--

Dt : 17/6/2022

Exp program_3:

JSP Application to demostrate accessing bean object.

(Using <jsp:useBean> <jsp:setProperty> <jsp:getProperty>)

ProductBean.java(Bean class)

```
package test;
import java.io.*;
@SuppressWarnings("serial")
public class ProductBean implements Serializable{
   private String code,name;
   private float price;
   private int qty;
   public ProductBean() {}
public String getCode() {
     return code;
}
public void setCode(String code) {
     this.code = code;
}
public String getName() {
     return name;
}
public void setName(String name) {
     this.name = name;
}
public float getPrice() {
     return price;
}
public void setPrice(float price) {
     this.price = price;
}
public int getQty() {
     return qty;
}
public void setQty(int qty) {
     this.qty = qty;
}
}
```

input.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <form action="LoadJSP.jsp" method="post">
  Enter the PCode:<input type="text" name="pcode"><br>
  <input type="submit" value="Display">
  </form>
</body>
</html>
```

LoadJSP.jsp

```
<%@ page language="java"
    contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"
    import="java.sql.*,test.*"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%
try{
     ResultSet rs = new RetrieveDAO().retrieve(request);
     if(rs.next()){
          %>
          <jsp:useBean id="ob" class="test.ProductBean"
scope="session"/>
          <jsp:setProperty property="code" name="ob"
value="<%=rs.getString(1) %>"/>
          <jsp:setProperty property="name" name="ob"
value="<%=rs.getString(2) %>"/>
          <jsp:setProperty property="price" name="ob"
value="<%=rs.getFloat(3) %>"/>
          <jsp:setProperty property="qty" name="ob"
value="<%=rs.getInt(4) %>"/>
```

```
          <a href="ViewJSP.jsp">ViewProductDetails</a>
          <%
     }else{
          out.println("Invalid ProdCode...<br>");
          %>
          <jsp:include page="input.html"/>
          <%
     }
}catch(Exception e){e.printStackTrace();}
%>
</body>
</html>
```

ViewJSP.jsp

```
<%@ page language="java"
         contentType="text/html; charset=ISO-8859-1"
         pageEncoding="ISO-8859-1"
         import="test.*"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<jsp:useBean id="ob" type="test.ProductBean" scope="session"/>
ProductCode:<jsp:getProperty property="code" name="ob"/><br>
ProductName:<jsp:getProperty property="name" name="ob"/><br>
ProductPrice:<jsp:getProperty property="price" name="ob"/><br>
ProductQty:<jsp:getProperty property="qty" name="ob"/><br>

</body>
</html>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
  <welcome-file-list>
    <welcome-file>input.html</welcome-file>
  </welcome-file-list>
</web-app>
```

DBConnection.java

```
package test;
import java.sql.*;
public class DBConnection {
    private static Connection con=null;//reference variable
    private DBConnection() {}
    static
    {
     try {
          Class.forName("oracle.jdbc.driver.OracleDriver");
          con = DriverManager.getConnection

("jdbc:oracle:thin:@localhost:1521:xe","system","manager");
     }catch(Exception e) {e.printStackTrace();}
    }
    public static Connection getCon()
    {
     return con;
    }
}
```

RetrieveDAO.java

```
package test;

import java.sql.*;

import javax.servlet.http.*;

public class RetrieveDAO {

 public ResultSet rs=null;

 public ResultSet retrieve(HttpServletRequest req)

 {

      try {

           Connection con = DBConnection.getCon();

                 //Accessing the Connection

           PreparedStatement ps = con.prepareStatement

           ("select * from Product45 where pcode=?");

           ps.setString(1,req.getParameter("pcode"));
```

```
                    rs = ps.executeQuery();
          }catch(Exception e) {e.printStackTrace();}
          return rs;
  }
}
```

===

Dt : 18/6/2022

*imp

Web Architecture Models:(Web Application Architectures)

Two types of development models are used in Java,for Web applications and these models are classified based on the different approaches used to develop Web applications.

These models are:

1. Model-1 Architecture

2. Model-2 Architecture(MVC)

1. Model-1 Architecture:

The Model-1 architecture was the first development model used to develop Web applications and this model uses JSP to design applications and, which is responsible for all the activities and functionalities provided by the application.

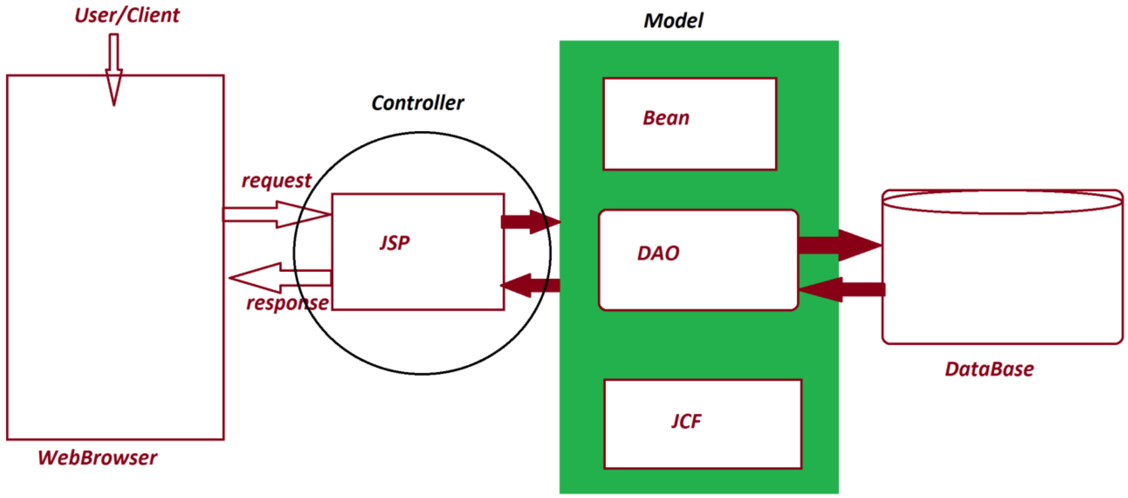
Limitations of the Model-1 Architecture:

(i).Applications are inflexible and difficult to maintain.

A single change in one page may cause changes in other pages, leading to unpredictable results.

(ii).Involves the developer at both the page development and the business logic implementation stages.

(iii).Increases the complexity of a program with the increase in the size of the JSP page.



==

2. Model-2 Architecture:

=>The draw backs in the Model-1 architecture led to the introduction of a new model called Model-2.

=>The Model-2 architecture was targeted at overcoming the drawbacks of Model-1 and helping developers to design more powerful Web applications and this Model-2 architecture is based on the MVC design model.

=>MVC Stands for Model View Controller.

Model: Represents enterprise data and business rules that specify how data is accessed and updated,and which is generally implemented by using JavaBeans.

View: Shows the contents of a Model.The View component accesses enterprise data through the Model component and specifies how that data should be presented and this View Component is designed by JSP.

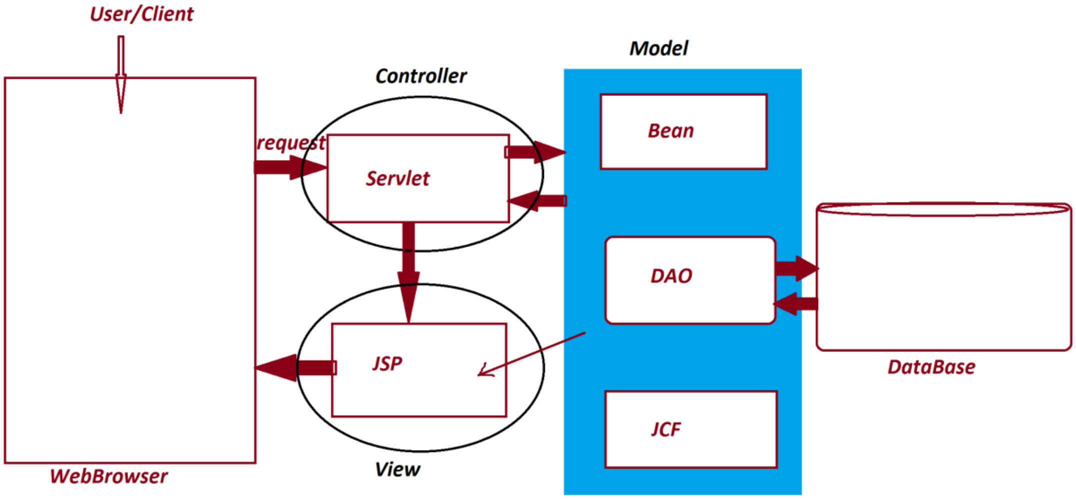
Controller: Receives HTTP requests. The Controller component receives requests from a client, determines the business logic to be performed, and delegates the responsibility for producing the next phase of the user interface to an appropriate view component. The Controller has complete control over each view, implying that any change in the Model component is immediately reflected in all the Views of an application.

The Controller component is implemented by servlets.

Advantages of Model-2 Architecture:

(i)Allows use of reusable software components to design the Business logic. Therefore, these components can be used in the business logic of other applications.

(ii)Offers great flexibility to the presentation logic, which can be modified without effecting the business logic.



==