Demonstration of MVC

Name: Sammith S B	SRN: PES1UG19CS427	

Problem Statement:

The problem statement is to build a login page that accepts username and password. Convert normal java code into MVC architecture.

Model Component:
//paste the implementation of model class

```java
public class LoginModel {
   private String userName;
   private String password;

   public LoginModel() {

   }

   public LoginModel(String username, String password) {
       this.userName = username;
       this.password = password;
   }

   public String getPassword() {
       return password;
   }

   public void setPassword(String password) {
       this.password = password;
   }

   public String getUserName() {
       return userName;
   }

   public void setUserName(String userName) {
       this.userName = userName;
   }
}
```

View Component:
//paste the implementation of view

```
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
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;

public class LoginView extends JFrame implements ActionListener {
   private JTextField txtUsername;
   private JPasswordField txtPassword;
   private JButton btnLogin;
   private LoginModel model;

   public LoginView() {
       super("PES1UG19CS427 Login MVC");

       txtUsername = new JTextField(15);
       txtPassword = new JPasswordField(15);
       txtPassword.setEchoChar('*');
       btnLogin = new JButton("PES1UG19CS427 Login");

       JPanel content = new JPanel();
       content.setLayout(new FlowLayout());
       content.add(new JLabel("Username:"));
       content.add(txtUsername);
       content.add(new JLabel("Password:"));
       content.add(txtPassword);
       content.add(btnLogin);

       btnLogin.addActionListener(this);

       this.setContentPane(content);
       this.pack();

       this.addWindowListener(new WindowAdapter() {
           public void windowClosing(WindowEvent e) {
               System.exit(0);
           }
       });
   }

   public void actionPerformed(ActionEvent e) {
   }

   public LoginModel getUser() {
       model = new LoginModel(txtUsername.getText(), txtPassword.getText());
       return model;
   }

   public void showMessage(String msg) {
       JOptionPane.showMessageDialog(this, msg);
   }

   public void addLoginListener(ActionListener log) {
       btnLogin.addActionListener(log);
   }
}
```

Controller Component:

//paste the implementation of controller

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.SQLException;

public class LoginController {
    private LoginModel model;
    private LoginView view;

    public LoginController(LoginView view) {
        this.view = view;

        view.addLoginListener(new LoginListener());
    }

    class LoginListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            try {
                model = view.getUser();
                System.out.println("Bye");
                if (checkUser(model)) {
                    view.showMessage("Login succesfully!");
                } else {
                    view.showMessage("Invalid username and/or password!");
                }
            } catch (Exception ex) {
                view.showMessage(ex.getStackTrace().toString());
            }
        }
    }
```

```
    public boolean checkUser(LoginModel user) throws Exception {
```

```
        String query = "Select * FROM users WHERE username ='" + user.getUserName()
                + "' AND password ='" + user.getPassword() + "'";
        System.out.println(query);
        try {
            Class.forName("org.postgresql.Driver");
        } catch (ClassNotFoundException e) {
            System.err.println(e);
            System.exit(-1);
        }
        try {
            Connection connection = DriverManager.getConnection(
                    "jdbc:postgresql://127.0.0.1:5433/postgres", "postgres",
"S01m13i20h");
```

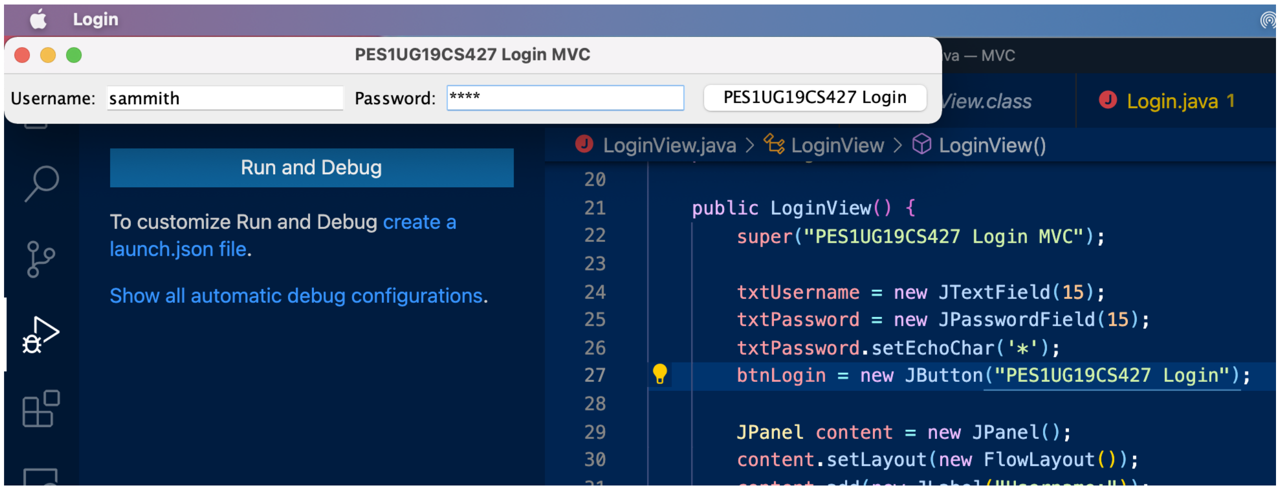
```
            // build query, here we get info about all databases"
```

```
            // execute query
            Statement statement = connection.createStatement();
            ResultSet rs = statement.executeQuery(query);
```

```
            // return query result
            while (rs.next()) {
                // display table name
                return true;
            }
            rs.close();
            statement.close();
            connection.close();
            return false;
        } catch (Exception e) {
            throw e;
            }

        }
}
```

Output Screen Shots:
//Run the application and take screen shots for demonstration and paste here.
//Ensure you include your SRN and name as an identity (in GUI or Model class)



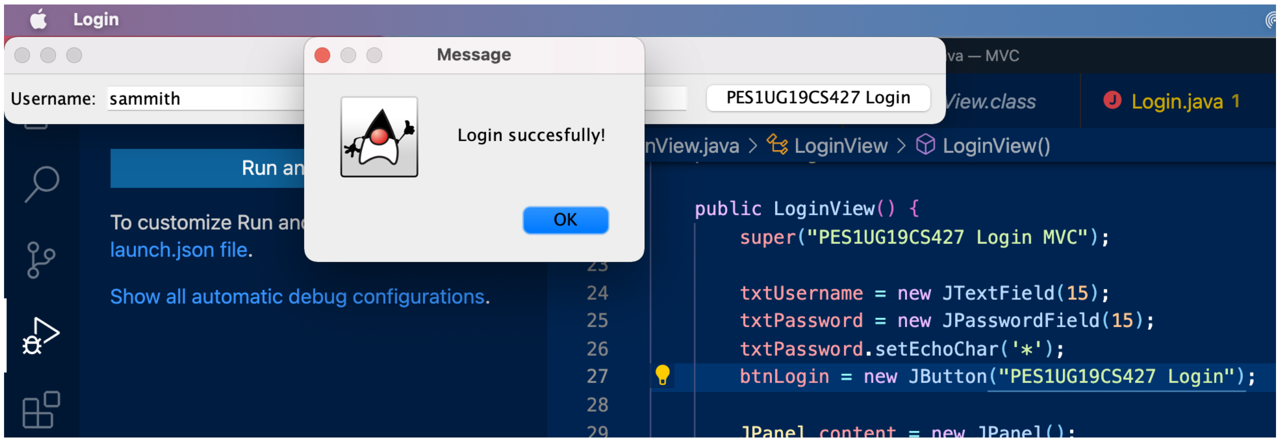
PES1UG19CS427 Login MVC

Username: sammith Password: **** PES1UG19CS427 Login

```java
    public LoginView() {
        super("PES1UG19CS427 Login MVC");

        txtUsername = new JTextField(15);
        txtPassword = new JPasswordField(15);
        txtPassword.setEchoChar('*');
        btnLogin = new JButton("PES1UG19CS427 Login");

        JPanel content = new JPanel();
        content.setLayout(new FlowLayout());
```



Message

Login succesfully!

OK

Database:
//In a line or two describe about the database wrt your application scenario.

I have used postgreSQL as the database for the assignment, there is postgres sql drive available to connect java apps with psql, have used that to store the username and password data.

Technologies /Tools used:
//Specify what technology/tools you have used for Model, view and controller.

PostgresSQL
Swing
java