**SCD Lab 08**

**Name:** Hafsa Salman

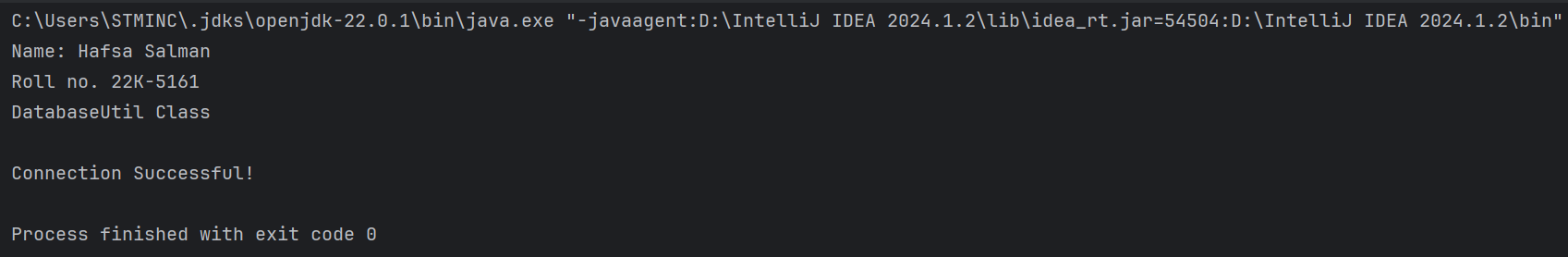
**Roll no.** 22K-5161

**DatabaseUtil Class**

Code:

//Hafsa Salman  
//22K-5161  
//DatabaseUtil Class  
  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
  
public class DatabaseUtil  
{  
 public static final String *JDBC\_URL* = "jdbc:oracle:thin:@localhost:1521:xe";  
 public static final String *USER* = "system";  
 public static final String *PASS* = "abc123";  
  
 public static Connection getConnection() throws SQLException  
 {  
 return DriverManager.*getConnection*(*JDBC\_URL*, *USER*, *PASS*);  
 }  
  
 public static void main(String[] args)  
 {  
 System.*out*.println("Name: Hafsa Salman");  
 System.*out*.println("Roll no. 22K-5161");  
 System.*out*.println("DatabaseUtil Class");  
 System.*out*.println();  
  
 Connection connection = null;  
  
 try  
 {  
 connection = *getConnection*();  
  
 if (connection != null)  
 {  
 System.*out*.println("Connection Successful!");  
 }  
 }  
  
 catch (SQLException e)  
 {  
 System.*out*.println("Connection Failed!");  
  
 e.printStackTrace();  
 }  
  
 finally  
 {  
 if (connection != null)  
 {  
 try  
 {  
 connection.close();  
 }  
  
 catch (SQLException e)  
 {  
 System.*out*.println("Failed to close the connection!");  
  
 e.printStackTrace();  
 }  
 }  
 }  
 }  
}

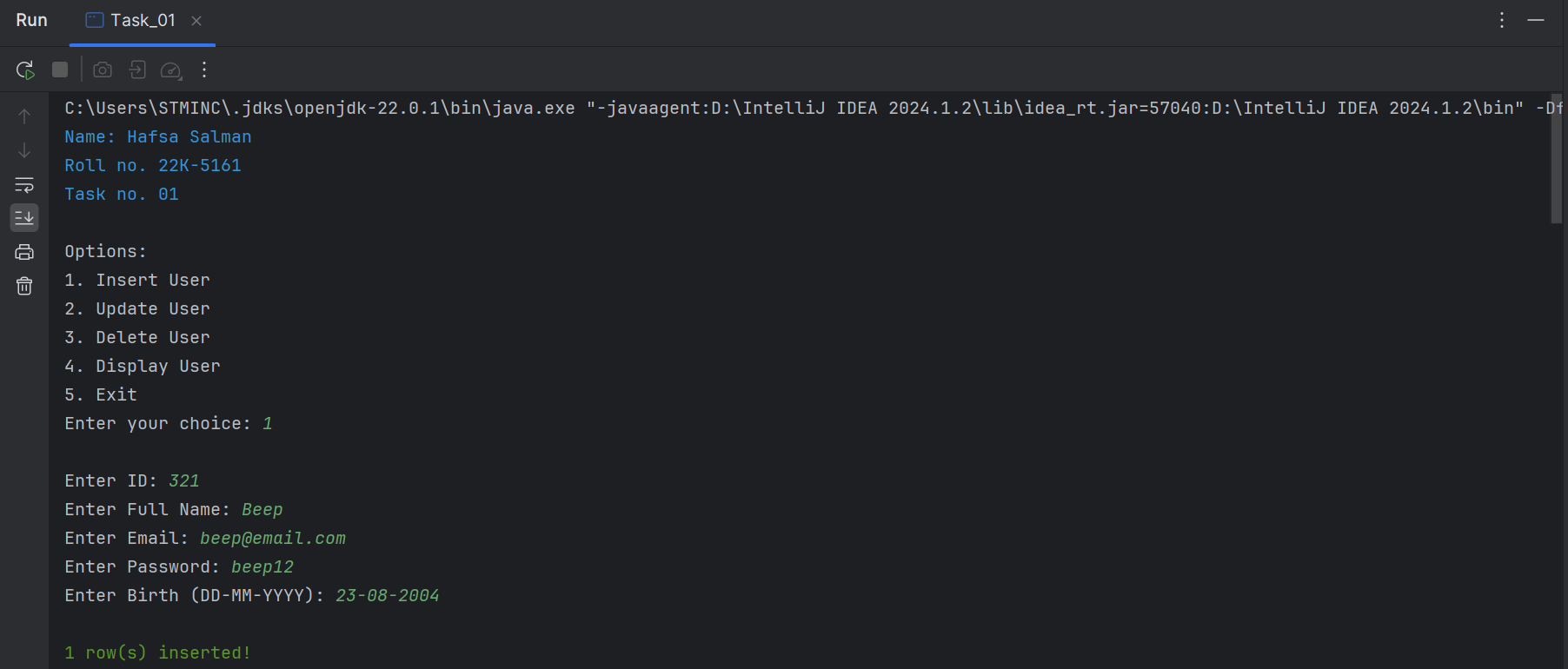
Output:

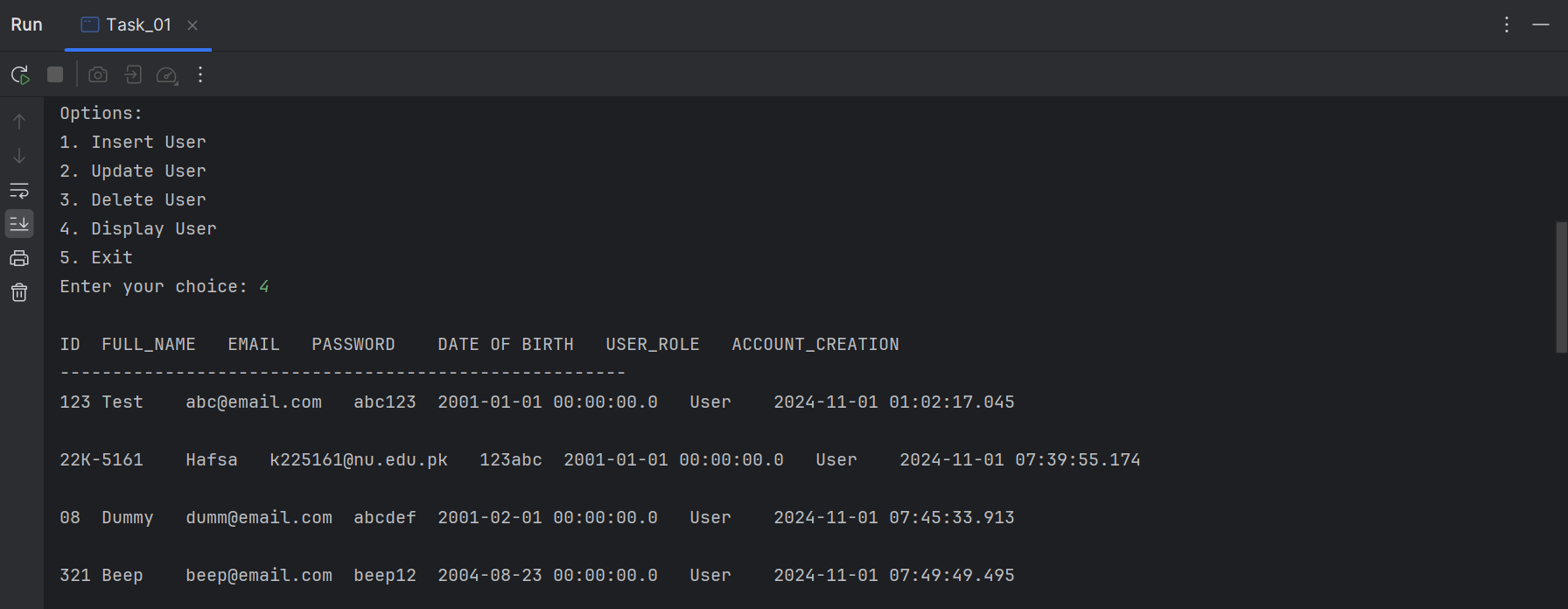


**Task 01**

Code:

//Hafsa Salman  
//22K-5161  
//Task no. 01  
  
import java.sql.\*;  
import java.util.Scanner;  
  
public class Task\_01  
{  
 public static void Insert (String ID, String full\_name, String email, String password, String birth)  
 {  
 String query = "INSERT INTO USERS (ID, FULL\_NAME, EMAIL, PASSWORD, BIRTH, USER\_ROLE, ACCOUNT\_CREATION) VALUES (?, ?, ?, ?, TO\_DATE(?, 'DD-MM-YYYY'), 'User', SYSTIMESTAMP)";  
  
 try (Connection connection = DatabaseUtil.*getConnection*())  
 {  
 PreparedStatement statement = connection.prepareStatement(query);  
  
 statement.setString(1, ID);  
 statement.setString(2, full\_name);  
 statement.setString(3, email);  
 statement.setString(4, password);  
 statement.setString(5, birth);  
  
 int rowsAffected = statement.executeUpdate();  
  
 System.*out*.println();  
 System.*out*.println("\u001B[32m" + rowsAffected + " row(s) inserted!\u001B[0m");  
 System.*out*.println();  
 }  
  
 catch (SQLException e)  
 {  
 System.*out*.println("Connection or insertion has failed!");  
  
 e.printStackTrace();  
 }  
 }  
  
 public static void Update (String ID, String attribute, String str)  
 {  
 if (attribute.equals("ID"))  
 {  
 System.*out*.println("ID can not be changed!");  
 }  
  
 else if (attribute.equals("Name") || attribute.equals("name") || attribute.equals("NAME"))  
 {  
 String query = "UPDATE USERS SET FULL\_NAME = ? WHERE ID = ?";  
  
 try (Connection connection = DatabaseUtil.*getConnection*())  
 {  
 PreparedStatement statement = connection.prepareStatement(query);  
  
 statement.setString(1, str);  
 statement.setString(2, ID);  
  
 int rowsAffected = statement.executeUpdate();  
  
 System.*out*.println();  
 System.*out*.println("\u001B[32m" + rowsAffected + " row(s) updated!\u001B[0m");  
 System.*out*.println();  
 }  
  
 catch (SQLException e)  
 {  
 System.*out*.println("Connection or update has failed!");  
 }  
 }  
  
 else if (attribute.equals("Email") || attribute.equals("email") || attribute.equals("EMAIL"))  
 {  
 String query = "UPDATE USERS SET EMAIL = ? WHERE ID = ?";  
  
 try (Connection connection = DatabaseUtil.*getConnection*())  
 {  
 PreparedStatement statement = connection.prepareStatement(query);  
  
 statement.setString(1, str);  
 statement.setString(2, ID);  
  
 int rowsAffected = statement.executeUpdate();  
  
 System.*out*.println();  
 System.*out*.println("\u001B[32m" + rowsAffected + " row(s) updated!\u001B[0m");  
 System.*out*.println();  
 }  
  
 catch (SQLException e)  
 {  
 System.*out*.println("Connection or update has failed!");  
 }  
 }  
  
 else if (attribute.equals("Password") || attribute.equals("password") || attribute.equals("PASSWORD"))  
 {  
 String query = "UPDATE USERS SET PASSWORD = ? WHERE ID = ?";  
  
 try (Connection connection = DatabaseUtil.*getConnection*())  
 {  
 PreparedStatement statement = connection.prepareStatement(query);  
  
 statement.setString(1, str);  
 statement.setString(2, ID);  
  
 int rowsAffected = statement.executeUpdate();  
  
 System.*out*.println();  
 System.*out*.println("\u001B[32m" + rowsAffected + " row(s) updated!\u001B[0m");  
 System.*out*.println();  
 }  
  
 catch (SQLException e)  
 {  
 System.*out*.println("Connection or update has failed!");  
 }  
 }  
  
 else if (attribute.equals("Date of Birth") || attribute.equals("date of birth") || attribute.equals("DATE OF BIRTH"))  
 {  
 String query = "UPDATE USERS SET BIRTH = ? WHERE ID = ?";  
  
 try (Connection connection = DatabaseUtil.*getConnection*())  
 {  
 PreparedStatement statement = connection.prepareStatement(query);  
  
 statement.setString(1, str);  
 statement.setString(2, ID);  
  
 int rowsAffected = statement.executeUpdate();  
  
 System.*out*.println();  
 System.*out*.println("\u001B[32m" + rowsAffected + " row(s) updated!\u001B[0m");  
 System.*out*.println();  
 }  
  
 catch (SQLException e)  
 {  
 System.*out*.println("Connection or update has failed!");  
 }  
 }  
  
 else  
 {  
 System.*out*.println("Invalid Choice!");  
 }  
 }  
  
 public static void Delete (String ID)  
 {  
 String query = "DELETE FROM USERS WHERE ID = ?";  
  
 try (Connection connection = DatabaseUtil.*getConnection*())  
 {  
 PreparedStatement statement = connection.prepareStatement(query);  
  
 statement.setString(1, ID);  
  
 int rowsAffected = statement.executeUpdate();  
  
 System.*out*.println();  
 System.*out*.println("\u001B[32m" + rowsAffected + " row(s) deleted!\u001B[0m");  
 System.*out*.println();  
 }  
  
 catch (SQLException e)  
 {  
 System.*out*.println("Connection or deletion has failed!");  
 }  
 }  
  
 public static void Display ()  
 {  
 try (Connection connection = DatabaseUtil.*getConnection*())  
 {  
 Statement statement = connection.createStatement();  
  
 String query = "SELECT \* FROM USERS";  
  
 ResultSet rs = statement.executeQuery(query);  
  
 System.*out*.println("ID\tFULL\_NAME\tEMAIL\tPASSWORD\tDATE OF BIRTH\tUSER\_ROLE\tACCOUNT\_CREATION" );  
 System.*out*.println("------------------------------------------------------");  
  
 while (rs.next())  
 {  
 String ID = rs.getString("ID");  
 String full\_name = rs.getString("FULL\_NAME");  
 String email = rs.getString("EMAIL");  
 String password = rs.getString("PASSWORD");  
 String birth = rs.getString("BIRTH");  
 String user\_role = rs.getString("USER\_ROLE");  
 String account = rs.getString("ACCOUNT\_CREATION");  
  
 System.*out*.println(ID + "\t" + full\_name + "\t" + email + "\t" + password + "\t" + birth + "\t" + user\_role + "\t" + account);  
 System.*out*.println();  
 }  
 }  
  
 catch (SQLException e)  
 {  
 System.*out*.println("Connection or query has failed!");  
 }  
 }  
  
 public static void main(String[] args)  
 {  
 System.*out*.println("\u001B[34mName: Hafsa Salman\u001B[0m");  
 System.*out*.println("\u001B[34mRoll no. 22K-5161\u001B[0m");  
 System.*out*.println("\u001B[34mTask no. 01\u001B[0m");  
 System.*out*.println();  
  
 Scanner s = new Scanner(System.*in*);  
  
 int choice;  
  
 do  
 {  
 System.*out*.println("Options: ");  
 System.*out*.println("1. Insert User");  
 System.*out*.println("2. Update User");  
 System.*out*.println("3. Delete User");  
 System.*out*.println("4. Display User");  
 System.*out*.println("5. Exit");  
  
 System.*out*.print("Enter your choice: ");  
 choice = s.nextInt();  
  
 s.nextLine();  
  
 System.*out*.println();  
  
 String ID, full\_name, email, password, birth;  
  
 switch(choice)  
 {  
 case 1:  
 {  
 System.*out*.print("Enter ID: ");  
 ID = s.nextLine();  
  
 System.*out*.print("Enter Full Name: ");  
 full\_name = s.nextLine();  
  
 System.*out*.print("Enter Email: ");  
 email = s.nextLine();  
  
 System.*out*.print("Enter Password: ");  
 password = s.nextLine();  
  
 System.*out*.print("Enter Birth (DD-MM-YYYY): ");  
 birth = s.nextLine();  
  
 *Insert*(ID, full\_name, email, password, birth);  
 }  
  
 break;  
  
 case 2:  
 {  
 System.*out*.print("Enter ID of the user to update: ");  
 ID = s.nextLine();  
  
 System.*out*.print("Enter the attribute (name, email, password, birth) you want to update: ");  
 String attribute = s.nextLine();  
  
 System.*out*.print("Enter " + attribute + ": ");  
 String value = s.nextLine();  
  
 *Update*(ID, attribute, value);  
 }  
  
 break;  
  
 case 3:  
 {  
 System.*out*.print("Enter ID of the user to delete: ");  
 ID = s.nextLine();  
  
 *Delete*(ID);  
 }  
  
 break;  
  
 case 4:  
 {  
 *Display*();  
 }  
  
 break;  
  
 default:  
 {  
 System.*out*.println("Invalid input!");  
 }  
  
 break;  
 }  
 }  
 while (choice != 5);  
  
 s.close();  
 }  
}

Output:  




A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

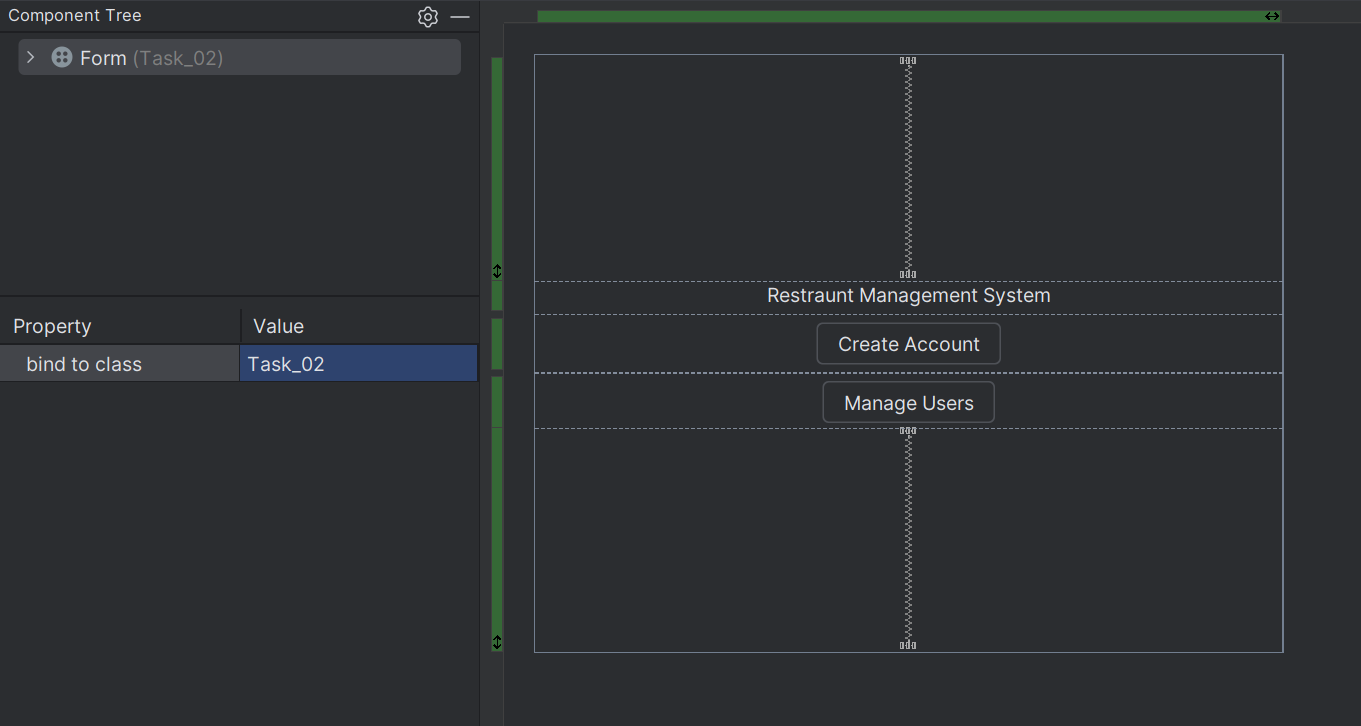
A black rectangle with white dots

Description automatically generated

**Task 02**

Code:

//Hafsa Salman  
//22K-5161  
//Task no. 02  
  
import javax.swing.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class Task\_02 extends JFrame  
{  
 private JPanel Task\_02;  
 private JLabel Label\_01;  
 private JButton createAccountButton;  
 private JButton manageUsersButton;  
  
 public Task\_02()  
 {  
 createAccountButton.addActionListener(new ActionListener()  
 {  
 @Override  
 public void actionPerformed(ActionEvent e)  
 {  
 new Create\_account();  
 }  
 });  
  
 manageUsersButton.addActionListener(new ActionListener()  
 {  
 @Override  
 public void actionPerformed(ActionEvent e)  
 {  
 new Login\_page();  
 }  
 });  
 }  
  
 public static void main(String[] args)  
 {  
 Task\_02 T2 = new Task\_02();  
 T2.setContentPane(T2.Task\_02);  
 T2.setTitle("22K-5161");  
 T2.setSize(500, 500);  
 T2.setVisible(true);  
  
 T2.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 }  
}

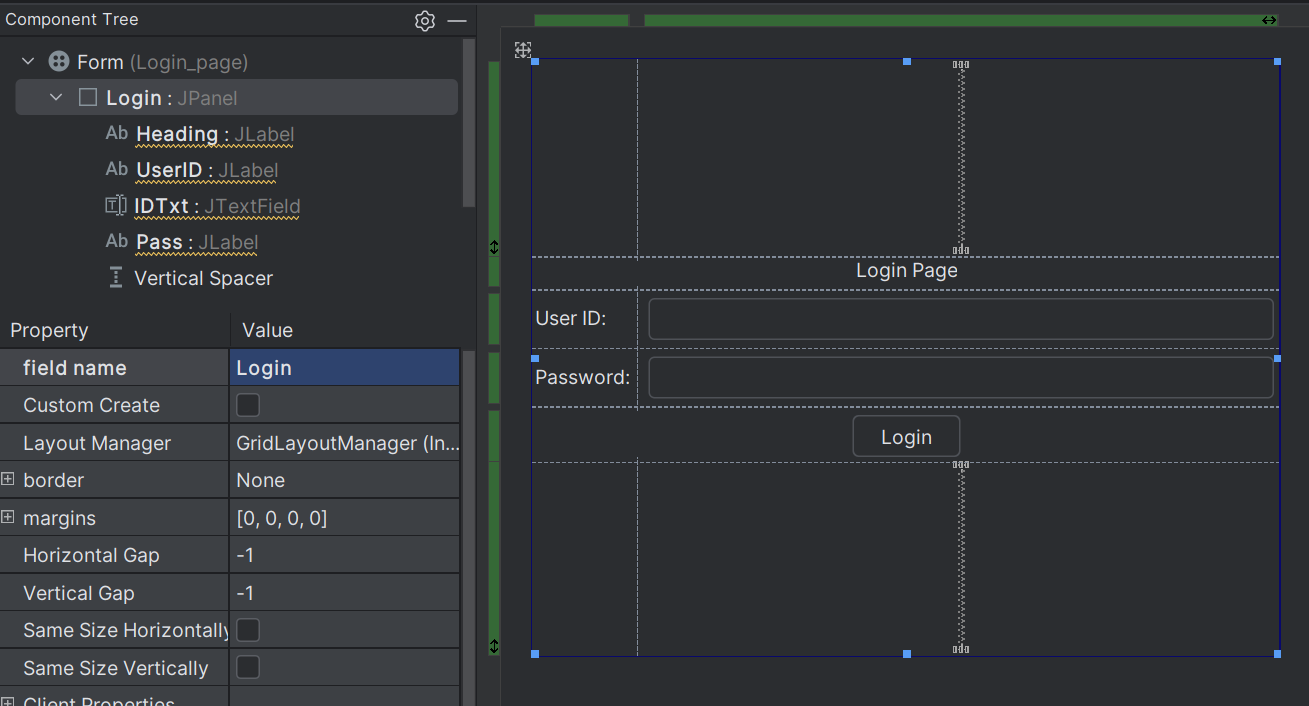


// Hafsa Salman  
// 22K-5161  
// Task no. 02  
  
import javax.swing.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.awt.event.FocusAdapter;  
import java.awt.event.FocusEvent;  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
  
public class Create\_account extends JFrame  
{  
 private JPanel CreateAccount;  
 private JLabel Label01;  
 private JTextField NameTxt;  
 private JLabel Name;  
 private JLabel UserID;  
 private JTextField IDTxt;  
 private JLabel Email;  
 private JTextField EmailTxt;  
 private JPasswordField passwordField1;  
 private JPasswordField passwordField2;  
 private JLabel Pass01;  
 private JLabel Pass02;  
 private JLabel DOBTxt;  
 private JTextField textField1;  
 private JButton createAccountButton;  
  
 public Create\_account()  
 {  
 setTitle("22K-5161 (Task no. 02)");  
 setSize(500, 500);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setContentPane(CreateAccount);  
 setVisible(true);  
  
 IDTxt.addFocusListener(new FocusAdapter()  
 {  
 @Override  
 public void focusLost(FocusEvent e)  
 {  
 String id = IDTxt.getText();  
 if (!id.isEmpty() && UserID(id))  
 {  
 JOptionPane.*showMessageDialog*(Create\_account.this, "UserID already exists! Please choose a different one.");  
 IDTxt.requestFocus();  
 }  
 }  
 });  
  
 EmailTxt.addFocusListener(new FocusAdapter()  
 {  
 @Override  
 public void focusLost(FocusEvent e)  
 {  
 String email = EmailTxt.getText();  
 if (!email.isEmpty() && isEmailTaken(email))  
 {  
 JOptionPane.*showMessageDialog*(Create\_account.this, "Email already exists! Please use a different email.");  
 EmailTxt.requestFocus();  
 }  
 }  
 });  
  
 createAccountButton.addActionListener(new ActionListener()  
 {  
 @Override  
 public void actionPerformed(ActionEvent e)  
 {  
 String id = IDTxt.getText();  
 String fullName = NameTxt.getText();  
 String email = EmailTxt.getText();  
 String password = new String(passwordField1.getPassword());  
 String confirmPassword = new String(passwordField2.getPassword());  
 String birth = DOBTxt.getText(); // No format checking  
  
 if (validateInput(id, fullName, email, password, confirmPassword))  
 {  
 createAccount(id, fullName, email, password, birth);  
 }  
 }  
 });  
 }  
  
 private boolean UserID(String id)  
 {  
 try (Connection conn = DatabaseUtil.*getConnection*())  
 {  
 String query = "SELECT ID FROM TASK\_02 WHERE ID = ?";  
 PreparedStatement ps = conn.prepareStatement(query);  
 ps.setString(1, id);  
 ResultSet rs = ps.executeQuery();  
  
 return rs.next();  
 }  
 catch (SQLException e)  
 {  
 JOptionPane.*showMessageDialog*(this, "Error checking UserID: " + e.getMessage());  
 }  
  
 return false;  
 }  
  
 private boolean isEmailTaken(String email)  
 {  
 try (Connection conn = DatabaseUtil.*getConnection*())  
 {  
 String query = "SELECT EMAIL FROM TASK\_02 WHERE EMAIL = ?";  
 PreparedStatement ps = conn.prepareStatement(query);  
 ps.setString(1, email);  
 ResultSet rs = ps.executeQuery();  
  
 return rs.next();  
 }  
  
 catch (SQLException e)  
 {  
 JOptionPane.*showMessageDialog*(this, "Error checking Email: " + e.getMessage());  
 }  
  
 return false;  
 }  
  
 private boolean validateInput(String id, String fullName, String email, String password, String confirmPassword)  
 {  
 if (id.isEmpty() || fullName.isEmpty() || email.isEmpty() || password.isEmpty())  
 {  
 JOptionPane.*showMessageDialog*(this, "All fields are required!");  
  
 return false;  
 }  
  
 if (!password.equals(confirmPassword))  
 {  
 JOptionPane.*showMessageDialog*(this, "Passwords do not match!");  
  
 return false;  
 }  
  
 return true;  
 }  
  
 private void createAccount(String id, String fullName, String email, String password, String birth)  
 {  
 try (Connection conn = DatabaseUtil.*getConnection*())  
 {  
 String query = "INSERT INTO TASK\_02 (ID, FULL\_NAME, EMAIL, PASSWORD, BIRTH, USER\_ROLE) VALUES (?, ?, ?, ?, ?, 'USER')";  
 PreparedStatement ps = conn.prepareStatement(query);  
 ps.setString(1, id);  
 ps.setString(2, fullName);  
 ps.setString(3, email);  
 ps.setString(4, password);  
 ps.setString(5, birth);  
  
 ps.executeUpdate();  
  
 JOptionPane.*showMessageDialog*(this, "Account created successfully!");  
 }  
  
 catch (SQLException e)  
 {  
 JOptionPane.*showMessageDialog*(this, "Account creation error: " + e.getMessage());  
 }  
 }  
}

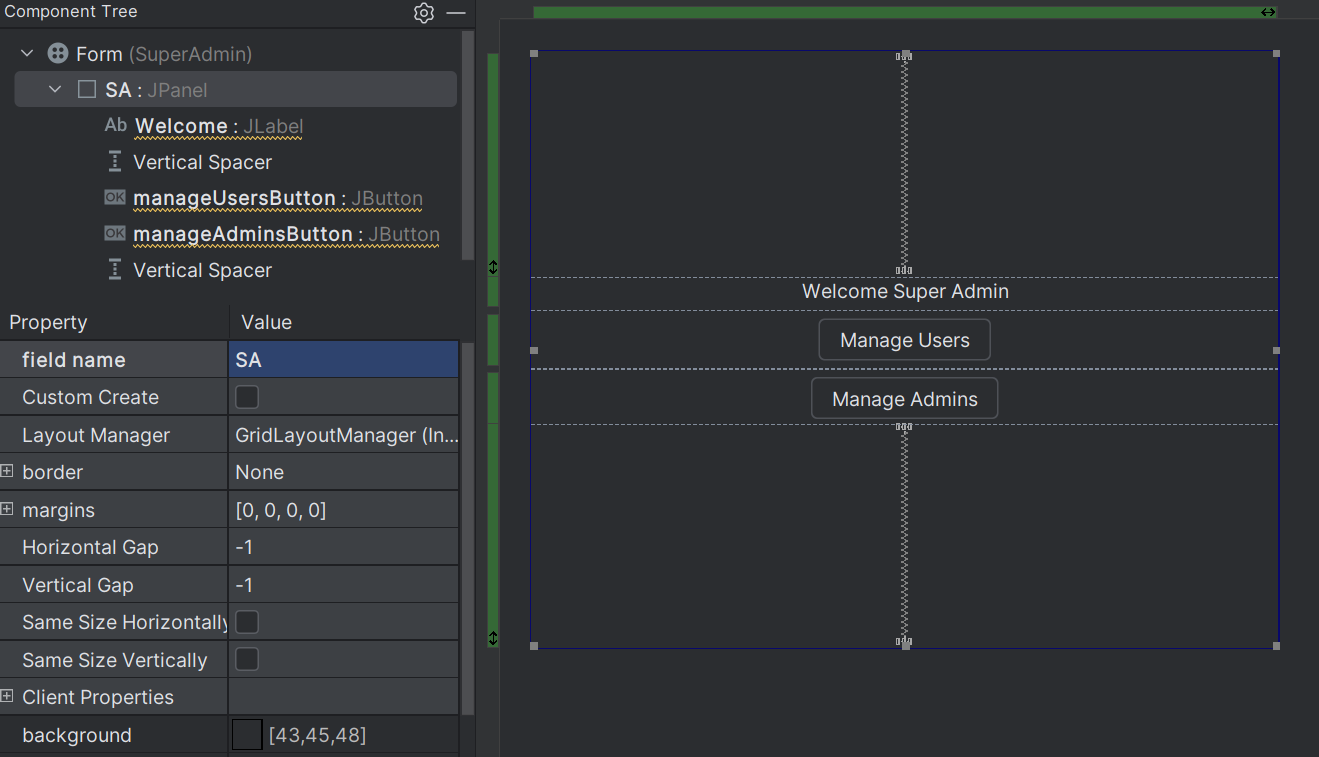
A screenshot of a computer

Description automatically generated

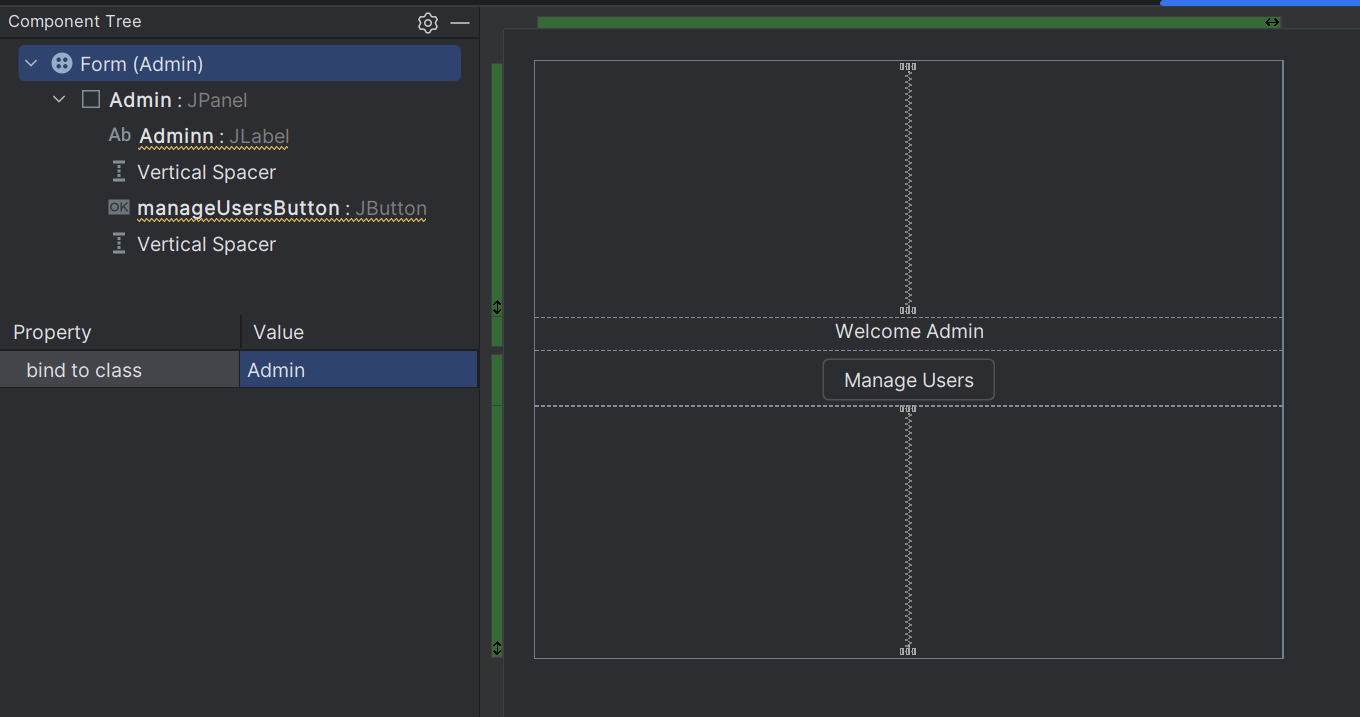
//Hafsa Salman  
//22K-5161  
//Task no. 02  
  
import javax.swing.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
  
public class Login\_page extends JFrame  
{  
 private JTextField IDTxt;  
 private JButton loginButton;  
 private JLabel Heading;  
 private JLabel UserID;  
 private JLabel Pass;  
 private JPasswordField passwordField1;  
 private JPanel Login;  
  
 public Login\_page()  
 {  
 setTitle("22K-5161 (Task no. 02)");  
 setSize(500, 500);  
 setContentPane(Login);  
 setVisible(true);  
  
 loginButton.addActionListener(new ActionListener()  
 {  
 @Override  
 public void actionPerformed(ActionEvent e)  
 {  
 String userID = IDTxt.getText();  
 String password = new String(passwordField1.getPassword());  
  
 if (userID.isEmpty() || password.isEmpty())  
 {  
 JOptionPane.*showMessageDialog*(Login\_page.this, "Please fill in both fields.");  
 }  
  
 else  
 {  
 if (authenticateUser(userID, password))  
 {  
 new SuperAdmin();  
 }  
  
 else  
 {  
 JOptionPane.*showMessageDialog*(Login\_page.this, "Invalid User ID or Password.");  
 }  
 }  
 }  
 });  
 }  
  
 private boolean authenticateUser(String userID, String password)  
 {  
 try (Connection conn = DatabaseUtil.*getConnection*())  
 {  
 String query = "SELECT ID, PASSWORD FROM TASK\_02 WHERE ID = ? AND PASSWORD = ?";  
 PreparedStatement ps = conn.prepareStatement(query);  
 ps.setString(1, userID);  
 ps.setString(2, password);  
  
 ResultSet rs = ps.executeQuery();  
 return rs.next();  
  
 }  
  
 catch (SQLException e)  
 {  
 JOptionPane.*showMessageDialog*(this, "Database error: " + e.getMessage());  
 }  
  
 return false;  
 }  
}

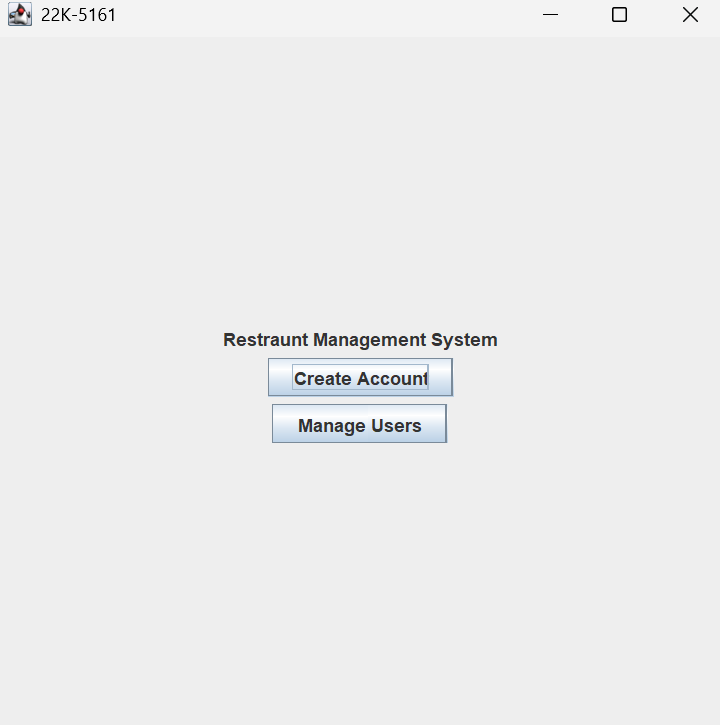


//Hafsa Salman  
//22K-5161  
//Task no. 02  
  
import javax.swing.\*;  
  
public class SuperAdmin extends JFrame  
{  
 private JLabel Welcome;  
 private JButton manageUsersButton;  
 private JButton manageAdminsButton;  
 private JPanel SA;  
  
 public SuperAdmin ()  
 {  
 setTitle("22K-5161 (Task no. 02)");  
 setSize(500, 500);  
 setContentPane(SA);  
 setVisible(true);  
 }  
}



//Hafsa Salman  
//22K-5161  
//Task no. 02  
  
import javax.swing.\*;  
  
public class Admin extends JFrame  
{  
 private JPanel Admin;  
 private JLabel Adminn;  
 private JButton manageUsersButton;  
  
 public Admin ()  
 {  
 setTitle("22K-5161 (Task no. 02)");  
 setSize(500, 500);  
 setContentPane(Admin);  
 setVisible(true);  
 }  
}



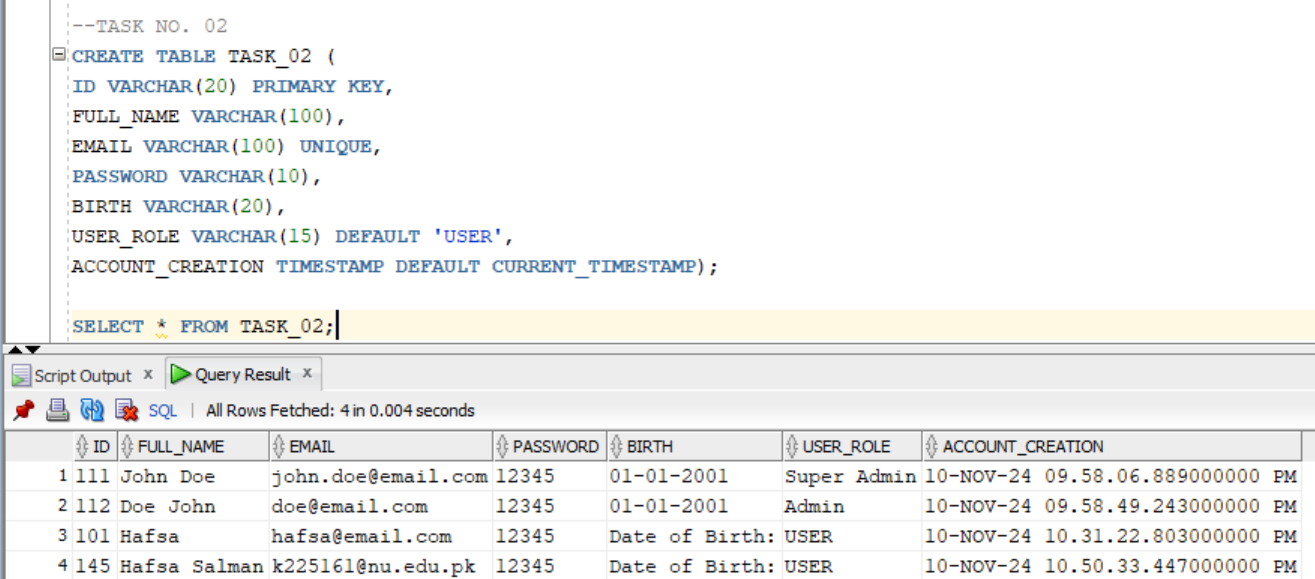
Output:  


A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



A screenshot of a login page

Description automatically generated

