

PROJECT REPORT
ON
QUIZ MANAGEMENT SYSTEM

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE

SUBMITTED BY:
TEETAS BHUIYA - 19070122181
TIRTH SHAH – 19070122182
BATCH: CS-C2



Under the Guidance of
PROF. RHYTHM BHATIA
SYMBIOSIS INSTITUTE OF TECHNOLOGY

**(A CONSTITUTENT OF SYMBIOSIS INTERNATIONAL
UNIVERSITY) PUNE - 412115**

2020 - 21

Software Requirement Specification

TABLE OF CONTENTS

1. Introduction
 - 1.1. Purpose
 - 1.2. Document Conventions
 - 1.3. Intended Audience
 - 1.4. Scope
2. Overall Description
 - 2.1. Product Perspective
 - 2.2. Product Functions
 - 2.3. User Classes and Characteristics
 - 2.4. Operating Environment
 - 2.5. Design and Implementation Constraints
 - 2.6. Assumptions and Dependencies
 - 2.7. User documentation
3. System Features
 - 3.1. Database
 - 3.1.1. Description
 - 3.1.2. Response Requests
 - 3.2. Functional Requirements
 - 3.2.1. Perform Login Validation
 - 3.2.2. Conduct Test
4. Non-functional Requirements

4.1. Performance Requirements

4.2. Safety Requirements

4.3. Security Requirements

5. Other Non-Functional Requirements

5.1. Performance Requirements

5.2. Security Requirements

5.3. Software quality attributes

6. Hardware and Software Requirements

6.1. Hardware Requirements

6.2. Software Requirements

7. Analysis Models

7.1. Use – Case Diagrams

7.2. ER diagram

8. Implementation screenshots

8.1. Code snippets

8.2. Output snippets

9. Conclusion and Future scope

9.1. Future Scope

9.2. Project outcomes

9.3. References

9.4. Conclusion

1. Introduction

1.1 Purpose

Online quiz management system is a software which is very helpful in current times when education is delivered through online channel. This software will reduce the complexity of taking tests online.

- a. It saves time as several students can together give a particular exam.
- b. The system automatically calculates the result, thus reducing the manual task of evaluator and results are available to students in no time.
- c. Administrator can add/modify/delete questions.
- d. Users must register, and they can give test with their ID, thus enabling authenticity.

1.2 Document Conventions

The following conventions are used throughout the SRS document:

- a. **Administrator** - A user with administration privileges of the software
- b. **User** - A general user of the software
- c. **Client** - Target users of the software, students/applicant in this case
- d. **Examiner** - Checks the answer of tests and gives the result.

1.3 Intended Audience

The main target audience for the software is the educational institutions which include the schools and colleges as well. It also includes corporates and MSME's intending to take surveys and customer feedback.

1.4 Scope

Scope of this software is very broad. It can be used in –

- a. Conducting online exams in institutions

- b. Conducting quiz & survey in corporate
- c. It can be accessed remotely at any location & at any time.
- d. Physical examiner need not be present

2. Overall Description

2.1 Product Perspective

Since this is at initial stage so, there is no cost for because this new idea to enhance the educational system.

So, the source code is made free for all. There are various reasons why should anyone use this program. First, it is a framework where you can create edit and store multiple choice questions.

Second is an easy and reliable testing program that is unique in its category where you can use the questions you have created in a way that represents test sessions. And third due to its open-source nature you can modify it according to your needs.

The Quiz Management System is an online exam cum survey platform. The process includes mainly 3 steps -

a. Login

The user has to login to their account when they first enter the site.

b. Test

It is the most important page of the system. It includes:

- i. Subject page to select the subject to give the exam.
- ii. After starting the test, the timer starts according to the allotted time. It has the option to skip, answer and flag the questions. It also shows the total answered and unanswered questions. Grid form is also available where user can go to any question without hassle.

2.2 Product Functions

There are 2 types of users –

a. Administrator/examiner

They have rights to do the following

- i. Create/delete user account
- ii. Change password
- iii. Can access any user account
- iv. Check the result/scores
- v. Modify/edit scores

b. Student/user account

- i. View tests available to them
- ii. View their test marks

2.3 User Classes and Characteristics

There will be 2 kinds of users that will visit the product for different reasons.

Physical actors:

- a. The administrator/examiner will visit for admin privileges mentioned in 2.2 a. and for the loading the test paper.
- b. The student/user account will visit to appear for the test and to check their marks.

System actors:

- a. Client: The client is the system that connects to the server and handles the tests based on the session and finally submits the information back to the server.
- b. Server: The server is the system that accepts multiple connections from clients and saves the results.

- c. Database Editor: The database editor handles all the question tools used to construct modify and save questions and make different tests for the students. Also, it is responsible for logging the session info and restores them for better study. The primary actor is the client that connects to the server and takes the test. The term student is to be used as a physical actor to describe the physical user of the client. The same term is used for the teacher who is the physical user of the database editor and server and is responsible for the good behaviour of the system and to make sure everything goes well.

2.4 Operating Environment

This product will work in almost all web browsers. The only basic requirement is an internet connection.

2.5 Design & Implementation Constraints

- MS SQL server will be used as SQL engine and database.
- Users may access from any computer that has Internet browsing capabilities and an Internet connection.
- Users must have their correct usernames and passwords to enter their online accounts and do actions.

2.6 Assumption and Dependencies

We assume all users have basic computer operating knowledge and our quiz system has a very interactive and user-friendly interface.

For creating the windows forms and setting up the core program my tools were used so far that effect overall rhythm of software here for better understanding the know-what to do rather than the know-how to do it we assume that the reader is not interested in knowing how to create and design windows forms and how the program is coded at that time.

2.7 User Documentation

The manual provided will guide the respective user through the product like technical details etc.

3. System Features

3.1 Database

3.1.1 Description

This database is supposed to store, retrieve, update, or edit information related to –

- Profile of both types of users
- Student/admin details
- Test results

3.1.2 Response requests

For Administrator - The admin will login and the credentials will be verified. If credentials are valid, the admin will be able to perform the tasks like add/delete student record, add exam details etc.

For user/student - The student will be given login credentials though which they can login. From the menu they can appear for tests, view scores, or view their profile.

3.2 Functional Requirements

Functional requirements the services provided to user by the product.

There are 2 subdivisions in the type of user.

- Administrator – The admin keeps track of the tests and scores and creates/modifies the student account as per requirement.
- Candidate – The candidate will login and take the test as assigned. The results will be displayed after completion of test.

3.2.1 Perform Validation

Input:

Input	Unit	Range
Username	Character	3-10
Password	Character	6-15

Output:

User will go to homepage.

3.2.2 Conduct Test

Input:

Input	Unit	Range
Question	Integer	1-99
Answer	Character	A-D or A-E

Output:

After successful evaluation & validation in the backend, the result will be displayed.

4. Non-Functional Requirements

4.1 Performance Requirements

Some performance requirements identified is listed below:

- The database shall be able to accommodate a minimum of 10,000 records of students.
- The software shall support use of multiple users at a time.

- There are no other specific performance requirements that will affect development.

4.2 Safety Requirements

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup.

4.3 Security Requirements

Some of the factors that are identified to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are described below. Keeps specific log or history data sets.

- Assign certain functions to different modules.
- Restrict communications between some areas of the program.
- Check data integrity for critical variables.
- Later version of the software will incorporate encryption techniques in the user/license authentication process.

Communication needs to be restricted when the application is validating the user or license (i.e., using https).

5. Other Non-Functional Requirements

5.1 Performance Requirements

Checking the fact that a system must perform as what every user expects.

So, in every action – response of the system, there are no immediate delays. In case of opening window forms, of popping error messages and saving the settings or sessions. Also, when connecting to the server the delay is based on the distance of the 2 systems and the configuration between them so there is high probability that there will be or not a successful connection in less than 20 seconds.

5.2 Security Requirements

This program uses object-oriented mechanisms to protect its data passed using methods also there is not currently a security schema of this program. Thus, the log files that are being created are readable using a simple text reader.

5.3 Software Quality Attributes

5.3.1 Availability

Checking that the system always has something to function and always pop-up error messages in case of component failure. In that case the error messages appear when something goes wrong so to prevail availability problems.

5.3.2 Usability

Checking that the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states.

5.3.3 Functionality

Checking that the system provides the right tools for editing question databases, creating session tests, and analysing the test sessions. In that case the tools.

6. Hardware and software Requirements

6.1 Hardware Requirements

- Systems having processors minimum 2.8.
- Printers may laser etc.

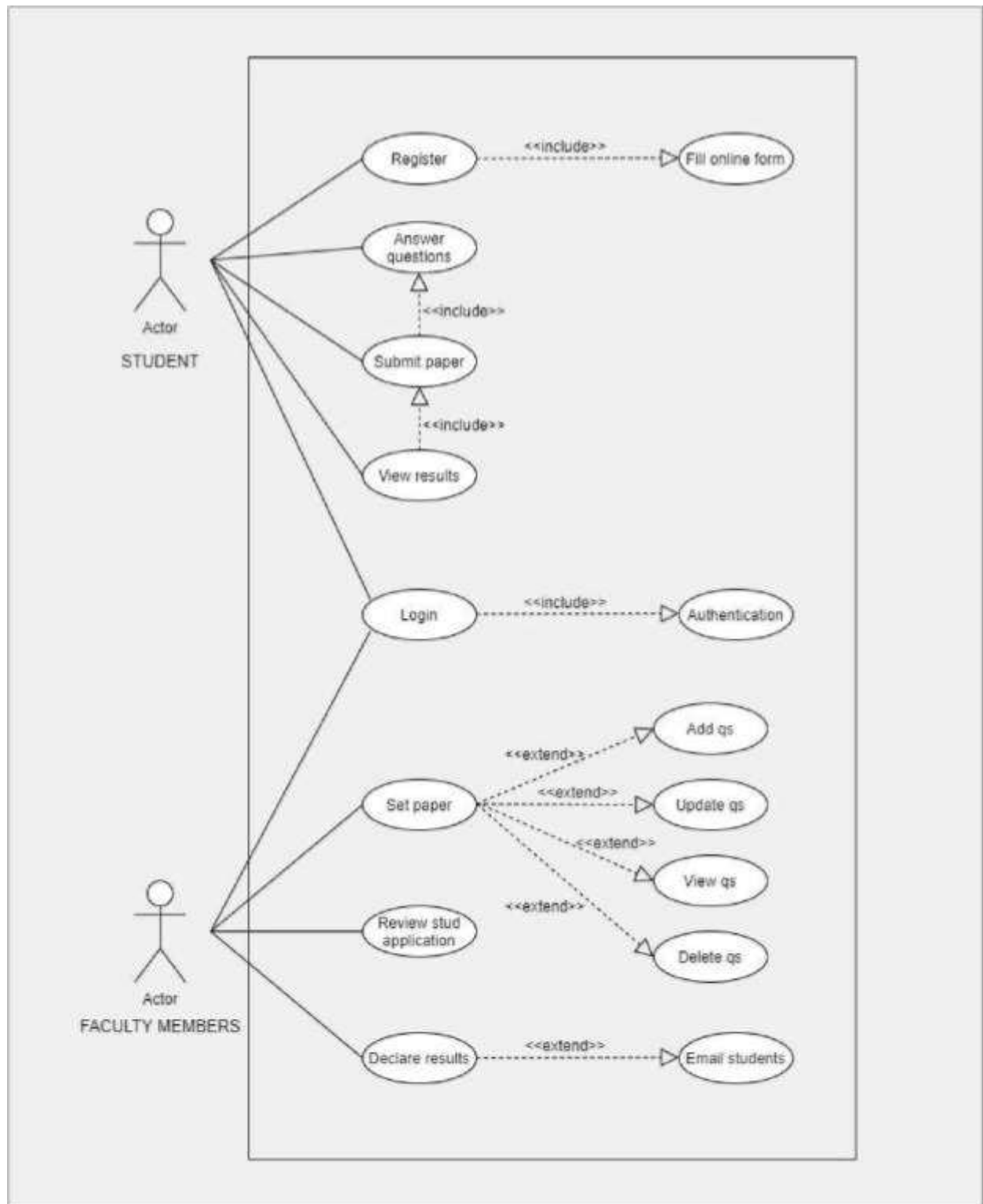
6.2 Software requirements

- Microsoft windows all versions, Linux, Mac, Solaris, Ubuntu etc.

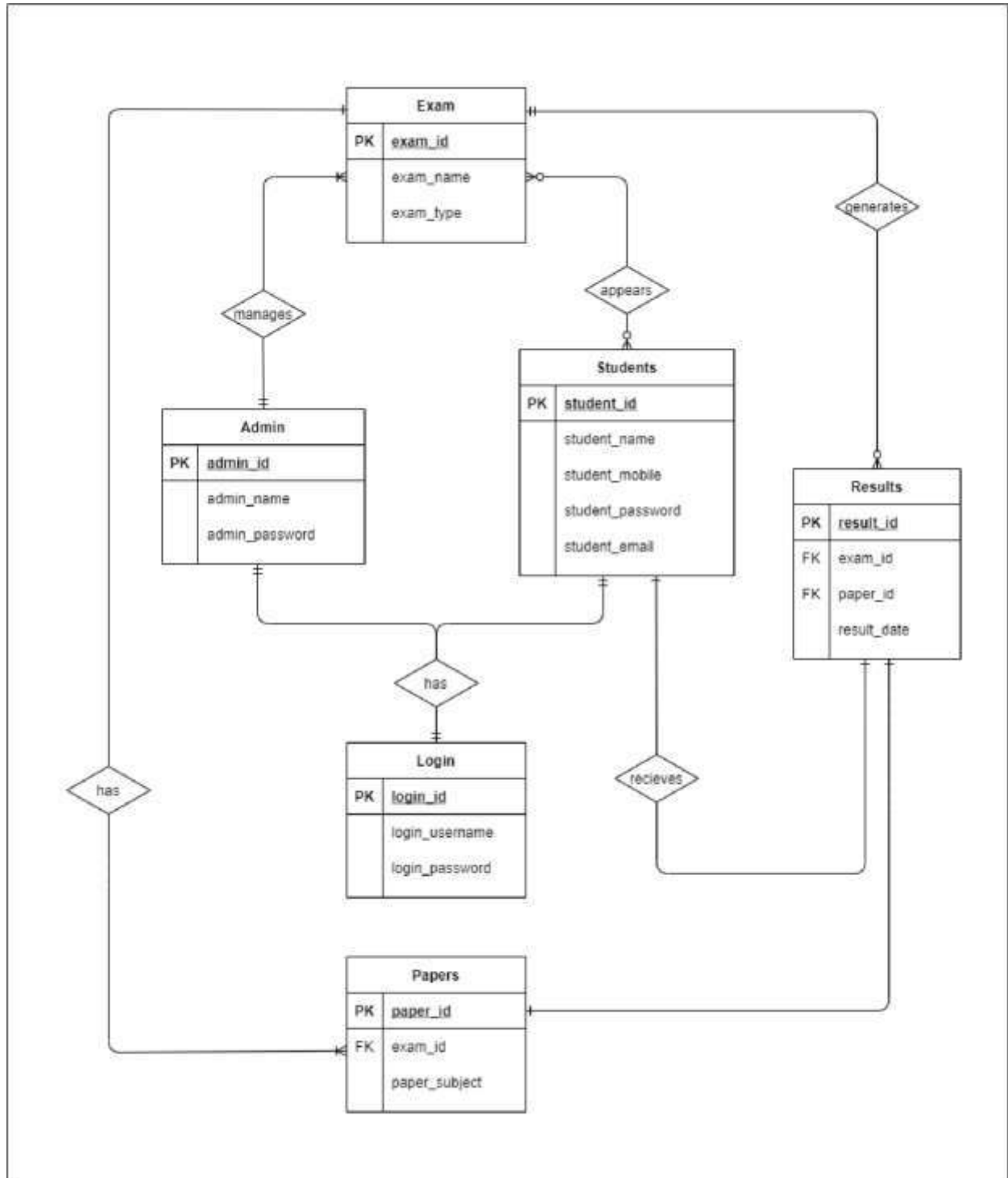
- Java
- Database with DBMS (database management system) tools as per requirements.

7. Analysis Model

7.1 Use Case diagram



7.2 ER (entity relationship) diagram:




```

import javax.swing.ImageIcon; import
javax.swing.JOptionPane;
public class loginAdmin extends javax.swing.JFrame {

    public loginAdmin() {
        initComponents();
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:        setVisible(false);
        new index().setVisible(true);
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
        if(jTextField1.getText().equals("qems") && jPasswordField1.getText().equals("admin"))
        {
            setVisible(false);
            new adminHome().setVisible(true);
        }
    else{
JOptionPane.showM
essageDia
log(null,
"INCORRE
CT
USERNAM
E OR
PASSWORD.", "Show",JOptionPane.INFORMATION_MESSAGE);
    }
}

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
        if (jCheckBox1.isSelected())
        {
            jPasswordField1.setEchoChar((char)0);

```



```

    }
else{
    jPasswordField1.setEchoChar('*');
}
}

```

//////////////////////////////////////**adminHome.java**

```

import javax.swing.JFrame; import
javax.swing.JOptionPane;

```

```

public class adminHome extends javax.swing.JFrame {
public static int open=0;

```

```

    public adminHome() {
        initComponents();
    }

```

```

    private void jMenuItem6MouseClicked(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:      JFrame jf=new JFrame();
jf.setAlwaysOnTop(true);
        int a=JOptionPane.showConfirmDialog(jf, "Do you really want to logout?", "Select",
JOptionPane.YES_NO_OPTION);
        if(a==0)
        {
            setVisible(false);
            new loginAdmin().setVisible(true);
        }
    }
}

```

```

    private void jMenuItem7MouseClicked(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:      JFrame jf=new JFrame();
jf.setAlwaysOnTop(true);
        int a=JOptionPane.showConfirmDialog(jf, "Do you really want to exit application?", "Select",
JOptionPane.YES_NO_OPTION);
        if(a==0)
        {
            System.exit(0);
        }
    }
}

```

```
private void jMenuItem1MouseClicked(java.awt.event.MouseEvent evt) {  
// TODO add your handling code here:  
    if(open==0){  
        new addNewQuestion().setVisible(true);  
open=1;  
    }  
    else{  
        JFrame jf=new JFrame();  
jf.setAlwaysOnTop(true);  
        JOptionPane.showMessageDialog(jf,"One form is already open");  
    }  
}
```

```
private void jMenuItem2MouseClicked(java.awt.event.MouseEvent evt) {  
// TODO add your handling code here:    if(open==0){  
        new updateQuestion().setVisible(true);  
open=1;  
    }  
    else{  
        JFrame jf=new JFrame();  
jf.setAlwaysOnTop(true);  
        JOptionPane.showMessageDialog(jf,"One form is already open");  
    }  
}
```

```
private void jMenuItem3MouseClicked(java.awt.event.MouseEvent evt) {  
// TODO add your handling code here:    if(open==0){  
        new allQuestion().setVisible(true);  
open=1;  
    }  
    else{  
        JFrame jf=new JFrame();  
jf.setAlwaysOnTop(true);  
        JOptionPane.showMessageDialog(jf,"One form is already open");  
    }  
}
```

```

    private void jMenuItem4MouseClicked(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
        if(open==0){
            new deleteQuestion().setVisible(true);
open=1;
        }
        else{
            JFrame jf=new JFrame();
jf.setAlwaysOnTop(true);
            JOptionPane.showMessageDialog(jf,"One form is already open");
        }
    }

```

```

    private void jMenuItem5MouseClicked(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
        if(open==0){
            new allStudentResult().setVisible(true);
open=1;
        }
        else{
            JFrame jf=new JFrame();
jf.setAlwaysOnTop(true);
            JOptionPane.showMessageDialog(jf,"One form is already open");
        }
    }
}

```

////////////////////////////////////addNewQuestion.java

```

import java.sql.*;
import Project.ConnectionProvider;
import javax.swing.JFrame; import
javax.swing.*;
public class addNewQuestion extends javax.swing.JFrame {

    public addNewQuestion() {
        initComponents();
        try{
            Connection con=ConnectionProvider.getCon();
            Statement st=con.createStatement();

```

```

        ResultSet rs=st.executeQuery("select count(id) from question");
if(rs.first())
    {
        int id=rs.getInt(1);
        id=id+1;
        String str=String.valueOf(id);
jLabel4.setText(str);
    }
    else{
        jLabel4.setText("1");
    }
}
catch(Exception e){
JFrame jf=new JFrame();
jf.setAlwaysOnTop(true);
JOptionPane.showMessageDialog(jf,e);

}
}

```

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
String id=jLabel4.getText();
String name=jTextField1.getText();
String opt1=jTextField2.getText();
String opt2=jTextField3.getText();
String opt3=jTextField4.getText();
String opt4=jTextField5.getText();
String answer=jTextField6.getText();
try{
    Connection con=ConnectionProvider.getCon();
    PreparedStatement ps=con.prepareStatement("insert into question values(?,?,?,?,?,?,?)");
ps.setString(1,id);      ps.setString(2,name);      ps.setString(3,opt1);
ps.setString(4,opt2);
    ps.setString(5,opt3);
ps.setString(6,opt4);
ps.setString(7,answer);
    ps.executeUpdate();
}
}

```

```

        JFrame jf=new JFrame();
jf.setAlwaysOnTop(true);
        JOptionPane.showMessageDialog(jf, "Successfully Updated");
setVisible(false);
        new addNewQuestion().setVisible(true);
    }
    catch(Exception e){
JFrame jf=new JFrame();
jf.setAlwaysOnTop(true);
        JOptionPane.showMessageDialog(jf, e);
    }

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:    adminHome.open=0;
    setVisible(false);
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    jTextField1.setText("");
jTextField2.setText("");    jTextField3.setText("");
jTextField4.setText("");    jTextField5.setText("");
jTextField6.setText("");
}

```

////////////////////////////////////updateQuestion.java

```

import java.sql.*;
import Project.ConnectionProvider; import
javax.swing.*;
public class updateQuestion extends javax.swing.JFrame {

    public updateQuestion() {
        initComponents();
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:    adminHome.open=0;

```

```

        setVisible(false);
    }
    private void
    jButton1ActionPerformed
    (java.awt.event.ActionEvent evt) {
        // TODO add your
        handling code here:
        String
        id=jTextField1.getText();
        try{
            Connection con=ConnectionProvider.getCon();
            Statement st=con.createStatement();
            ResultSet rs=st.executeQuery("select *from question where id='"+id+"'");
            if(rs.first())
            {
                jTextField2.setText(rs.getString(2));
                jTextField3.setText(rs.getString(3));
                jTextField4.setText(rs.getString(4));
                jTextField5.setText(rs.getString(5));
                jTextField6.setText(rs.getString(6));
                jTextField7.setText(rs.getString(7));                jTextField1.setEditable(false);
            }
            else{
                JFrame jf=new JFrame();
                jf.setAlwaysOnTop(true);
                JOptionPane.showMessageDialog(jf,"Question id does not exist!");
            }
        }
        catch(Exception e){
            JFrame jf=new JFrame();
            jf.setAlwaysOnTop(true);
            JOptionPane.showMessageDialog(jf,e);
        }
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String id=jTextField1.getText();
        String name=jTextField2.getText();
    }

```

```

String opt1=jTextField3.getText();
String opt2=jTextField4.getText();
String opt3=jTextField5.getText();
String opt4=jTextField6.getText();
String answer=jTextField7.getText();
try{
    Connection con=ConnectionProvider.getCon();
    PreparedStatement ps=con.prepareStatement("update question set
name=?,opt1=?,opt2=?,opt3=?,opt4=?,answer=? where id=?");
ps.setString(1, name);      ps.setString(2, opt1);
ps.setString(3, opt2);      ps.setString(4, opt3);
    ps.setString(5, opt4);
ps.setString(6, answer);
ps.setString(7, id);
ps.executeUpdate();        JFrame
jf=new JFrame();
jf.setAlwaysOnTop(true);
    JOptionPane.showMessageDialog(jf,"successfully updated!");
setVisible(false);
    new updateQuestion().setVisible(true);
}
catch(Exception e){
JFrame jf=new JFrame();
jf.setAlwaysOnTop(true);
    JOptionPane.showMessageDialog(jf,e);
}
}

```

```

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    jTextField1.setText("");
jTextField2.setText("");    jTextField3.setText("");
jTextField4.setText("");    jTextField5.setText("");
jTextField6.setText("");    jTextField7.setText("");
jTextField1.setEditable(true);
}

```

//allQuestion.java

```
import java.sql.*;
```

```

import Project.ConnectionProvider;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import net.proteanit.sql.DbUtils;
public class allQuestion extends javax.swing.JFrame {

    public allQuestion() {
        initComponents();
        try{
            Connection con=ConnectionProvider.getCon();
            Statement st=con.createStatement();
            ResultSet rs=st.executeQuery("select *from question");
jTable1.setModel(DbUtils.resultSetToTableModel(rs));
        }
        catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
adminHome.open=0;
        setVisible(false);
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
String id=jTextField1.getText();    try{
        Connection con=ConnectionProvider.getCon();
        Statement st=con.createStatement();
        ResultSet rs=st.executeQuery("select *from question where id='"+id+"'");
if(rs.first())
    {
        jTextField2.setText(rs.getString(2));
jTextField3.setText(rs.getString(3));
jTextField4.setText(rs.getString(4));
jTextField5.setText(rs.getString(5));
    }
}
    }

```



```

jTextField6.setText(rs.getString(6));
jTextField7.setText(rs.getString(7));          jTextField1.setEditable(false);
    }
    else{
        JFrame jf=new JFrame();
jf.setAlwaysOnTop(true);
        JOptionPane.showMessageDialog(jf,"Question id does not exist!");
    }
}
    catch(Exception e){
JFrame jf=new JFrame();
jf.setAlwaysOnTop(true);
        JOptionPane.showMessageDialog(jf,e);
    }
}

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:      adminHome.open=0;
        setVisible(false);
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:      String id=jTextField1.getText();      try{
        Connection con=ConnectionProvider.getCon();
        PreparedStatement ps=con.prepareStatement("delete from question where id=?");
ps.setString(1, id);      ps.executeUpdate();      JFrame jf=new JFrame();
jf.setAlwaysOnTop(true);
        JOptionPane.showMessageDialog(jf,"Successfully deleted!");
setVisible(false);
        new deleteQuestion().setVisible(true);
    }
    catch(Exception e){
JFrame jf=new JFrame();
jf.setAlwaysOnTop(true);
        JOptionPane.showMessageDialog(jf,e);
    }
}

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

```

```

        jTextField1.setText("");
jTextField2.setText("");    jTextField3.setText("");
jTextField4.setText("");    jTextField5.setText("");
jTextField6.setText("");    jTextField7.setText("");
jTextField1.setEditable(true);
    }

```

//allStudentResult.java

```

import java.sql.*; import
Project.ConnectionProvider;
import javax.swing.JOptionPane;
import
javax.swing.table.DefaultTableModel;
l;
import net.proteanit.sql.DbUtils; import
javax.swing.JFrame;
public class allStudentResult extends javax.swing.JFrame {

```

```

    public allStudentResult() {
        initComponents();
        try{
            Connection con=ConnectionProvider.getCon();
            Statement st=con.createStatement();
            ResultSet rs=st.executeQuery("select *from student");
jTable1.setAutoResizeMode(jTable1.AUTO_RESIZE_OFF);
jTable1.setModel(DbUtils.resultSetToTableModel(rs));
        }
        catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
    }

```

```

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here
adminHome.open=0;
        setVisible(false);
    }

```

```

    private void jTextField1KeyReleased(java.awt.event.KeyEvent evt) {
// TODO add your handling code here:      int marks;
        if (jTextField1.getText().equals(""))
marks=0;      else
            marks=Integer.parseInt(jTextField1.getText());

try{
    Connection con=ConnectionProvider.getCon();
    Statement st=con.createStatement();
    ResultSet rs=st.executeQuery("select *from student where marks >= "+marks+"");
jTable1.setAutoResizeMode(jTable1.AUTO_RESIZE_OFF);
jTable1.setModel(DbUtils.resultSetToTableModel(rs));
    }
    catch(Exception e){
JFrame jf=new JFrame();
jf.setAlwaysOnTop(true);
    JOptionPane.showMessageDialog(null, e);
    }
    }
}

```

////////////////////////////////////studentDetails.java

```

import java.text.SimpleDateFormat; import
java.sql.*;
import Project.ConnectionProvider; import
java.awt.Color; import java.util.Date;
import javax.swing.JOptionPane; import
java.util.Properties; import
javax.mail.Message; import
javax.mail.Session; import
javax.mail.PasswordAuthentication; import
javax.mail.internet.InternetAddress; import
javax.mail.internet.MimeMessage; import
javax.mail.Transport;
public class studentDetails extends javax.swing.JFrame {

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:      setVisible(false);
        new index().setVisible(true);
    }
}

```

```
}
```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
// TODO add your handling code here:    int a = JOptionPane.showConfirmDialog(null, "Do you  
really want to exit?", "Select", JOptionPane.YES_NO_OPTION);    if(a==0)  
    {  
        System.exit(0);  
    }  
}
```

```
private void jTextField7FocusGained(java.awt.event.FocusEvent evt) {  
// TODO add your handling code here:    if(jTextField7.getText().equals("Enter University  
name"))  
    {  
        jTextField7.setText("");  
        jTextField7.setForeground(new Color(0,0,0));  
    }  
}
```

```
private void jTextField10FocusGained(java.awt.event.FocusEvent evt) {  
// TODO add your handling code here:    if(jTextField10.getText().equals("Enter University  
name"))  
    {  
        jTextField10.setText("");  
        jTextField10.setForeground(new Color(0,0,0));  
    }  
}
```

```
private void jTextField13FocusGained(java.awt.event.FocusEvent evt) {  
// TODO add your handling code here:    if(jTextField13.getText().equals("Enter University  
name"))  
    {  
        jTextField13.setText("");  
        jTextField13.setForeground(new Color(0,0,0));  
    }  
}
```

```
private void jTextField8FocusGained(java.awt.event.FocusEvent evt) {  
// TODO add your handling code here:    if(jTextField8.getText().equals("Enter Percentage"))
```

```
{
    jTextField8.setText("");
    jTextField8.setForeground(new Color(0,0,0));
}
}
```

```
private void jTextField11FocusGained(java.awt.event.FocusEvent evt) {
// TODO add your handling code here:    if(jTextField11.getText().equals("Enter Percentage"))
{
    jTextField11.setText("");
    jTextField11.setForeground(new Color(0,0,0));
}
}
```

```
private void jTextField14FocusGained(java.awt.event.FocusEvent evt) {
// TODO add your handling code here:    if(jTextField14.getText().equals("Enter Percentage"))
{
    jTextField14.setText("");
    jTextField14.setForeground(new Color(0,0,0));
}
}
```

```
private void jTextField9FocusGained(java.awt.event.FocusEvent evt) {
// TODO add your handling code here:    if(jTextField9.getText().equals("Enter Passout
Year"))
{
    jTextField9.setText("");
    jTextField9.setForeground(new Color(0,0,0));
}
}
```

```
private void jTextField12FocusGained(java.awt.event.FocusEvent evt) {
// TODO add your handling code here:    if(jTextField12.getText().equals("Enter Passout
Year"))
{
    jTextField12.setText("");
    jTextField12.setForeground(new Color(0,0,0));
}
}
```

```
private void jTextField15FocusGained(java.awt.event.FocusEvent evt) {  
// TODO add your handling code here:    if(jTextField15.getText().equals("Enter Passout  
Year"))  
{  
    jTextField15.setText("");  
    jTextField15.setForeground(new Color(0,0,0));  
}  
}
```

```
private void jTextField7FocusLost(java.awt.event.FocusEvent evt) {  
// TODO add your handling code here:  
    if(jTextField7.getText().equals(""))  
    {  
        jTextField7.setText("Enter University name");  
jTextField7.setForeground(new Color(153,153,153));  
    }  
}
```

```
private void jTextField10FocusLost(java.awt.event.FocusEvent evt) {  
// TODO add your handling code here:    if(jTextField10.getText().equals(""))  
{  
    jTextField10.setText("Enter University name");  
jTextField10.setForeground(new Color(153,153,153));  
}  
}
```

```
private void jTextField13FocusLost(java.awt.event.FocusEvent evt) {  
// TODO add your handling code here:    if(jTextField13.getText().equals(""))  
{  
    jTextField13.setText("Enter University name");  
jTextField13.setForeground(new Color(153,153,153));  
}  
}
```

```
private void jTextField8FocusLost(java.awt.event.FocusEvent evt) {  
// TODO add your handling code here:  
    if(jTextField8.getText().equals(""))  
    {  
        jTextField8.setText("Enter Percentage");  
    }  
}
```

```

        jTextField8.setForeground(new Color(153,153,153));
    }
}

private void jTextField11FocusLost(java.awt.event.FocusEvent evt) {
    // TODO add your handling code here:
    if(jTextField11.getText().equals(""))
    {
        jTextField11.setText("Enter Percentage");
        jTextField11.setForeground(new Color(153,153,153));
    }
}

private void jTextField14FocusLost(java.awt.event.FocusEvent evt) {
// TODO add your handling code here:    if(jTextField14.getText().equals(""))
{
    jTextField14.setText("Enter Percentage");
    jTextField14.setForeground(new Color(153,153,153));
}
}

private void jTextField9FocusLost(java.awt.event.FocusEvent evt) {
// TODO add your handling code here:
    if(jTextField9.getText().equals(""))
    {
        jTextField9.setText("Enter Passout Year");
        jTextField9.setForeground(new Color(153,153,153));
    }
}

private void jTextField12FocusLost(java.awt.event.FocusEvent evt) {
// TODO add your handling code here:    if(jTextField12.getText().equals(""))
{
    jTextField12.setText("Enter Passout Year");
    jTextField12.setForeground(new Color(153,153,153));
}
}

private void jTextField15FocusLost(java.awt.event.FocusEvent evt) {
// TODO add your handling code here:    if(jTextField15.getText().equals(""))

```

```

{
    jTextField15.setText("Enter Passout Year");
    jTextField15.setForeground(new Color(153,153,153));
}
}

```

```

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

```

```

    String rollNo=jTextField1.getText();
    String name=jTextField2.getText();
    String fatherName=jTextField3.getText();
    String motherName=jTextField4.getText();
    String gender=(String)jComboBox1.getSelectedItem();
    String contactNo=jTextField5.getText();
    String email=jTextField6.getText(); //TOemail
    String tenthUniversityName=jTextField7.getText();
    String tenthPercentage=jTextField8.getText();
    String tenthPassoutYear=jTextField9.getText();
    String twelveUniversityName=jTextField10.getText();
    String twelvePercentage=jTextField11.getText();
    String twelvePassoutYear=jTextField12.getText();
    String graduationUniversityName=jTextField13.getText();
    String graduationPercentage=jTextField14.getText();
    String graduationPassoutYear=jTextField15.getText();
    String address=jTextField16.getText();
    String marks="0";

```

```

try{
    Connection con=ConnectionProvider.getCon();
    PreparedStatement ps=con.prepareStatement("insert into student
values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
    ps.setString(1, rollNo);      ps.setString(2,
name);      ps.setString(3, fatherName);
ps.setString(4, motherName);      ps.setString(5,
gender);      ps.setString(6, contactNo);
ps.setString(7, email);      ps.setString(8,
tenthUniversityName);      ps.setString(9,

```



```

tenthPercentage);      ps.setString(10,
tenthPassoutYear);      ps.setString(11,
twelveUniversityName);    ps.setString(12,
twelvePercentage);      ps.setString(13,
twelvePassoutYear);      ps.setString(14,
graduationUniversityName);    ps.setString(15,
graduationPercentage);    ps.setString(16,
graduationPassoutYear);

        ps.setString(17, address);
ps.setString(18, marks);
ps.executeUpdate();

        setVisible(false);
        new instructionStudent(rollNo).setVisible(true);

    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null,e);
    }

    //String email=jTextField6.getText(); //TOemail
    String fromEmail="examinationmanager@gmail.com";
    String fromEmailPassword="pblproject1";
    String subject="Online Examination alert";

    Properties properties=new Properties();
    properties.put("mail.smtp.auth","true");
    properties.put("mail.smtp.starttls.enable", "true");
    properties.put("mail.smtp.host", "smtp.gmail.com");
    properties.put("mail.smtp.port", 465);    properties.put("mail.smtp.debug",
"true");    properties.put("mail.smtp.socketFactory.port", 465);
        properties.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
    properties.put("mail.smtp.socketFactory.fallback", "false");

    Session session=Session.getDefaultInstance(properties,new javax.mail.Authenticator() {
protected PasswordAuthentication getPasswordAuthentication(){          return new
PasswordAuthentication(fromEmail, fromEmailPassword);

    }
    });
    try{

        MimeMessage message=new MimeMessage(session);
        message.setFrom(new InternetAddress(fromEmail));

```

```

        message.addRecipient(Message.RecipientType.TO, new InternetAddress(email));
message.setSubject(subject);
        message.setText("Hi, online examination alert message!");
message.saveChanges();
        Transport transport= session.getTransport("smtp");
transport.connect("smtp.gmail.com",fromEmail,fromEmailPassword);
transport.sendMessage(message, message.getAllRecipients());        transport.close();
    }
    catch(Exception e){
System.out.println(""+e);
    }

```

////////////////////////////////////instructionStudent.java

```
import javax.swing.JOptionPane;
```

```

public class instructionStudent extends javax.swing.JFrame {
public String rollNo;
    public String email;
    public instructionStudent() {
initComponents();
        jTextArea1.setEditable(false);
    }

```

```

        public instructionStudent(String rollNo1) {
initComponents();
jTextArea1.setEditable(false);        rollNo=rollNo1;
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
        setVisible(false);
        new quizExamStudent(rollNo).setVisible(true);
    }

```

////////////////////////////////////quizExamStudent.java

```

import Project.ConnectionProvider;
import java.sql.*; import
java.awt.event.ActionEvent; import

```

```

java.awt.event.ActionListener; import
java.util.Date; import
javax.swing.Timer; import
java.text.SimpleDateFormat; import
javax.swing.JOptionPane;
public class quizExamStudent extends javax.swing.JFrame {
    public String questionId="1";
    public String answer;
    public int min=0;
    public int sec=0;
    public int marks=0;
    public String email2;
    public String rollNo;

    public void answerCheck()
    {
        String studentAnswer="";
        if(jRadioButton1.isSelected()){
            studentAnswer=jRadioButton1.getText();
        }
        else if(jRadioButton2.isSelected()){
            studentAnswer=jRadioButton2.getText();
        }
        else if(jRadioButton3.isSelected()){
            studentAnswer=jRadioButton3.getText();
        }
        else{
            studentAnswer=jRadioButton4.getText();
        }

        if(studentAnswer.equals(answer)){
            marks=marks+3;
            String marks1=String.valueOf(marks);
            //    jLabel18.setText(marks1);
        }

        //question num change
        int questionId1=Integer.parseInt(questionId);
        questionId1=questionId1+1;
        questionId=String.valueOf(questionId1);
    }
}

```

```

        //clear radiobuttons
jRadioButton1.setSelected(false);
jRadioButton2.setSelected(false);
jRadioButton3.setSelected(false);
jRadioButton4.setSelected(false);

        //last question.. hide next button
if(questionId.equals("10")){
    jButton1.setVisible(false);
}
}
public void question()
{
try{
    Connection con=ConnectionProvider.getCon();
    Statement st=con.createStatement();

    ResultSet rs1=st.executeQuery("select *from question where id='"+questionId+"'");
    while(rs1.next()){
        jLabel16.setText(rs1.getString(1));
jLabel19.setText(rs1.getString(2));        jRadioButton1.setText(rs1.getString(3));
jRadioButton2.setText(rs1.getString(4));        jRadioButton3.setText(rs1.getString(5));
jRadioButton4.setText(rs1.getString(6));        answer=rs1.getString(7);
    }
}
catch(Exception e){
    JOptionPane.showMessageDialog(null, e);
}
}
public void submit()
{
    String rollNo=jLabel11.getText();
answerCheck();
    try{
        Connection con=ConnectionProvider.getCon();
        Statement st=con.createStatement();
        st.executeUpdate("update student set marks= '"+marks+"' where rollNo='"+rollNo+"'");
String marks1=String.valueOf(marks);

```

```

        setVisible(false);
        new successfullySubmitted(marks1).setVisible(true);
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
}

/**
 * Creates new form quizExamStudent
 */
public quizExamStudent() {
    initComponents();

    public quizExamStudent(String rollNo, String email1){
email2=email1;
    }

    //successfullySubmitted(rollNo, email2);

    Timer time;
    public quizExamStudent(String rollNo) {
        initComponents();
        jLabel11.setText(rollNo);

        //date
        SimpleDateFormat dFormat=new SimpleDateFormat("dd-MM-yyyy");
        Date date=new Date();
        jLabel3.setText(dFormat.format(date));

        //first question and stud details
    try{
        Connection con=ConnectionProvider.getCon();
        Statement st=con.createStatement();
        ResultSet rs=st.executeQuery("select *from student where rollNo='"+rollNo+"'");
        while(rs.next()){
            jLabel10.setText(rs.getString(2));
        }
    }

```

```

        ResultSet rs1=st.executeQuery("select *from question where id='"+questionId+"'");
while(rs1.next()){
    jLabel16.setText(rs1.getString(1));
jLabel19.setText(rs1.getString(2));
    jLabelRadioButton1.setText(rs1.getString(3));
        jLabelRadioButton2.setText(rs1.getString(4));
jRadioButton3.setText(rs1.getString(5));
jRadioButton4.setText(rs1.getString(6));
        answer=rs1.getString(7);
    }
}
catch(Exception e){
    JOptionPane.showMessageDialog(null, e);
}

//timer
setLocationRelativeTo(this);
time=new Timer(1000,new ActionListener(){
    @Override
    public void
actionPerformed(ActionEvent e){
jLabel8.setText(String.valueOf(sec));
        jLabel7.setText(String.valueOf(min));

        if(sec==60){
sec=0;
            min++;
if(min==10){
time.stop();
answerCheck();
submit();
        }
    }
        sec++;

    }
});
time.start();
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    answerCheck();
    question();
}

```

```

        private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:      int a=JOptionPane.showConfirmDialog(null, "Do you
really want to submit?", "Select",JOptionPane.YES_NO_OPTION);      if(a==0){
answerCheck();
        submit();
    }

}

```

```

        private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:      if(jRadioButton1.isSelected()){
jRadioButton2.setSelected(false);      jRadioButton3.setSelected(false);
jRadioButton4.setSelected(false);
    }
}

```

```

        private void jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:      if(jRadioButton2.isSelected()){
jRadioButton1.setSelected(false);      jRadioButton3.setSelected(false);
jRadioButton4.setSelected(false);
    }

}

```

```

        private void jRadioButton3ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:      if(jRadioButton3.isSelected()){
jRadioButton2.setSelected(false);      jRadioButton1.setSelected(false);
jRadioButton4.setSelected(false);
    }
}

```

```

        private void jRadioButton4ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:      if(jRadioButton4.isSelected()){
jRadioButton2.setSelected(false);      jRadioButton3.setSelected(false);
jRadioButton1.setSelected(false);
}

```

///successfullySubmitted.java

```

import java.text.SimpleDateFormat; import
java.sql.*;

```

```

import Project.ConnectionProvider; import
java.awt.Color; import java.util.Date;
import javax.swing.JOptionPane; import
java.util.Properties; import
javax.mail.Message; import
javax.mail.Session; import
javax.mail.PasswordAuthentication; import
javax.mail.internet.InternetAddress; import
javax.mail.internet.MimeMessage; import
javax.mail.Transport;
public class successfullySubmitted extends javax.swing.JFrame {
public String toEmail;

    public successfullySubmitted() {
initComponents();
    }

    public successfullySubmitted(String marks5) {
initComponents();
//    String marks1="Marks Obtained: "+marks5;
        jLabel1.setText(marks5);

        String result=marks5;
//    toEmail=email2;
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

        String toEmail=jTextField1.getText();

        String fromEmail="examinationmanager@gmail.com";
        String fromEmailPassword="pblproject1";
        String subject="Online Examination result alert";

        Properties properties=new Properties();
properties.put("mail.smtp.auth","true");
properties.put("mail.smtp.starttls.enable", "true");
properties.put("mail.smtp.host", "smtp.gmail.com");
properties.put("mail.smtp.port", 465);    properties.put("mail.smtp.debug",
"true");    properties.put("mail.smtp.socketFactory.port", 465);

```



```
properties.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
properties.put("mail.smtp.socketFactory.fallback", "false");
```

```
Session session=Session.getDefaultInstance(properties,new javax.mail.Authenticator() {
protected PasswordAuthentication getPasswordAuthentication(){ return new
PasswordAuthentication(fromEmail, fromEmailPassword);
}
});
try{
    MimeMessage message=new MimeMessage(session);
message.setFrom(new InternetAddress(fromEmail));
    message.addRecipient(Message.RecipientType.TO, new InternetAddress(toEmail));
message.setSubject(subject);
    message.setText("Hi, online examination result message!\nYou have scored
"+jLabel1.getText()+" marks.\nThank you for registering!");
message.saveChanges();
    Transport transport= session.getTransport("smtp");
transport.connect("smtp.gmail.com",fromEmail,fromEmailPassword);
transport.sendMessage(message, message.getAllRecipients());    transport.close();
}
catch(Exception e){
    System.out.println(""+e);
}

setVisible(false);
new index().setVisible(true);
}
}
```

////////////////////////////////////ConnectionProvider.java

```
package Project;
import java.sql.*;
```

```
public class ConnectionProvider {
    public static Connection getCon()
    {
        try{
            Class.forName("com.mysql.jdbc.Driver");
```

```

        Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/qems","root","Symbiosis@123");
        return con;
    }
    catch(Exception e){
        return null;
    }
}
}

```

//Quizmanagement.java

```
package quiz.management;
```

```

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.StackPane;

```

```
public class QuizManagement extends Application {
```

```
    @Override
```

```
    public void start(Stage primaryStage) {
```

```
        Button btn = new Button();
```

```
        btn.setText("Say 'Hello World'");
```

```
        btn.setOnAction(new EventHandler<ActionEvent>() {
```

```
            @Override
```

```
            public void handle(ActionEvent event) {
```

```
                System.out.println("Hello World!");
```

```
            }
```

```
        });
```

```
        StackPane root = new StackPane();
```

```
        root.getChildren().add(btn);
```

```
Scene scene = new Scene(root, 300, 250);
```

```
primaryStage.setTitle("Hello World!");
```

```
primaryStage.setScene(scene);
```

```
primaryStage.show();
```

```
}
```

```
}
```

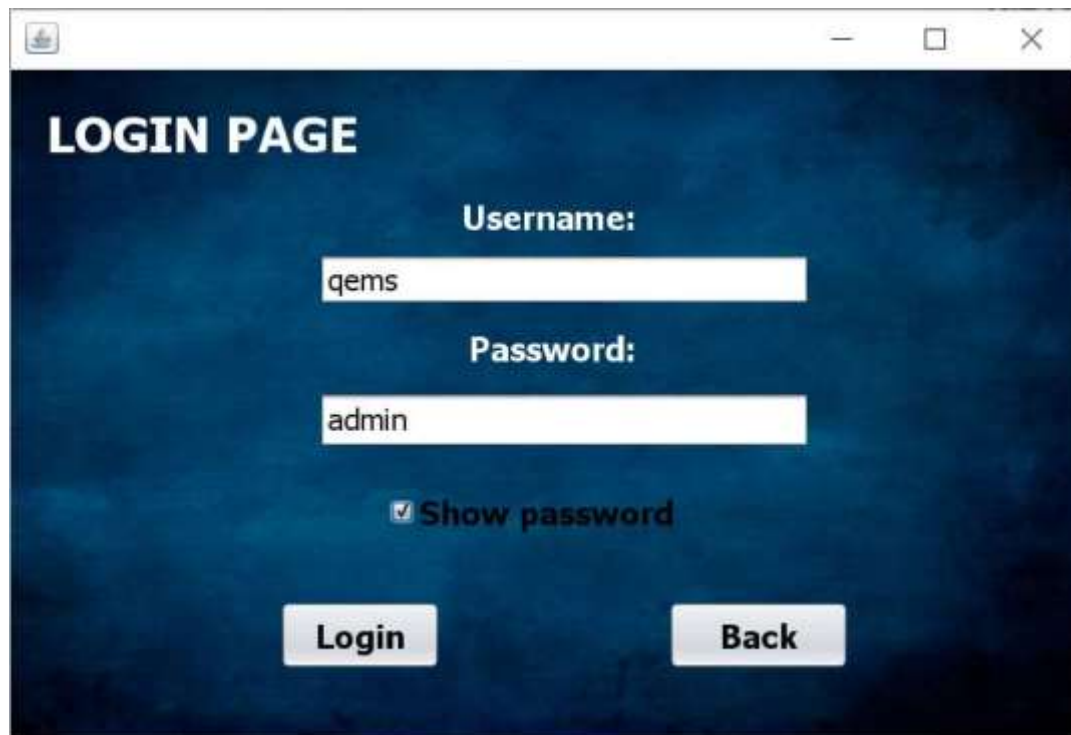
(output snippets next page)

8.2 Output Snippets

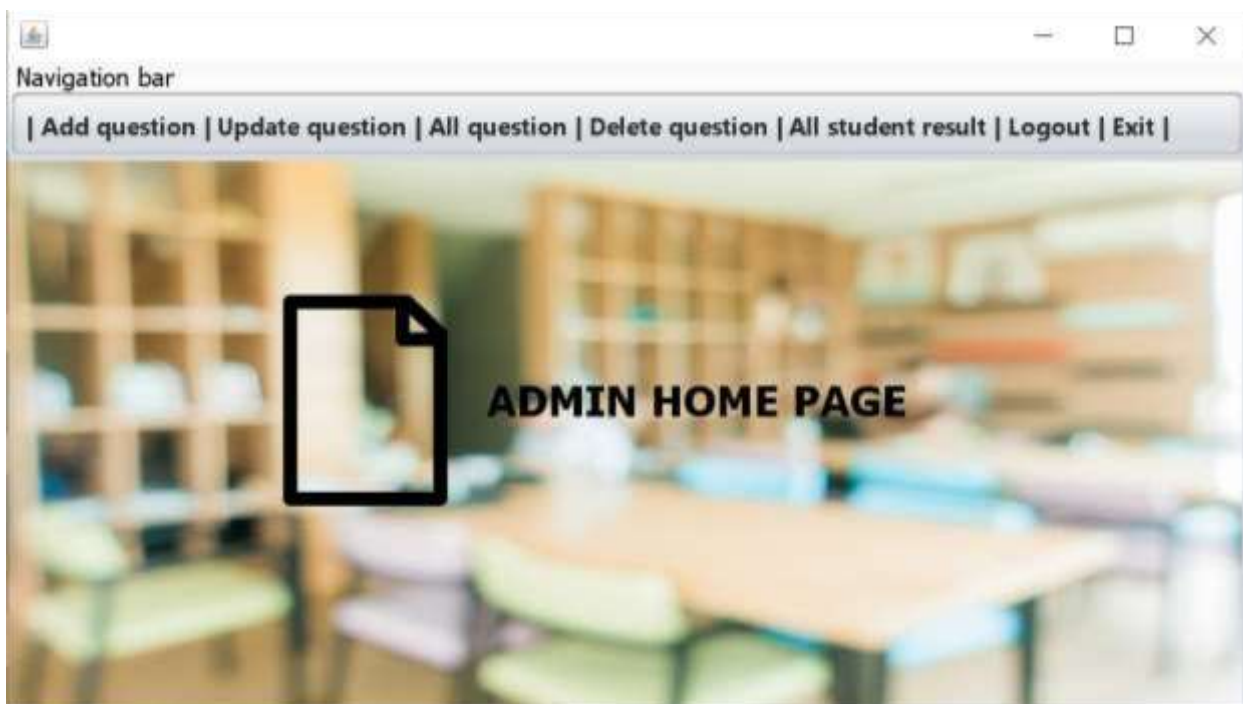
I. INDEX PAGE




II. LOGIN PAGE



III. ADMIN HOME PAGE



IV. ADDING QUESTION

— □ ×

ADD NEW QUESTION

Exit

Question ID: 11

Question: Which of the following is an operating system?

Option 1: linux


Option 2: ubuntu

Option 3: java

Option 4: HP

Answer: ubuntu

SaveClear

— □ ×

ADD NEW QUESTION

Exit

Question ID: 11

Question: Which of the following is an operating system?

Option 1: linux

Option 2: ubuntu


Option 3: java

Option 4: HP

Answer: ubuntu

SaveClear

Message ×

 Successfully Updated

OK

V. DELETE QUESTION

The screenshot shows a web application window titled "DELETE QUESTION". It features an "Exit" button in the top right corner. Below the title, there is a "Question ID:" label followed by a text input field containing the number "22" and a "Search" button. Further down, there are five labels: "Question:", "Option 1:", "Option 2:", "Option 3:", and "Option 4:", each followed by a text input field. At the bottom, there is an "Answer:" label followed by a text input field. Below the input fields, there are two buttons: "Delete" and "Clear". A modal message box is displayed in the center of the form, titled "Message", with an information icon and the text "Question id does not exist". It has an "OK" button at the bottom right.

The screenshot shows the same "DELETE QUESTION" web application window. In this instance, the "Question ID:" field contains the number "11". The "Question:" field contains the text "Which operating system?", "Option 1:" contains "linux", "Option 2:" contains "ubuntu", "Option 3:" contains "java", "Option 4:" contains "HP", and the "Answer:" field contains "ubuntu". The "Delete" button is highlighted. A modal message box is displayed in the center, titled "Message", with an information icon and the text "Successfully deleted!". It has an "OK" button at the bottom right.

VI. UPDATE QUESTION

UPDATE QUESTION

Exit

Question ID: 22

Search

Question:

Option 1:

Option 2:

Option 3:

Option 4:

Answer:

Update

Clear

Message

Question id does not exist!

OK

UPDATE QUESTION

Exit

Question ID: 4

Search

Question: What is the capital of India?

Option 1: Mumbai

Option 2: Calcutta

Option 3: New Delhi

Option 4: Chandigarh

Answer: New Delhi

Update

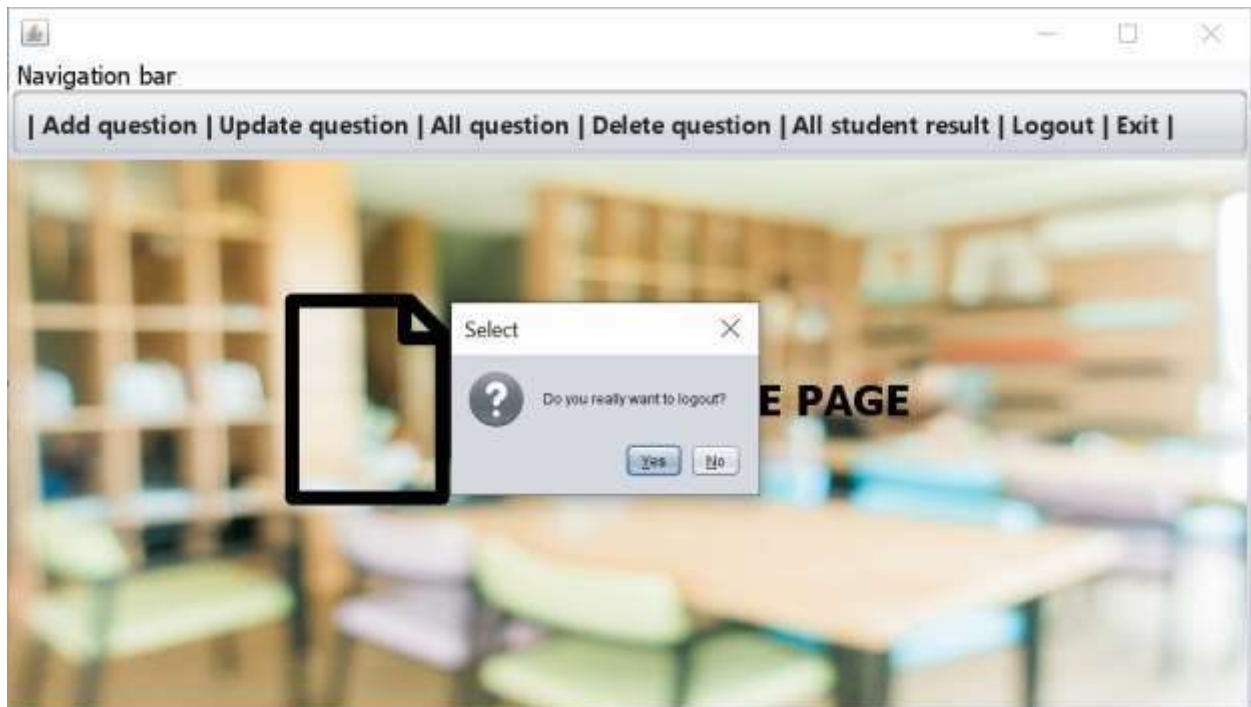
Clear

Message

successfully updated!

OK

VII. LOGOUT



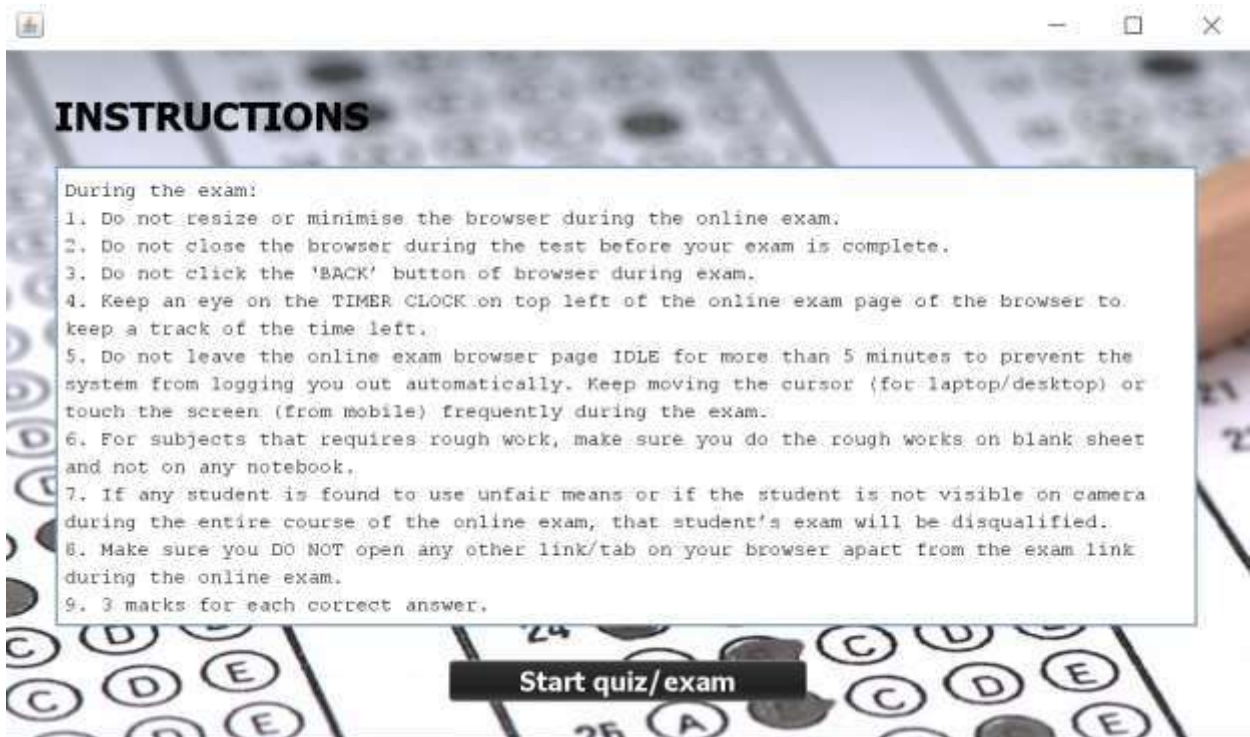
VI. STUDENT DETAILS

The screenshot shows a web application interface for filling up a form. The form is titled "FILL UP THE FORM" and includes a date field showing "23-05-2021". There are "Back" and "Exit" buttons at the top right. The form fields are as follows:

Roll number:	18123		
Name:	Priya Anand		
Father's name:	PC Anand		
Mother's name:	Manisha Anand		
Gender:	Female		
Contact number:	9324446677		
Email:	bteetas14@gmail.com		
10th :	KV	90	2016
12th:	PACE	90	2019
Graduation:	SIT	92	2023
Address:	Mumbai, India		

At the bottom, there is a "Save and next" button. A text box on the right side of the form contains the following text: "Online Quiz management system is managing all the activities related to quiz management right from the receipt of enrollment forms and examination forms through the processing of results and certificates and grading reports."

VIII. INSTRUCTIONS PAGE



IX. QUIZ PAGE

WELCOME Date: 23-05-2021 Total time: 10 min
Time taken: 0 58

Roll number: 18123
Name: Priya Anand
Total Question: 10
Question number: 10

What do you call a shape that has 5 sides?

- ☐ Circle
- ☐ Pentagon
- ☐ Octagon
- ☐ Nonagon

Submit

WELCOME Date: 23-05-2021 Total time: 10 min
Time taken: 1 47

Roll number: 18123
Name: Priya Anand
Total Question: 10
Question number: 10

What do you call a shape that has 5 sides?

- ☐ Circle
- ☒ Pentagon
- ☐ Octagon
- ☐ Nonagon

Select
? Do you really want to submit?
Yes No

Submit

X. EXAM SUBMITTED PAGE

Enter the email where you want to recieve your marks:

Successfully Updated!

27

XI. COMPLETE RESULT LIST

ALL STUDENT RESULT

Filter Students by marks:

contactNo	email	tenth/inter	tenthPer	tenthPass	twelve/inter	twelvePer	twelvePas	graduabo	graduabo	graduabo	address	marks
123	bteeta...	cbse	99	15	cbse	99	12	cbse	99	40	aa	18
bteeta...	bteeta...	ss	33	33	ss	33	33	ss	33	33	aa	18

XII. ALL QUESTIONS

ALL QUESTIONS							Exit
id	name	opt1	opt2	opt3	opt4	answer	
1	How many ho...	24	67	32	41	24	
2	Which is the l...	Earth	Jupiter	Mars	Pluto	Jupiter	
3	Which month ...	May	February	June	October	February	
4	What is the ca...	Mumbai	Calcutta	New Delhi	Chandigarh	New Delhi	
5	How many co...	8	6	5	7	7	
6	Which planet i...	Earth	Mars	Mercury	Venus	Mars	
7	Unit of curren...	Ohm	Ampere	Watt	None of the a...	Ampere	
8	First prime mi...	Indira Gandhi	Manmohan Si...	Narendra Modi	Jawaharlal Ne...	Jawaharlal Ne...	
9	How many col...	5	6	7	8	7	
10	What do you ...	Circle	Pentagon	Octagon	Nonaon	Octagon	

XIV. SQL DATA

```

MySQL 8.0 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 222
Server version: 8.0.25 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use gems;
Database changed
mysql> show tables;
+-----+
| Tables_in_gems |
+-----+
| question       |
| student        |
+-----+
2 rows in set (0.30 sec)

```

```

MySQL 8.0 Command Line Client

mysql> desc question;
+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+
| id     | varchar(10)   | YES  |     | NULL    |       |
| name   | varchar(500)  | YES  |     | NULL    |       |
| opt1   | varchar(500)  | YES  |     | NULL    |       |
| opt2   | varchar(500)  | YES  |     | NULL    |       |
| opt3   | varchar(500)  | YES  |     | NULL    |       |
| opt4   | varchar(500)  | YES  |     | NULL    |       |
| answer | varchar(500)  | YES  |     | NULL    |       |
+-----+
7 rows in set (0.03 sec)

```


MySQL 8.0 Command Line Client

7 rows in set (0.03 sec)

```
mysql> desc student;
```

Field	Type	Null	Key	Default	Extra
rollNo	varchar(10)	YES		NULL	
name	varchar(100)	YES		NULL	
fatherName	varchar(100)	YES		NULL	
motherName	varchar(100)	YES		NULL	
gender	varchar(50)	YES		NULL	
contactNo	varchar(10)	YES		NULL	
email	varchar(100)	YES		NULL	
tenthUniversityName	varchar(200)	YES		NULL	
tenthPercentage	varchar(10)	YES		NULL	
tenthPassoutYear	varchar(5)	YES		NULL	
twelveUniversityName	varchar(200)	YES		NULL	
twelvePercentage	varchar(10)	YES		NULL	
twelvePassoutYear	varchar(5)	YES		NULL	
graduationUniversityName	varchar(200)	YES		NULL	
graduationPercentage	varchar(10)	YES		NULL	
graduationPassoutYear	varchar(5)	YES		NULL	
address	varchar(500)	YES		NULL	
marks	int	YES		NULL	

18 rows in set (0.01 sec)

MySQL 8.0 Command Line Client

```
mysql> select *from question;
```

id	name	opt1	opt2	opt3	opt4
1	How many hours do we have in a day?	24	67	32	41
2	Which is the largest planet in our solar system?	Earth	Jupiter	Mars	Pluto
3	Which month has the least number of days?	May	February	June	October
4	What is the capital of India?	Mumbai	Calcutta	New Delhi	Chandigarh
5	How many continents are there in the world?	8	6	5	7
6	Which planet is known as the red planet?	Earth	Mars	Mercury	Venus
7	Unit of current is?	Ohm	Ampere	Watt	None of the
8	First prime minister of india?	Indira Gandhi	Mannohan Singh	Narendra Modi	Jawaharlal
9	How many colors in a rainbow?	5	6	7	8
10	What do you call a shape that has 5 sides?	Circle	Pentagon	Octagon	Nonagon

MySQL 8.0 Command Line Client

```
mysql> select *from student;
```

rollNo	name	fatherName	motherName	gender	contactNo	email	tenthUniversityName
tenthPercentage	tenthPassoutYear	twelveUniversityName	twelvePercentage	twelvePassoutYear	graduationUniversityName	graduationPercentage	graduationPassoutYear
address	marks						
123	123	123	123	123	123	123	123
89	89	89	89	89	89	89	89
99	99	99	99	99	99	99	99
2016	2016	2016	2016	2016	2016	2016	2016
2023	2023	2023	2023	2023	2023	2023	2023
Male	Male	Male	Male	Male	Male	Male	Male
1234567	1234567	1234567	1234567	1234567	1234567	1234567	1234567
wert	wert	wert	wert	wert	wert	wert	wert
2019	2019	2019	2019	2019	2019	2019	2019
fjv	fjv	fjv	fjv	fjv	fjv	fjv	fjv
erv	erv	erv	erv	erv	erv	erv	erv
123	123	123	123	123	123	123	123
89	89	89	89	89	89	89	89
99	99	99	99	99	99	99	99
2016	2016	2016	2016	2016	2016	2016	2016
2023	2023	2023	2023	2023	2023	2023	2023
Male	Male	Male	Male	Male	Male	Male	Male
1234567	1234567	1234567	1234567	1234567	1234567	1234567	1234567
wert	wert	wert	wert	wert	wert	wert	wert
2019	2019	2019	2019	2019	2019	2019	2019
fjv	fjv	fjv	fjv	fjv	fjv	fjv	fjv
erv	erv	erv	erv	erv	erv	erv	erv
123	123	123	123	123	123	123	123
89	89	89	89	89	89	89	89
99	99	99	99	99	99	99	99
2016	2016	2016	2016	2016	2016	2016	2016
2023	2023	2023	2023	2023	2023	2023	2023
Female	Female	Female	Female	Female	Female	Female	Female
123456789	123456789	123456789	123456789	123456789	123456789	123456789	123456789
bteetas@example.com	bteetas@example.com	bteetas@example.com	bteetas@example.com	bteetas@example.com	bteetas@example.com	bteetas@example.com	bteetas@example.com
cbse	cbse	cbse	cbse	cbse	cbse	cbse	cbse
sit	sit	sit	sit	sit	sit	sit	sit
12345	12345	12345	12345	12345	12345	12345	12345
quer	quer	quer	quer	quer	quer	quer	quer
2016	2016	2016	2016	2016	2016	2016	2016
2023	2023	2023	2023	2023	2023	2023	2023
Female	Female	Female	Female	Female	Female	Female	Female
987654321	987654321	987654321	987654321	987654321	987654321	987654321	987654321
.example.com	.example.com	.example.com	.example.com	.example.com	.example.com	.example.com	.example.com
cbse	cbse	cbse	cbse	cbse	cbse	cbse	cbse
sit	sit	sit	sit	sit	sit	sit	sit
123	123	123	123	123	123	123	123
quer	quer	quer	quer	quer	quer	quer	quer
adsf	adsf	adsf	adsf	adsf	adsf	adsf	adsf
Female	Female	Female	Female	Female	Female	Female	Female
123456789	123456789	123456789	123456789	123456789	123456789	123456789	123456789
quert	quert	quert	quert	quert	quert	quert	quert
quer	quer	quer	quer	quer	quer	quer	quer

XV. EMAILS RECIEVED



9. Future scope and conclusion

9.1 Future scope

Technological advancements in this era of digitization along with being a boon to the world have been advantageous to the educational sector too. The introduction of online exam system replaced the conventional system of assessment.

Before proceeding further let us understand the concept of online examination software. Exam software allows users to take online tests and automatically generate results based on the answers marked by the users.

- Safe and secure data:

Various tools offered by exam software have enabled the assessment conducting agencies to manage the crucial data related to examination questions and test-takers safely.

- Reduce administrative burden:

Organizing and running exams online not only reduces an organization's administrative burden but also saves cost and time. Online quiz management with its objective to make evaluation massive but simple, cost-effective, and faster has replaced the pen paper-based assessment.

- i. As is the case with all the applications, we will try to make this more user friendly.
- ii. Captcha verification can be added at login page.
- iii. We can add a search option for the admin to easily view, edit, delete questions
- iv. The option to download the results in .pdf/.csv/.xlsx can be added.
- v. The option of switching and attempting any question at any time can be added.

9.2 Project Outcomes

With the help of this project, we were able to learn the following concepts:

- i. JavaFX – It is one of the very important features with which we can enhance the Graphical User Interface (GUI) which in turn improves the user experience.
- ii. Classes & Constructors – it is widely used in the project
- iii. Inheritance – Many of the code blocks need to be inherited to avoid long lines of code. It turns out to be very useful
- iv. SQL – it is a very important feature, which helps us to store data and conduct a quiz. The questions, login credentials, result etc. can be accessed and verified using SQL.

9.3 References

<https://www.javatpoint.com/exception-handling-in-java>

<https://stackoverflow.com/questions/27679867/jtable-how-to-use-rs2xml/27680067>

<https://www.youtube.com/>

<https://docs.oracle.com/javase/7/docs/api/javafx/swing/JFrame.html>

9.4 Conclusion

The whole project is designed in such a way that future improvements are easy to implement. We can conclude that:

- i. Only authorized users get access to the admin functionalities.
- ii. Updating information is easier
- iii. Conducting a quiz through a software improves the efficiency and ease with which we can conduct a quiz.

Online Quiz Management System is significantly superior among the other exams. We have come to result that the problems can be solved by introducing new security systems using biometrics, we can identify the student's identity by analysing digital signature or by fingerprint mechanism and by providing web cameras in the examination hall. Although web cameras sometimes get failed. We conclude that no mechanism is ideal. Each mechanism has some

restriction on its own. Key concepts are to develop paperless environment and to convert all the documentation in digital form.