### JAVA SWING BASED –CODEC GUIDE-SQL CONNECTIVITY USING JDBC

*A*

*Report*

*Submitted in partial fulfilment of the Requirements for the award of the Degree of*

***BACHELOR OF TECHNOLOGY***

***IN***

#### INFORMATION TECHNOLOGY

By

**G LIKHITH REDDY**

**1602-20-737-021**

##### Under the Guidance of

**B. Leelavathy**

Diagram

Description automatically generated with low confidence

### Department of Information Technology Vasavi College of Engineering (Autonomous) (Affiliated to Osmania University) Ibrahimbagh, Hyderabad-31

### 2021-2022

BONAFIDE CERTIFICATE

This to Certify that the project report titled

### “CODEC GUIDE” project work of Mr. G LIKHITH REDDY bearing Roll.no:1602-20-737-021 who carried out this project under my supervision in the IV semester for the academic year 2021-2022.

*Signature Signature*

*External examiner Internal examiner*

**ABSTRACT**

Codec Guide is a console-based project designed with sql and java that suggests hobbies for students to be maintained to become successful developer. This is a very interactive project which will be useful for the students

**REQUIREMENT ANALYSIS:**

**LIST OF TABLES:**

Students

Hobbies

Developers

Tests

**LIST OF ATTRIBUTES WITH THEIR DOMAIN TYPES:**

**Students:**

sname varchar2(10)

college varchar2(10)

sid number(10)

**Developers:**

dname varchar2(10)

organisation varchar2(10)

did number(10)

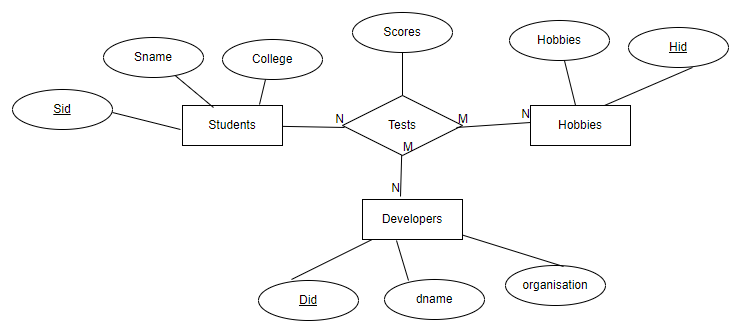
**HobbieSuggest:**

Hobbies varchar2(10)

**Test:**

**Scores Number(10)**

**ER DIAGRAM:**



**RELATIONAL MODEL:**

**DDL OPERATIONS:**

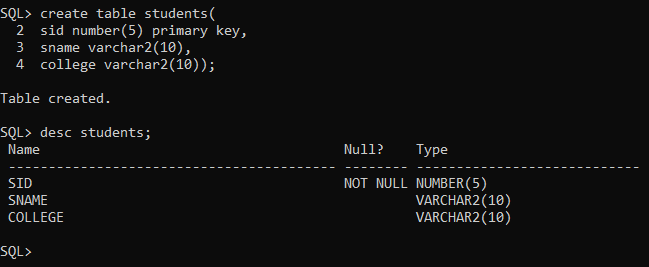
**Student:**

**Create table Students(**

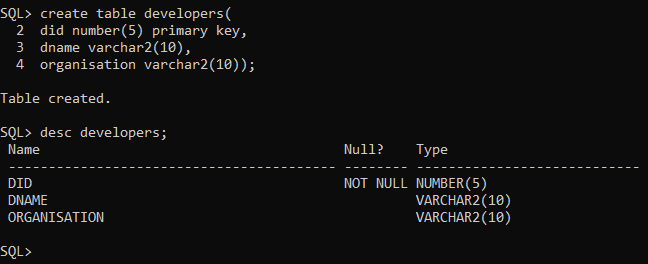
**Sid number(5) primary key,**

**Sname varchar2(10),**

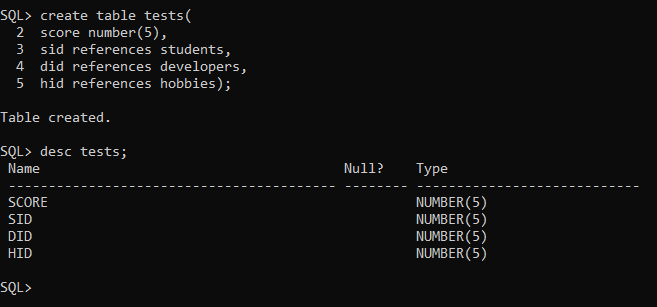
**College Varchar2(10));**



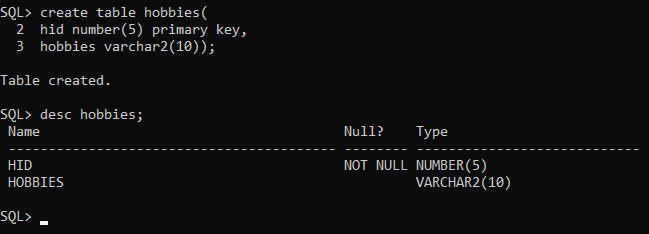
**Developers:**



**Tests:**

****

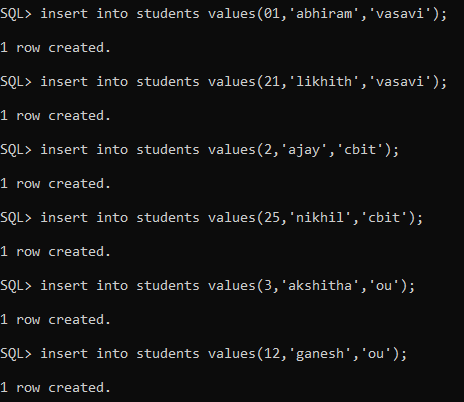
**Hobbies:**

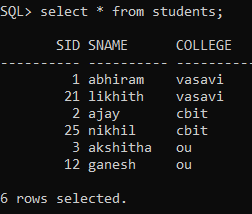
****

**DML OPERATIONS:**

**Students:**

**Insert into Students values(sid,’sname’,’college’);**

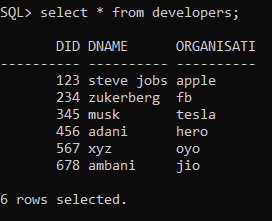
****

****

**Developers:**

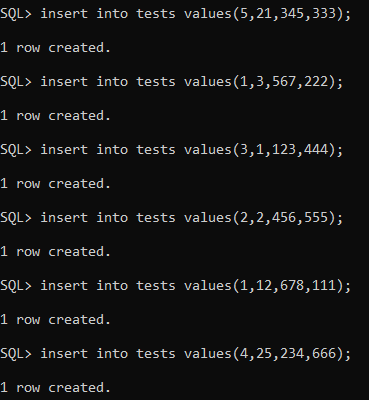
**Insert into developers values(did,’dname’,’Organisation’);**

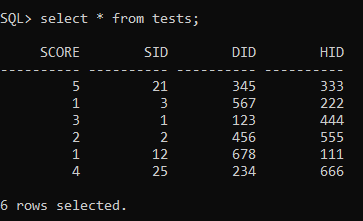
****

****

**Tests:**

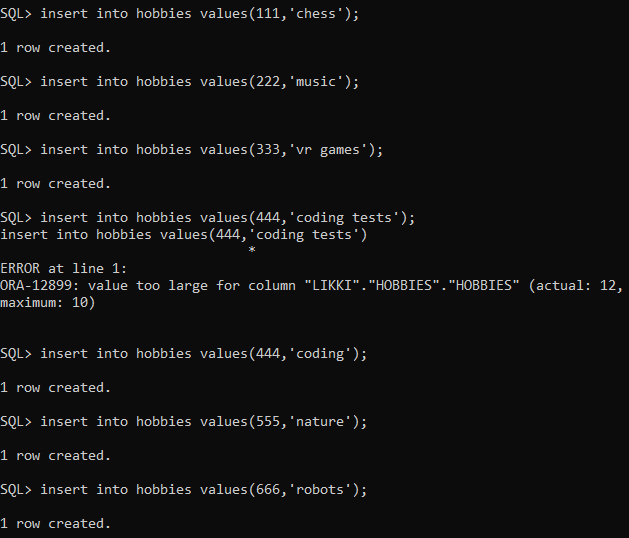
**Insert into Tests values(score,sid,did,hid);**

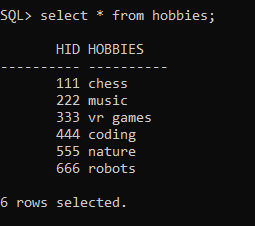
****

****

**Hobbies:**

**Insert into Hobbies values(hid,’hobbies’);**





**ARCHITECTURE AND TECHNOLOGY USED:**

### SOFTWARE USED:

Java Eclipse, Oracle 11g Database, Java SE version 8, SQL Plus.

### Java SWING:

**Swing** is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs.

Swing was developed to provide a more sophisticated set of GUI components than the

earlier AWT. Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

### SQL:

Structure Query Language (SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's **Relational** model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

### Java-SQL Connectivity using JDBC:

**Java Database Connectivity** (**JDBC**) is an application programming

interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform,

from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

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

Graphical user interface, text, application, email

Description automatically generated

**IMPLEMENTATION:**

MY FRAME

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class MyFrame extends JFrame implements ActionListener{

private JButton b\_Developer;

private JButton b\_Student;

MyFrame(){

b\_Developer = new JButton("Developer");

b\_Student = new JButton("Student");

b\_Student.setBounds(50,50,100,50);

b\_Developer.setBounds(200,50,100,50);

this.setSize(420,200);

this.setTitle("Codec Guide");

this.setLayout(null);

this.setVisible(true);

b\_Developer.addActionListener(this);

b\_Student.addActionListener(this);

this.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

add(b\_Developer);

add(b\_Student);

}

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource()==b\_Developer){

System.out.println("Developer");

dispose();

new Dev\_login();

}

if(e.getSource()==b\_Student) {

System.out.println("Student");

dispose();

new Stu\_login();

}

}

}

STU\_login:

import java.awt.Color;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.\*;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JPasswordField;

import javax.swing.JTextField;

import javax.swing.border.EmptyBorder;

public class Stu\_login extends JFrame

{

private QuizTest qt;

private static final long serialVersionUID = 1L;

private JTextField textField;

private JPasswordField passwordField;

private JButton btnNewButton;

private JLabel label;

private JPanel contentPane;

public Stu\_login()

{

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(450, 190, 1014, 597);

setResizable(false);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JLabel lblNewLabel = new JLabel("Login");

lblNewLabel.setForeground(Color.BLACK);

lblNewLabel.setFont(new Font("Times New Roman", Font.PLAIN, 46));

lblNewLabel.setBounds(423, 13, 273, 93);

contentPane.add(lblNewLabel);

textField = new JTextField();

textField.setFont(new Font("Tahoma", Font.PLAIN, 32));

textField.setBounds(481, 170, 281, 68);

contentPane.add(textField);

textField.setColumns(10);

passwordField = new JPasswordField();

passwordField.setFont(new Font("Tahoma", Font.PLAIN, 32));

passwordField.setBounds(481, 286, 281, 68);

contentPane.add(passwordField);

JLabel lblUsername = new JLabel("Username");

lblUsername.setBackground(Color.BLACK);

lblUsername.setForeground(Color.BLACK);

lblUsername.setFont(new Font("Tahoma", Font.PLAIN, 31));

lblUsername.setBounds(250, 166, 193, 52);

contentPane.add(lblUsername);

JLabel lblPassword = new JLabel("Password");

lblPassword.setForeground(Color.BLACK);

lblPassword.setBackground(Color.CYAN);

lblPassword.setFont(new Font("Tahoma", Font.PLAIN, 31));

lblPassword.setBounds(250, 286, 193, 52);

contentPane.add(lblPassword);

this.setVisible(true);

btnNewButton = new JButton("Login");

btnNewButton.setFont(new Font("Tahoma", Font.PLAIN, 26));

btnNewButton.setBounds(545, 392, 162, 73);

btnNewButton.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

if (e.getActionCommand().equals("Login")) {

try

{

int s1 = Integer.parseInt(textField.getText());

String s2 = passwordField.getText();

System.out.println(s1+s2);

qt.setId(s1);

try {

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

Connection con;

con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "likhith", "likhith");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery("select \* from log\_stu where sid='" + s1 + "'");

if (rs.next()) {

if (s2.equals(rs.getString(2))) {

System.out.println("yess");

qt = new QuizTest("Student Quiz");

} else {

throw new Exception("Enter valid credentials");

}

} else {

throw new Exception("Email not found");

}

con.close();

dispose();

} catch (Exception e1) {

System.out.println(e1);

JOptionPane.showMessageDialog(qt, e1.getMessage(), "ERROR", JOptionPane.ERROR\_MESSAGE);

}

}catch(Exception e1)

{

System.out.println("integer only");

}

}

}

});

contentPane.add(btnNewButton);

label = new JLabel("");

label.setBounds(0, 0, 1008, 562);

contentPane.add(label);

}

}

DEV\_Login:

import java.awt.Color;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JPasswordField;

import javax.swing.JTextField;

import javax.swing.border.EmptyBorder;

public class Dev\_login extends JFrame

{

private static final long serialVersionUID = 1L;

private JTextField textField;

private JPasswordField passwordField;

private JButton btnNewButton;

private JLabel label;

private JPanel contentPane;

public Dev\_login()

{

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(450, 190, 1014, 597);

setResizable(false);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JLabel lblNewLabel = new JLabel("Login");

lblNewLabel.setForeground(Color.BLACK);

lblNewLabel.setFont(new Font("Times New Roman", Font.PLAIN, 46));

lblNewLabel.setBounds(423, 13, 273, 93);

contentPane.add(lblNewLabel);

textField = new JTextField();

textField.setFont(new Font("Tahoma", Font.PLAIN, 32));

textField.setBounds(481, 170, 281, 68);

contentPane.add(textField);

textField.setColumns(10);

passwordField = new JPasswordField();

passwordField.setFont(new Font("Tahoma", Font.PLAIN, 32));

passwordField.setBounds(481, 286, 281, 68);

contentPane.add(passwordField);

JLabel lblUsername = new JLabel("Username");

lblUsername.setBackground(Color.BLACK);

lblUsername.setForeground(Color.BLACK);

lblUsername.setFont(new Font("Tahoma", Font.PLAIN, 31));

lblUsername.setBounds(250, 166, 193, 52);

contentPane.add(lblUsername);

JLabel lblPassword = new JLabel("Password");

lblPassword.setForeground(Color.BLACK);

lblPassword.setBackground(Color.CYAN);

lblPassword.setFont(new Font("Tahoma", Font.PLAIN, 31));

lblPassword.setBounds(250, 286, 193, 52);

contentPane.add(lblPassword);

this.setVisible(true);

btnNewButton = new JButton("Login");

btnNewButton.setFont(new Font("Tahoma", Font.PLAIN, 26));

btnNewButton.setBounds(545, 392, 162, 73);

btnNewButton.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

String userName = textField.getText();

String password = passwordField.getText();

}

});

contentPane.add(btnNewButton);

label = new JLabel("");

label.setBounds(0, 0, 1008, 562);

contentPane.add(label);

}

}

QuizTest:

import java.awt.event.\*;

import java.awt.event.ActionEvent;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import javax.swing.\*;

public class QuizTest extends JFrame implements ActionListener

{

JLabel label;

JRadioButton radioButtons[]= new JRadioButton[5];

JButton btnNext, btnResult;

ButtonGroup bg;

int sid;

int count=0, current=0, x=1, y=1, now=0,rating;

int m[] = new int[5];

public void setId(int sid) {

this.sid=sid;

}

QuizTest(String s)

{

super(s);

label=new JLabel();

add(label);

bg = new ButtonGroup();

for(int i=0; i<5; i++)

{

radioButtons[i]= new JRadioButton();

add(radioButtons[i]);

bg.add(radioButtons[i]);

}

btnNext = new JButton("Next");

btnResult = new JButton("Result");

btnResult.setVisible(false);

btnNext.addActionListener(this);

btnResult.addActionListener(this);

add(btnNext);

add(btnResult);

setData();

label.setBounds(30,40,450,20);

radioButtons[0].setBounds(50,80,450,20);

radioButtons[1].setBounds(50,110,200,20);

radioButtons[2].setBounds(50,140,200,20);

radioButtons[3].setBounds(50,170,200,20);

btnNext.setBounds(100,240,100,30);

btnResult.setBounds(270,240,100,30);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(null);

setLocation(250,100);

setVisible(true);

setSize(600,350);

}

void setData()

{

radioButtons[4].setSelected(true);

if(current==0)

{

label.setText("How familiar are you with Programming");

radioButtons[0].setText("Very Familiar and know Advanced Concepts");

radioButtons[1].setText("Familiar but I often get struck");

radioButtons[2].setText("A Little familiar, Can print basic things");

radioButtons[3].setText("Don't know at all!");

}

if(current==1)

{

label.setText("Are you Good with your logic while programming?");

radioButtons[0].setText("Yes...");

radioButtons[1].setText("Mostly but cant get through with tougher questions");

radioButtons[2].setText("A little ");

radioButtons[3].setText("No not at all");

}

if(current==2)

{

label.setText("How punctual are you?");

radioButtons[0].setText("Very punctual");

radioButtons[1].setText("Punctual at most times");

radioButtons[2].setText("Sometimes this Sometimes that..");

radioButtons[3].setText("Not at all");

}

if(current==3)

{

label.setText("How good are you with debugging your own code?");

radioButtons[0].setText("Very good, rarely make a mistake");

radioButtons[1].setText("Can debug a lot myself");

radioButtons[2].setText("I need someones help");

radioButtons[3].setText("Cannot debug at all");

}

if(current==4)

{

label.setText("Do you enjoy yourself while coding?");

radioButtons[0].setText("Yes I enjoy a lot");

radioButtons[1].setText("Yes at most times but harder problems frustrate me");

radioButtons[2].setText("Cant say!");

radioButtons[3].setText("I feel CS is not my cup of coffee");

}

label.setBounds(30,40,450,20);

for(int i=0,j=0; i<=90; i+=30,j++)

{

radioButtons[j].setBounds(50,80+i,200,20);

}

}

int checkAnswer()

{

if(current==0)

{

if(radioButtons[0].isSelected()) count=count+4;

if(radioButtons[1].isSelected()) count=count+3;

if(radioButtons[2].isSelected()) count=count+2;

if(radioButtons[3].isSelected()) count=count+1;

}

if(current==1)

{

if(radioButtons[0].isSelected()) count=count+4;

if(radioButtons[1].isSelected()) count=count+3;

if(radioButtons[2].isSelected()) count=count+2;

if(radioButtons[3].isSelected()) count=count+1;

}

if(current==2)

{

if(radioButtons[0].isSelected()) count=count+4;

if(radioButtons[1].isSelected()) count=count+3;

if(radioButtons[2].isSelected()) count=count+2;

if(radioButtons[3].isSelected()) count=count+1;

}

if(current==3)

{

if(radioButtons[0].isSelected()) count=count+4;

if(radioButtons[1].isSelected()) count=count+3;

if(radioButtons[2].isSelected()) count=count+2;

if(radioButtons[3].isSelected()) count=count+1;

}

if(current==4)

{

if(radioButtons[0].isSelected()) count=count+4;

if(radioButtons[1].isSelected()) count=count+3;

if(radioButtons[2].isSelected()) count=count+2;

if(radioButtons[3].isSelected()) count=count+1;

}

if(0<=count&&count<3) return 1;

if(3<=count&&count<6) return 2;

if(6<=count&&count<9) return 3;

if(9<=count&&count<12) return 4;

if(12<=count&&count<15) return 5;

if(15<=count&&count<18) return 6;

if(18<=count&&count<20) return 7;

return rating;

}

public static void main(String args[])

{

new QuizTest("Simple Quiz App");

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==btnNext)

{

rating=checkAnswer();

current++;

setData();

if(current==4)

{

btnResult.setVisible(true);

btnNext.setEnabled(false);

btnResult.setText("Result");

}

}

if(e.getActionCommand().equals("Result"))

{

rating=checkAnswer();

current++;

}

if(current==5)

{

JOptionPane.showMessageDialog(this,"Your total score is "+count+" Your rating level is "+rating);

System.exit(0);

try {

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

Connection con;

con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "likhith", "likhith");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery("select \* from log\_stu where sid='" + s1 + "'");

if (rs.next()) {

} else {

throw new Exception("Enter valid credentials");

}

} else {

throw new Exception("Email not found");

}

con.close();

dispose();

} catch (Exception e1) {

System.out.println(e1);

JOptionPane.showMessageDialog(qt, e1.getMessage(), "ERROR", JOptionPane.ERROR\_MESSAGE);

}

}

}

}

Dev\_frame:

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.sql.SQLOutput;

public class Dev\_frame extends JFrame implements ItemListener, TextListener{

JComboBox rating;

JTextField Hobbie;

JLabel greetings;

JButton submit;

Color c;

Dev\_frame(){

this.setSize(1350,450);

this.setTitle("Developer");

this.setLayout(null);

this.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

this.setVisible(true);

String scores[]={" Rating 1 "," Rating 2 "," Rating 3 "," Rating 4 "," Rating 5 "," Rating 6 "," Rating 7 "};

rating = new JComboBox(scores);

rating.setBounds(75,150,220,75);

rating.addItemListener(this);

rating.setFont(new Font("Serief",Font.BOLD,40));

rating.setBackground(new Color(253, 207, 169, 194));

rating.setForeground(new Color(49, 49, 49));

c=new Color(253, 207, 169);

getContentPane().setBackground(c);

Hobbie=new JTextField(" Enter a new Hobbie");

Hobbie.setFont(new Font("Serief",Font.BOLD,35));

Hobbie.setBackground(new Color(254, 211, 1));

Hobbie.setForeground(new Color(49, 49, 49, 255));

Hobbie.setEditable(false);

greetings=new JLabel("Hello \"Developer\" & \"DevId\" Select Rating And then Enter the Hobby");

greetings.setFont(new Font("Serief",Font.BOLD,40));

greetings.setBounds(25,35,10000,100);

greetings.setForeground(new Color(49, 49, 49));

this.add(greetings);

this.add(rating);

this.add(Hobbie);

Hobbie.setBounds(800,150,400,75);

submit=new JButton("Submit");

submit.setBounds(450,250,220,75);

submit.setFont(new Font("Serief",Font.BOLD,40));

submit.setBackground(new Color(59, 59, 79, 255));

submit.setForeground(new Color(255, 255, 255));

this.add(submit);

Hobbie.setCaretColor(Color.black);

}

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getSource()==rating) {

System.out.println(rating.getSelectedIndex());

rating.setEnabled(false);

Hobbie.setEditable(true);

}

}

@Override

public void textValueChanged(TextEvent e) {

//if(e.getSource()==Hobbie)

String s =Hobbie.getText();

System.out.println(s);

}

}

MAIN:

public class main {

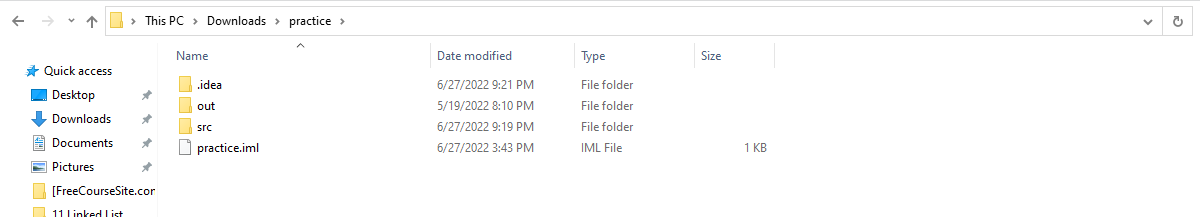
public static void main(String[] args) {

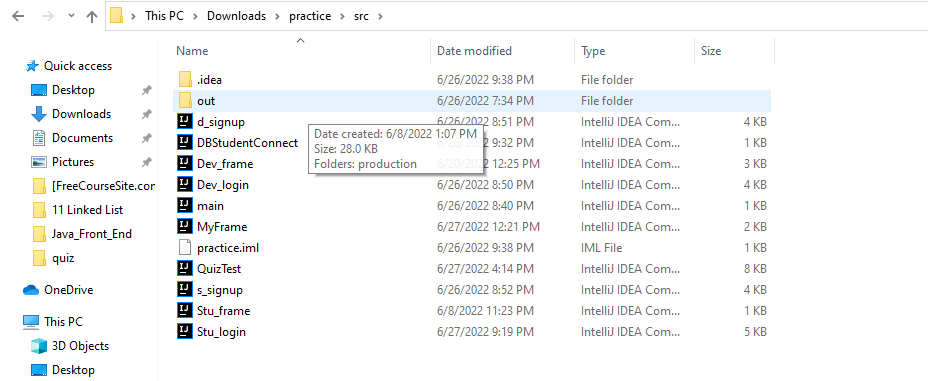
new MyFrame();

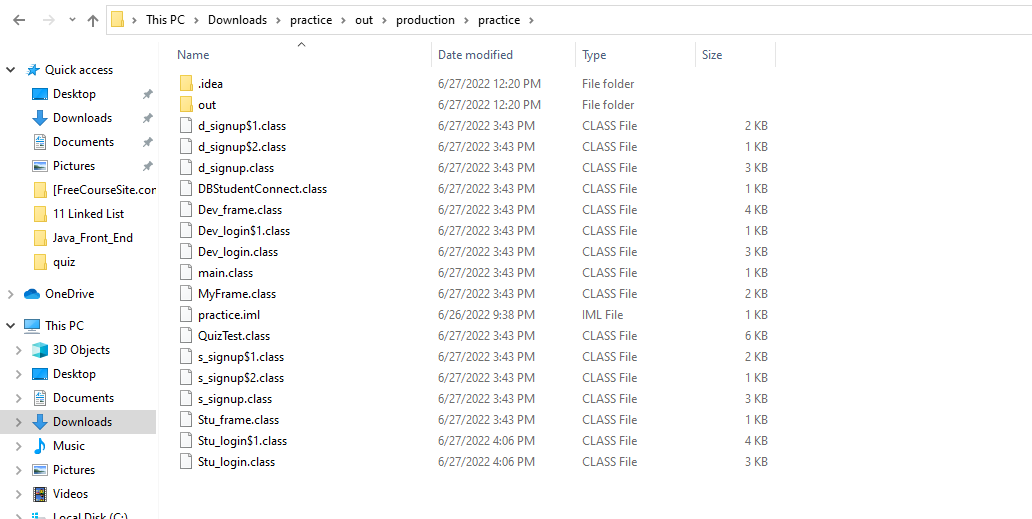
}

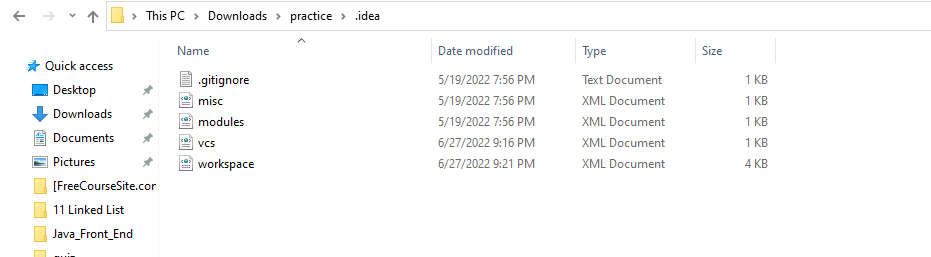
}

**GITHUB LINK AND FOLDER STRUCTURE:**

****

****

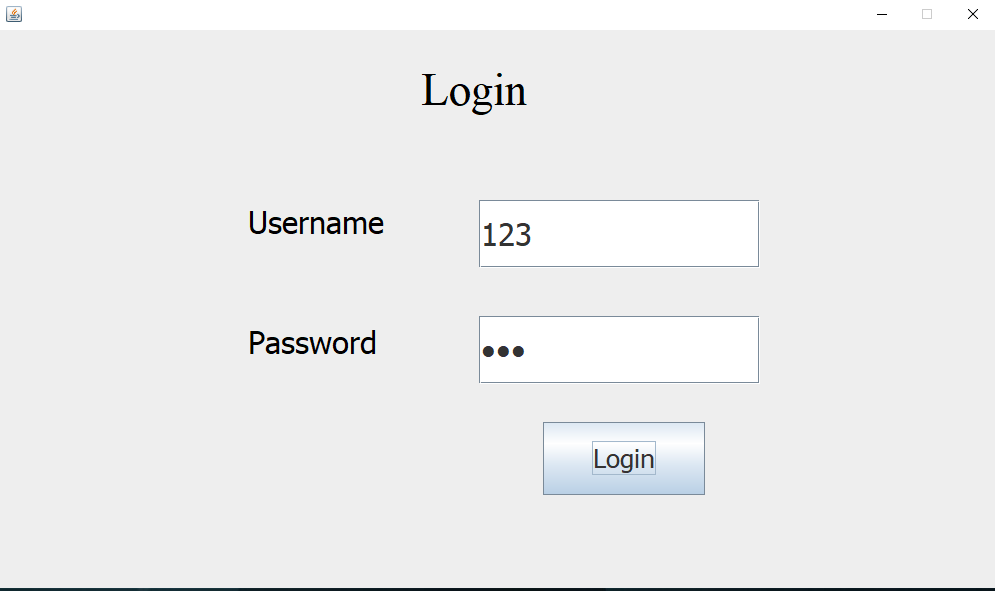
****

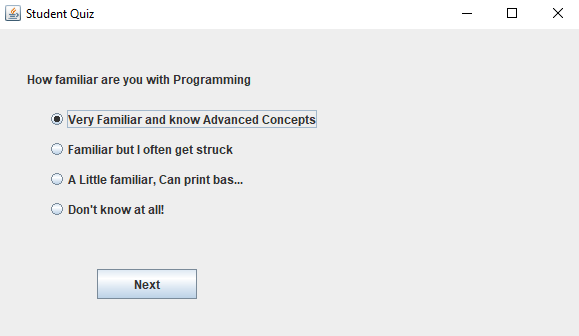
****

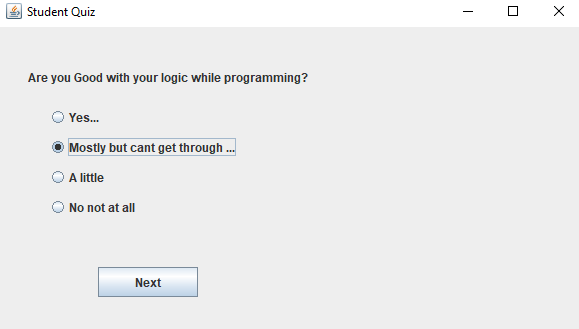
**TESTING:**

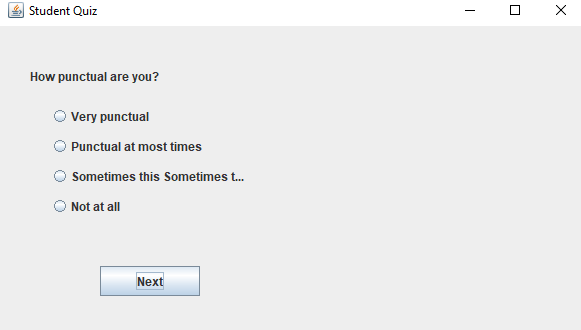
****

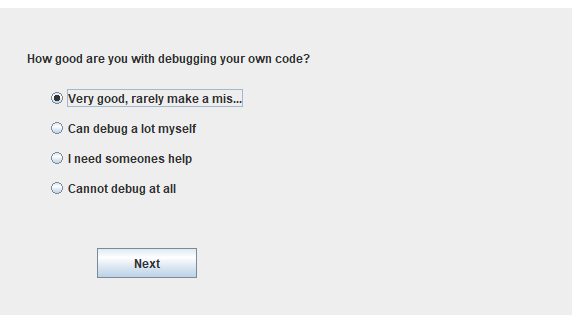
**Student login:**

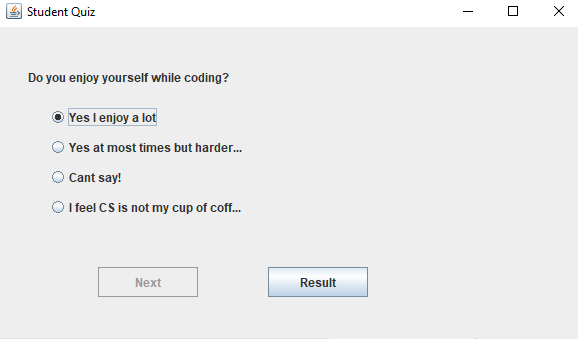
****

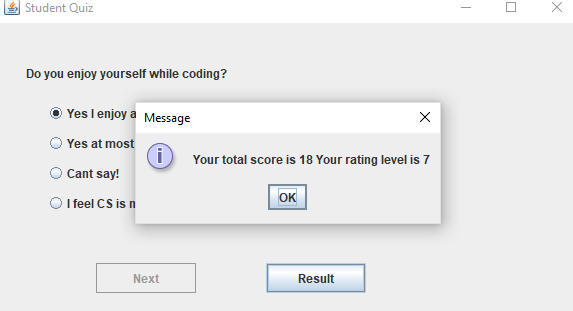
****

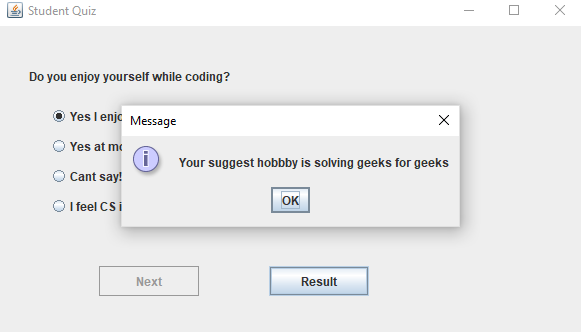
****

****

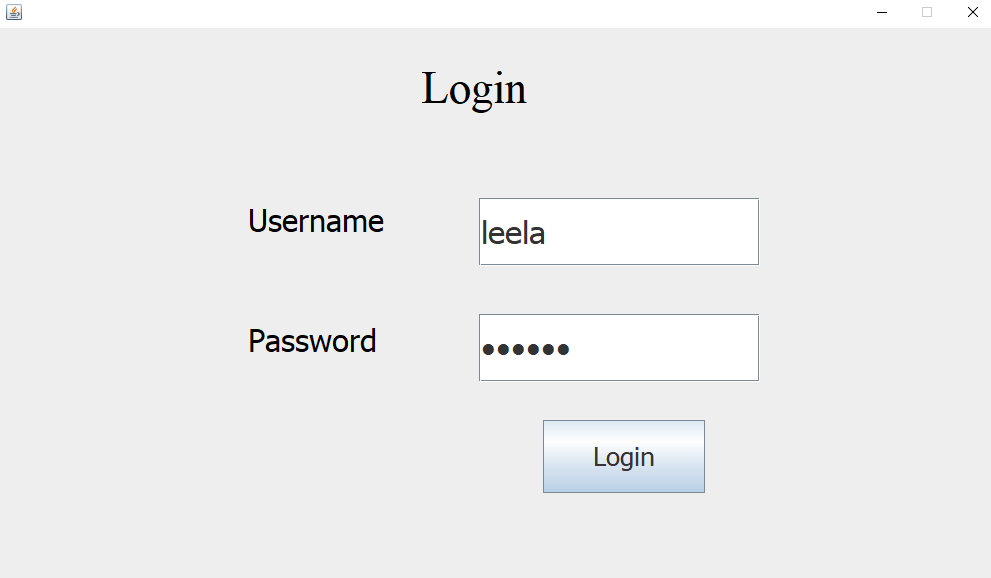
****

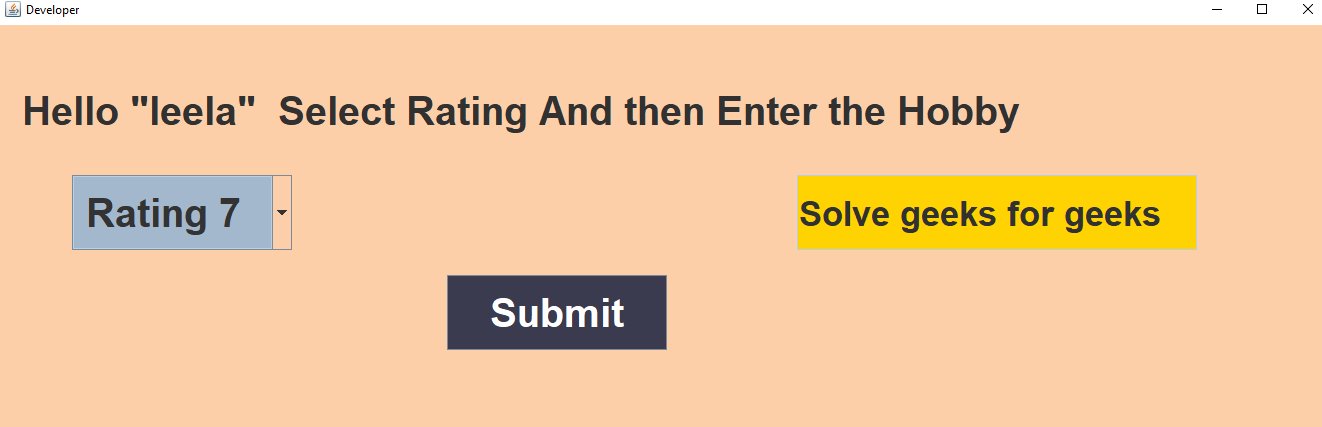
****

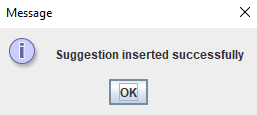
****

****

**Developer Login:**

****

****

****

### Results:

I successfully completed this MINI PROJECT “Codec Guide”.

### Discussion and Future work

While doing this project I got new ideas I understood how to work on projects. This project has been a huge learning curve for me as I have learnt many new technologies. I also learnt how to write queries efficiently and I have a thorough understanding of DBMS. A huge credit goes to my mentor **Ms.B.Leelavathy** whose guidance and motivation has lead me to keep improving the project.

Now to further extend this project I want to create Android app so that this will be very handy for both students and tros.

### References:

* [https://www.academia.edu/36893248/Ramakrishnan\_-](https://www.academia.edu/36893248/Ramakrishnan_-_Database_Management_Systems_3rd_Edition)

[\_Database\_Management\_Systems\_3rd\_Edition](https://www.academia.edu/36893248/Ramakrishnan_-_Database_Management_Systems_3rd_Edition)

* <https://docs.oracle.com/javase/7/docs/index.html>
* <https://www.javatpoint.com/dbms-tutorial>
* <http://www.sqlines.com/articles/java/sql_server_jdbc_connection>