**JAVA SWING BASED – DBMS QUIZ APPLICATION- SQL CONNECTIVITY**

**USING JDBC**

*A*

*Report*

*Submitted in partial fulfillment of theRequirementsfortheawardoftheDegreeof*

**BACHELOROFENGINEERINGIN**

**INFORMATIONTECHNOLOGY**

**By**

### P.Sneha Anjali<1602-20-737-302>

**Undertheguidanceof MsB.Leelavathy**



### Department of Information TechnologyVasaviCollegeofEngineering(Autonomous)(Affiliated to Osmania University)Ibrahimbagh,Hyderabad-31

**2020-2021**

BONAFIDECERTIFICATE

This is to certify that this project report titled **‘DBMS Quiz Application’** is a project work of Ms. P.Sneha Anjali bearing roll no. 1602-20-737-302 who carried out the project under my supervision in the IV semester for the academic year 2021- 2022.

Signature Signature

ExternalExaminer InternalExaminer

# ABSTRACT

ABSTRACT

The project: “Quiz Application” is a collection of number of different types of quizzes

like technical, games, sports, etc. A user can access/play all of the quiz and can attempt any of

the one. There will be limited number of questions and for each correct answer user will get a

credit score. User can see answers as well as can ask a query related to it. There are many quiz

applications available currently on internet. But there are few Which provide better

understanding between users and the application like, providing proper answers, user query

solving, uploading user questions as well as answer to it, etc.

To develop a user friendly quiz application which will contain : Numbers of quiz ,

Answers to every question, Query solving regarding any question, Uploading of user question

and answer, and to improve the knowledge level of users. To develop an application which will

contain solution to the above problems. By this application the user will come to know about

his/her level and can learn additional knowledge. Also by this application a user can expand

Quiz Application - As an administrator can create number quiz application and user can take any quiz, which helps to improve the skillset, identify the gaps in knowledge and build the confidence.

Quiz Application - As an administrator can create number quiz application and user can take any quiz, which helps to improve the skillset, identify the gaps in knowledge and build the confidence.

Administrator -  
• Create the Quiz in control panel under Content & Data Section.  
• Edit and Update Quiz Details.  
• Create Question and Answers.  
• Edit and Update Questions and Answers.

User -  
• User can choose any Quiz Exam.  
• Answer the Question within specified time.  
• If not submit the Quiz within the time, then automatically submit the Quiz Exam.  
• Once finish the exam then User able to see the Results.

ABSTRACT

The project: “Quiz Application” is a collection of number of different types of quizzes

like technical, games, sports, etc. A user can access/play all of the quiz and can attempt any of

the one. There will be limited number of questions and for each correct answer user will get a

credit score. User can see answers as well as can ask a query related to it. There are many quiz

applications available currently on internet. But there are few Which provide better

understanding between users and the application like, providing proper answers, user query

solving, uploading user questions as well as answer to it, etc.

To develop a user friendly quiz application which will contain : Numbers of quiz ,

Answers to every question, Query solving regarding any question, Uploading of user question

and answer, and to improve the knowledge level of users. To develop an application which will

contain solution to the above problems. By this application the user will come to know about

his/her level and can learn additional knowledge. Also by this application a user can expand

his/her knowledge among the world

ABSTRACT

The project: “Quiz Application” is a collection of number of different types of quizzes

like technical, games, sports, etc. A user can access/play all of the quiz and can attempt any of

the one. There will be limited number of questions and for each correct answer user will get a

credit score. User can see answers as well as can ask a query related to it. There are many quiz

applications available currently on internet. But there are few Which provide better

understanding between users and the application like, providing proper answers, user query

solving, uploading user questions as well as answer to it, etc.

To develop a user friendly quiz application which will contain : Numbers of quiz ,

Answers to every question, Query solving regarding any question, Uploading of user question

and answer, and to improve the knowledge level of users. To develop an application which will

contain solution to the above problems. By this application the user will come to know about

his/her level and can learn additional knowledge. Also by this application a user can expand

his/her knowledge among the world

# RequirementAnalysis

## ListofTables:

* users
* question

## ListofAttributeswiththeirDomainTypes:

users

* + userid number(45)
  + username varchar2(45)
  + email varchar2(45)

question

* + questionid number(20)
  + name varchar2(45)
  + option1 varchar2(45)
  + option2 varchar2(45)
  + option3 varchar2(45)
  + option4 varhar2(45)
  + answer varchar2(45)
  + quizid varchar2(45)

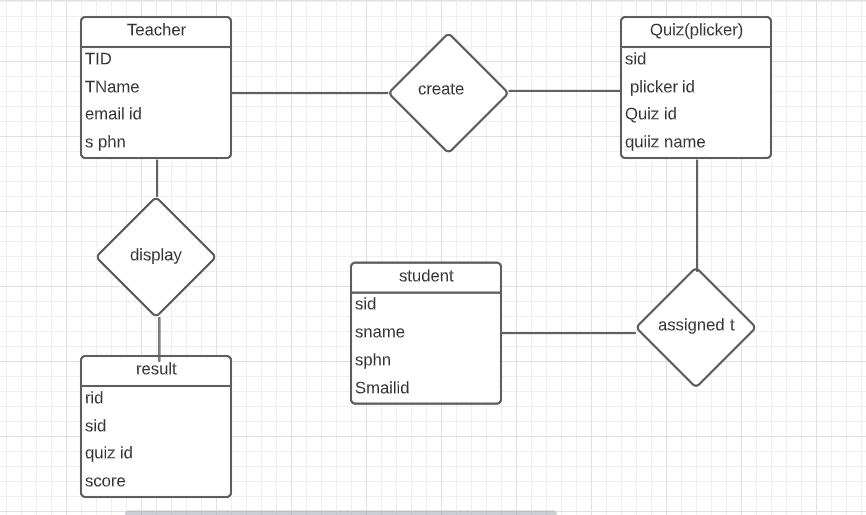
# AIMAND PRIORITYOFTHE PROJECT

To create a **Javaswing based quiz application.** It helps to write a quiz which contain four options.And after writing the quiz the result will be displayed.

.

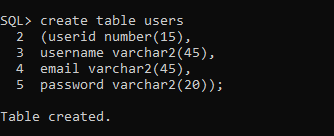
**DESIGN**

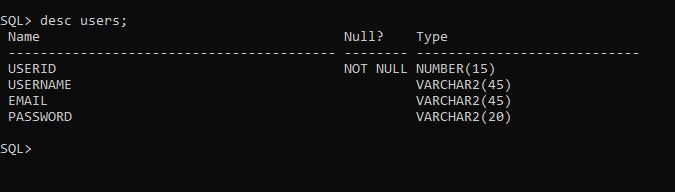
**EntityRelationshipDiagram**

****

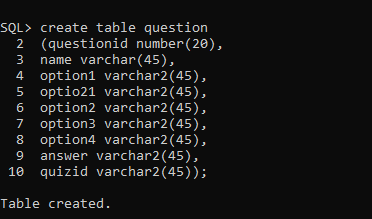
# DATABASEDESIGN

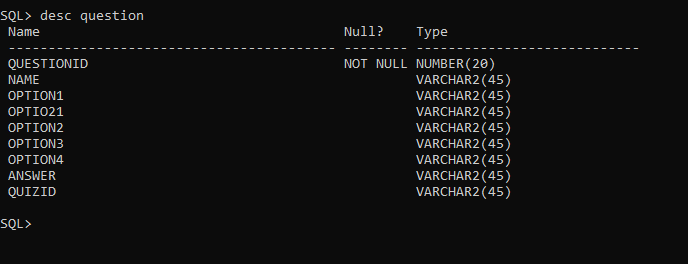
### DDLOperations



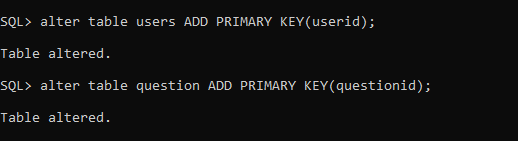


2.

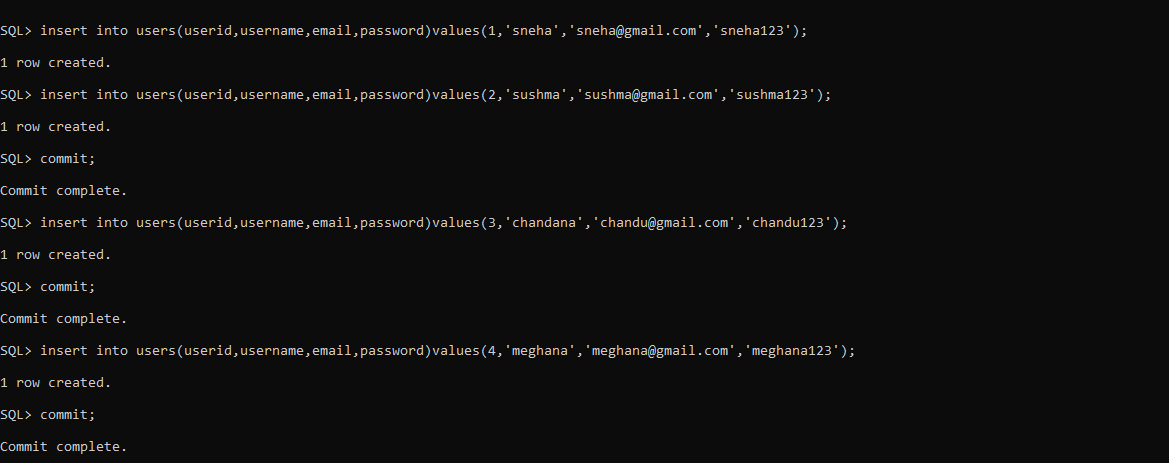


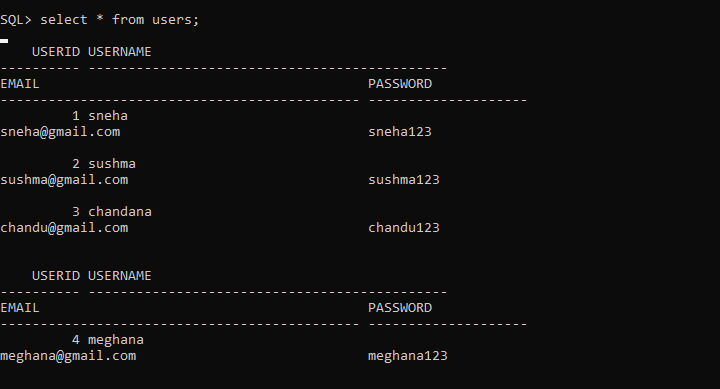


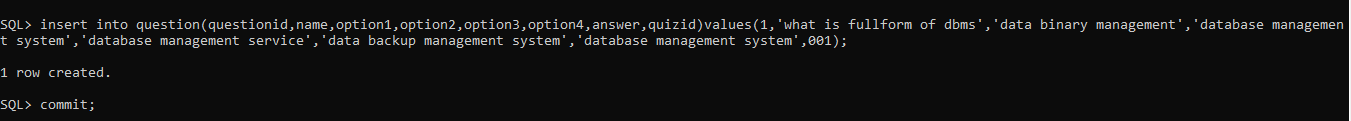
3. Integrity constraints



### DMLOperations

1.



2.

# IMPLEMENTATION

### JAVA-SQLConnectivityusingJDBC:

**JavaDatabaseConnectivity(JDBC)**isanapplicationprogramminginterface (API) for the programming language Java, which defines how aclient may access a database. It is a Java-based data access technology usedforJavadatabaseconnectivity.ItispartoftheJavaStandardEditionplatform, from Oracle Corporation. It provides methods to query and updatedata in a database and is oriented towards relational databases.

The connection to the database can be performed using Java programming(JDBCAPI) as:

**import** java.sql.\*;

**public** **class** ConnectionProvider {

**private** **static** Connection *con*;

**public** **static** Connection getConnection() {

**try** {

**if** (*con* == **null**) {

//driver class load

Class.*forName*("oracle.jdbc.driver.OracleDriver");

//create a connection..

*con* = DriverManager.

*getConnection*("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it20737302","vasavi");

}

} **catch** (Exception e) {

e.printStackTrace();

}

**return** *con*;

}

}

**Main class:**

**public** **class** Main {

**public** **static** **void** main(String s[])

{

System.***out***.println("connection"+ ConnectionProvider.*getConnection*());

**new** Registration();

}

}

**Login and Logout Page:**

**import** java.sql.\*;

**import** java.awt.\*;

**import** java.awt.GridLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.\*;

**import** javax.swing.border.TitledBorder;

**public** **class** Registration **implements** ActionListener {

JFrame frame;

JPanel signup\_panel, front, registration;

JPanel login\_panel;

JLabel signup\_username, welcome;

JLabel signup\_email;

JLabel signup\_password, signup\_confirm\_password;

JLabel login\_email;

JLabel login\_password;

JTextField tf\_signup\_username;

JTextField tf\_signup\_email;

JPasswordField tf\_signup\_password;

JPasswordField tf\_signup\_confirm\_password;

JTextField tf\_login\_email;

JPasswordField tf\_login\_password;

JButton btn\_signup, btn\_login, login, signup;

Boolean isLogined;

**public** Registration() {

frame = **new** JFrame();

signup\_panel = **new** JPanel();

front = **new** JPanel();

registration = **new** JPanel();

login\_panel = **new** JPanel();

signup\_username = **new** JLabel("User Name : ");

welcome = **new** JLabel("Welcome to Quiz");

signup\_email = **new** JLabel("Email : ");

signup\_password = **new** JLabel("Password : ");

signup\_confirm\_password = **new** JLabel("Confirm "

+ "Password : ");

tf\_signup\_username = **new** JTextField();

tf\_signup\_email = **new** JTextField();

tf\_signup\_password = **new** JPasswordField();

tf\_signup\_confirm\_password = **new** JPasswordField();

btn\_signup = **new** JButton("Sign Up");

signup = **new** JButton("Sign Up");

login = **new** JButton("Login");

login\_email = **new** JLabel("Email:");

login\_password = **new** JLabel("Password :");

tf\_login\_email = **new** JTextField();

tf\_login\_password = **new** JPasswordField();

btn\_login = **new** JButton("Login");

signup\_username.setBounds(10, 100, 120, 30);

signup\_email.setBounds(10, 140, 120, 30);

signup\_password.setBounds(10, 180, 120, 30);

signup\_confirm\_password.setBounds(10, 220, 180, 30);

tf\_signup\_username.setBounds(180, 100, 120, 30);

tf\_signup\_email.setBounds(180, 140, 120, 30);

tf\_signup\_password.setBounds(180, 180, 120, 30);

tf\_signup\_confirm\_password.setBounds(180, 220, 120, 30);

btn\_signup.setBounds(180, 300, 80, 40);

login\_email.setBounds(10, 100, 120, 30);

login\_password.setBounds(10, 140, 120, 30);

//tf\_signup\_username.setBounds(80, 100, 120, 30);

tf\_login\_email.setBounds(80, 100, 120, 30);

tf\_login\_password.setBounds(80, 140, 120, 30);

btn\_login.setBounds(100, 220, 80, 40);

login.setBounds(200, 400, 80, 40);

signup.setBounds(400, 400, 80, 40);

login.setForeground(Color.***blue***);

signup.setForeground(Color.***blue***);

welcome.setBounds(30, 50, 700, 200);

welcome.setFont(**new** Font("Jokerman", Font.***PLAIN***, 26));

welcome.setForeground(Color.***BLUE***);

registration.add(welcome);

registration.add(login);

registration.add(signup);

signup\_panel.setBorder(BorderFactory.*createTitledBorder*(BorderFactory.*createEtchedBorder*(), "SIGN UP FORM", TitledBorder.***CENTER***, TitledBorder.***TOP***));

login\_panel.setBorder(BorderFactory.*createTitledBorder*(BorderFactory.*createEtchedBorder*(), "LOGIN FORM", TitledBorder.***CENTER***, TitledBorder.***TOP***));

signup\_panel.add(signup\_username);

signup\_panel.add(tf\_signup\_username);

signup\_panel.add(signup\_email);

signup\_panel.add(tf\_signup\_email);

signup\_panel.add(signup\_password);

signup\_panel.add(tf\_signup\_password);

signup\_panel.add(signup\_confirm\_password);

signup\_panel.add(tf\_signup\_confirm\_password);

signup\_panel.add(btn\_signup);

login\_panel.add(login\_email);

login\_panel.add(login\_password);

login\_panel.add(tf\_login\_email);

login\_panel.add(tf\_login\_password);

login\_panel.add(btn\_login);

signup\_panel.setLayout(**null**);

login\_panel.setLayout(**null**);

registration.setLayout(**null**);

front.setLayout(**null**);

front.setBackground(Color.***BLUE***);

frame.add(front);

frame.add(registration);

btn\_signup.addActionListener(**this**);

btn\_login.addActionListener(**this**);

login.addActionListener(**this**);

signup.addActionListener(**this**);

frame.setExtendedState(JFrame.***MAXIMIZED\_BOTH***);

frame.setSize(600, 400);

frame.setLayout(**new** GridLayout(1, 2));

frame.setTitle("Welcome Page");

frame.setVisible(**true**);

}

**public** **void** actionPerformed(ActionEvent e) {

**if** (e.getSource() == signup) {

JInternalFrame f = **new** JInternalFrame();

f.add(signup\_panel);

f.setSize(600, 800);

f.setTitle("Signup Form");

f.setClosable(**true**);

signup.~~disable~~();

front.add(f);

f.setVisible(**true**);

}

**if** (e.getSource() == login) {

JInternalFrame f = **new** JInternalFrame();

f.add(login\_panel);

f.setSize(600, 800);

f.setTitle("Login Form");

f.setClosable(**true**);

front.add(f);

f.setVisible(**true**);

}

**if** (e.getSource() == btn\_signup) {

SignUp();

}

**if** (e.getSource() == btn\_login) {

Login();

}

}

**private** **void** SignUp() {

**int** x = 0;

Connection con = ConnectionProvider.*getConnection*();

String Username = tf\_signup\_username.getText();

String Email = tf\_signup\_email.getText();

**char**[] s1 = tf\_signup\_password.getPassword();

String Password = **new** String(s1);

**char**[] s2 = tf\_signup\_confirm\_password.getPassword();

String CPassword = **new** String(s2);

**if** (Password.equals(CPassword)) {

**try** {

PreparedStatement ps = con.prepareStatement("insert into users(username,email,password) values(?,?,?)");

ps.setString(1, Username);

ps.setString(2, Email);

ps.setString(3, CPassword);

ps.executeUpdate();

x++;

**if** (x > 0) {

JOptionPane.*showMessageDialog*(btn\_signup, "Data Saved Successfully");

}

} **catch** (Exception ex) {

System.***out***.println(ex);

}

} **else** {

JOptionPane.*showMessageDialog*(btn\_signup, "Password Does Not Match");

}

}

**private** **void** Login() {

Connection con = ConnectionProvider.*getConnection*();

JFrame f1 = **new** JFrame();

JLabel l, l0;

String Email = tf\_login\_email.getText();

**char**[] p = tf\_login\_password.getPassword();

String Password = **new** String(p);

**if** (Email.equals("admin@gmail.com") && Password.equals("Admin@123")) {

**new** QuizApplication();

} **else** {

**try** {

PreparedStatement ps = con.prepareStatement("select username from users where email=? and password=?");

ps.setString(1, Email);

ps.setString(2, Password);

ResultSet rs = ps.executeQuery();

**if** (rs.next()) {

**new** UserProfile(rs.getString(1));

frame.setVisible(**false**);

} **else** {

JOptionPane.*showMessageDialog*(**null**,

"Incorrect Email-Id or Password.."

+ "Try Again with correct detail");

}

} **catch** (Exception ex) {

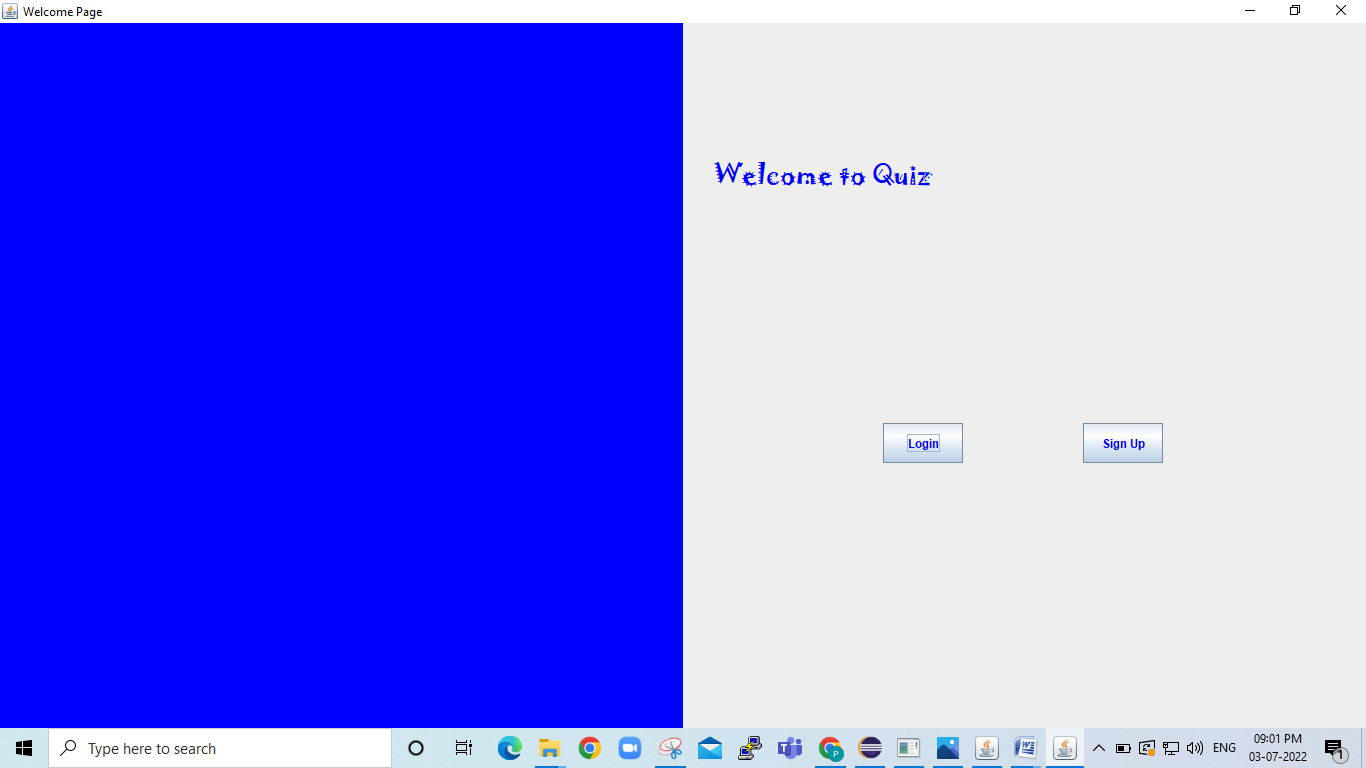
System.***out***.println(ex);

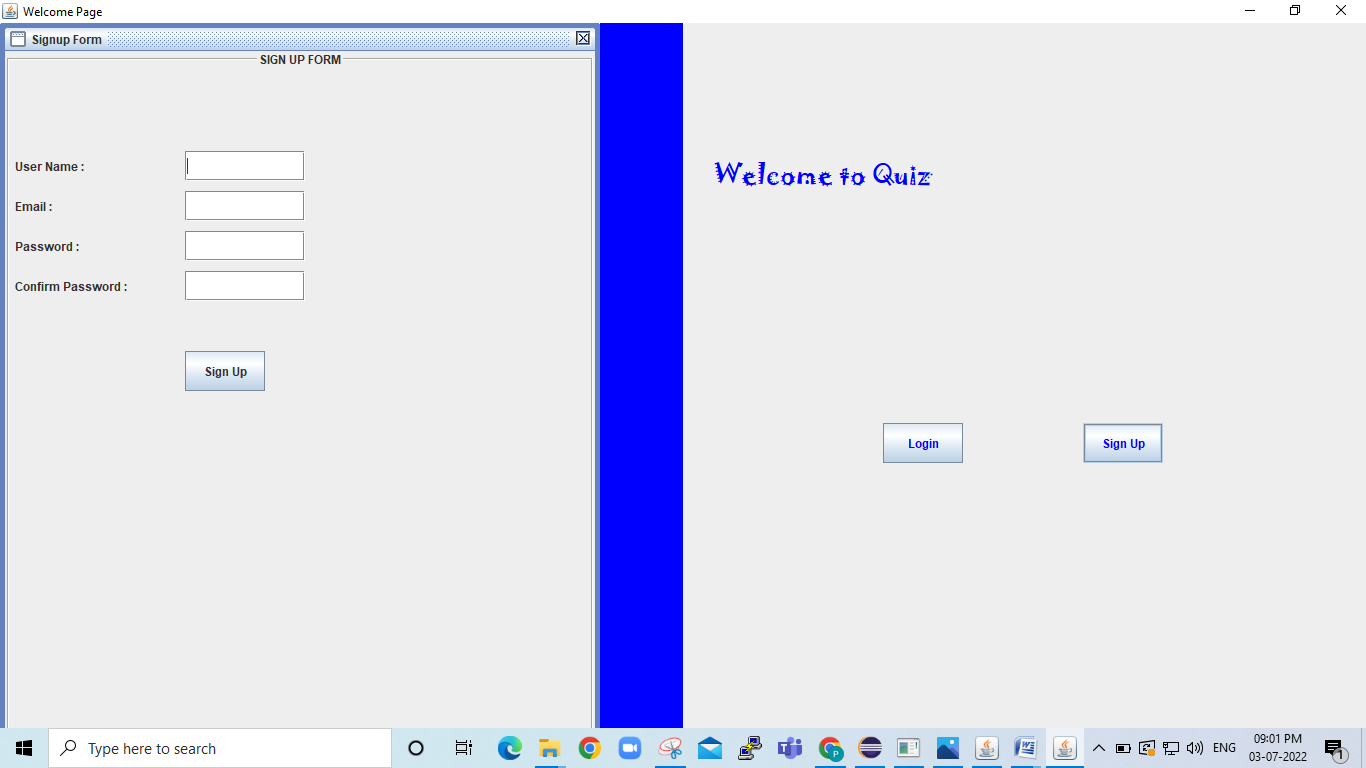
}

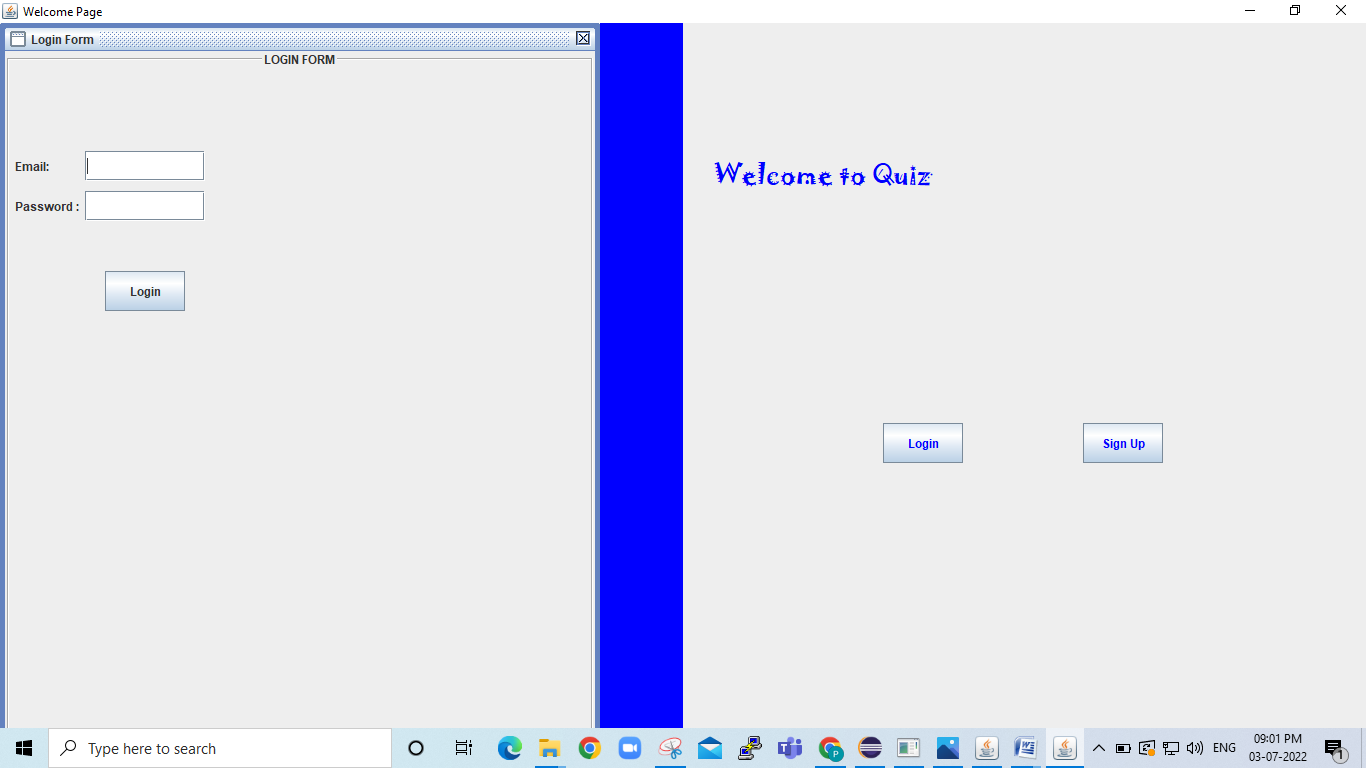
}

}

}

****





**Users Profile:**

**import** java.awt.BorderLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JPanel;

**public** **class** UserProfile {

JFrame f;

JPanel panel,top\_panel;

JLabel l = **new** JLabel();

UserProfile(String Username) {

f = **new** JFrame();

panel = **new** JPanel();

top\_panel = **new** JPanel();

l.setText("Welcome to Quiz World , " + Username);

JButton b1 = **new** JButton("NORTH");;

JButton b2 = **new** JButton("SOUTH");;

JButton b3 = **new** JButton("EAST");;

JButton b4 = **new** JButton("WEST");;

JButton b5 = **new** JButton("Take Quiz");

b5.setBounds(200, 200, 150, 30);

// b1.setBounds(900,0,40,80);

l.setBounds(500, 50, 200, 30);

panel.add(b5);

panel.add(l);

top\_panel.add(b1);

panel.setLayout(**null**);

top\_panel.setLayout(**null**);

f.add(top\_panel, BorderLayout.***NORTH***);

//f.add(b2, BorderLayout.SOUTH);

// f.add(b3, BorderLayout.EAST);

// f.add(b4, BorderLayout.WEST);

f.add(panel, BorderLayout.***CENTER***);

b5.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

**new** Test(Username);

f.setVisible(**false**);

}

});

f.setExtendedState(JFrame.***MAXIMIZED\_BOTH***);

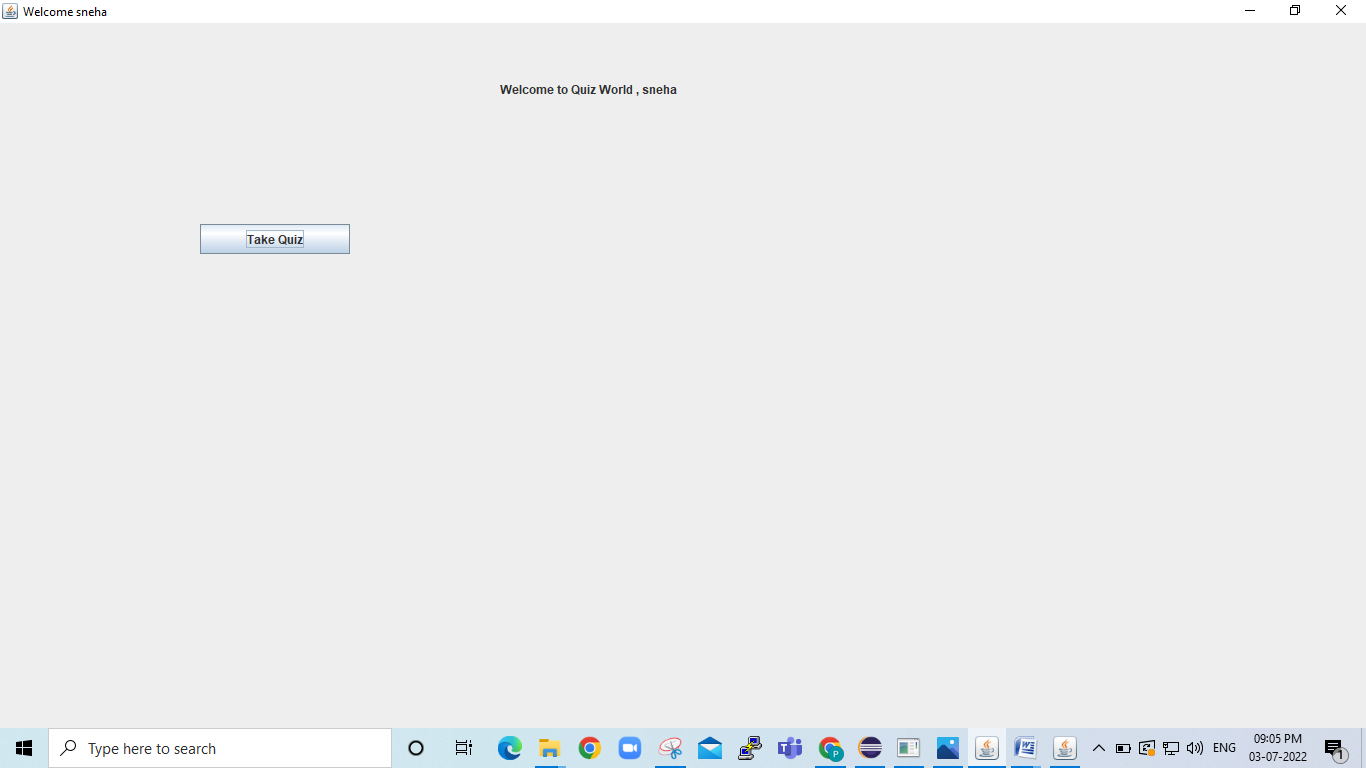
f.setTitle("Welcome " +Username);

f.setSize(300, 300);

f.setVisible(**true**);

}

}

****

**Test page:**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.sql.\*;**

**import javax.swing.\*;**

**public class Test implements ActionListener{**

**NumberOfQuestions numberOfQuestions = new NumberOfQuestions();**

**public int numberofQuestion = numberOfQuestions.getCount();**

**JFrame frame;**

**JLabel question;**

**JRadioButton option1, option2, option3, option4;**

**JButton save,submit;**

**String Questions[][] = new String[numberofQuestion][5];**

**String Answers[][] = new String[numberofQuestion][1];**

**String pa[][] = new String[numberofQuestion][1];**

**String Username;**

**ButtonGroup options;**

**public static int i=0;**

**public static int count = 0;**

**public static int timer = 15;**

**public static int ans\_given = 0;**

**public int score = 0;**

**public Test(String Username) {**

**this.Username = Username;**

**frame = new JFrame();**

**question = new JLabel();**

**option1 = new JRadioButton();**

**option2 = new JRadioButton();**

**option3 = new JRadioButton();**

**option4 = new JRadioButton();**

**save = new JButton("Save & Next");**

**submit = new JButton("Submit");**

**question.setBounds(10, 10, 200, 50);**

**option1.setBounds(10, 60, 200, 50);**

**option2.setBounds(10, 110, 200, 50);**

**option3.setBounds(10, 160, 200, 50);**

**option4.setBounds(10, 210, 200, 50);**

**save.setBounds(40, 260, 80, 40);**

**submit.setBounds(140, 260, 80, 40);**

**options = new ButtonGroup();**

**options.add(option1);**

**options.add(option2);**

**options.add(option3);**

**options.add(option4);**

**Connection con = ConnectionProvider.getConnection();**

**try {**

**PreparedStatement ps = con.prepareStatement("select \* from question");**

**ResultSet rs = ps.executeQuery();**

**while(rs.next()) {**

**System.out.print("i " +i);**

**Questions[i][0] = rs.getString(2);**

**Questions[i][1] = rs.getString(3);**

**Questions[i][2] = rs.getString(4);**

**Questions[i][3] = rs.getString(5);**

**Questions[i][4] = rs.getString(6);**

**Answers[i][0] = rs.getString(7);**

**i = i+1;**

**}**

**} catch (Exception ex) {**

**System.out.println(ex);**

**}**

**save.addActionListener(this);**

**submit.addActionListener(this);**

**frame.add(question);**

**frame.add(option1);**

**frame.add(option2);**

**frame.add(option3);**

**frame.add(option4);**

**frame.add(save);**

**frame.add(submit);**

**frame.setLayout(null);**

**frame.setSize(600, 600);**

**frame.setVisible(true);**

**frame.setTitle("Test is Running");**

**start(0);**

**}**

**public Test() {**

**}**

**public void actionPerformed(ActionEvent ae){**

**if(ae.getSource() == save){**

**//repaint();**

**option1.setEnabled(true);**

**option2.setEnabled(true);**

**option3.setEnabled(true);**

**option4.setEnabled(true);**

**ans\_given = 1;**

**if(options.getSelection() == null){**

**pa[count][0] = "";**

**}else {**

**pa[count][0] = options.getSelection().getActionCommand();**

**}**

**if(count == (numberofQuestion-2)){**

**save.setEnabled(false);**

**submit.setEnabled(true);**

**}**

**count++;**

**start(count);**

**}else if(ae.getSource() == submit){**

**ans\_given = 1;**

**if(options.getSelection() == null){**

**pa[count][0] = "";**

**}else {**

**pa[count][0] = options.getSelection().getActionCommand();**

**}**

**for(int ii =0 ; ii < pa.length ; ii++){**

**System.out.println(ii);**

**if(pa[ii][0].equals(Answers[ii][0])){**

**score+=10;**

**}else{**

**score+=0;**

**}**

**}**

**System.out.println(score);**

**frame.setVisible(false);**

**new Score(Username, score).setVisible(true);**

**}**

**}**

**public void start(int count){**

**/\*for(int i =0;i<10;i++)**

**{**

**System.out.println("Question " + Questions[i][0]);**

**System.out.println("Option1 "+Questions[i][1]);**

**System.out.println("Option2 "+Questions[i][2]);**

**System.out.println("Option3 "+Questions[i][3]);**

**System.out.println("Option4 "+Questions[i][4]);**

**System.out.println("Answer "+Answers[i][0]);**

**}\*/**

**//qno.setText("" + (count + 1) + ". ");**

**question.setText(Questions[count][0]);**

**option1.setText(Questions[count][1]);**

**option1.setActionCommand(Questions[count][1]);**

**option2.setText(Questions[count][2]);**

**option2.setActionCommand(Questions[count][2]);**

**option3.setText(Questions[count][3]);**

**option3.setActionCommand(Questions[count][3]);**

**option4.setText(Questions[count][4]);**

**option4.setActionCommand(Questions[count][4]);**

**options.clearSelection();**

**}**

**}**

**Score.java**

**import** java.awt.Color;

**import** java.awt.Font;

**import** java.awt.Image;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** javax.swing.ImageIcon;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**public** **class** Score **extends** JFrame **implements** ActionListener {

Score(String username, **int** score) {

setBounds(600, 150, 750, 550);

getContentPane().setBackground(Color.***WHITE***);

setLayout(**null**);

JLabel l2 = **new** JLabel("Thankyou " + username + " for Playing Simple Minds");

l2.setBounds(45, 30, 700, 30);

l2.setFont(**new** Font("RALEWAY", Font.***PLAIN***, 26));

add(l2);

JLabel l3 = **new** JLabel("Your Score is " + score);

l3.setBounds(350, 200, 300, 30);

l3.setFont(**new** Font("Jokerman", Font.***PLAIN***, 26));

l3.setForeground(**new** Color(199, 21, 133));

add(l3);

JButton b1 = **new** JButton("Play Again");

b1.setBackground(Color.***BLUE***);

b1.setForeground(Color.***WHITE***);

b1.addActionListener(**this**);

b1.setBounds(400, 270, 120, 30);

add(b1);

}

**public** **void** actionPerformed(ActionEvent ae) {

**this**.setVisible(**false**);

**new** Test("User");

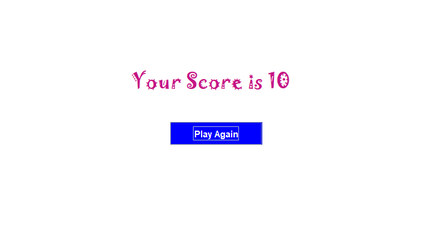
}

**public** **static** **void** main(String[] args) {

**new** Score("", 0).setVisible(**true**);

}

}

****

**Quiz Application**

**import** java.awt.BorderLayout;

**import** java.awt.CardLayout;

**import** java.awt.Color;

**import** java.awt.Container;

**import** java.awt.FlowLayout;

**import** java.awt.GridLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

//import static java.util.Collections.copy;

**import** javax.swing.JButton;

**import** javax.swing.JDesktopPane;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JPanel;

**public** **class** QuizApplication **implements** ActionListener {

JFrame f;

JPanel p, p1;

CardLayout card;

JDesktopPane desktop;

Container c;

JButton addQuiz = **new** JButton("Add Question");;

JButton updateQuiz = **new** JButton("Update Or Remove Question");;

JButton viewQuiz = **new** JButton("View Questions");;

JButton exit = **new** JButton("Exit");;

JButton logout = **new** JButton("Logout");;

QuizApplication() {

f = **new** JFrame();

p = **new** JPanel();

p1 = **new** JPanel();

// p.add(addQuiz);

p.add(updateQuiz);

p.add(viewQuiz);

p.add(exit);

p.add(logout);

p.setLayout(**new** GridLayout(5, 1));

//p1.setSize(900,900);

p1.setLayout(**null**);

f.add(p, BorderLayout.***LINE\_START***);

//p1.setBackground(Color.white);

// addQuiz.addActionListener(this);

updateQuiz.addActionListener(**this**);

viewQuiz.addActionListener(**this**);

exit.addActionListener(**this**);

logout.addActionListener(**this**);

f.setExtendedState(JFrame.***MAXIMIZED\_BOTH***);

f.setSize (300,300);

f.setVisible (**true**);

}

**public** **void** actionPerformed(ActionEvent e) {

**if** (e.getSource() == addQuiz) {

Thread runner = **new** Thread() {

**public** **void** run() {

// CheckBoxExample cl = new CheckBoxExample();

// AddQuiz ot = new AddQuiz();

// p1.add(ot);

// f.add(ot, BorderLayout.CENTER);

}

};

runner.start();

}

**if** (e.getSource() == updateQuiz) {

Thread runner = **new** Thread() {

**public** **void** run() {

// CheckBoxExample cl = new CheckBoxExample();

EditQuiz ot = **new** EditQuiz();

p1.add(ot);

f.add(ot, BorderLayout.***CENTER***);

}

};

runner.start();

}

**if** (e.getSource() == viewQuiz) {

ViewQuiz vq = **new** ViewQuiz();

f.add(vq, BorderLayout.***CENTER***);

}

**if** (e.getSource() == exit) {

f.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

f.setVisible(**false**);

}

**if** (e.getSource() == logout) {

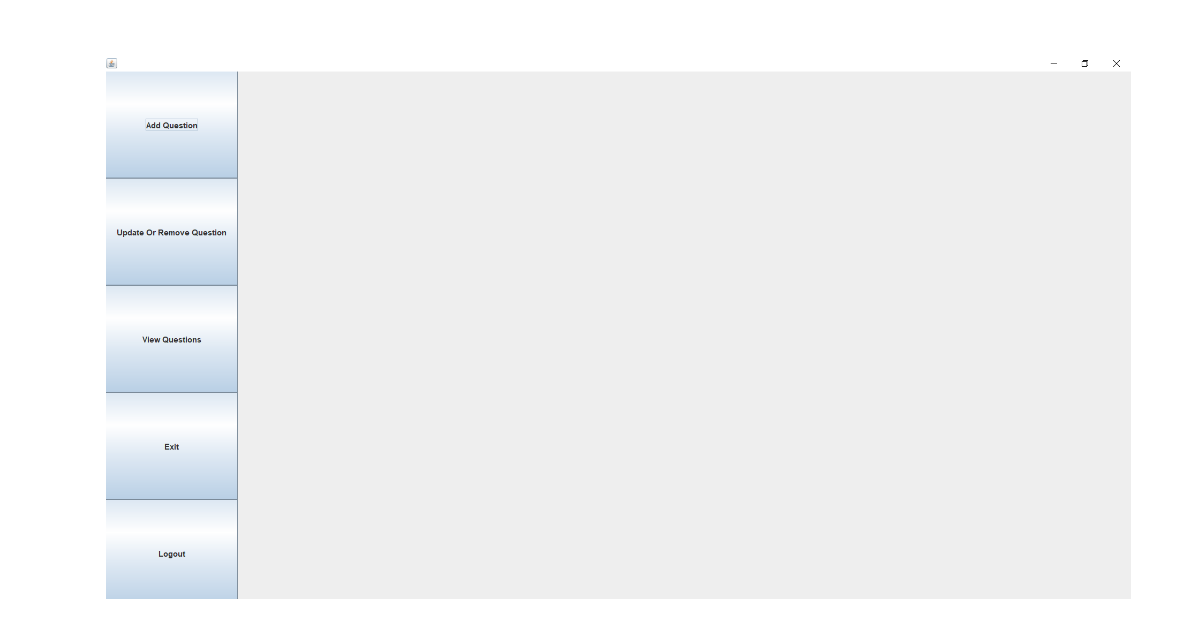
f.setVisible(**false**);

**new** Registration();

}

}

}



NumberOfQuestions:

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

public class NumberOfQuestions {

public int count;

public int getCount() {

return count;

}

public NumberOfQuestions() {

Connection con = ConnectionProvider.getConnection();

try {

PreparedStatement ps = con.prepareStatement("SELECT Count(\*) FROM question ");

ResultSet rs = ps.executeQuery();

while(rs.next()) {

count = rs.getInt(1);

}

} catch (Exception ex) {

System.out.println(ex);

}

}

public static void main(String args[]) {

new NumberOfQuestions();

// System.out.print("Count "+count);

}

}

**AddQuestions:**

**import** java.sql.\*;

**import** java.awt.FlowLayout;

**import** java.awt.GridLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** javax.swing.\*;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JPanel;

**public** **class** AddQuiz **extends** JInternalFrame{

NumberOfQuestions numberOfQuestions = **new** NumberOfQuestions();

**int** i;

**int** numberofQuestion = numberOfQuestions.getCount();

JPanel panel = **new** JPanel();

JLabel question = **new** JLabel("Question");

JPanel[] quizes = **new** JPanel[10];

JLabel[] questions = **new** JLabel[10];

JLabel options1 = **new** JLabel("Option1");

JLabel options2 = **new** JLabel("Option2");

JLabel options3 = **new** JLabel("Option3");

JLabel options4 = **new** JLabel("Option4");

JLabel answer = **new** JLabel("Answer");

JTextField tf\_question = **new** JTextField();

JTextField tf\_option1 = **new** JTextField();

JTextField tf\_option2 = **new** JTextField();

JTextField tf\_option3 = **new** JTextField();

JTextField tf\_option4 = **new** JTextField();

JTextField tf\_answer = **new** JTextField();

JSeparator sep[] = **new** JSeparator[10];

JButton btn\_add\_question = **new** JButton("Add Question");

String Quiz[] = **new** String[10];

JComboBox cb\_quiz;

// JComboBox.setTitle("Select Quiz");

AddQuiz quiz = **new** AddQuiz();

AddQuiz()

{

AddQuizes();

cb\_quiz=**new** JComboBox(Quiz);

question.setBounds(10,10,80,30);

options1.setBounds(10, 40,80,30);

options2.setBounds(10,70,80,30);

options3.setBounds(10,110,80,30);

options4.setBounds(10,150,80,30);

answer.setBounds(10,190,80,30);

tf\_question.setBounds(100,10,200,30);

tf\_option1.setBounds(100, 40,80,30);

tf\_option2.setBounds(100,70,80,30);

tf\_option3.setBounds(100,110,80,30);

tf\_option4.setBounds(100,150,80,30);

tf\_answer.setBounds(100,190,80,30);

cb\_quiz.setBounds(40,230,100,40);

btn\_add\_question.setBounds(40,280,80,30);

panel.add(question);

panel.add(options1);

panel.add(options2);

panel.add(options3);

panel.add(options4);

panel.add(answer);

panel.add(tf\_question);

panel.add(tf\_option1);

panel.add(tf\_option2);

panel.add(tf\_option3);

panel.add(tf\_option4);

panel.add(tf\_answer);

panel.add(btn\_add\_question);

panel.add(cb\_quiz);

/\*for(int i =0;i<10;i++)

{

add(questions[i] =questions[i] = new JLabel(Quiz[i]) );

// add(quizes[i] = new JPanel());

//add(sep[i]);

} \*/

btn\_add\_question.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e){

AddQuestion();

}

});

panel.setLayout(**null**);

add(panel);

setSize(600, 600);

setClosable(**true**);

setVisible(**true**);

setTitle("Add New Question");

//setLayout(null);

pack();

}

**private** **void** AddQuestion() {

**int** x=0;

String Question = tf\_question.getText();

String Option1 = tf\_option1.getText();

String Option2 = tf\_option2.getText();

String Option3 = tf\_option3.getText();

String Option4 = tf\_option4.getText();

String Answer = tf\_answer.getText();

Connection con = ConnectionProvider.*getConnection*();

// question.connection("select \* from question where Name = " + cb\_quiz.getItemAt(cb\_quiz.getSelectedIndex()));

**try**

{

PreparedStatement ps = con.prepareStatement("insert into question (Name,Option1,Option2,Option3,Option4,Answer,QuizID) values(?,?,?,?,?,?,?)");

ps.setString(1, Question);

ps.setString(2, Option1);

ps.setString(3, Option2);

ps.setString(4, Option3);

ps.setString(5, Option4);

ps.setString(6, Answer);

// ps.setInt(7, quiz.getQuizID());

ps.executeUpdate();

x++;

**if** (x > 0) {

JOptionPane.*showMessageDialog*(btn\_add\_question, "Question Saved Successfully");

}

}

**catch** (Exception ex)

{

System.***out***.println(ex);

}

}

**private** **void** AddQuizes() {

Connection con = ConnectionProvider.*getConnection*();

//To change body of generated methods, choose Tools | Templates.

**try**

{

PreparedStatement ps = con.prepareStatement("Select \* from quiz");

ResultSet rs = ps.executeQuery();

**while**(rs.next())

{

Quiz[i] = rs.getString("Name");

System.***out***.println(Quiz[i]);

i++;

}

}

**catch** (Exception ex)

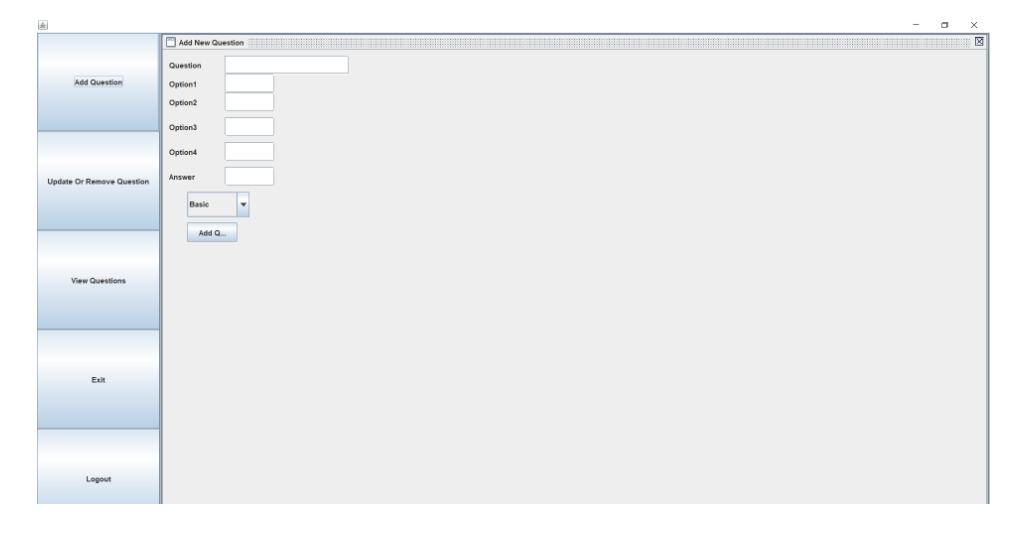
{

System.***out***.println(ex);

}

}

}



Edit quiz:

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.\*;

public class EditQuiz extends JInternalFrame {

NumberOfQuestions numberOfQuestions = new NumberOfQuestions();

int numberofQuestion = numberOfQuestions.getCount();

JPanel panel = new JPanel();

JLabel question = new JLabel("Question");

JLabel questionID = new JLabel("Question ID : ");

JLabel options1 = new JLabel("Option1");

JLabel options2 = new JLabel("Option2");

JLabel options3 = new JLabel("Option3");

JLabel options4 = new JLabel("Option4");

JLabel answer = new JLabel("Answer");

JTextField tf\_question = new JTextField();

JTextField tf\_option1 = new JTextField();

JTextField tf\_option2 = new JTextField();

JTextField tf\_option3 = new JTextField();

JTextField tf\_option4 = new JTextField();

JTextField tf\_answer = new JTextField();

JTextField tf\_questionID = new JTextField();

JButton btn\_edit\_question = new JButton("Update Question");

JButton btn\_search\_question = new JButton("Search");

JButton btn\_delete\_question = new JButton("Delete");

EditQuiz() {

questionID.setBounds(10, 10, 80, 30);

btn\_search\_question.setBounds(200, 10, 80, 30);

question.setBounds(10, 60, 80, 30);

options1.setBounds(10, 100, 80, 30);

options2.setBounds(10, 140, 80, 30);

options3.setBounds(10, 180, 80, 30);

options4.setBounds(10, 220, 80, 30);

answer.setBounds(10, 260, 80, 30);

tf\_questionID.setBounds(100, 10, 80, 30);

tf\_question.setBounds(100, 60, 80, 30);

tf\_option1.setBounds(100, 100, 80, 30);

tf\_option2.setBounds(100, 140, 80, 30);

tf\_option3.setBounds(100, 180, 80, 30);

tf\_option4.setBounds(100, 220, 80, 30);

tf\_answer.setBounds(100, 260, 80, 30);

btn\_edit\_question.setBounds(40, 300, 80, 30);

btn\_delete\_question.setBounds(140, 300, 80, 30);

panel.add(question);

panel.add(options1);

panel.add(options2);

panel.add(options3);

panel.add(options4);

panel.add(answer);

panel.add(tf\_question);

panel.add(tf\_option1);

panel.add(tf\_option2);

panel.add(tf\_option3);

panel.add(tf\_option4);

panel.add(tf\_answer);

panel.add(btn\_edit\_question);

panel.add(btn\_search\_question);

panel.add(btn\_delete\_question);

panel.add(questionID);

panel.add(tf\_questionID);

btn\_search\_question.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

EditQuestion();

}

});

btn\_edit\_question.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

UpdateQuestion();

}

});

btn\_delete\_question.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

DeleteQuestion();

}

});

panel.setLayout(null);

add(panel);

setSize(300, 300);

setClosable(true);

// frame.setLocationRelativeTo(null);

//frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setVisible(true);

setTitle("Update Questions");

}

private void EditQuestion() {

int x = 0;

String QuestionID = tf\_questionID.getText();

Connection con = ConnectionProvider.getConnection();

try {

PreparedStatement ps = con.prepareStatement("select \* from question where QuestionID = ?");

ps.setString(1, QuestionID);

ResultSet rs = ps.executeQuery();

if (rs.next()) {

tf\_question.setText(rs.getString(2));

tf\_option1.setText(rs.getString(3));

tf\_option2.setText(rs.getString(4));

tf\_option3.setText(rs.getString(5));

tf\_option4.setText(rs.getString(6));

tf\_answer.setText(rs.getString(7));

// .setText("Welcome " + rs.getString(1));

//l.setForeground(Color.red);

//l.setFont(new Font("Serif", Font.BOLD, 30));

} else {

JOptionPane.showMessageDialog(null,

"Question Id is wrong");

}

} catch (Exception ex) {

System.out.println(ex);

}

//To change body of generated methods, choose Tools | Templates.

/\* try {

PreparedStatement ps = con.prepareStatement("insert into question (Name,Option1,Option2,Option3,Option4,Answer) values(?,?,?,?,?,?)");

ps.setString(1, Question);

ps.setString(2, Option1);

ps.setString(3, Option2);

ps.setString(4, Option3);

ps.setString(5, Option4);

ps.setString(6, Answer);

ps.executeUpdate();

x++;

if (x > 0) {

JOptionPane.showMessageDialog(btn\_add\_question, "Question Saved Successfully");

}

} catch (Exception ex) {

System.out.println(ex);

}

}\*/

}

private void UpdateQuestion() {

int x = 0;

String Question = tf\_question.getText();

String option1 = tf\_option1.getText();

String option2 = tf\_option2.getText();

String option3 = tf\_option3.getText();

String option4 = tf\_option4.getText();

String answer = tf\_answer.getText();

Connection con = ConnectionProvider.getConnection();

//To change body of generated methods, choose Tools | Templates.

try {

PreparedStatement ps = con.prepareStatement("UPDATE question SET Name = '" + Question + "',Option1 = '" + option1 + "',Option2 = '" + option2

+ "',Option3 = '" + option3 + "',Option4 = '" + option4 + "',Answer = '" + answer

+ "' WHERE QuizID =" + tf\_questionID.getText());

ps.executeUpdate();

x++;

if (x > 0) {

JOptionPane.showMessageDialog(btn\_edit\_question, "Question Upadate Successfully");

}

} catch (Exception ex) {

System.out.println(ex);

}

}

private void DeleteQuestion() {

int x = 0;

String QuestionID = tf\_questionID.getText();

Connection con = ConnectionProvider.getConnection();

try {

PreparedStatement ps = con.prepareStatement("Delete from question where QuestionID =" + tf\_questionID.getText());

//ps.setString(1, QuestionID);

ps.executeUpdate();

JOptionPane.showMessageDialog(null,

"Deleted Successfully");

} catch (Exception ex) {

System.out.println(ex);

}

throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

**}**

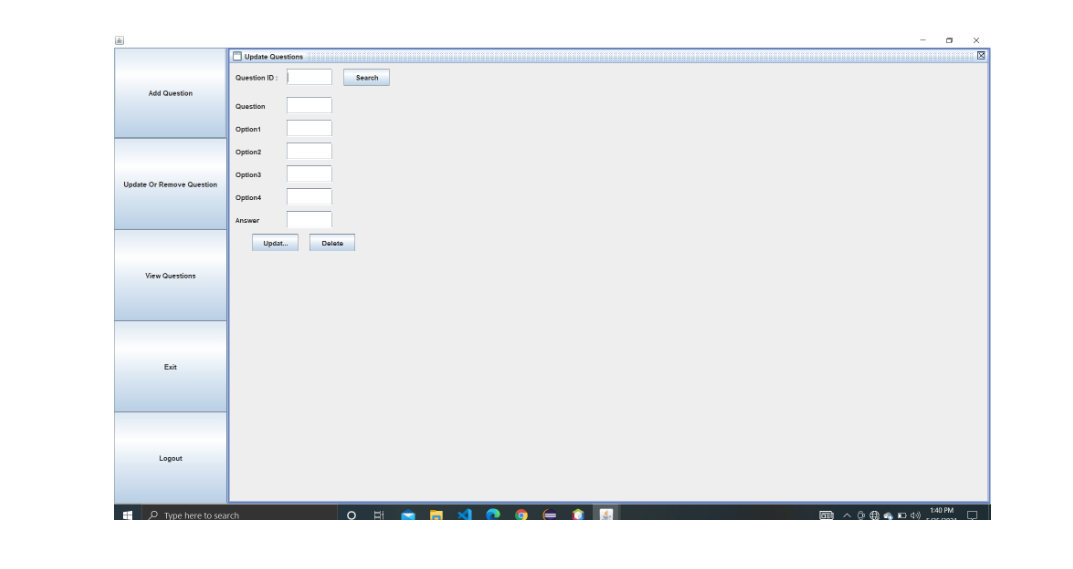
public static void main(String s[]) {

//System.out.println("connection" + ConnectionProvider.getConnection());

new EditQuiz();

}

}



**View Quiz:**

**import** java.awt.Button;

**import** java.awt.Frame;

**import** java.awt.GridLayout;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** javax.swing.\*;

**public** **class** ViewQuiz **extends** JInternalFrame {

NumberOfQuestions numberOfQuestions = **new** NumberOfQuestions();

**public** **int** numberofQuestion = numberOfQuestions.getCount();

//JFrame frame = new JFrame();

String Questions[][] = **new** String[numberofQuestion][5];

String Answers[][] = **new** String[numberofQuestion][1];

JLabel[] question,option1,option2,option3,option4;

**public** ViewQuiz() {

Test test = **new** Test();

question = **new** JLabel[numberofQuestion];

option1 = **new** JLabel[numberofQuestion];

option2 = **new** JLabel[numberofQuestion];

option3 = **new** JLabel[numberofQuestion];

option4 = **new** JLabel[numberofQuestion];

Connection con = ConnectionProvider.*getConnection*();

**try** {

PreparedStatement ps = con.prepareStatement("select \* from question");

ResultSet rs = ps.executeQuery();

**while**(rs.next()) {

/\* System.out.print("i " +i);

Questions[i][0] = rs.getString(2);

Questions[i][1] = rs.getString(3);

Questions[i][2] = rs.getString(4);

Questions[i][3] = rs.getString(5);

Questions[i][4] = rs.getString(6);

Answers[i][0] = rs.getString(7);

i = i+1;

\*/

}

} **catch** (Exception ex) {

System.***out***.println(ex);

}

**for**(**int** i=0;i<numberofQuestion;i++)

{

add(question[i] = **new** JLabel(Questions[i][0]));

add(option1[i]= **new** JLabel(Questions[i][1]));

add(option2[i]= **new** JLabel(Questions[i][2]));

add(option3[i]= **new** JLabel(Questions[i][3]));

add(option4[i]= **new** JLabel(Questions[i][4]));

}

setSize(500,500);

setLayout(**new** GridLayout(numberofQuestion,1));

setClosable(**true**);

setVisible(**true**);

setTitle("All Questions");

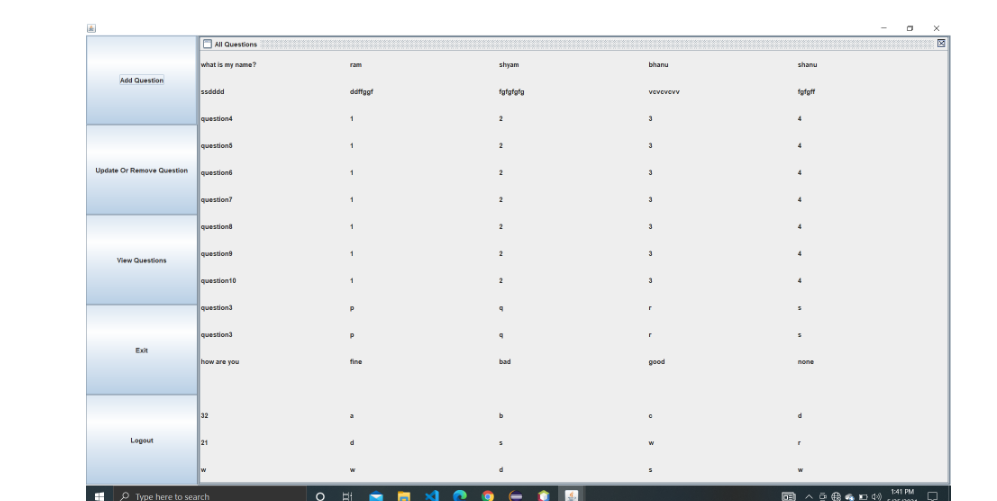
}

**public** **static** **void** main(String args[]) {

**new** ViewQuiz();

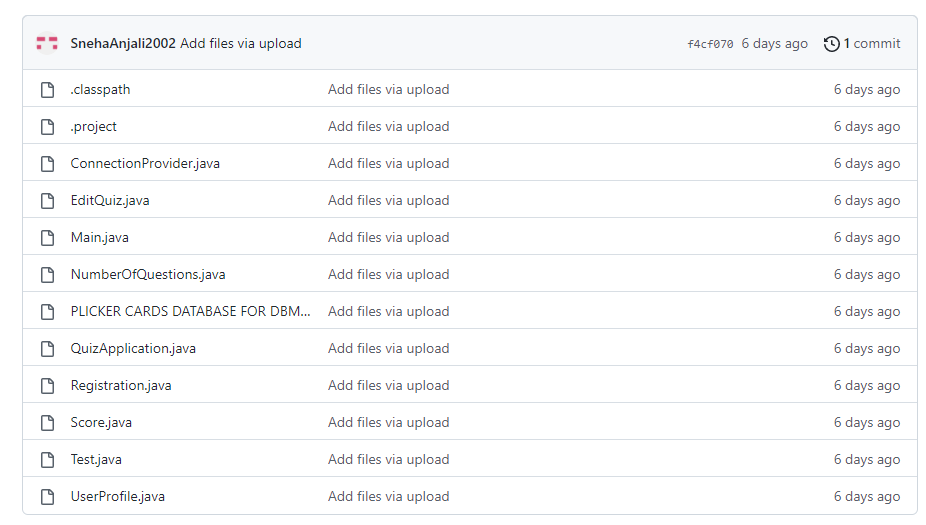
}

}

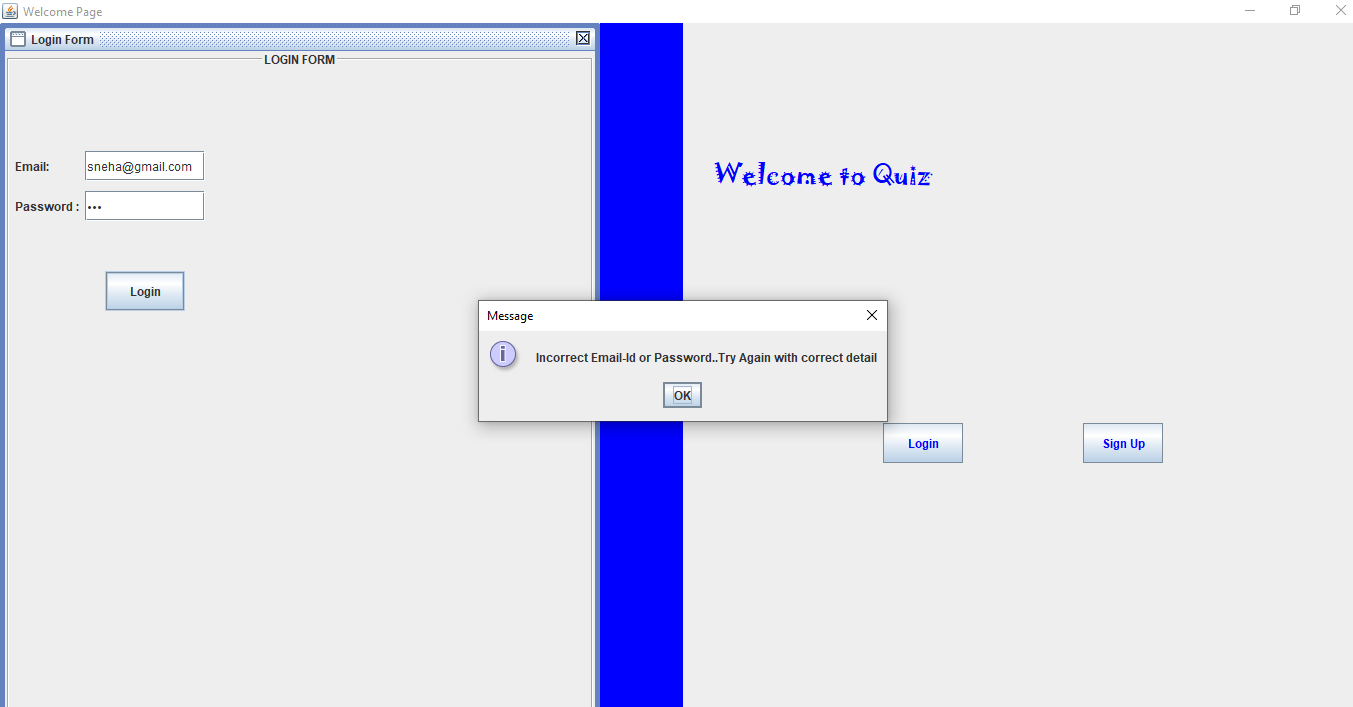
****

**GitHubLinksandFolderStructure**

**FolderStructure:** **https://github.com/SnehaAnjali2002/DBMS-QUIZ.git**



# TESTING



**RESULTS**

I have successfully completed the mini-project ***“DBMS Quiz Application ”*** .

## DISCUSSIONAND FUTUREWORK

In future the design of the application will be improved.And adding of new featues

To the project will be done such as time limit for taking quiz,The scores displayed in

The form of graphs.

## REFERENCES

* <https://docs.oracle.com/javase/7/docs/api/>
* <https://www.javatpoint.com/java-swing>
* <https://stackoverflow.com/>