



Demonstration of MVC

Name:PRAVEEN OBLI VENKATESH	SRN:PES1UG19CS350	

Problem Statement:

Adding the details of an employee

Database Class:

```
import java.sql.*;
```

```
public class conn
```

```
{
```

```
    public Connection c = null;
```

```
    public Statement stmt = null;
```

```
    public conn()
```

```
    {
```

```
        try
```

```
        {
```

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

```
            c =
```

```
DriverManager.getConnection("jdbc:postgresql://localhost:5432/prsystem", "postgres", "");
```

```
            stmt = c.createStatement();
```

```
        }
```

```
        catch (Exception e)
```

```
        {
```

```
            e.printStackTrace();
```

```
            System.err.println(e.getClass().getName()+": "+e.getMessage());
```

```
            System.exit(0);
```

```
        }
```

```
    }
```

```
}
```

Demonstration of MVC

Model Component:

public class checker

{

 public int spaceCheck(String str)

 {

 int ctr = 0;

 char[] ch = str.toCharArray();

 for(char c: ch)

 if(Character.isWhitespace(c))

 ctr++;

 return ctr;

 }

 public boolean singleSpaceCheck(String str)

 {

 int ctr = 0;

 char[] ch = str.toCharArray();

 for(char c: ch)

 if(Character.isWhitespace(c))

 return true;

 return false;

 }

 public boolean isNotNum(String str)

 {

 int ctr = 0;

 char[] ch = str.toCharArray();

 for(char c: ch)

 if(Character.isDigit(c))

 continue;

 else

 return true;

 return false;

 }

}

Demonstration of MVC

View and Controller Component:

```
import javax.swing.*;
import java.sql.*;
import java.awt.event.*;
import java.awt.*;
import java.util.*;

public class addEmployee implements ActionListener
{
    JFrame frame;
    JPanel panel;
    JLabel idL;
    JTextField idT;
    JLabel idE;
    JLabel nameL;
    JTextField nameT;
    JLabel nameE;
    JLabel passL;
    JPasswordField passT;
    JLabel passE;
    JLabel confL;
    JPasswordField confT;
    JLabel confE;
    JLabel contactL;
    JTextField contactT;
    JLabel contactE;
    JLabel genderL;
    Choice genderT;
    JLabel stateL;
    Choice stateT;
    JLabel cityL;
    JTextField cityT;
    JLabel cityE;
    JButton addE;

    public addEmployee()
    {
        panel = new JPanel();
```

Demonstration of MVC

```
frame = new JFrame();
frame.setSize(400, 450);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

frame.add(panel);

panel.setLayout(null);

idL = new JLabel("Employee ID :");
idL.setBounds(20, 20, 150, 25);
panel.add(idL);

idT = new JTextField();
idT.setBounds(150, 20, 200, 25);
panel.add(idT);

idE = new JLabel("");
idE.setBounds(180, 40, 200, 25);
idE.setForeground(Color.RED);
panel.add(idE);

nameL = new JLabel("Employee Name :");
nameL.setBounds(20, 60, 150, 25);
panel.add(nameL);

nameT = new JTextField();
nameT.setBounds(150, 60, 200, 25);
panel.add(nameT);

nameE = new JLabel("");
nameE.setBounds(180, 80, 150, 25);
nameE.setForeground(Color.RED);
panel.add(nameE);

passL = new JLabel("Password :");
passL.setBounds(20, 100, 150, 25);
panel.add(passL);
```

Demonstration of MVC

```
passT = new JPasswordField();  
passT.setBounds(150, 100, 200, 25);  
panel.add(passT);
```

```
passE = new JLabel("");  
passE.setBounds(180, 120, 150, 25);  
passE.setForeground(Color.RED);  
panel.add(passE);
```

```
confL = new JLabel("Confirm Password :");  
confL.setBounds(20, 140, 150, 25);  
panel.add(confL);
```

```
confT = new JPasswordField();  
confT.setBounds(150, 140, 200, 25);  
panel.add(confT);
```

```
confE = new JLabel("");  
confE.setBounds(180, 160, 150, 25);  
confE.setForeground(Color.RED);  
panel.add(confE);
```

```
contactL = new JLabel("Contact :");  
contactL.setBounds(20, 180, 150, 25);  
panel.add(contactL);
```

```
contactT = new JTextField();  
contactT.setBounds(150, 180, 200, 25);  
panel.add(contactT);
```

```
contactE = new JLabel("");  
contactE.setBounds(180, 200, 150, 25);  
contactE.setForeground(Color.RED);  
panel.add(contactE);
```

```
genderL = new JLabel("Gender :");  
genderL.setBounds(20, 225, 120, 25);  
panel.add(genderL);
```

Demonstration of MVC

```
genderT = new Choice();
genderT.setBounds(150, 225, 100, 25);
genderT.add("Male");
genderT.add("Female");
panel.add(genderT);
```

```
stateL = new JLabel("State :");
stateL.setBounds(20, 265, 120, 25);
panel.add(stateL);
```

```
stateT = new Choice();
stateT.setBounds(150, 265, 200, 25);
stateT.add("Andaman and Nicobar");
stateT.add("Andhra Pradesh");
stateT.add("Arunachal Pradesh");
stateT.add("Assam");
stateT.add("Bihar");
stateT.add("Chandigarh");
stateT.add("Chhattisgarh");
stateT.add("Dadra and Nagar Haveli");
stateT.add("Daman and Diu");
stateT.add("Delhi");
stateT.add("Goa");
stateT.add("Gujarat");
stateT.add("Haryana");
stateT.add("Himachal Pradesh");
stateT.add("Jammu and Kashmir");
stateT.add("Jharkhand");
stateT.add("Karnataka");
stateT.add("Kerala");
stateT.add("Lakshadweep");
stateT.add("Madhya Pradesh");
stateT.add("Maharashtra");
stateT.add("Manipur");
stateT.add("Meghalaya");
stateT.add("Mizoram");
stateT.add("Nagaland");
```

Demonstration of MVC

```
stateT.add("Orissa");
stateT.add("Puducherry");
stateT.add("Punjab");
stateT.add("Rajasthan");
stateT.add("Sikkim");
stateT.add("Tamil Nadu");
stateT.add("Telangana");
stateT.add("Tripura");
stateT.add("Uttar Pradesh");
stateT.add("Uttarakhand");
stateT.add("West Bengal");
panel.add(stateT);

cityL = new JLabel("City :");
cityL.setBounds(20, 305, 150, 25);
panel.add(cityL);

cityT = new JTextField();
cityT.setBounds(150, 305, 200, 25);
panel.add(cityT);

cityE = new JLabel("");
cityE.setBounds(180, 325, 150, 25);
cityE.setForeground(Color.RED);
panel.add(cityE);

addE = new JButton("Add Employee");
addE.setBounds(120, 345, 120, 25);
panel.add(addE);
addE.addActionListener(this);

frame.setVisible(true);
}

@Override
public void actionPerformed(ActionEvent event)
{
    int ctr = 0;
```

Demonstration of MVC

```
checker check = new checker();
String id = idT.getText();
String name = nameT.getText();
String pass = passT.getText();
String conf = confT.getText();
String contact = contactT.getText();
String gender = genderT.getSelectedItem();
String state = stateT.getSelectedItem();
String city = cityT.getText();
String q = "insert into employee values('"+id+"', '"+name+"', encrypt('"+pass+"',
'salty', 'aes'), '"+contact+"', '"+gender+"', '"+state+"', '"+city+"');";

if(check.singleSpaceCheck(id) || id.length() == 0 || check.isNotNum(id))
    idE.setText("Enter valid id");
else
{
    idE.setText("");
    ctr++;
}
if(check.spaceCheck(name) == name.length() ||
Character.isWhitespace(name.charAt(0)))
    nameE.setText("Enter valid name");
else
{
    nameE.setText("");
    ctr++;
}
if(check.singleSpaceCheck(pass) || pass.length() == 0)
    passE.setText("Enter valid password");
else
{
    passE.setText("");
    ctr++;
}
if(!pass.equals(conf))
    confE.setText("Passwords don't match");
else
{
```


Demonstration of MVC

```
        confE.setText("");
        ctr++;
    }
    if(check.singleSpaceCheck(contact) || contact.length() != 10 ||
check.isNotNum(contact))
        contactE.setText("Enter valid contact");
    else
    {
        contactE.setText("");
        ctr++;
    }
    if(check.spaceCheck(city) == city.length() ||
Character.isWhitespace(city.charAt(0)))
        cityE.setText("Enter valid City");
    else
    {
        cityE.setText("");
        ctr++;
    }

    if(ctr == 6)
    {
        boolean present = false;
        try
        {
            conn c1 = new conn();
            ResultSet rs = c1.stmt.executeQuery("select * from employee;");

            while(rs.next())
            {
                String eid = rs.getString("id");
                if(eid.equals(id))
                {
                    present = true;
                    idE.setText("Employee id must be unique");
                    break;
                }
            }
        }
    }
```

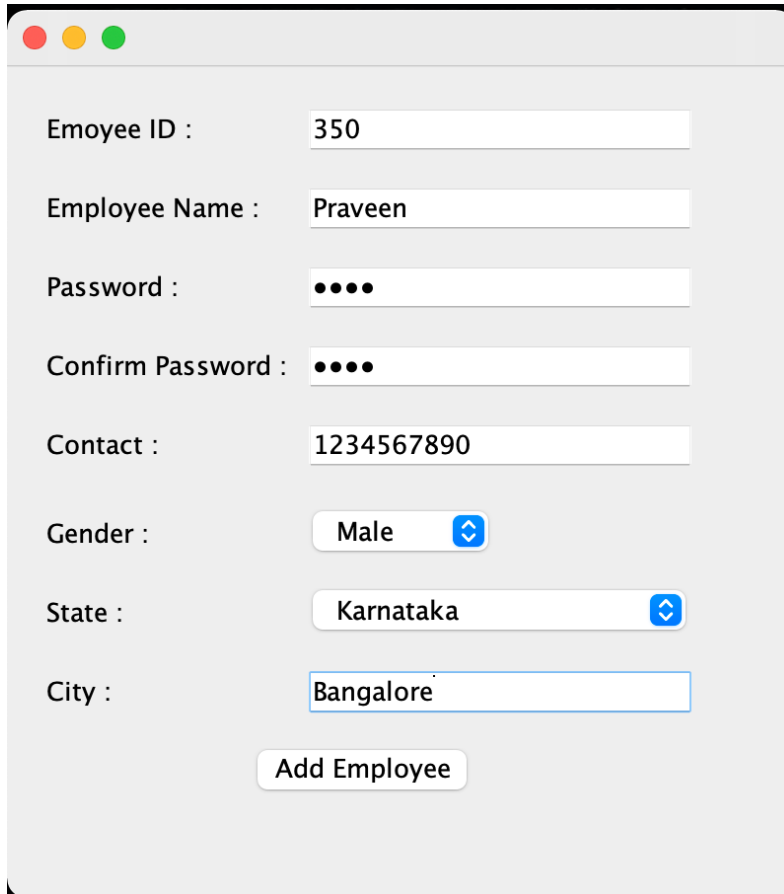
Demonstration of MVC

```
        if(!present)
        {
            c1.stmt.executeUpdate(q);
            idE.setText("");
            JOptionPane.showMessageDialog(null,"Employee
Added");
        }
    }
    catch(Exception e)
    {
        e.printStackTrace();
        System.err.println(e.getClass().getName()+": "+e.getMessage());
        System.exit(0);
    }
}

public static void main(String[] args)
{
    new addEmployee();
}
}
```

Demonstration of MVC

Output Screen Shots:



The screenshot shows a Java Swing window with a light gray background and a standard macOS-style title bar with red, yellow, and green buttons. The form contains the following fields and controls:

- Employee ID :** A text input field containing the value "350".
- Employee Name :** A text input field containing the value "Praveen".
- Password :** A password input field with four black dots.
- Confirm Password :** A password input field with four black dots.
- Contact :** A text input field containing the value "1234567890".
- Gender :** A dropdown menu with "Male" selected and a blue arrow icon.
- State :** A dropdown menu with "Karnataka" selected and a blue arrow icon.
- City :** A text input field containing the value "Bangalore".
- Add Employee** : A button located at the bottom center of the form.

Database:

I used postgres database and used jdbc to implement it. JDBC is an open source driver for java applications(postgres in this case).

Technologies /Tools used:

- swing for GUI
- Awt for event listener
- Postgres database