Title
Create a login form with database verification.

Objective
To learn to verify a user login from the existing records in the database.

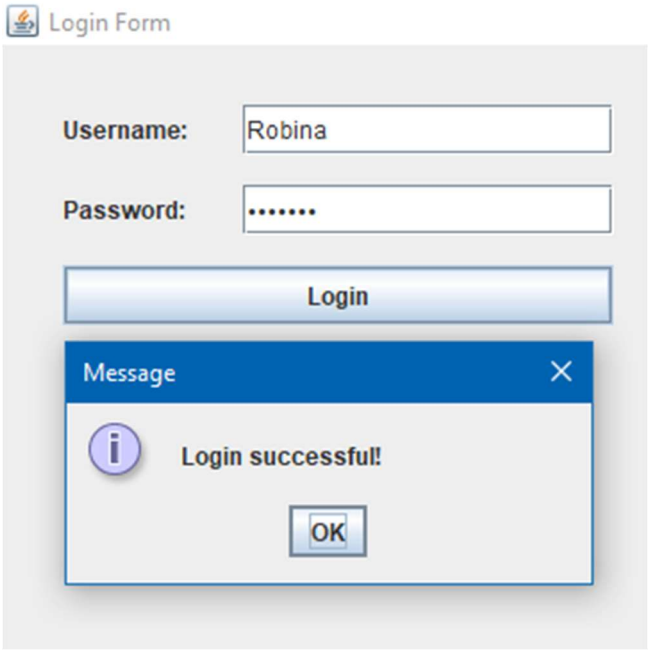
Program Code:

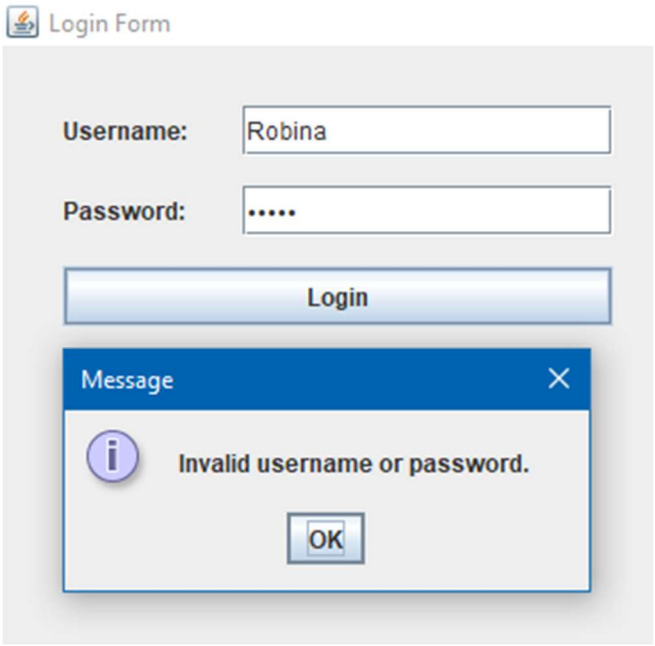
```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class DBLogin extends JFrame {

    private JTextField usernameField;
    private JPasswordField passwordField;
    private JButton loginButton;
    private JPanel mainPanel;

    public DBLogin() {
        setTitle("Login Form");
        setSize(350, 250);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        mainPanel = new JPanel();
        mainPanel.setLayout(null);

        JLabel userLabel = new JLabel("Username:");
        userLabel.setBounds(30, 30, 80, 25);
        mainPanel.add(userLabel);

        usernameField = new JTextField(20);
        usernameField.setBounds(120, 30, 185, 25);
        mainPanel.add(usernameField);

        JLabel passwordLabel = new JLabel("Password:");
```

```
        passwordLabel.setBounds(30, 70, 80, 25);
        mainPanel.add(passwordLabel);

        passwordField = new JPasswordField(20);
        passwordField.setBounds(120, 70, 185, 25);
        mainPanel.add(passwordField);

        loginButton = new JButton("Login");
        loginButton.setBounds(30, 110, 275, 30);
        mainPanel.add(loginButton);

        add(mainPanel);

        loginButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/userdetails",
"root", "");
                    String query = "SELECT * FROM userdata WHERE
username=? AND password=?";
                    PreparedStatement preparedStatement =
connection.prepareStatement(query);
                    preparedStatement.setString(1, usernameField.getText());
                    preparedStatement.setString(2, new
String(passwordField.getPassword()));
                    ResultSet resultSet = preparedStatement.executeQuery();

                    if (resultSet.next()) {
                        JOptionPane.showMessageDialog(null, "Login successful!");
                    } else {
                        JOptionPane.showMessageDialog(null, "Invalid username or
password.");
                    }

                    connection.close();
                } catch (Exception ex) {
                    System.out.println("Error");;
                }
```

```
            }
        });
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                new DBLogin().setVisible(true);
            }
        });
    }
}
```

OUTPUT:





Discussion

The practice for validating an user login from the existing records in the database which was accessed through the SQLConnector library imported in the program was understood and performed successfully.

Conclusion:

By completing this program, I learnt to validate an user login with the existing records in the database.

Title

Create a registration form containing text fields, passwords, radio button, checkbox, label, button with confirm password field which should be validated.

Objective

To learn to confirm the two user input passwords and register the user only if the two passwords match each other in the form along with other details.

Program Code:

```
import javax.swing.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;


public class RegistrationForm  extends JFrame {


   private JTextField usernameField;

   private JPasswordField passwordField;

   private JPasswordField confirmPasswordField;

   private JRadioButton maleRadioButton;

   private JRadioButton femaleRadioButton;

   private JCheckBox agreeCheckBox;

   private JButton registerButton;

   private JPanel mainPanel;


   public RegistrationForm() {

      setTitle("RegistrationForm");

      setSize(400, 350);
```

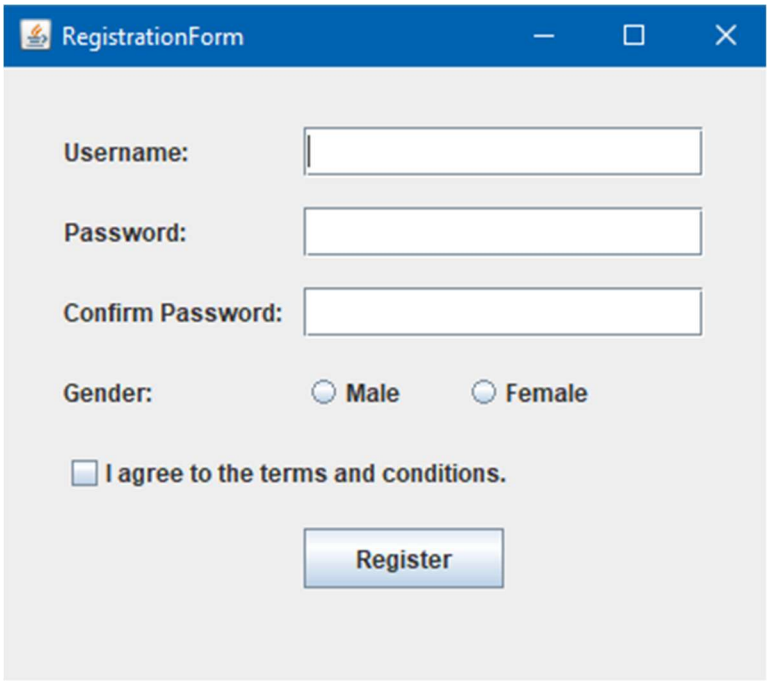
```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLocationRelativeTo(null);

mainPanel = new JPanel();
mainPanel.setLayout(null);

JLabel userLabel = new JLabel("Username:");
userLabel.setBounds(30, 30, 80, 25);
mainPanel.add(userLabel);

usernameField = new JTextField(20);
usernameField.setBounds(150, 30, 200, 25);
mainPanel.add(usernameField);

JLabel passwordLabel = new JLabel("Password:");
passwordLabel.setBounds(30, 70, 80, 25);
mainPanel.add(passwordLabel);

passwordField = new JPasswordField(20);
passwordField.setBounds(150, 70, 200, 25);
mainPanel.add(passwordField);

JLabel confirmPasswordLabel = new JLabel("Confirm Password:");
confirmPasswordLabel.setBounds(30, 110, 120, 25);
mainPanel.add(confirmPasswordLabel);
```

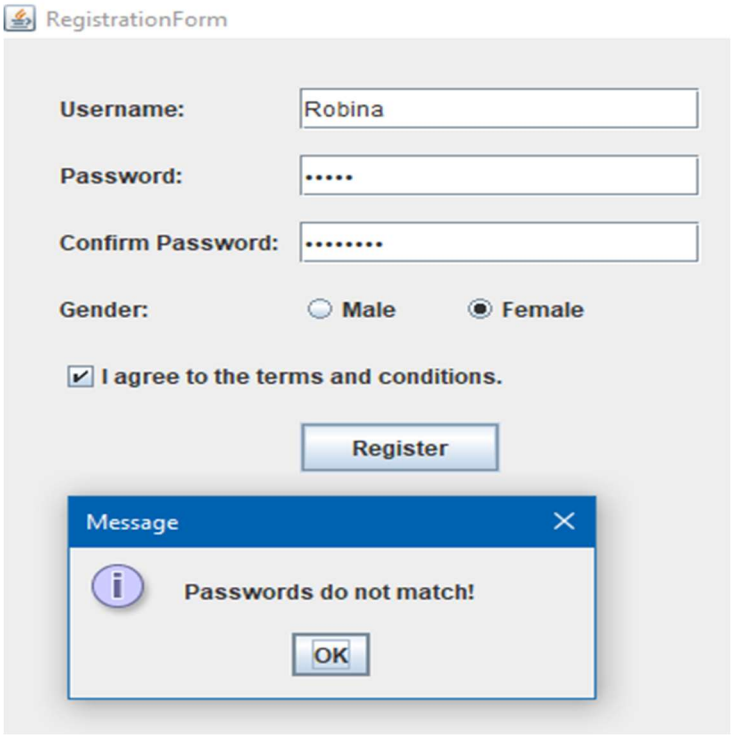
```
        confirmPasswordField = new JPasswordField(20);
        confirmPasswordField.setBounds(150, 110, 200, 25);
        mainPanel.add(confirmPasswordField);

        JLabel genderLabel = new JLabel("Gender:");
        genderLabel.setBounds(30, 150, 80, 25);
        mainPanel.add(genderLabel);

        maleRadioButton = new JRadioButton("Male");
        maleRadioButton.setBounds(150, 150, 80, 25);
        mainPanel.add(maleRadioButton);

        femaleRadioButton = new JRadioButton("Female");
        femaleRadioButton.setBounds(230, 150, 80, 25);
        mainPanel.add(femaleRadioButton);

        ButtonGroup genderGroup = new ButtonGroup();
        genderGroup.add(maleRadioButton);
        genderGroup.add(femaleRadioButton);

        agreeCheckBox = new JCheckBox("I agree to the terms and
conditions.");
        agreeCheckBox.setBounds(30, 190, 320, 25);
        mainPanel.add(agreeCheckBox);
```

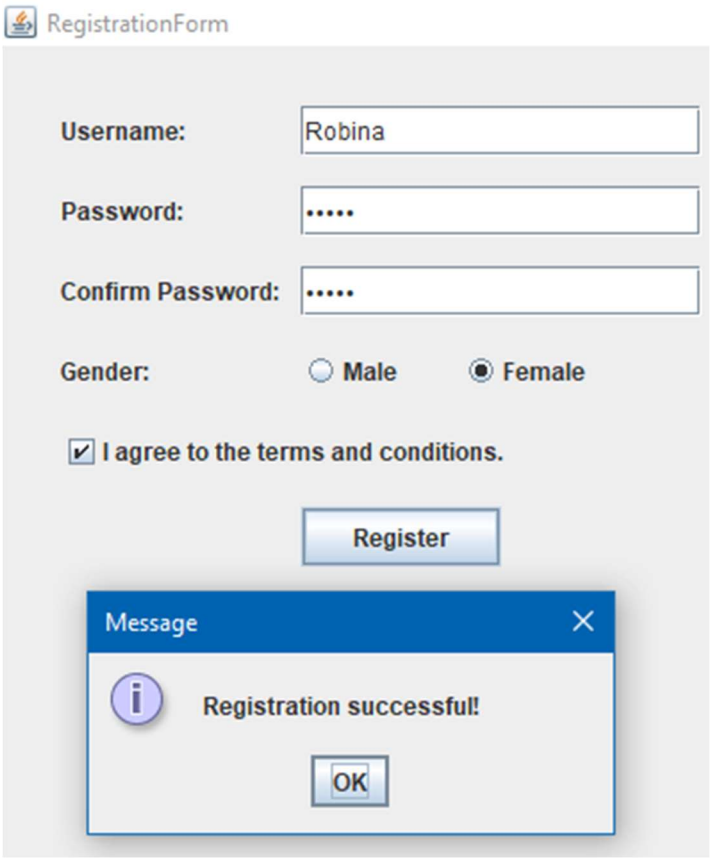
```
        registerButton = new JButton("Register");
        registerButton.setBounds(150, 230, 100, 30);
        mainPanel.add(registerButton);

        add(mainPanel);

        registerButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String password = new String(passwordField.getPassword());
                String confirmPassword = new
String(confirmPasswordField.getPassword());

                if (!password.equals(confirmPassword)) {
                    JOptionPane.showMessageDialog(null, "Passwords do not
match!");
                } else if (!agreeCheckBox.isSelected()) {
                    JOptionPane.showMessageDialog(null, "You must agree to the
terms and conditions.");
                } else {
                    JOptionPane.showMessageDialog(null, "Registration
successful!");
                }
            }
        });
    }
```

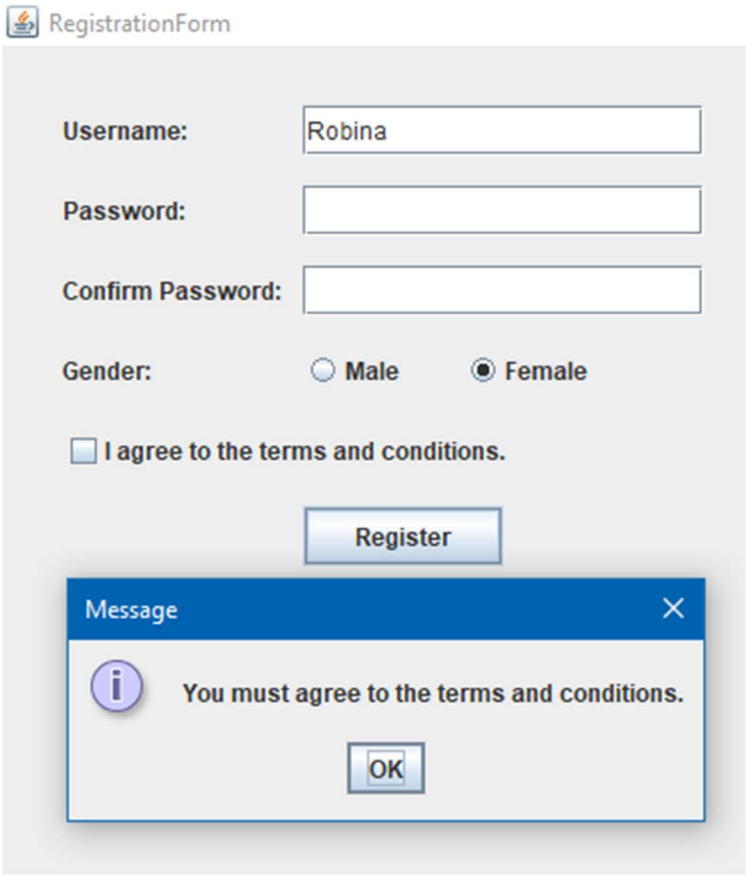
```
        public static void main(String[] args) {
            SwingUtilities.invokeLater(new Runnable() {
                @Override
                public void run() {
                    new RegistrationForm().setVisible(true);
                }
            });
        }
    }
```

OUTPUT:




RegistrationForm
Username:
Robina
Password:
Confirm Password:
Gender:
Male
Female
I agree to the terms and conditions.
Register
Message
Passwords do not match!
OK


RegistrationForm
Username:
Robina
Password:
Confirm Password:
Gender:
Male
Female
I agree to the terms and conditions.
Register
Message
Registration successful!
OK



Discussion

The practice for registering an user only and only if the two passwords input in the password field and confirm password field match each other and if they don't match return the error message was understood and performed successfully.

Conclusion:

By completing this program, I learnt to validate the user password inputs and confirm passwords input comparing them to each other and only registering in the case of successful password inputs.

Title
Create a GUI that displays record from database.

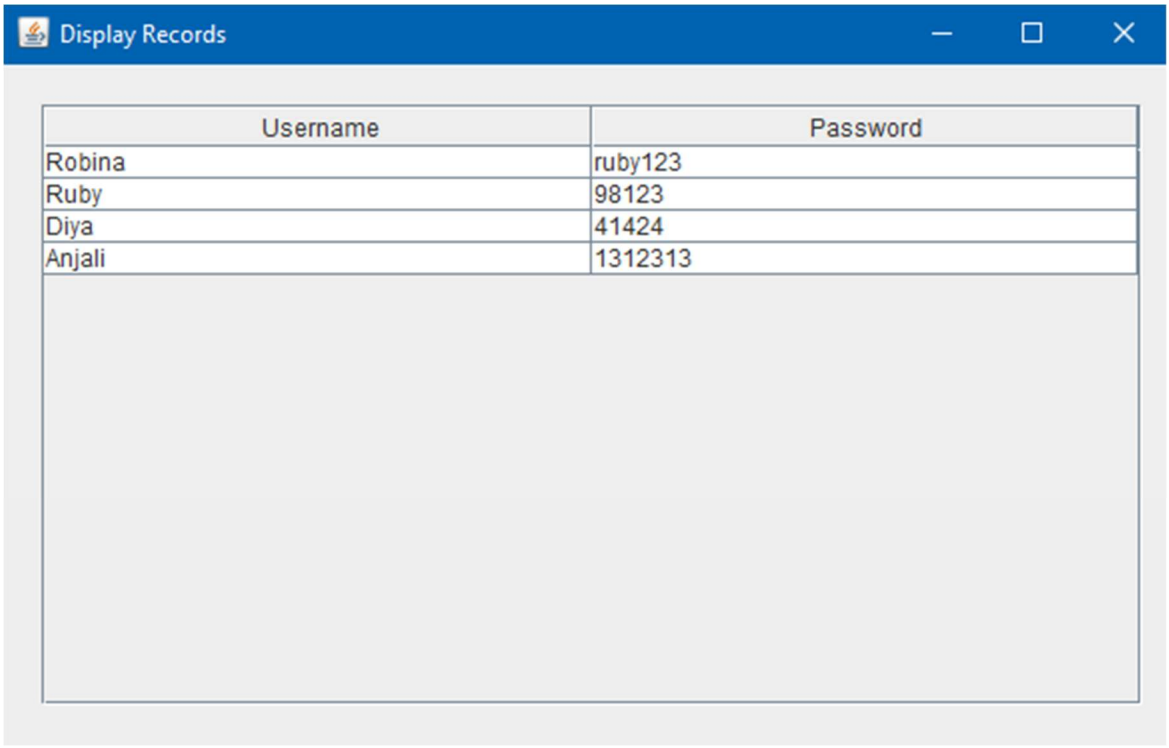
Objective
To learn to access the database and view the database existing records.

Program Code:

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class DisplayRecordProgram extends JFrame {

    public  JTable table;
    public DefaultTableModel tableModel;

    public DisplayRecordProgram() {
        setTitle("Display Records");
        setSize(600, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        String[] columnNames = {"Username", "Password"};
        tableModel = new DefaultTableModel(columnNames, 0);
        table = new JTable(tableModel);

        JScrollPane scrollPane = new JScrollPane(table);
        scrollPane.setBounds(20, 20, 550, 300);

        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(null);
        mainPanel.add(scrollPane);

        add(mainPanel);

        loadDataFromDatabase();
```

```
    }

    private void loadDataFromDatabase() {
        String url =
"jdbc:mysql://localhost:3306/userdetails?zeroDateTimeBehavior=convertTo
Null";
        String user = "root";
        String password = "";

        try {
            Connection connection = DriverManager.getConnection(url, user,
password);
            Statement statement = connection.createStatement();
            String query = "SELECT * FROM userdata";
            ResultSet resultSet = statement.executeQuery(query);

            while (resultSet.next()) {
                String username = resultSet.getString("username");
                String passwords = resultSet.getString("password");
                tableModel.addRow(new Object[]{username, passwords});
            }

            connection.close();
        } catch (Exception ex) {
            System.out.println(ex.getMessage());;
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                new DisplayRecordProgram().setVisible(true);
            }
        });
    }
}
```

OUTPUT:



Discussion

The practice for accessing the database with the help of SQL connector imported in the library and making the view for the existing database records in a tabular form was understood and performed successfully.

Conclusion:

By completing this program, I learnt to retrieve the data existing in the database with the help of required libraries and present it in a tabular form.

Title
Create a GUI that will receive password of an existing user and change it in database when button is clicked.

Objective
To learn to retrieve the password of an existing user in the database and change the existing password to a new password in the database through the input textfield.

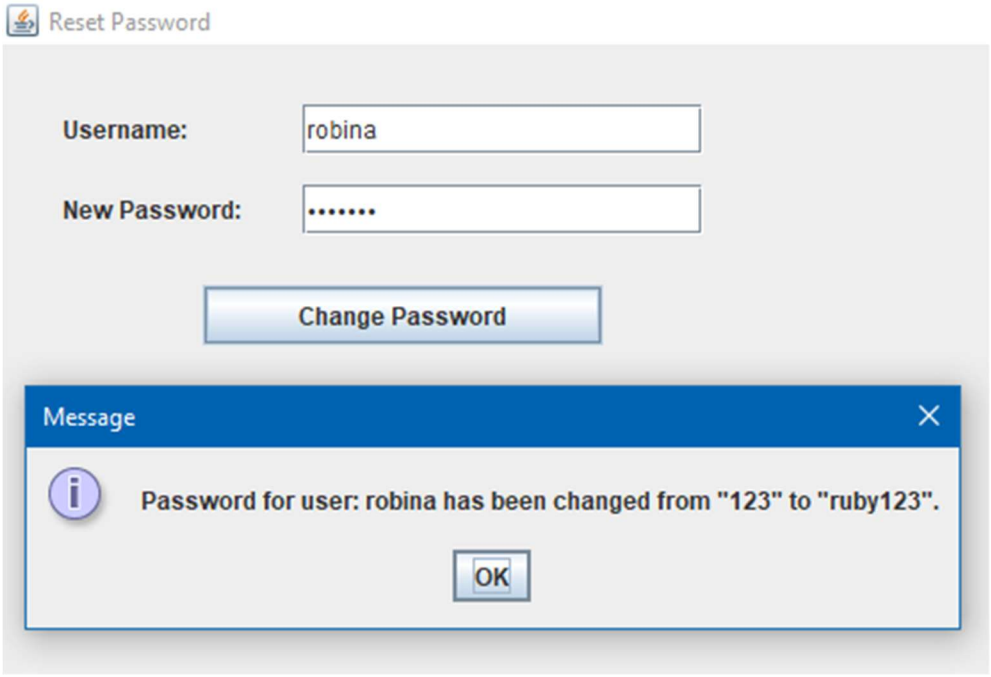
Program Code:

```
import javax.swing.*;
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;
public class ChangePasswordProgram  extends JFrame {

    private JTextField usernameField;
    private JPasswordField newPasswordField;
    private JButton changePasswordButton;
    private JPanel mainPanel;

    public ChangePasswordProgram() {
        setTitle("Reset Password");
        setSize(400, 250);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        mainPanel = new JPanel();
        mainPanel.setLayout(null);

        JLabel usernameLabel = new JLabel("Username:");
        usernameLabel.setBounds(30, 30, 100, 25);
        mainPanel.add(usernameLabel);

        usernameField = new JTextField(20);
        usernameField.setBounds(150, 30, 200, 25);
```

```
        mainPanel.add(usernameField);

        JLabel newPasswordLabel = new JLabel("New Password:");
        newPasswordLabel.setBounds(30, 70, 100, 25);
        mainPanel.add(newPasswordLabel);

        newPasswordField = new JPasswordField(20);
        newPasswordField.setBounds(150, 70, 200, 25);
        mainPanel.add(newPasswordField);

        changePasswordButton = new JButton("Change Password");
        changePasswordButton.setBounds(100, 120, 200, 30);
        mainPanel.add(changePasswordButton);

        add(mainPanel);

        changePasswordButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String username = usernameField.getText();
                String newPassword = new
String(newPasswordField.getPassword());

                String url =
"jdbc:mysql://localhost:3306/userdetails?zeroDateTimeBehavior=convertTo
Null";
                String user = "root";
                String password = "";

                try {
                    Connection connection = DriverManager.getConnection(url,
user, password);

                    String fetchQuery = "SELECT password FROM userdata
WHERE username=?";
                    PreparedStatement fetchStmt =
connection.prepareStatement(fetchQuery);
                    fetchStmt.setString(1, username);
                    ResultSet resultSet = fetchStmt.executeQuery();
```

```
                if (resultSet.next()) {
                    String oldPassword = resultSet.getString("password");

                    String updateQuery = "UPDATE userdata SET password=? WHERE username=?";
                    PreparedStatement updateStmt = connection.prepareStatement(updateQuery);
                    updateStmt.setString(1, newPassword);
                    updateStmt.setString(2, username);
                    int rowsUpdated = updateStmt.executeUpdate();

                    if (rowsUpdated > 0) {
                        JOptionPane.showMessageDialog(null, "Password for user: " + username + " has been changed from \"" + oldPassword + "\" to \"" + newPassword + "\".");
                    } else {
                        JOptionPane.showMessageDialog(null, "Failed to update password. User not found.");
                    }
                } else {
                    JOptionPane.showMessageDialog(null, "User not found.");
                }

                connection.close();
            } catch (Exception ex) {
                ex.printStackTrace();
            }
        }
    });
  }

  public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            new ChangePasswordProgram().setVisible(true);
        }
    });
  }
}
```

OUTPUT:



Discussion

The practice for accessing the database with the help of SQLConnector imported in the library retrieving the password of existing user and changing it to a new password was understood and performed successfully.

Conclusion:

By completing this program, I learnt to retrieve the password of an existing user in the database and make desired changes to it.

Title
Create a GUI to ask username and delete that user from the database.

Objective
To delete an existing user in the database with the help of username.

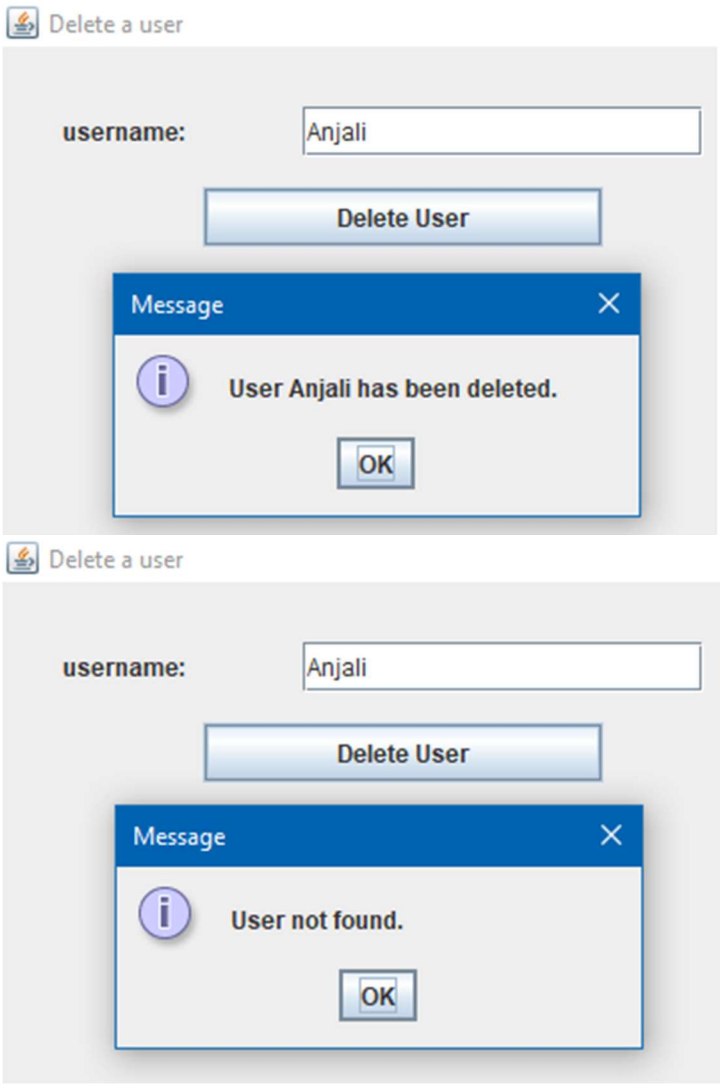
Program Code:

```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
public class DeleteUser extends JFrame {
    private JTextField usernameField;
    private JButton deleteButton;
    private JPanel mainPanel;
    public DeleteUser() {
        setTitle("Delete a user");
        setSize(400, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        mainPanel = new JPanel();
        mainPanel.setLayout(null);
        JLabel usernameLabel = new JLabel("username:");
        usernameLabel.setBounds(30, 30, 100, 25);
        mainPanel.add(usernameLabel);
        usernameField = new JTextField(20);
        usernameField.setBounds(150, 30, 200, 25);
        mainPanel.add(usernameField);
```

```
        deleteButton = new JButton("Delete User");
        deleteButton.setBounds(100, 70, 200, 30);
        mainPanel.add(deleteButton);
        add(mainPanel);

        deleteButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String username = usernameField.getText();

                String url =
"jdbc:mysql://localhost:3306/userdetails?zeroDateTimeBehavior=convertTo
Null";
                String user = "root";
                String password = "";

                try {
                    Connection connection = DriverManager.getConnection(url,
user, password);
                    String deleteQuery = "DELETE FROM userdata WHERE
username=?";
                    PreparedStatement deleteStmt =
connection.prepareStatement(deleteQuery);
                    deleteStmt.setString(1, username);
                    int rowsDeleted = deleteStmt.executeUpdate();

                    if (rowsDeleted > 0) {
```

```
                    JOptionPane.showMessageDialog(null, "User " + username +
" has been deleted.");
                } else {
                    JOptionPane.showMessageDialog(null, "User not found.");
                }

                connection.close();
            } catch (Exception ex) {
                ex.printStackTrace();
                JOptionPane.showMessageDialog(null, "Error occurred while
deleting user.");
            }
        }
    });
  }

  public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
      @Override
      public void run() {
        new DeleteUser().setVisible(true);
      }
    });
  }
}
```

OUTPUT:



Discussion

The practice for accessing the database with the help of SQLConnector imported in the library and delete the existing user from the database with the help of username to that corresponding user was understood and performed successfully.

Conclusion:

By completing this program, I learnt to delete an existing user in the database with the help of username corresponding to that user.