**STUDENT RIGISTRATION FORM USING JAVA GUI AND POSTGRE SQL DATABASE**

**Code:**

import java.util.\*;

import javax.swing.\*;

import javax.swing.border.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.sql.ResultSet;

import javax.swing.ImageIcon;

import javax.swing.JToggleButton;

import java.net.URI;

import java.net.URL;

import java.net.URLClassLoader;

import java.lang.reflect.Constructor;

public class project{

private static final String DB\_URL="jdbc:postgresql://192.168.102.2:5432/java gui";

private static final String DB\_USER="postgres";

private static final String DB\_PASSWORD="123456";

static {

try {

// Load the PostgreSQL driver class dynamically

initializeDriver();

} catch (Exception e) {

e.printStackTrace();

}

}

private static void initializeDriver() {

try {

// First check if driver is already available

Class.forName("org.postgresql.Driver");

} catch (ClassNotFoundException initialEx) {

try {

// If not available, download driver from Maven repository

URI uri = new URI("https://repo1.maven.org/maven2/org/postgresql/postgresql/42.7.1/postgresql-42.7.1.jar");

URL url = uri.toURL();

URLClassLoader cl = new URLClassLoader(new URL[]{url});

Class<?> driverClass = cl.loadClass("org.postgresql.Driver");

Constructor<?> constructor = driverClass.getDeclaredConstructor();

constructor.newInstance();

} catch (Exception e) {

e.printStackTrace();

JOptionPane.showMessageDialog(null, "Failed to initialize database driver. Please contact administrator.");

}

}

}

public static void main(String[] args){

JFrame frame = new JFrame("Student Registration Form");

frame.setSize(600,600);

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

frame.setLayout(null);

frame.getContentPane().setBackground(new Color(230, 230, 250));

JButton bt = new JButton("Register");

bt.setBounds(160,80,100,40);

bt.setCursor(new Cursor(Cursor.HAND\_CURSOR));

JButton bt1 = new JButton("Login");

bt1.setBounds(300,80,100,40);

bt1.setCursor(new Cursor(Cursor.HAND\_CURSOR));

bt.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

frame.dispose();

showRegistrationPage();

}

});

bt1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

frame.dispose();

showLoginPage();

}

});

frame.add(bt);

frame.add(bt1);

frame.setVisible(true);

}

public static void showRegistrationPage(){

JFrame rframe = new JFrame("Registration Page");

rframe.setSize(1000,1000);

rframe.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

rframe.setLayout(null);

JButton back = new JButton("🔙 Back");

back.setBounds(5,15,100,20);

back.setCursor(new Cursor(Cursor.HAND\_CURSOR));

back.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

rframe.dispose();

String[] args = {};

main(args);

}

});

JLabel rlb = new JLabel(" STUDENT REGISTRATION FORM");

rlb.setBounds(390,20,350,30);

rlb.setOpaque(true);

rlb.setBackground(Color.RED);

rlb.setForeground(Color.BLACK);

ImageIcon icon = new ImageIcon("C:\\practice\\Java Project\\logo.png");

int w = 80;

int h = 30;

Image scaledImage = icon.getImage().getScaledInstance(w,h,Image.SCALE\_SMOOTH);

ImageIcon scaledIcon = new ImageIcon(scaledImage);

JLabel rlb1 = new JLabel(scaledIcon);

rlb1.setBounds(300,20,w,h);

JLabel rlb2 = new JLabel("Full Name : ");

rlb2.setBounds(120,80,120,25);

JTextField rtf = new JTextField();

rtf.setBounds(240,80,180,25);

rtf.setBackground(new Color(255,213,128));

JLabel rlb3 = new JLabel("Hall no : ");

rlb3.setBounds(120,120,120,25);

JTextField rtf1 = new JTextField();

rtf1.setBounds(240,120,180,25);

rtf1.setBackground(new Color(255,213,128));

JLabel rlb4 = new JLabel("Phone No : ");

rlb4.setBounds(120,150,120,25);

JTextField rtf2 = new JTextField();

rtf2.setBounds(240,150,180,25);

rtf2.setBackground(new Color(255,213,128));

JLabel rlb5 = new JLabel("Father Name : ");

rlb5.setBounds(120,180,120,25);

JTextField rtf3 = new JTextField();

rtf3.setBounds(240,180,180,25);

rtf3.setBackground(new Color(255,213,128));

JLabel rlb6 = new JLabel("Father Occupation : ");

rlb6.setBounds(420,180,150,25);

JTextField rtf4 = new JTextField();

rtf4.setBounds(550,180,180,25);

rtf4.setBackground(new Color(255,213,128));

JLabel rlb7 = new JLabel("Mother Name : ");

rlb7.setBounds(120,220,120,25);

JTextField rtf5 = new JTextField();

rtf5.setBounds(240,220,180,25);

rtf5.setBackground(new Color(255,213,128));

JLabel rlb8 = new JLabel("Mother Occupation : ");

rlb8.setBounds(420,220,150,25);

JTextField rtf6 = new JTextField();

rtf6.setBounds(550,220,180,25);

rtf6.setBackground(new Color(255,213,128));

JLabel rlb9 = new JLabel("Gender : ");

rlb9.setBounds(120,250,120,25);

JTextField rtf7 = new JTextField();

rtf7.setBounds(240,250,180,25);

rtf7.setBackground(new Color(255,213,128));

JRadioButton rbt = new JRadioButton("Male");

rbt.setBounds(180,280,120,25);

rbt.setCursor(new Cursor(Cursor.HAND\_CURSOR));

JRadioButton rbt1 = new JRadioButton("Female");

rbt1.setBounds(300,280,120,25);

rbt1.setCursor(new Cursor(Cursor.HAND\_CURSOR));

ButtonGroup group = new ButtonGroup();

group.add(rbt);

group.add(rbt1);

rbt.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

rtf7.setText("Male");

}

});

rbt1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

rtf7.setText("Female");

}

});

JLabel rlb10 = new JLabel("Password : ");

rlb10.setBounds(120,310,120,25);

JPasswordField rtf8 = new JPasswordField();

rtf8.setBounds(240,310,180,25);

rtf8.setBackground(new Color(255,213,128));

JLabel rlb11 = new JLabel("Re-enter Password : ");

rlb11.setBounds(440,310,120,25);

JTextField rtf9 = new JTextField();

rtf9.setBounds(580,310,180,25);

rtf9.setBackground(new Color(255,213,128));

JButton rbt2 = new JButton("Create");

rbt2.setBounds(850,310,100,25);

rbt2.setCursor(new Cursor(Cursor.HAND\_CURSOR));

JTextField rtf10 = new JTextField();

rtf10.setBounds(240,350,340,25);

rtf10.setBackground(new Color(255,213,128));

rbt2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

String pass1 = new String(rtf8.getPassword());

String pass2 = new String(rtf9.getText());

if(pass1.isEmpty() || pass2.isEmpty()){

rtf10.setText("Please Enter Password");

}

else if(pass1.equals(pass2)){

rtf10.setText("Password Created");

}

else{

rtf10.setText("Password Not Matched Please Re enter password");

rtf8.setText("");

rtf9.setText("");

}

}

});

JLabel rlb12 = new JLabel("Email : ");

rlb12.setBounds(120,380,120,25);

JTextField rtf11 = new JTextField();

rtf11.setBounds(240,380,250,25);

rtf11.setBackground(new Color(255,213,128));

JLabel rlb13 = new JLabel("Date-Of-Birth: ");

rlb13.setBounds(120,420,120,25);

JTextField rtf12 = new JTextField();

rtf12.setBounds(240,420,100,25);

rtf12.setBackground(new Color(255,213,128));

JLabel rlb14 = new JLabel("Day: ");

rlb14.setBounds(360,420,80,25);

Scanner sc = new Scanner(System.in);

String[] days= new String[31];

for(int i=0;i<31;i++){

days[i] = String.valueOf(i+1);

}

JComboBox<String> rcb = new JComboBox<>(days);

rcb.setBounds(410,420,60,25);

JLabel rlb15 = new JLabel("Month : ");

rlb15.setBounds(490,420,80,25);

String[] months = { "Jan", "Feb", "Mar", "Apr", "May", "June", "July", "Aug", "Sep", "Oct", "Nov", "Dec"};

JComboBox<String> rcb1 = new JComboBox<>(months);

rcb1.setBounds(560,420,60,25);

String[] years = new String[100];

int cur = 2024;

for(int i=0;i<100;i++){

years[i]= String.valueOf(cur-i);

}

JLabel rlb16 = new JLabel("Year : ");

rlb16.setBounds(650,420,60,25);

JComboBox<String> rcb2 = new JComboBox<>(years);

rcb2.setBounds(710,420,60,25);

JButton rbt3 = new JButton("Enter");

rbt3.setBounds(800,420,120,25);

rbt3.setCursor(new Cursor(Cursor.HAND\_CURSOR));

rbt3.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

String dd = (String) rcb.getSelectedItem();

String mm = (String) rcb1.getSelectedItem();

String yy = (String) rcb2.getSelectedItem();

rtf12.setText(" " + dd + " - "+ mm + " - "+yy);

}

});

JLabel rlb18 = new JLabel("Country : ");

rlb18.setBounds(200,460,100,25);

JTextField rtf13 = new JTextField();

rtf13.setBounds(290,460,120,25);

rtf13.setBackground(new Color(255,213,128));

JLabel rlb19 = new JLabel("State : ");

rlb19.setBounds(450,460,100,25);

JTextField rtf14 = new JTextField();

rtf14.setBounds(550,460,120,25);

rtf14.setBackground(new Color(255,213,128));

JLabel rlb20 = new JLabel("Pincode : ");

rlb20.setBounds(700,460,100,25);

JTextField rtf15 = new JTextField();

rtf15.setBounds(810,460,120,25);

rtf15.setBackground(new Color(255,213,128));

JLabel rlb21 = new JLabel("City : ");

rlb21.setBounds(200,500,100,25);

JTextField rtf16 = new JTextField();

rtf16.setBounds(290,500,120,25);

rtf16.setBackground(new Color(255,213,128));

JLabel rlb22 = new JLabel("Village : ");

rlb22.setBounds(450,500,100,25);

JTextField rtf17 = new JTextField();

rtf17.setBounds(550,500,120,25);

rtf17.setBackground(new Color(255,213,128));

JLabel rlb23 = new JLabel("House.no : ");

rlb23.setBounds(700,500,100,25);

JTextField rtf18 = new JTextField();

rtf18.setBounds(810,500,120,25);

rtf18.setBackground(new Color(255,213,128));

JButton rbt4 = new JButton("Enter");

rbt4.setBounds(480,530,120,25);

rbt4.setCursor(new Cursor(Cursor.HAND\_CURSOR));

JLabel rlb17 = new JLabel("Address: ");

rlb17.setBounds(120,570,100,25);

JTextField rtf19 = new JTextField();

rtf19.setBounds(200,570,390,25);

rtf19.setBackground(new Color(255,213,128));

rbt4.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

String cou = new String(rtf13.getText());

String sta = new String(rtf14.getText());

String pin = new String(rtf15.getText());

String cit = new String(rtf16.getText());

String vil = new String(rtf17.getText());

String hsno = new String(rtf18.getText());

if(cou.isEmpty() || sta.isEmpty() || pin.isEmpty() || cit.isEmpty() || vil.isEmpty() || hsno.isEmpty()){

JOptionPane.showMessageDialog(null,"Please fill all address blocks !!");

}

else{

rtf19.setText(" "+cou+", "+sta+", "+cit+"(" +pin+ "), "+vil+ ", " +hsno);

}

}

});

JLabel rlb24 = new JLabel("Religion : ");

rlb24.setBounds(120,600,120,25);

JTextField rtf20 = new JTextField();

rtf20.setBounds(240,600,100,25);

rtf20.setBackground(new Color(255,213,128));

JLabel rlb25 = new JLabel("Caste : ");

rlb25.setBounds(380,600,120,25);

JTextField rtf21 = new JTextField();

rtf21.setBounds(440,600,120,25);

rtf21.setBackground(new Color(255,213,128));

JLabel rlb26 = new JLabel("Name of Caste : ");

rlb26.setBounds(580,600,120,25);

JTextField rtf22 = new JTextField();

rtf22.setBounds(690,600,120,25);

rtf22.setBackground(new Color(255,213,128));

JLabel rlb27 = new JLabel("Blood Group:");

rlb27.setBounds(120,640,120,25);

JTextField rtf23 = new JTextField();

rtf23.setBounds(230,640,80,25);

rtf23.setBackground(new Color(255,213,128));

JLabel rlb28 = new JLabel("Identification Mark1:");

rlb28.setBounds(120,680,120,25);

JTextField rtf24 = new JTextField();

rtf24.setBounds(260,680,250,25);

rtf24.setBackground(new Color(255,213,128));

JLabel rlb29 = new JLabel("Identification Mark2:");

rlb29.setBounds(120,720,120,25);

JTextField rtf25 = new JTextField();

rtf25.setBounds(260,720,250,25);

rtf25.setBackground(new Color(255,213,128));

JButton rbt5 = new JButton("Rigester");

rbt5.setBounds(250,780,120,40);

rbt5.setCursor(new Cursor(Cursor.HAND\_CURSOR));

JButton rbt6 = new JButton("Cancel");

rbt6.setBounds(400,780,120,40);

rbt6.setCursor(new Cursor(Cursor.HAND\_CURSOR));

JTextField rtf26 = new JTextField();

rtf26.setBounds(580,790,480,30);

rtf26.setBackground(new Color(255,213,128));

JLabel rlb30 = new JLabel("Select Your Security Question❓ 🙋");

rlb30.setBounds(670,640,250,30);

String[] questions = { "1.Your First School Name ?", "2. Your Pet Name ?", "3.Your Favourite Food?", "4. Your Favourite Place?"};

JComboBox<String> rcb4 = new JComboBox<>(questions);

rcb4.setBounds(630,680,280,30);

JLabel rlb31 = new JLabel("Answer: ");

rlb31.setBounds(590,720,150,30);

JTextField rtf27 = new JTextField();

rtf27.setBounds(650,720,200,30);

rtf27.setBackground(new Color(255,213,128));

rbt6.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

rtf.setText("");

rtf1.setText("");

rtf2.setText("");

rtf3.setText("");

rtf4.setText("");

rtf5.setText("");

rtf6.setText("");

rtf7.setText("");

rtf8.setText("");

rtf9.setText("");

rtf10.setText("");

rtf11.setText("");

rtf12.setText("");

rtf13.setText("");

rtf14.setText("");

rtf15.setText("");

rtf16.setText("");

rtf17.setText("");

rtf18.setText("");

rtf19.setText("");

rtf20.setText("");

rtf21.setText("");

rtf22.setText("");

rtf23.setText("");

rtf24.setText("");

rtf25.setText("");

rtf27.setText("");

}

});

rbt5.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

if (rtf.getText().isEmpty() || rtf1.getText().isEmpty() || rtf2.getText().isEmpty() || rtf3.getText().isEmpty() || rtf4.getText().isEmpty() || rtf5.getText().isEmpty() || rtf6.getText().isEmpty() || rtf7.getText().isEmpty() || rtf8.getPassword().length == 0 || rtf9.getText().isEmpty() || rtf10.getText().isEmpty() || rtf11.getText().isEmpty() || rtf12.getText().isEmpty() || rtf13.getText().isEmpty() || rtf14.getText().isEmpty() || rtf15.getText().isEmpty() || rtf16.getText().isEmpty() || rtf17.getText().isEmpty() || rtf18.getText().isEmpty() || rtf19.getText().isEmpty() || rtf20.getText().isEmpty() || rtf21.getText().isEmpty() || rtf22.getText().isEmpty() || rtf23.getText().isEmpty() || rtf24.getText().isEmpty() || rtf25.getText().isEmpty() || rtf27.getText().isEmpty()) {

JOptionPane.showMessageDialog(null,"!! MAKE SURE YOU FILLED ALL CONTENTS !!");

}else{

rtf26.setText("");

String name = new String(rtf.getText());

String hall = new String(rtf1.getText());

String phone = new String(rtf2.getText());

String father = new String(rtf3.getText());

String focp = new String(rtf4.getText());

String mother = new String(rtf5.getText());

String mocp = new String(rtf6.getText());

String gender = new String(rtf7.getText());

String password = new String(rtf9.getText());

String email = new String(rtf11.getText());

String dob = new String(rtf12.getText());

String country = new String(rtf13.getText());

String state = new String(rtf14.getText());

String pin = new String(rtf15.getText());

String city = new String(rtf16.getText());

String village = new String(rtf17.getText());

String hsno = new String(rtf18.getText());

String religion = new String(rtf20.getText());

String caste = new String(rtf21.getText());

String castename = new String(rtf22.getText());

String blood = new String(rtf23.getText());

String idfm1 = new String(rtf24.getText());

String idfm2 = new String(rtf25.getText());

String question = (String) rcb4.getSelectedItem();

String answer = new String(rtf27.getText());

saveToDatabase(name, hall, phone, father, focp, mother, mocp, gender, password, email, dob, country, state, pin, city, village, hsno, religion, caste, castename, blood, idfm1, idfm2,question,answer);

rframe.dispose();

showThankyoupage();

}

}

});

rframe.add(back);

rframe.add(rcb2);

rframe.add(rcb1);

rframe.add(rcb);

rframe.add(rcb4);

rframe.add(rtf);

rframe.add(rtf1);

rframe.add(rtf2);

rframe.add(rtf3);

rframe.add(rtf4);

rframe.add(rtf5);

rframe.add(rtf6);

rframe.add(rtf7);

rframe.add(rtf8);

rframe.add(rtf9);

rframe.add(rtf10);

rframe.add(rtf11);

rframe.add(rtf12);

rframe.add(rtf13);

rframe.add(rtf14);

rframe.add(rtf15);

rframe.add(rtf16);

rframe.add(rtf17);

rframe.add(rtf18);

rframe.add(rtf19);

rframe.add(rtf20);

rframe.add(rtf21);

rframe.add(rtf22);

rframe.add(rtf23);

rframe.add(rtf24);

rframe.add(rtf25);

rframe.add(rtf27);

rframe.add(rlb1);

rframe.add(rlb2);

rframe.add(rlb3);

rframe.add(rlb4);

rframe.add(rlb5);

rframe.add(rlb6);

rframe.add(rlb7);

rframe.add(rlb8);

rframe.add(rlb9);

rframe.add(rlb10);

rframe.add(rlb11);

rframe.add(rlb12);

rframe.add(rlb13);

rframe.add(rlb14);

rframe.add(rlb15);

rframe.add(rlb16);

rframe.add(rlb17);

rframe.add(rlb18);

rframe.add(rlb19);

rframe.add(rlb20);

rframe.add(rlb21);

rframe.add(rlb22);

rframe.add(rlb23);

rframe.add(rlb24);

rframe.add(rlb25);

rframe.add(rlb26);

rframe.add(rlb27);

rframe.add(rlb28);

rframe.add(rlb29);

rframe.add(rlb30);

rframe.add(rlb31);

rframe.add(rlb);

rframe.add(rbt);

rframe.add(rbt1);

rframe.add(rbt2);

rframe.add(rbt3);

rframe.add(rbt4);

rframe.add(rbt5);

rframe.add(rbt6);

rframe.setVisible(true);

}

private static void saveToDatabase(String name, String hall, String phone, String father, String focp, String mother, String mocp, String gender, String password, String email, String dob, String country, String state, String pin, String city, String village, String hsno, String religion, String caste, String castename, String blood, String idfm1, String idfm2,String question,String answer){

try{

Class.forName("org.postgresql.Driver");

try(Connection conn = DriverManager.getConnection(DB\_URL,DB\_USER,DB\_PASSWORD)){

String sql = "INSERT INTO Students (\"Full Name\", \"Hall No\", \"Phone No\", \"Father Name\", " + "\"Father Occupation\", \"Mother Name\", \"Mother Occupation\", \"Gender\", \"Password\", " + "\"Email\", \"Date of Birth\", \"Country\", \"State\", \"City\", \"Village\", \"Pincode\", " + "\"House No\", \"Religion\", \"Caste\", \"Caste Name\", \"Blood Group\", " + "\"Identification Marks1\", \"Identification Marks2\", \"security\_question\", \"answer\") " + "VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

try(PreparedStatement pstmt = conn.prepareStatement(sql)){

pstmt.setString(1, name);

pstmt.setString(2, hall);

pstmt.setString(3, phone);

pstmt.setString(4, father);

pstmt.setString(5, focp);

pstmt.setString(6, mother);

pstmt.setString(7, mocp);

pstmt.setString(8, gender);

pstmt.setString(9, password);

pstmt.setString(10, email);

pstmt.setString(11, dob);

pstmt.setString(12, country);

pstmt.setString(13, state);

pstmt.setString(14, pin);

pstmt.setString(15, city);

pstmt.setString(16, village);

pstmt.setString(17, hsno);

pstmt.setString(18, religion);

pstmt.setString(19, caste);

pstmt.setString(20, castename);

pstmt.setString(21, blood);

pstmt.setString(22, idfm1);

pstmt.setString(23, idfm2);

pstmt.setString(24, question);

pstmt.setString(25, answer);

pstmt.executeUpdate();

JOptionPane.showMessageDialog(null,"Data Saved Successfully!");

}

}

}catch (ClassNotFoundException e){

JOptionPane.showMessageDialog(null, "PsotgreSQL JDBC Driver Not Found");

e.printStackTrace();

}catch (SQLException e){

JOptionPane.showMessageDialog(null, "DataBase error: " +e.getMessage());

e.printStackTrace();

}

}

private static Connection getConnection() throws SQLException {

try {

return DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

} catch (SQLException e) {

JOptionPane.showMessageDialog(null,

"Unable to connect to the database. Please check your internet connection and try again.");

throw e;

}

}

public static void showThankyoupage(){

JFrame sframe = new JFrame("Login Page");

sframe.setSize(1000,1000);

sframe.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

sframe.setLayout(null);

sframe.setVisible(true);

JLabel sbl1 = new JLabel("!! You Have Successfully Rigestered !!");

sbl1.setBounds(380,280,500,80);

JLabel sbl2 = new JLabel("-> Click Next and Enter Your Mail And Password To Login <-");

sbl2.setBounds(340,350,400,80);

JButton sbt = new JButton("NEXT");

sbt.setBounds(430,420,150,40);

sbt.setCursor(new Cursor(Cursor.HAND\_CURSOR));

sbt.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

sframe.dispose();

showLoginPage();

}

});

sframe.add(sbt);

sframe.add(sbl1);

sframe.add(sbl2);

}

public static void showLoginPage(){

JFrame lframe = new JFrame("Login Page");

lframe.setSize(1000,1000);

lframe.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

lframe.setLayout(null);

lframe.setVisible(true);

JLabel llb = new JLabel(" STUDENT LOGIN");

llb.setBounds(150,30,550,50);

llb.setFont(new Font("Arial", Font.BOLD, 18));

llb.setOpaque(true);

llb.setBackground(Color.RED);

lframe.add(llb);

JButton lbtn = new JButton("🔙 Back");

lbtn.setBounds(10,05,100,30);

lframe.add(lbtn);

lbtn.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

String[] args={};

lframe.dispose();

main(args);

}

});

JLabel llb1 = new JLabel("Email: ");

llb1.setFont(new Font("Times New Roman", Font.BOLD, 18));

llb1.setBounds(140,200,80,30);

lframe.add(llb1);

JTextField ltf = new JTextField();

ltf.setBounds(270,200,200,30);

ltf.setBorder(new LineBorder(new Color(255, 213, 128), 5, true));

lframe.add(ltf);

JLabel llb2 = new JLabel("Password: ");

llb2.setFont(new Font("Times New Roman", Font.BOLD, 18));

llb2.setBounds(140,300,190,30);

lframe.add(llb2);

JPasswordField ltf1 = new JPasswordField();

ltf1.setBounds(270,300,200,30);

ltf1.setBorder(new LineBorder(new Color(255, 213, 128), 5, true));

lframe.add(ltf1);

ImageIcon icon = new ImageIcon("C:\\Users\\SRIVARSHAN\\OneDrive\\Pictures\\Screenshots\\Screenshot 2024-12-31 200513.png");

int w1=50;

int h1=30;

Image scaledImage = icon.getImage().getScaledInstance(w1,h1,Image.SCALE\_SMOOTH);

ImageIcon scaledIcon = new ImageIcon(scaledImage);

JLabel llb4 = new JLabel(scaledIcon);

llb4.setBounds(400,300,200,30);

lframe.add(llb4);

llb4.setCursor(Cursor.getPredefinedCursor(Cursor.HAND\_CURSOR));

llb4.addMouseListener(new MouseAdapter() {

boolean isVisible = false;

public void mouseClicked(MouseEvent e){

isVisible = !isVisible;

if(isVisible) {

ltf1.setEchoChar((char)0);

} else {

ltf1.setEchoChar('•');

}

}});

JLabel llb3 = new JLabel("Forgot Password?");

llb3.setBounds(370,335,180,20);

llb3.setForeground(Color.BLUE);

lframe.add(llb3);

llb3.setCursor(Cursor.getPredefinedCursor(Cursor.HAND\_CURSOR));

llb3.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

String email = new String(ltf.getText());

if(email.isEmpty()) {

JOptionPane.showMessageDialog(null, "Please Enter Email!!");

} else {

try {

Class.forName("org.postgresql.Driver");

Connection conn = DriverManager.getConnection(

"jdbc:postgresql://localhost:5432/java gui", "postgres", "123456");

String query = "SELECT \* FROM Students WHERE \"Email\" = ?";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setString(1, email);

ResultSet rs = stmt.executeQuery();

if (!rs.next()) {

JOptionPane.showMessageDialog(null, "This email is not registered. Please register first.");

} else {

lframe.dispose();

showForgotPassWordPage(email);

}

conn.close();

} catch (Exception ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, "Error connecting to database: " + ex.getMessage());

}

}

}

});

JButton lbt = new JButton("Login");

lbt.setBounds(230,380,120,30);

lbt.setCursor(new Cursor(Cursor.HAND\_CURSOR));

lframe.add(lbt);

lbt.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String email = new String(ltf.getText());

String password = new String(ltf1.getPassword());

if(email.isEmpty() || password.isEmpty()) {

JOptionPane.showMessageDialog(null, "Please Enter Username and Password");

} else {

try {

Class.forName("org.postgresql.Driver");

Connection conn = DriverManager.getConnection(

"jdbc:postgresql://localhost:5432/java gui", "postgres", "123456");

String query = "SELECT \* FROM Students WHERE \"Email\" = ?";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setString(1, email);

ResultSet rs = stmt.executeQuery();

if (!rs.next()) {

JOptionPane.showMessageDialog(null, "This email is not registered. Please register first.");

} else {

String storedPassword = rs.getString("Password");

if (!password.equals(storedPassword)) {

JOptionPane.showMessageDialog(null, "Wrong password!", "Login Error", JOptionPane.ERROR\_MESSAGE);

} else {

lframe.dispose();

showDetailsPage(email);

}

}

conn.close();

} catch (Exception ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, "Error connecting to database: " + ex.getMessage());

}

}

}

});

JButton lbt2 = new JButton("Cancel");

lbt2.setBounds(430,380,120,30);

lbt2.setCursor(new Cursor(Cursor.HAND\_CURSOR));

lframe.add(lbt2);

}

public static void showForgotPassWordPage(String email) {

JFrame pframe = new JFrame("Reset Password");

pframe.setSize(1000, 1000);

pframe.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

pframe.setLayout(null);

JPanel securityPanel = new JPanel(null);

securityPanel.setBounds(200, 200, 600, 300);

securityPanel.setBorder(BorderFactory.createTitledBorder("Security Verification"));

JLabel securityQuestionLabel = new JLabel("Security Question:");

securityQuestionLabel.setBounds(50, 50, 500, 30);

securityQuestionLabel.setFont(new Font("Arial", Font.BOLD, 14));

JLabel answerLabel = new JLabel("Your Answer:");

answerLabel.setBounds(50, 100, 150, 30);

JTextField answerField = new JTextField();

answerField.setBounds(200, 100, 300, 30);

JButton verifyButton = new JButton("Verify Answer");

verifyButton.setBounds(250, 150, 120, 30);

JButton fbt = new JButton("🔙 Back");

fbt.setBounds(20,20,120,20);

pframe.add(fbt);

fbt.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

pframe.dispose();

showLoginPage();

}

});

securityPanel.add(securityQuestionLabel);

securityPanel.add(answerLabel);

securityPanel.add(answerField);

securityPanel.add(verifyButton);

JPanel resetPanel = new JPanel(null);

resetPanel.setBounds(200, 200, 600, 300);

resetPanel.setBorder(BorderFactory.createTitledBorder("Reset Password"));

resetPanel.setVisible(false);

JLabel newPassLabel = new JLabel("New Password:");

newPassLabel.setBounds(50, 50, 150, 30);

JPasswordField newPassField = new JPasswordField();

newPassField.setBounds(200, 50, 300, 30);

JLabel confirmPassLabel = new JLabel("Confirm Password:");

confirmPassLabel.setBounds(50, 100, 150, 30);

JTextField visiblePassField = new JTextField();

visiblePassField.setBounds(200, 100, 300, 30);

JButton updateButton = new JButton("Update Password");

updateButton.setBounds(250, 150, 150, 30);

resetPanel.add(newPassLabel);

resetPanel.add(newPassField);

resetPanel.add(confirmPassLabel);

resetPanel.add(visiblePassField);

resetPanel.add(updateButton);

pframe.add(securityPanel);

pframe.add(resetPanel);

try {

Class.forName("org.postgresql.Driver");

// Get security question

Connection initialConn = DriverManager.getConnection(

"jdbc:postgresql://localhost:5432/java gui", "postgres", "123456");

String query = "SELECT security\_question, answer FROM Students WHERE \"Email\" = ?";

PreparedStatement stmt = initialConn.prepareStatement(query);

stmt.setString(1, email);

ResultSet rs = stmt.executeQuery();

if (rs.next()) {

String securityQuestion = rs.getString("security\_question");

String correctAnswer = rs.getString("answer");

securityQuestionLabel.setText("Security Question: " + securityQuestion);

initialConn.close(); // Close initial connection after getting security question

verifyButton.addActionListener(e -> {

String userAnswer = answerField.getText().trim();

if (userAnswer.equalsIgnoreCase(correctAnswer)) {

securityPanel.setVisible(false);

resetPanel.setVisible(true);

} else {

JOptionPane.showMessageDialog(pframe, "Incorrect answer. Please try again.");

}

});

updateButton.addActionListener(e -> {

String newPass = new String(newPassField.getPassword());

String confirmPass = visiblePassField.getText();

if (newPass.isEmpty() || confirmPass.isEmpty()) {

JOptionPane.showMessageDialog(pframe, "Please fill in both password fields.");

return;

}

if (!newPass.equals(confirmPass)) {

JOptionPane.showMessageDialog(pframe, "Passwords do not match!");

return;

}

// Create new connection for update

try (Connection updateConn = DriverManager.getConnection(

"jdbc:postgresql://localhost:5432/java gui", "postgres", "123456")) {

String updateQuery = "UPDATE Students SET \"Password\" = ? WHERE \"Email\" = ?";

try (PreparedStatement updateStmt = updateConn.prepareStatement(updateQuery)) {

updateStmt.setString(1, newPass);

updateStmt.setString(2, email);

int rowsAffected = updateStmt.executeUpdate();

if (rowsAffected > 0) {

JOptionPane.showMessageDialog(pframe, "Password updated successfully!");

pframe.dispose();

showLoginPage();

} else {

JOptionPane.showMessageDialog(pframe, "Failed to update password. Please try again.");

}

}

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(pframe, "Error updating password: " + ex.getMessage());

}

});

}

} catch (Exception ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(pframe, "Error retrieving security question: " + ex.getMessage());

}

JButton backButton = new JButton("Back");

backButton.setBounds(20, 900, 120, 30);

backButton.addActionListener(e -> {

pframe.dispose();

showLoginPage();

});

pframe.add(backButton);

pframe.setVisible(true);

}

public static void showDetailsPage(String email) {

JFrame dframe = new JFrame("Details Page");

dframe.setSize(2000, 2000);

dframe.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

dframe.setLayout(null);

JButton close = new JButton("✖ CLOSE");

close.setBounds(1900,40,100,30);

dframe.add(close);

JTextArea detailsArea = new JTextArea();

detailsArea.setEditable(false);

detailsArea.setFont(new Font("Arial", Font.PLAIN, 16));

detailsArea.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

JScrollPane scrollPane = new JScrollPane(detailsArea);

scrollPane.setBounds(20, 20, 700, 700);

dframe.add(scrollPane);

JButton dbt = new JButton("🔙 BACK");

dbt.setBounds(1100,1100,100,30);

dframe.add(dbt);

dbt.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

dframe.dispose();

showLoginPage();

}

});

try {

Class.forName("org.postgresql.Driver");

Connection conn = DriverManager.getConnection(

"jdbc:postgresql://localhost:5432/java gui", "postgres", "123456");

String query = "SELECT \* FROM Students WHERE \"Email\" = ?";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setString(1, email);

ResultSet rs = stmt.executeQuery();

if (rs.next()) {

StringBuilder details = new StringBuilder();

details.append("=============== PERSONAL INFORMATION ===============\n\n");

details.append(String.format("%-25s %s\n", "Full Name:", rs.getString("Full Name")));

details.append(String.format("%-25s %s\n", "Hall No:", rs.getString("Hall No")));

details.append(String.format("%-25s %s\n", "Phone No:", rs.getString("Phone No")));

details.append(String.format("%-25s %s\n", "Gender:", rs.getString("Gender")));

details.append(String.format("%-25s %s\n", "Date of Birth:", rs.getString("Date of Birth")));

details.append(String.format("%-25s %s\n", "Blood Group:", rs.getString("Blood Group")));

details.append("\n");

details.append("=============== FAMILY INFORMATION ===============\n\n");

details.append(String.format("%-25s %s\n", "Father Name:", rs.getString("Father Name")));

details.append(String.format("%-25s %s\n", "Father Occupation:", rs.getString("Father Occupation")));

details.append(String.format("%-25s %s\n", "Mother Name:", rs.getString("Mother Name")));

details.append(String.format("%-25s %s\n", "Mother Occupation:", rs.getString("Mother Occupation")));

details.append("\n");

details.append("=============== CONTACT INFORMATION ===============\n\n");

details.append(String.format("%-25s %s\n", "Email:", rs.getString("Email")));

details.append(String.format("%-25s %s\n", "Country:", rs.getString("Country")));

details.append(String.format("%-25s %s\n", "State:", rs.getString("State")));

details.append(String.format("%-25s %s\n", "City:", rs.getString("City")));

details.append(String.format("%-25s %s\n", "Village:", rs.getString("Village")));

details.append(String.format("%-25s %s\n", "Pincode:", rs.getString("Pincode")));

details.append(String.format("%-25s %s\n", "House No:", rs.getString("House No")));

details.append("\n");

details.append("=============== OTHER INFORMATION ===============\n\n");

details.append(String.format("%-25s %s\n", "Religion:", rs.getString("Religion")));

details.append(String.format("%-25s %s\n", "Caste:", rs.getString("Caste")));

details.append(String.format("%-25s %s\n", "Caste Name:", rs.getString("Caste Name")));

details.append(String.format("%-25s %s\n", "Identification Mark 1:", rs.getString("Identification Marks1")));

details.append(String.format("%-25s %s\n", "Identification Mark 2:", rs.getString("Identification Marks2")));

detailsArea.setText(details.toString());

detailsArea.setBackground(new Color(255, 213, 128));

}

conn.close();

} catch (Exception e) {

e.printStackTrace();

JOptionPane.showMessageDialog(dframe, "Error retrieving data: " + e.getMessage());

}

JButton backButton = new JButton("Back to Login");

backButton.setBounds(20, 930, 120, 30);

backButton.setFont(new Font("Arial", Font.PLAIN, 20));

backButton.addActionListener(e -> {

dframe.dispose();

showLoginPage();

});

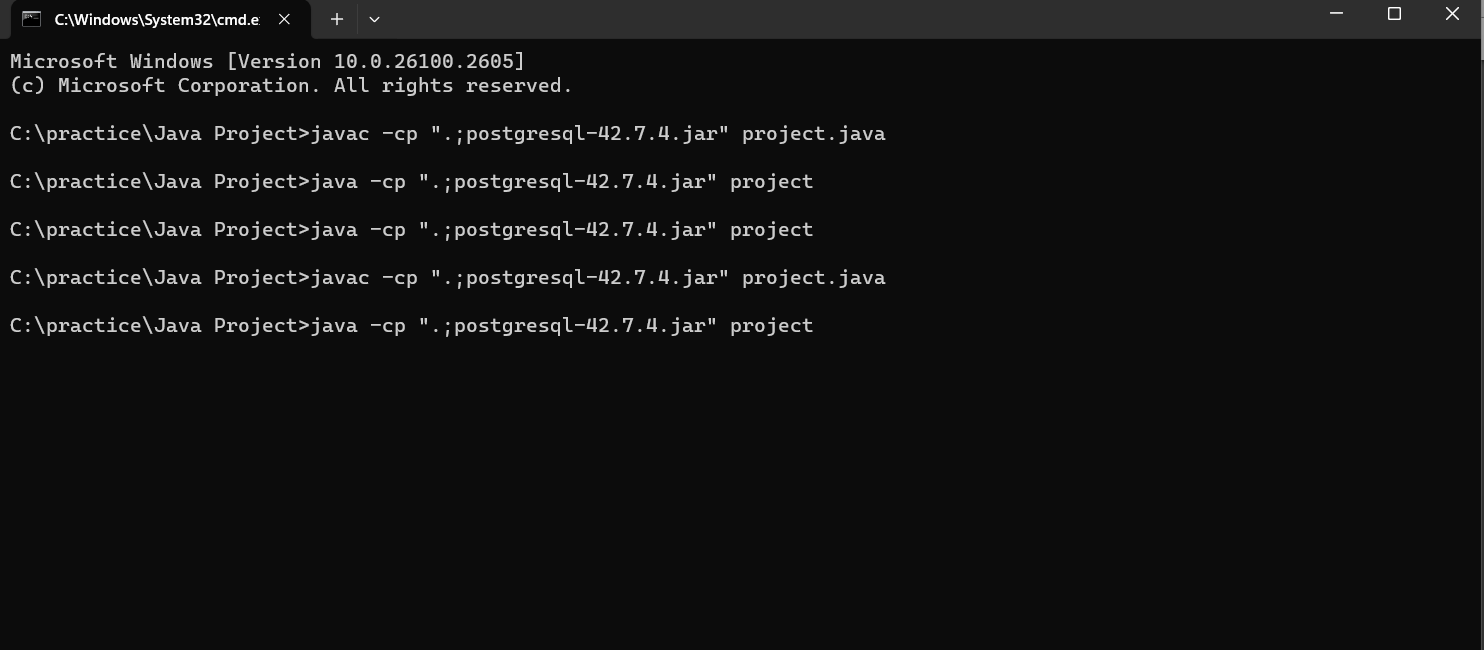
dframe.add(backButton);

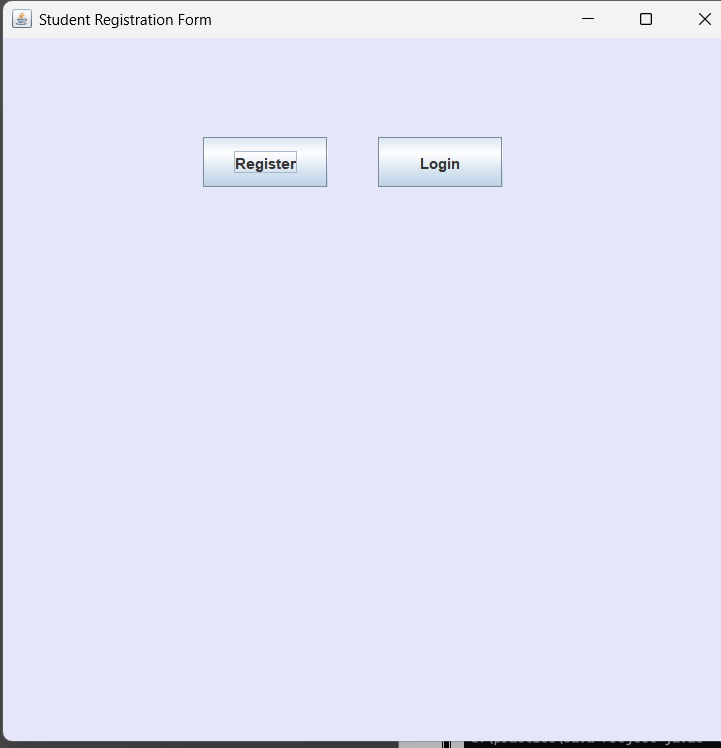
dframe.setVisible(true);

}

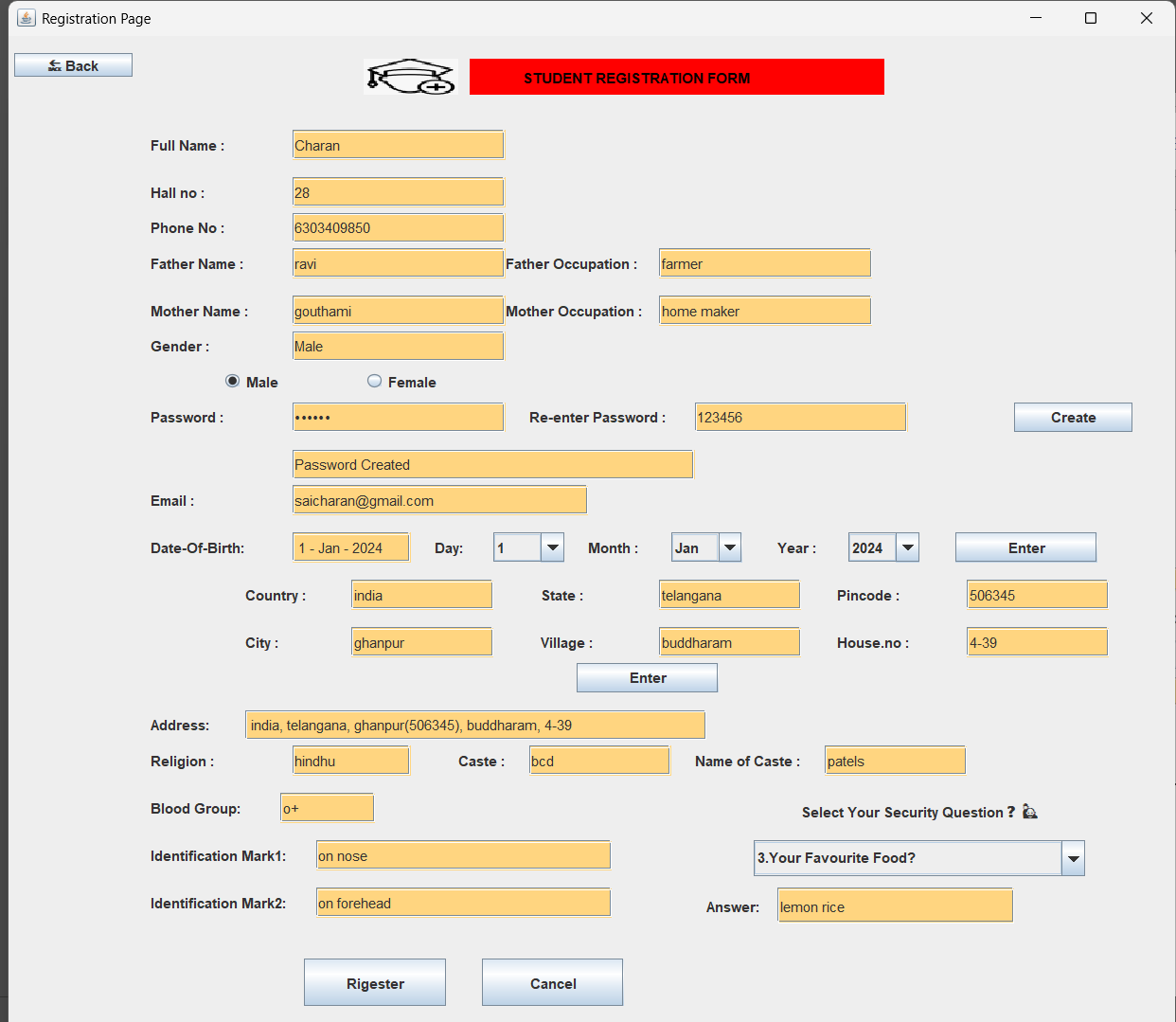
}

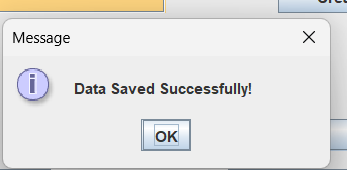
**Output Screenshots:**

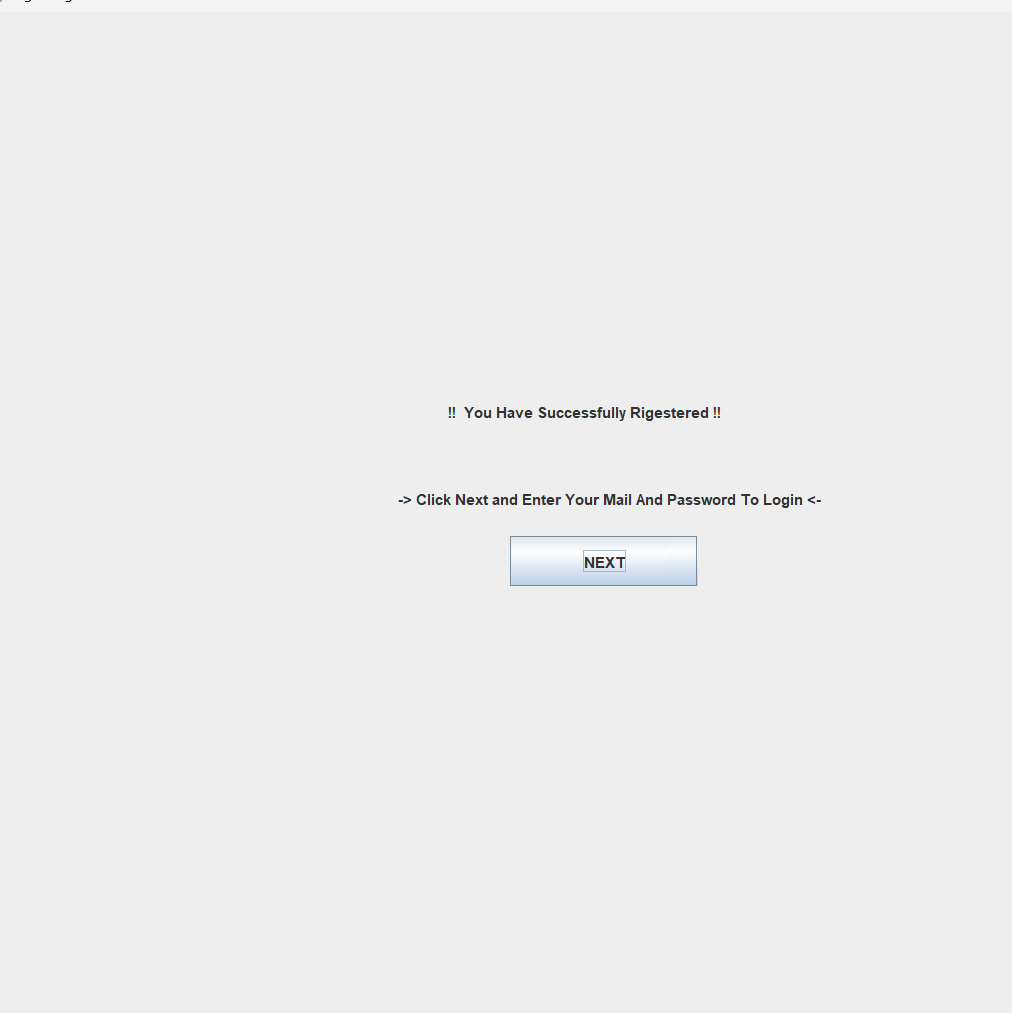
****

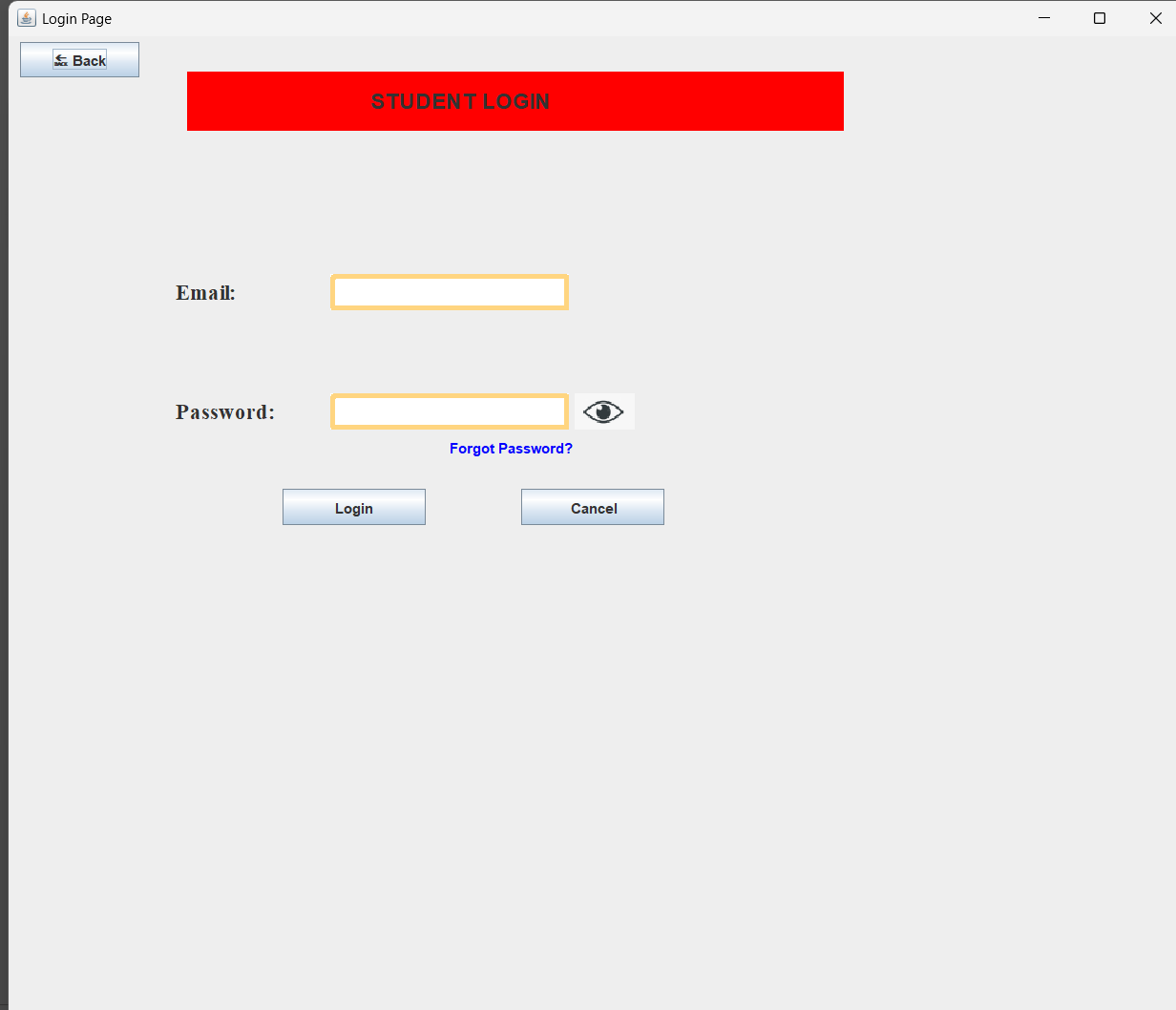
****

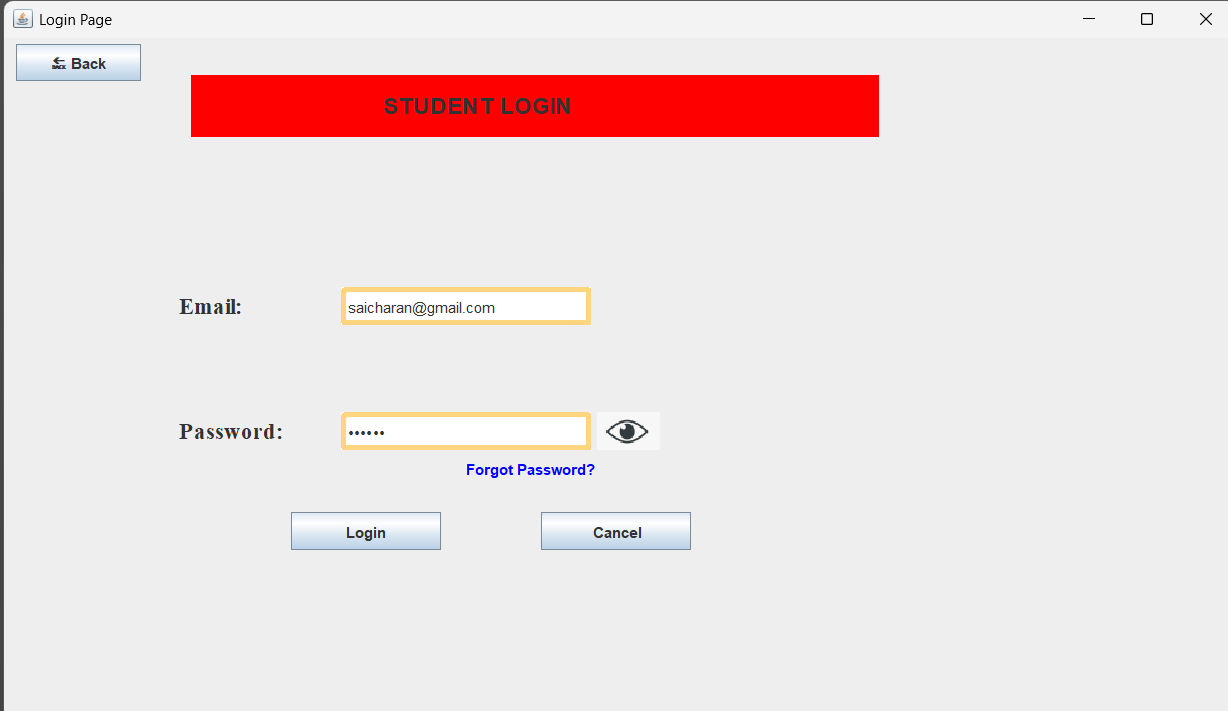
****

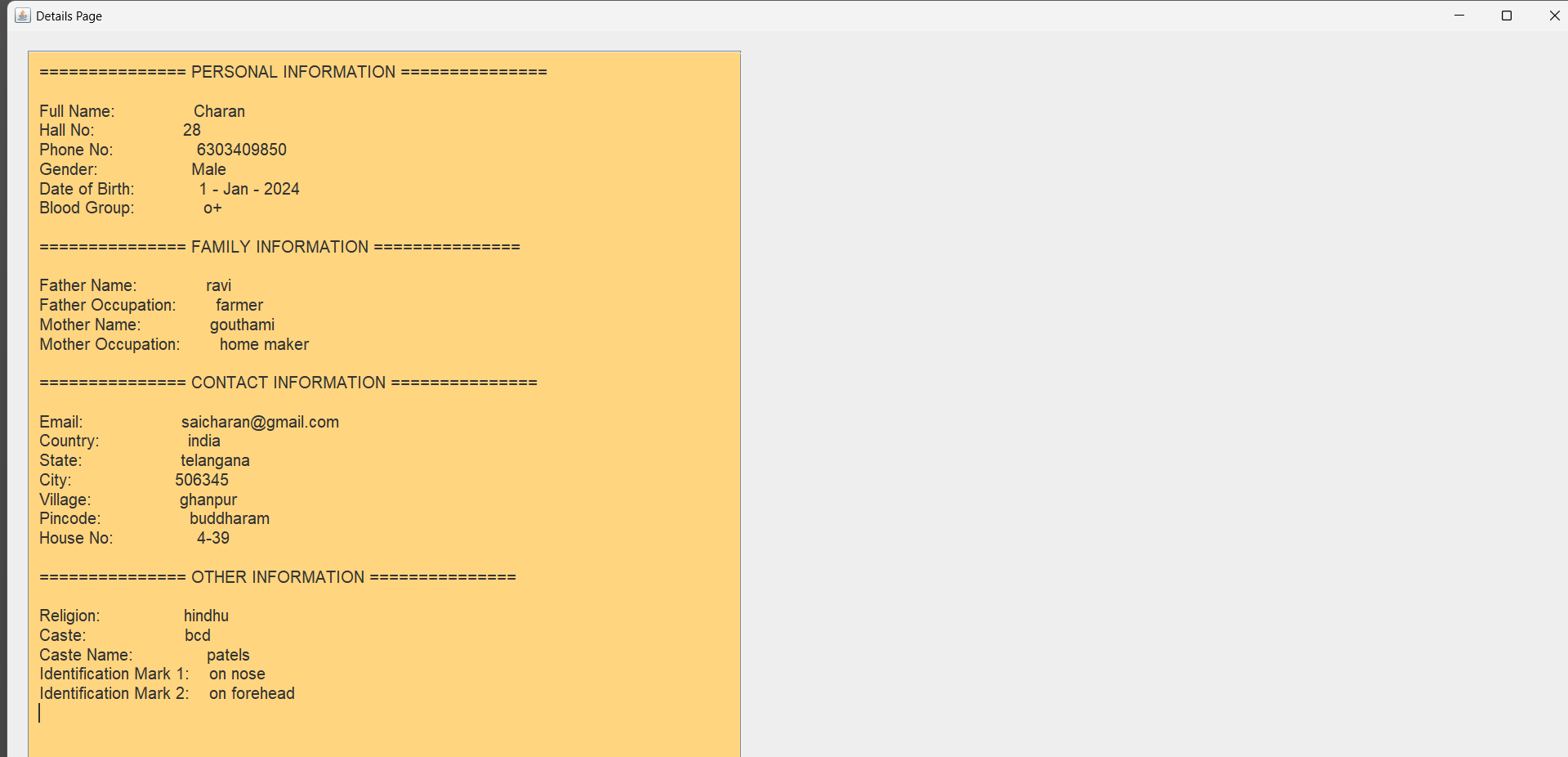
****

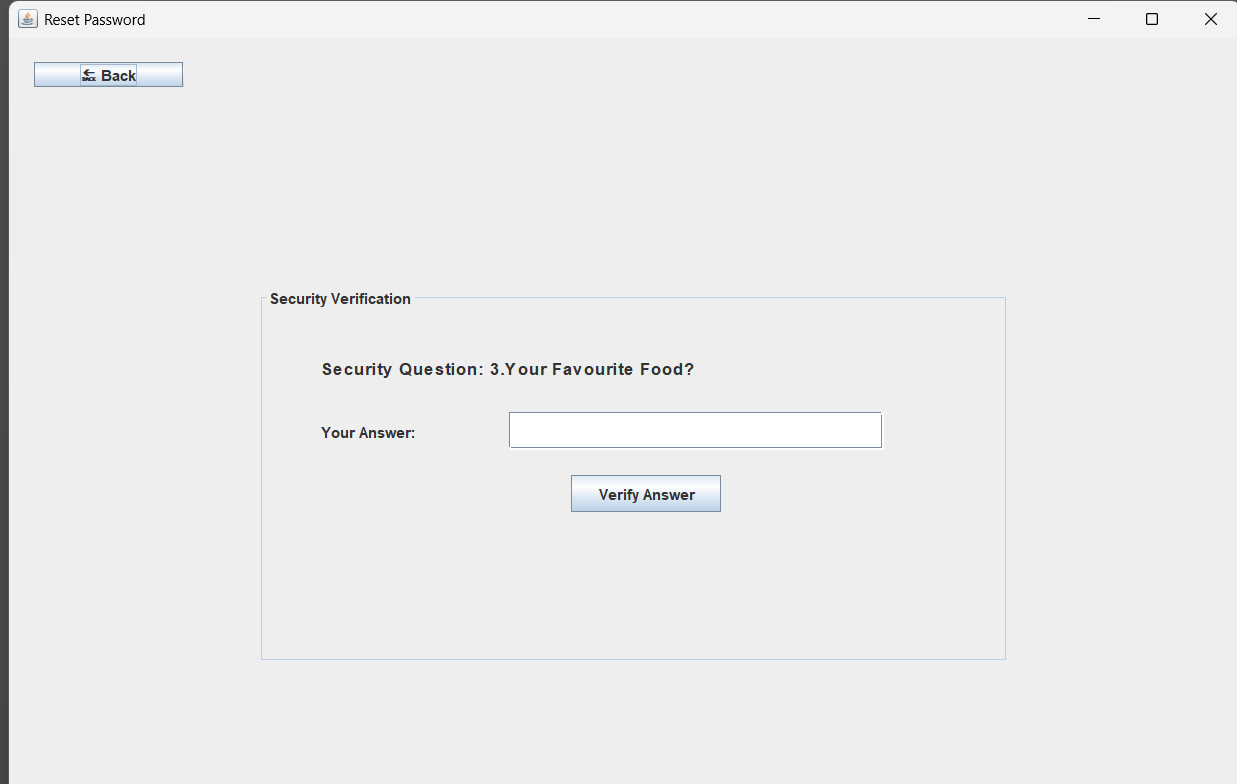
****

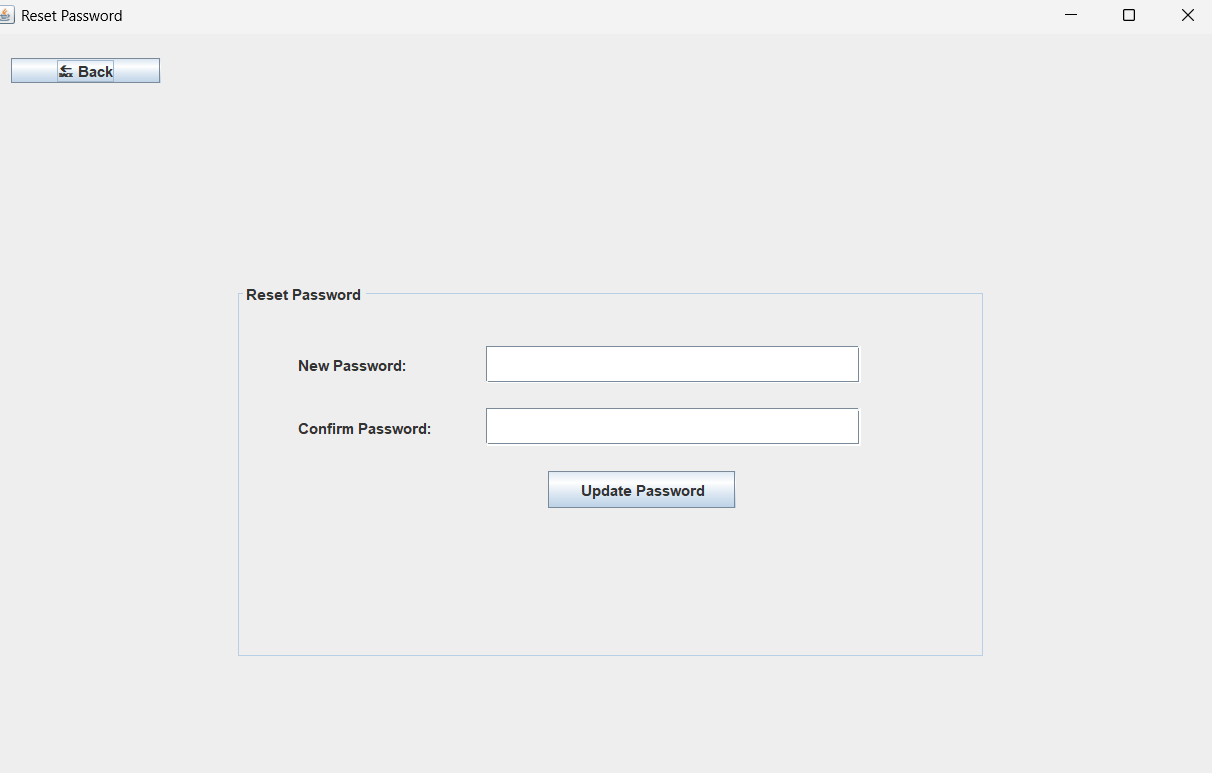
****

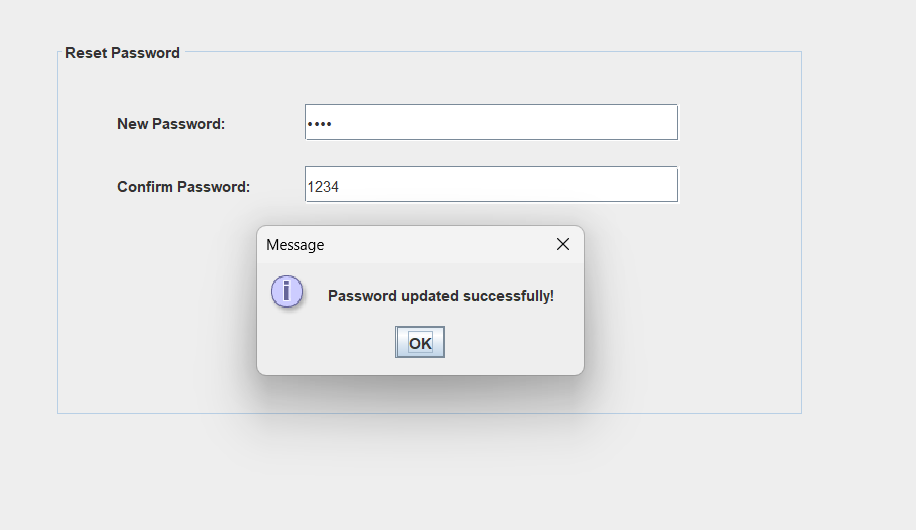
****

****

****

****

****

****