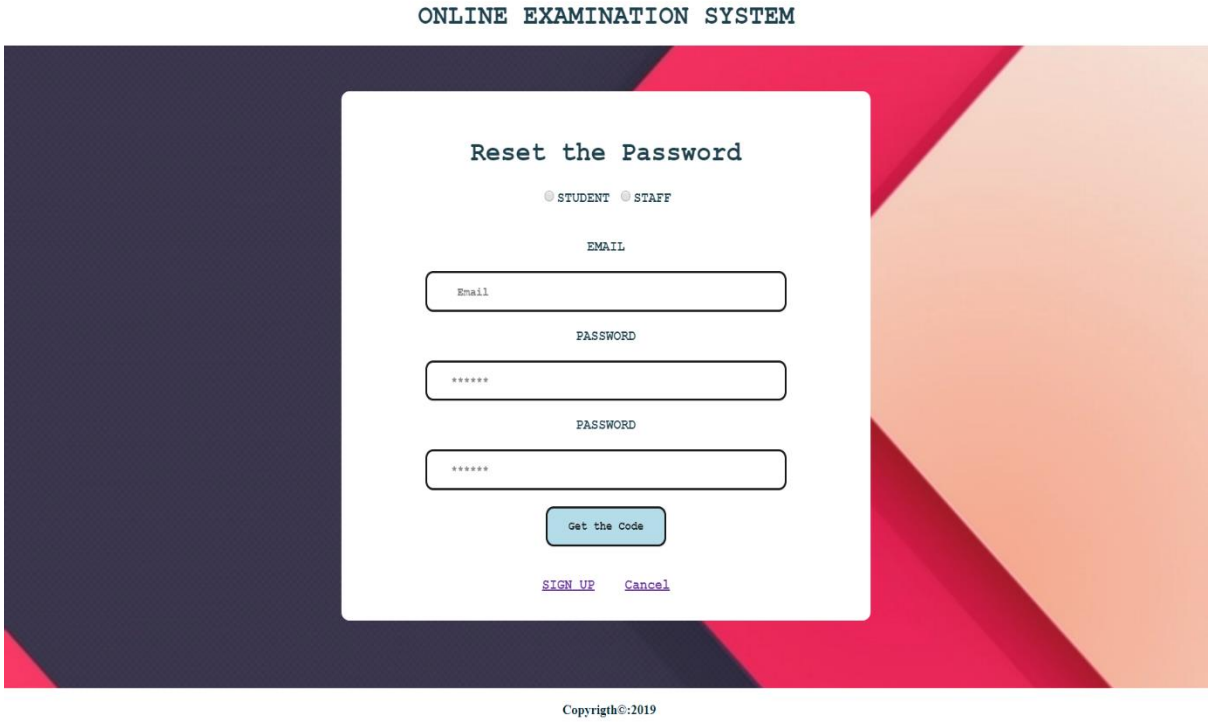


Figure 8.10: Request for security code

Requesting security code for resetting the password.

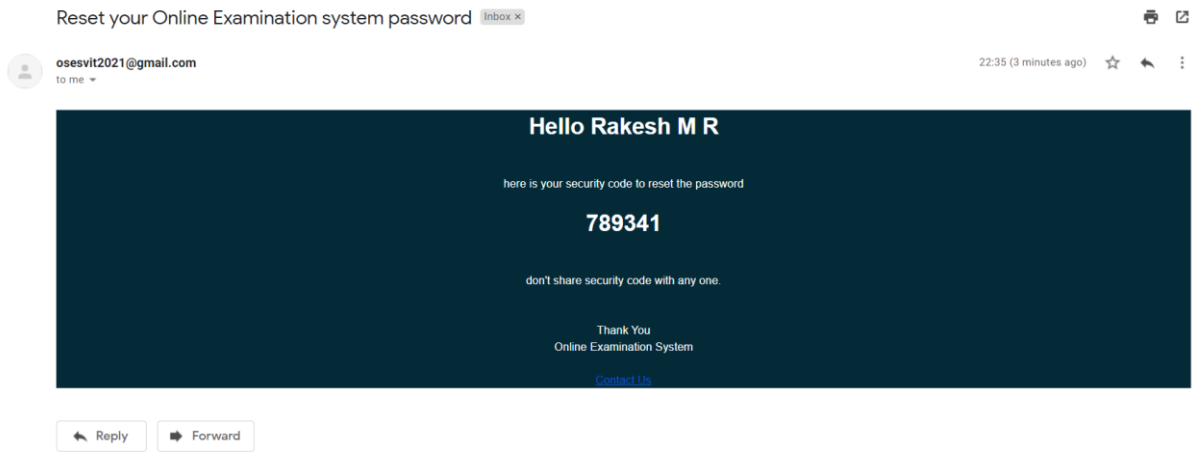


Figure 8.11: mail received by user with security code

Mail sent to the user for verification required for changing of password.

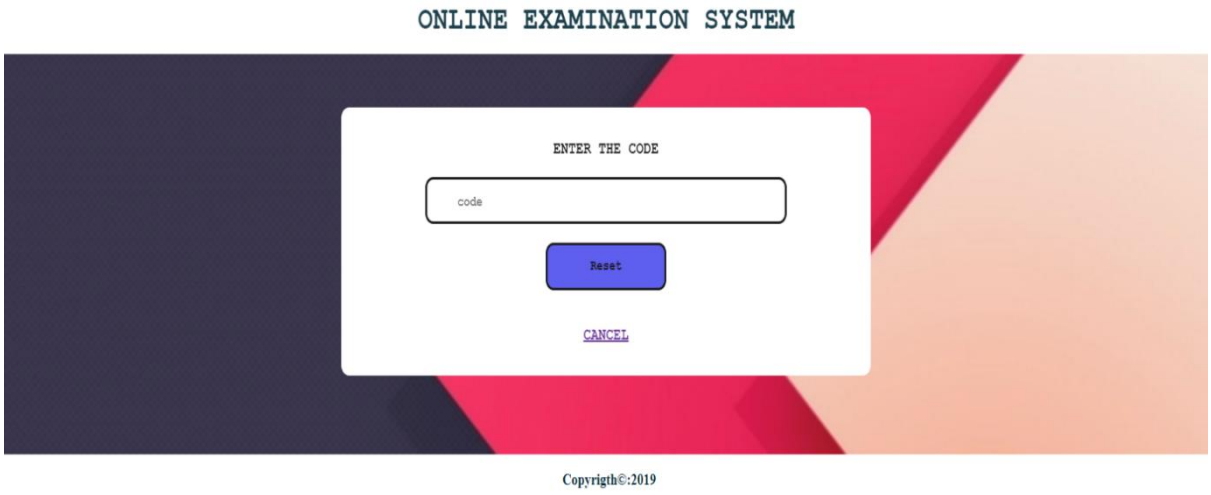


Figure 8.12: verifying the security code and update password

Verifies the security code and requests for change of password.

CHAPTER 9

CONCLUSION & SCOPE

The online examination system provides better functionality for an examination to be more efficient and reduce manual paperwork in order to automate all possible tasks. For implementing this system, PHP, HTML, CSS, JavaScript and MySql are used.

The system comprises of following features:

- Management of quiz.
- Automated grading.
- Adding/deleting quizzes and questions.

SCOPE OF ENHANCEMENT

There are also few features which can be integrated with the system to make it more flexible. Below list shows the future points to be considered:

- Implementing the timer for the quiz.
- Sending mails on sign up and when student takes the quiz.
- Supporting all type of questions including MCQ’s

REFERENCES

BOOK REFERENCE

[1] Database System Models, Languages, Ramez Elmasri and Sham Kant B. Navathe, 7th Edition, 2017 Pearson.

[2] Fundamentals of Web Development, Randy Connolly and Ricardo Hoar ,First Impression, 2016 Pearson

WEBSITES

www.stackoverflow.com

www.youtube.com

https://www.php.net

www.google.com

https://www.w3schools.com