

**A PROJECT REPORT  
ON  
“ONLINE EXAMINATION SYSTEM”  
SUBMITTED  
FOR THE DEGREE OF  
BACHELOR OF ENGINEERING  
(Computer Engineering)**

**BY**

**Mr. Faizan Shiklagar  
Mr. Shivam Singh  
Mr. Mansing Padvi  
Mr. Abhimanyu Karche**

**Under the Guidance of  
Prof. K S Hangargi**



**DEPARTMENT OF COMPUTER ENGINEERING  
P. K. TECHNICAL CAMPUS  
GAT NO. 714 KADACHIWADI ,TAL: KHED, DIST: PUNE.**

**AFFILIATED TO**



**SAVITRIBAI PHULE PUNE UNIVERSITY  
Year: 2022-23**



**P K TECHNICAL CAMPUS  
THIRD-YEAR COMPUTER ENGINEERING**

***CERTIFICATE***

**THIS IS TO CERTIFY THAT THE PROJECT REPORT ENTITLES**

**“Online Examination System”**

**IN PARTICULAR FULFILLMENT FOR THE AWARD OF THE S.E COMPUTER  
ENGINEERING THROUGH THE SAVITRIBAI PHULE PUNE UNIVERSITY DURING THE  
ACADEMIC YEAR 2021-2022**

**Prof. K. S. Hangargi  
Project guide**

**Prof. S. R. Bhujbal  
HOD**

## ACKNOWLEDGEMENT

The completion of Project brings with and sense of satisfaction, but it is never completed without thanking the persons who are all responsible for its successful completion. First and foremost, we wish to express our deep sincere feelings of gratitude to our Institution, **P.K Technical Campus**, for providing us an opportunity to do our education.

We would like to thank the **Management** and **Prof. K S Hangargi**, for providing the facilities.

I would like to express my deep and sincere gratitude to my subject teacher Prof. K.S.Hangargi for giving us the opportunity to do this project and provide invaluable guidance throughout this project. Her dynamism, vision, sincerity and motivation have deeply inspired us. She has taught us the methodology to carry out the research and to present the project works as clearly as possible. It was a great privilege and honor to work and study under her guidance. We are extremely grateful for what she has offered us. We would also like to thank her for her friendship, empathy, and great sense of humor.

We are extremely grateful to all group members for the dedication and consistency towards this mini project. And also thankful for all the resources which are provided by each group member and which played a very crucial role in the accomplishment of this project.

Students Name

# **ABSTRACT**

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.

Functional Specification:

1. Registering new Student
2. Getting the student and staff info and storing it to databases
3. Getting the type and number of question paper
4. Generating result after the exam.

# CONTENTS

<b>Chapters</b>	<b>Page No</b>
<b>ACKNOWLEDGEMENT</b>	<b>i</b>
<b>ABSTRACT</b>	<b>ii</b>
<b>1. INTRODUCTION TO DBMS</b>	<b>1</b>
<b>2. SYSTEM SPECIFICATIONS</b>	<b>4</b>
<b>3. SYSTEM DESIGN</b>	
<b>4. FUNCTIONS and Implementations</b>	<b>9</b>
<b>5. Graphical User Interface</b>	<b>11</b>
<b>6. CONCLUSION &amp; FUTURE ENHANCEMENT</b>	<b>13</b>

# CHAPTER 1

## INTRODUCTION

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.

### 1.1 What is the need of DBMS?

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
- vendor support

## 1.2 Design and Modeling:

The first task of a database designer is to produce a conceptual datamodel 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.

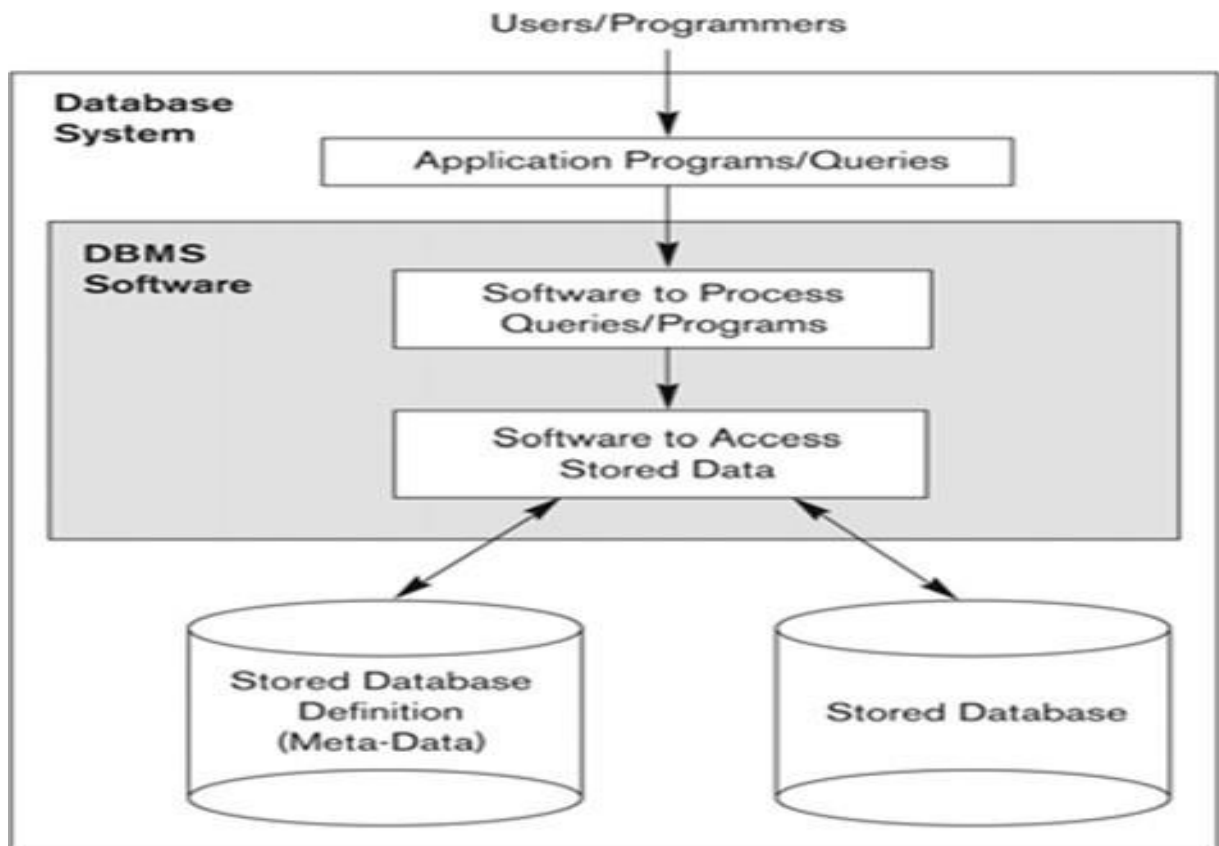


Figure 1.1: Simplified database system environment

### **1.3 Objective**

The main objective of this project is to determine how an interactive inventory management system helps in the smooth functioning of a warehouse compared to traditional inventory management, by digitizing all the records and transacting everything on a computer rather than on paper. This project is a two-tier architecture application.

### **1.4 Problem Statement**

Existing systems for Warehouse Inventory Management are very inefficient and mostly involve a lot of manual work to be done by the manager of the system. With this project, we want to automate as many tasks as possible using the available technology and the internet.

### **1.5 Scope of the report**

The essential framework of this report would be to elaborate the design of E.R-diagram, Schema Diagram and to display how the modules of the program work in order to achieve the automation.



## **CHAPTER 2**

# **SYSTEM AND SOFTWARE REQUIREMENTS AND SPECIFICATIONS**

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

### **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.

#### **2.1.1 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.

### **2.2 NON-FUNCTIONAL REQUIREMENTS:**

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

#### **2.2.1 Usability**

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

#### **2.2.2 Security**

Security requirements are included in a system to ensure:

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

### **2.2.3 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.

## **2.3 SOFTWARE REQUIREMENTS**

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

## **2.4 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: 3

# ER DIAGRAM, SCHEMA DIAGRAM

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

### 3.1 ER Diagram with relationships and cardinality ratio

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 regard 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 following are the notations of the ER diagram:








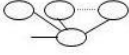

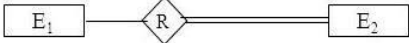
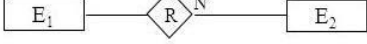
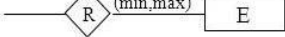
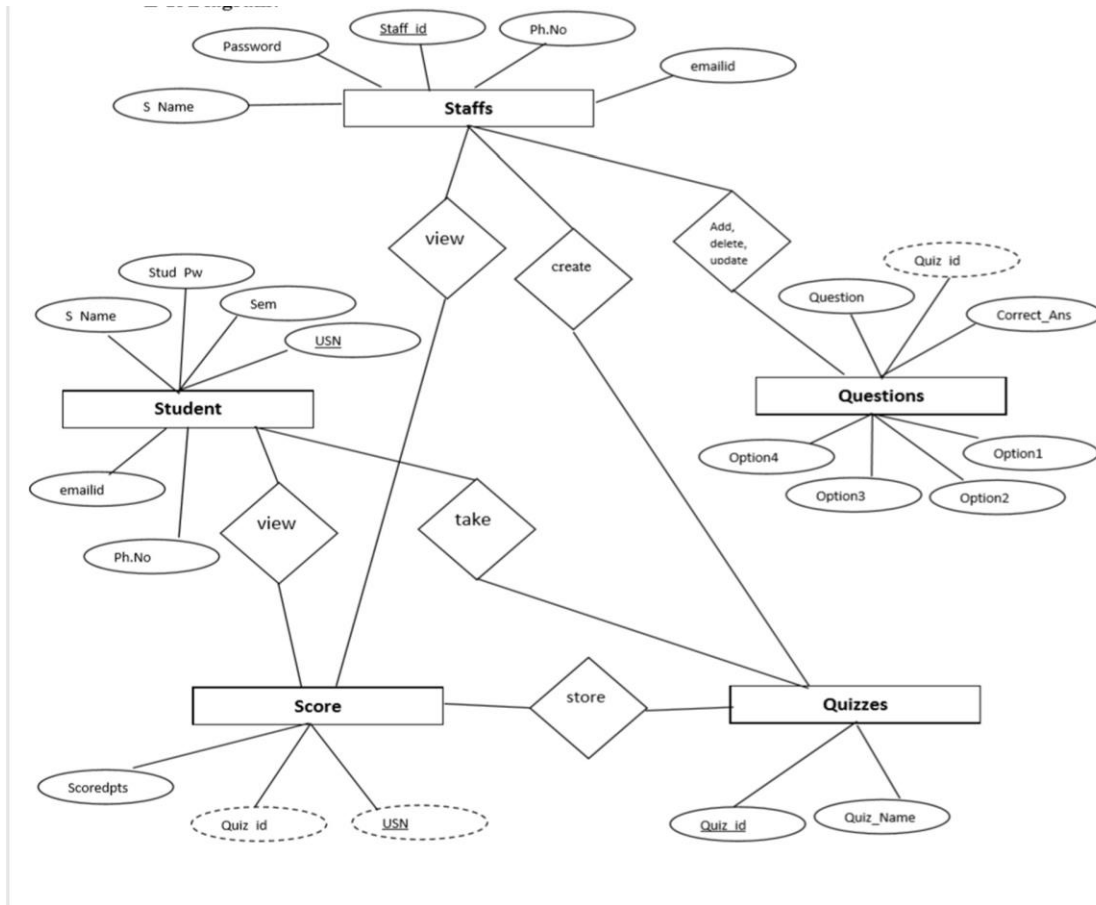
Symbol	Meaning
	ENTITY TYPE
	WEAK ENTITY TYPE
	RELATIONSHIP TYPE
	IDENTIFYING RELATIONSHIP TYPE
	ATTRIBUTE
	KEY ATTRIBUTE
	MULTIVALUED ATTRIBUTE
	COMPOSITE ATTRIBUTE
	DERIVED ATTRIBUTE
	TOTAL PARTICIPATION OF $E_2$ IN R
	CARDINALITY RATIO 1:N FOR $E_1:E_2$ IN R
	STRUCTURAL CONSTRAINT (min, max) ON PARTICIPATION OF E IN R

Fig 3.1: Notations for ER Diagrams

The ER diagram below shows the relationship between the many tables that exist in the database for the functioning of Warehouse Inventory Management System.

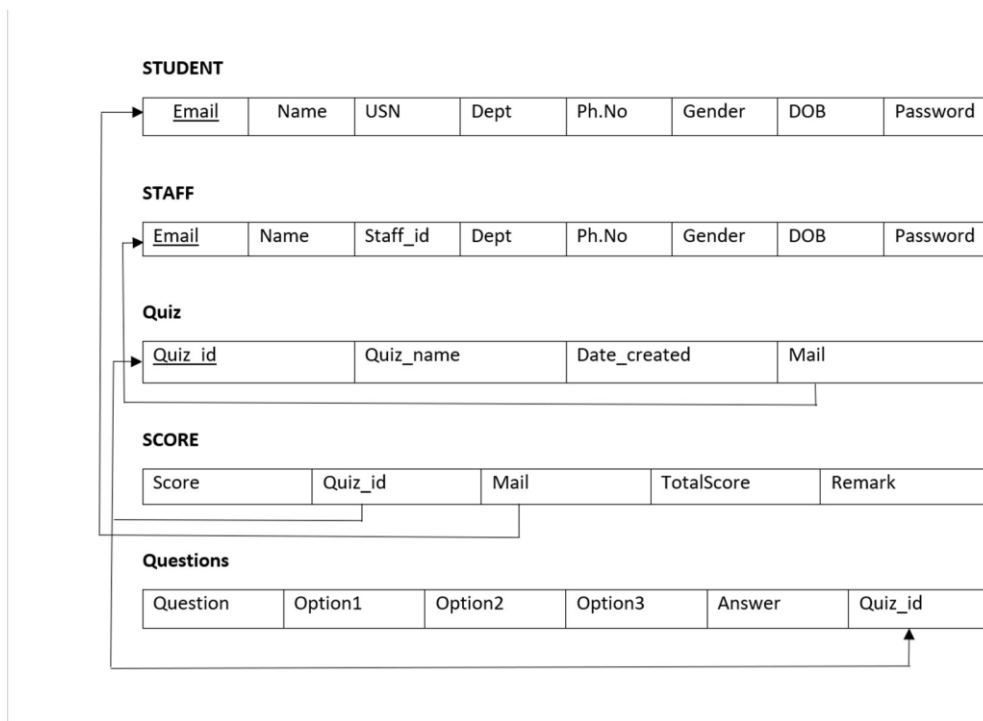


**Fig 3.2: ER Diagram of Online Examination System**

### 3.3 Schema Diagram

In any data model it is important to distinguish between the description of the database and the database itself. The description of a database is called the database schema, which is specified during database design and is not expected to change frequently.

A displayed schema is called a schema diagram. A schema diagram displays only some aspects of a schema, such as the names of record types and data items, and some types of constraints.



**Fig 3.3: Schema Diagram**

## **CHAPTER: 4**

# **IMPLEMENTATIONS**

This chapter of the report describes the Functions, packages and modules used in the project:

### **4.1 Libraries and Frameworks**

#### **PHP**

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

#### **HTML**

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.

#### **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

#### **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.

#### **PHP Mailer**

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. So, we used PHP Mailer

## **4.2 Functional Modules**

The functional modules included in the project are listed below:

### **Insert Module:**

This module provides the functionality of collecting the required data from the designed interface and transmitting it to the appropriate table present in the database designed for this project. If the provided data does not satisfy the given constraints, it must refrain from storing it into the database.

### **Update Module:**

This module again has the functionality of collecting the data from the designed interface, but it updates the already existing tuple that matches the provided primary key of the tuple to be updated, by replacing the existing attribute values with the newly collected data. Again, if the newly provided data does not satisfy the given constraints, it must refrain from updating the corresponding tuple.

### **Delete Module:**

The delete counterpart is loaded with the ability to delete a single or multiple records from the table. It searches for the tuple, in the query specified table, based on the provided value for an attribute. Admin can delete in the interface, based on which delete module searches for the record corresponding to that provided attribute value and deletes the record.

### **Retrieve Module:**

The retrieve module has a basic functionality of accessing the entire specified table from the database and displays it.

### **Trigger Module:**

Trigger in database is set of statements that are executed after an event occurs on the specified table. This is useful for logs wherein every change in database can be logged which helps keep a track of all changes/transactions on the database.

### 4.3 Source Code

```
-- phpMyAdmin SQL Dump
-- version 4.1.14
-- http://www.phpmyadmin.net
-- Host: 127.0.0.1
-- Generation Time: Nov 20, 2019 at 04:29 PM
-- Server version: 5.6.17
-- PHP Version: 5.5.12
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

-- Database: `project`

DELIMITER $$

Procedures

CREATE DEFINER=`root`@`localhost` PROCEDURE `leaderboard`()
NO SQL
select q.quizname,s.score,s.totalscore,st.name,s.mail from score s,student st,quiz q where s.mail=st.mail and
q.quizid=s.quizid order by score DESC$$

DELIMITER ;

--
-- Table structure for table `dept`
--

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

--
-- Dumping data for table `dept`
--

INSERT INTO `dept` (`dept_id`,`dept_name`) VALUES
(1, 'CSE'),
(2, 'ISE'),
(3, 'ECE'),
(4, 'EEE');
```

-- -----



```
--
-- Table structure for 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`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
-- Dumping data for table `questions`
--
```

```
INSERT INTO `questions` (`qs`, `op1`, `op2`, `op3`, `answer`, `quizid`) VALUES
('/* Assume that integers take 4 bytes.<br> #include<iostream> <br> using namespace std; <br> class Test
{ <br> static int i;<br> int j;<br> }; <br> int Test::i; <br> int main() { ', '1', '2', '3', '4', 5),
('C primarily developed as..', 'General purpose language', 'Data processing language D.', 'None of the above.',
'System programming language ', 4),
('C programs converted into machine language with the help of..', 'An Editor ', 'An operating system', 'None of
these.', 'A compiler ', 4),
('No. of consonant in english language is..', '20', '22', '28', '21', 6),
('No. of vowels in english language is..', '3', '4', '7', '5', 6),
('Total no of letters in english language is..', '23', '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',
'When an object is constructed ', 'All of the above', 5),
('Which of the following functions must use reference.', 'Assignment operator function', 'Destructor',
'Parameterized constructor', 'Copy Constructor', 5),
('Which of the following is FALSE about references in C++', 'References cannot be NULL', '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', ':: (Scope Resolution Operator)', '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', 'A constructor without any para', 'All of the above', 5),
('Who is the father of C language?', 'Bjarne Stroustrup', 'James A. Gosling ', 'Dr. E.F. Codd', 'Dennis Ritchie ',
4);
```

```
-- -----
```

```
--
-- Table structure for 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`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;

--
-- Dumping data for table `quiz`
--

INSERT INTO `quiz` (`quizid`, `quizname`, `date_created`, `mail`) VALUES
(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');

--
-- Triggers `quiz`
--
DROP TRIGGER IF EXISTS `ondeleteqs`;
DELIMITER //
CREATE TRIGGER `ondeleteqs` AFTER DELETE ON `quiz`
FOR EACH ROW delete from questions where questions.quizid=old.quizid
//
DELIMITER ;

-- -----
--
-- Table structure for 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`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=15 ;

--
-- Dumping data for table `score`
--

INSERT INTO `score` (`slno`, `score`, `quizid`, `mail`, `totalscore`, `remark`) VALUES
(13, 6, 5, 'rakeshmariyaplar1@gmail.com', 6, 'good'),
(14, 2, 4, 'rakeshmariyaplar1@gmail.com', 3, 'good');

--
-- Triggers `score`
--
DROP TRIGGER IF EXISTS `remarks`;

```

```
DELIMITER //
CREATE TRIGGER `remarks` BEFORE INSERT ON `score`
FOR EACH ROW set NEW.remark = if(NEW.score = 0, 'bad', 'good')
//
DELIMITER ;
```

-----

```
--
-- Table structure for 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`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
-- Dumping data for table `staff`
--
```

```
INSERT INTO `staff` (`staffid`, `name`, `mail`, `phno`, `gender`, `DOB`, `pw`, `dept`) VALUES
('svit1', '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');
```

-----

```
--
-- Table structure for 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`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--  
-- Dumping data for table `student`  
--
```

```
INSERT INTO `student` (`usn`,`name`,`mail`,`phno`,`gender`,`DOB`,`pw`,`dept`) VALUES  
('1va17cs010', 'B G VINAYAK', 'BHATVINAYAK94@GMAIL.COM', '9740834260', 'M', '1999-09-23',  
'ral7gku4rfhLk', 'CSE'),  
('1va17cs140', 'Rakesh Mariyaplar', 'rakeshmariyaplar1@gmail.com', '6360300095', 'M', '1999-10-07',  
'rajJYeVNCiGD2', 'CSE'),  
('1va17cs040', 'Rakesh M R', 'rakeshmr723@gmail.com', '9901735897', 'M', '2000-10-07', 'rajJYeVNCiGD2',  
'CSE'),  
('1va17cs051', 'Siddhanth Sipoliya', 'siddhanthsipoliya@saividya.ac.', '7619360459', 'M', '1999-11-15',  
'ray.whoA8HjCQ', 'CSE');
```

```
--  
-- Constraints for dumped tables  
--
```

```
--  
-- Constraints for table `quiz`  
--
```

```
ALTER TABLE `quiz`  
  ADD CONSTRAINT `quiz_ibfk_1` FOREIGN KEY (`mail`) REFERENCES `staff` (`mail`) ON DELETE  
  CASCADE;
```

```
--  
-- Constraints for table `score`  
--
```

```
ALTER TABLE `score`  
  ADD CONSTRAINT `score_ibfk_1` FOREIGN KEY (`quizid`) REFERENCES `quiz` (`quizid`) ON  
  DELETE CASCADE,  
  ADD CONSTRAINT `score_ibfk_2` FOREIGN KEY (`mail`) REFERENCES `student` (`mail`) ON  
  DELETE CASCADE ON UPDATE CASCADE;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

# Graphical User Interface

ONLINE EXAMINATION SYSTEM

☒ STUDENT ☐ STAFF

USERNAME

PASSWORD

[Forgot password?](#) [New user! SIGN UP](#)

Snapshot 1: Start page of the project

ONLINE EXAMINATION SYSTEM

☒ STUDENT ☐ STAFF

**Sign-Up as Student**

NAME

USEN

Email

Ph. No.

Department

DOB

Gender ☒ MALE ☐ FEMALE

Password

Snapshot 2: Sign Up Page

ONLINE EXAMINATION SYSTEM

DashbordprofileQuiz'sSign Out

Welcome to Online Examination System Rakesh M R

Dashbord

Add QuizDelete QuizView Quiz

Leaderboard

Quiz Title	Student name	score obtained	Max Score
c++ quiz	Rakesh Mariyaplar	6	6

Snapshot 2: Dashboard for Staff

ONLINE EXAMINATION SYSTEM

DashbordprofileScoreSign Out

Take any Quiz

Quiz Title	Created on	Created By	
c quiz	2019-11-18 21:43:50	BHATVINAYAK94@GMAIL.COM	Take_Quiz
c++ quiz	2019-11-18 21:47:13	rakeshmr723@gmail.com	Take_Quiz

Leaderboard

Quiz Title	Score	Total Score	Student name	Student Mail ID
c++ quiz	6	6	Rakesh Mariyaplar	rakeshmariyaplar1@gmail.com
c quiz	2	3	Rakesh Mariyaplar	rakeshmariyaplar1@gmail.com

Copyright©:2019

Snapshot 3: Dashboard for Students

```
1. / Assume that integers t
#include
using namespace std;
class Test {
static int i;
int j;
};
int Test::i;
int main() {
    • 1
    • 2
    • 3
    • 4

2. When a copy constructor may be called?
    • When an object of the class is
    • When an object of the class is
    • When an object is constructed
    • All of the above

3. Which of the following functions must use reference.
    • Assignment operator function
    • Destructor
    • Parameterized constructor
    • Copy Constructor

4. Which of the following is FALSE about references in C++
    • References cannot be NULL
```

Type of User : student  
NAME : Rakesh Mariyaplar  
EMAIL : rakeshmariyaplar1@gmail.com  
Ph No. : 6360300095  
USN : lval7cs140  
GENDER : M  
DOB : 1999-10-07  
Dept. : CSE

#### Snapshot 4: Profile view of Student

```
1. / Assume that integers t
#include
using namespace std;
class Test {
static int i;
int j;
};
int Test::i;
int main() {
    • 1
    • 2
    • 3
    • 4

2. When a copy constructor may be called?
    • When an object of the class is
    • When an object of the class is
    • When an object is constructed
    • All of the above

3. Which of the following functions must use reference.
    • Assignment operator function
    • Destructor
    • Parameterized constructor
    • Copy Constructor

4. Which of the following is FALSE about references in C++
    • References cannot be NULL
```

#### Scoreboard

Quiz Title	Score Obtained	Total Score
c quiz	2	3
c++ quiz	6	6

#### Snapshot 5: Scoreboard of user

```
1. / Assume that integers take 4 bytes.
#include
using namespace std;
class Test {
static int i;
int j;
};
int Test::i;
int main() {
```

- 1
- 2
- 3
- 4

2. When a copy constructor may be called?

- When an object of the class is
- When an object of the class is
- When an object is constructed
- All of the above

3. Which of the following functions must use reference.

- Assignment operator function
- Destructor
- Parameterized constructor
- Copy Constructor

4. Which of the following is FALSE about references in C++

- References cannot be NULL

### Snapshot 6: Taking Quiz by Student

Wellcome

List of Quiz added by U

Quiz id	Quiz Title	Created on
5	c++ quiz	2019-11-18 21:47:13

[Add Quiz](#) [Delete Quiz](#) [View Quiz](#)

**Add quiz**

Quiz name

[submit](#)

## Leaderboard

Quiz Title	Student name	score obtained	Max Score
c++ quiz	Rakesh Mariyaplar	6	6

### Snapshot 7: Quiz Added by The Staff



ONLINE EXAMINATION SYSTEM
Dashboard
profile
Quiz's
Sign Out

Welcome to Online Examination System Rakesh M R

### Dashbord

Add Quiz
Delete Quiz
View Quiz

Add quiz

Quiz name

### Leaderboard

Quiz Title	Student name	score obtained	Max Score
c++ quiz	Rakesh Mariyaplar	6	6

Snapshot 8: Adding quiz page by staff

ONLINE EXAMINATION SYSTEM

### Reset the Password

☒ STUDENT
☐ STAFF

EMAIL

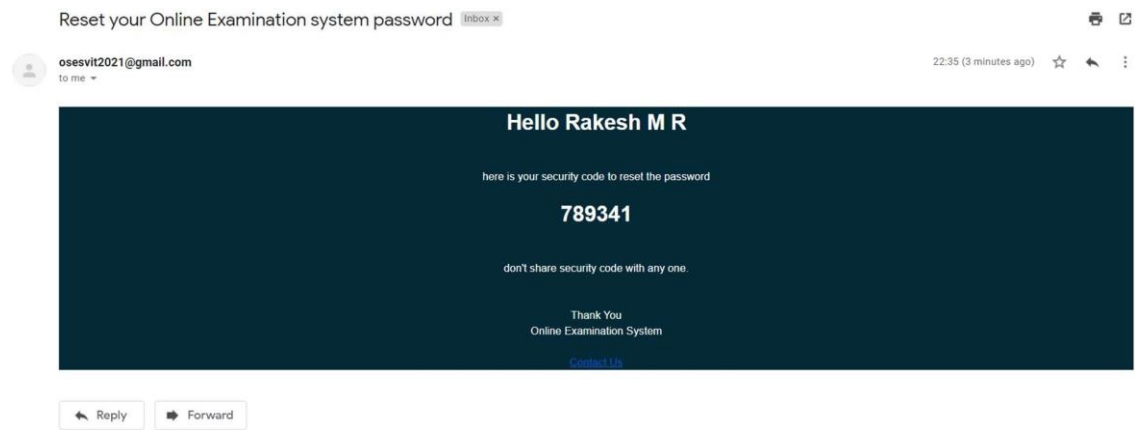
PASSWORD

PASSWORD

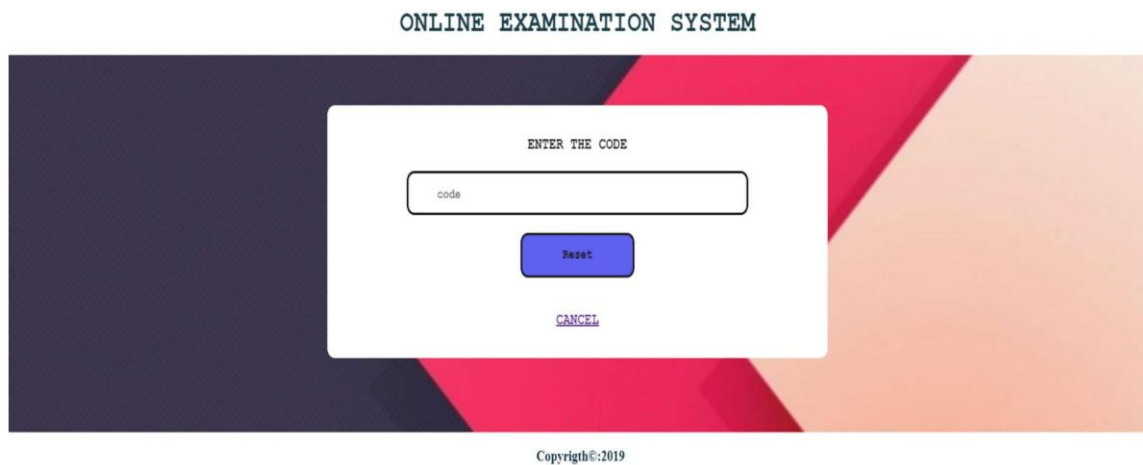
[SIGN UP](#)
[Cancel](#)

Copyright©:2019

Snapshot 9: Request for security code



**Snapshot 10: mail received by user with security code**



**Snapshot 11: verifying the security code and update password**

## **CHAPTER 6**

### **CONCLUSION**

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.
- ❑ Including Programming Questiond where user can compile or interpret on site only.

# **BIBLIOGRAPHY**

## **BOOKS REFERENCE**

- [1] Database System Models, Languages, Ramez Elmasri and Sham Kant B. Navathe, 7<sup>th</sup> Edition, 2017 Pearson.
- [2] Fundamentals of Web Development, Randy Connolly and Ricardo Hoar ,First Impression, 2016 Pearson

## **WEBSITE REFERENCES**

- [1] [www.stackoverflow.com](http://www.stackoverflow.com)
- [2] [www.youtube.com](http://www.youtube.com)
- [3] <https://www.php.net>
- [4] [www.google.com](http://www.google.com)
- [5] <https://www.w3school.com>

