

Malla Reddy Engineering College (Autonomous)

Maisammaguda, Dhulapally (Post via Kompally), Secunderabad – 500 100.

II B.TECH – II Semester (MR20) II Mid Assignment

Subject: Web Technologies

Branch: II- CSE C& D sections

Subject code: A1202

Name of the Faculty: Mr. S Ajay Kumar

Project: Develop MVC in JSP CRUD application for User Management Systems

The **Model-View-Controller (MVC)** is an architectural pattern that separates an application into three main logical components: the **model**, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.

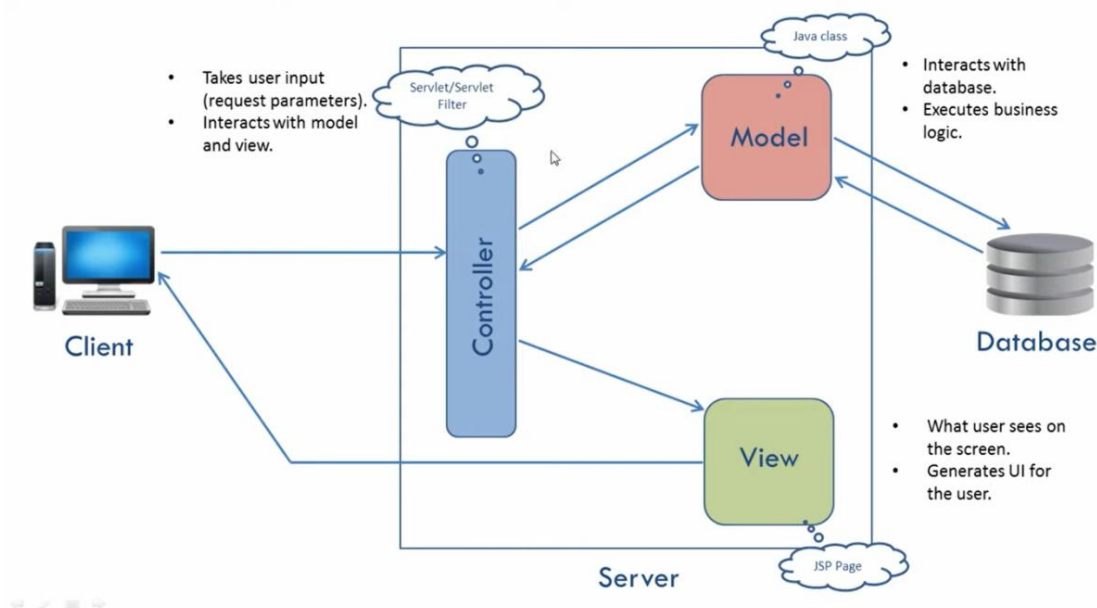


Figure: Model-View-Controller Architecture

Model: The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data.

View: The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

Controller: Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.

Tools and Technologies used: JDK, Eclipse IDE, Apache Tomcat, JSP, Servlet and MySQL.

Development Steps:

1. Create an Eclipse Dynamic web project
2. Add Dependencies
3. Create Project Structure
4. MySQL Database setup
5. Create Java Bean class (UserBean.java in model package)
6. Create a DAO class (UserDao.java in dao package)
7. Create web pages such as index.jsp, adduserform.jsp, userform.html, adduser.jsp, adduser-success.jsp, adduser-error.jsp, viewusers.jsp, editform.jsp, edituser.jsp and deleteuser.jsp.
8. Deploying and Testing the Application demo.

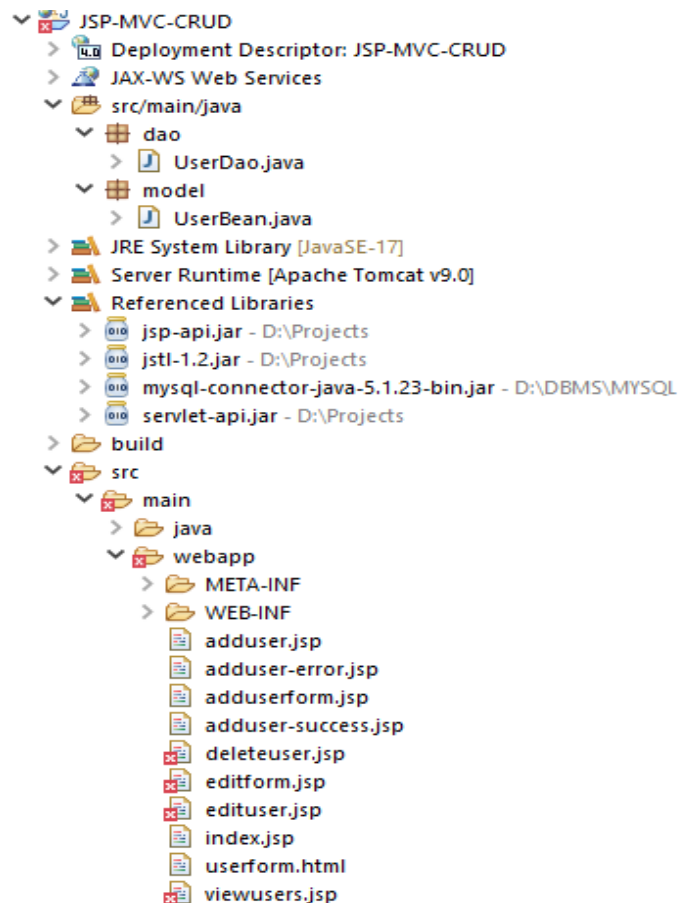
1. Create an Eclipse Dynamic web project:

- To create a new dynamic Web project in Eclipse:
- a) On the main menu select File > New > Project....
 - b) In the upcoming wizard choose Web > Dynamic Web Project.
 - c) Click Next.
 - d) Enter project name as “JSP-MVC-CRUD”
 - e) Make sure that the target runtime is set to Apache Tomcat with the currently supported version.

2. Add Dependencies:

- Add the latest release of below jar files to the lib folder.
- a) jsp-api.2.3.1.jar
 - b) jstl-1.2.jar
 - c) mysql-connector-java-8.0.13.jar
 - d) servlet-api.2.3.jar

3. Create Project Structure:



4. MySQL Database setup:

Mysql> Create table registration (userid int(4) primary key auto_increment, username varchar(20), password varchar(20), email varchar(20)); //Registration table is created.

Mysql> select *from registration; //no rows selected. Performs CRUD operations and check.

5. Create Java Bean class (UserBean.java class in model package)

package model;

```
public class UserBean {
    public int userid;
    public String username,password,email;
    public int getUserId() {
        return userid;
    }
    public void setUserid(int userid) {
        this.userid = userid;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
}
```

6. Create a DAO class (UserDao.java class in dao package)

package dao;

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;
```

import model.UserBean;

```
public class UserDao {

    public static Connection getConnection() {
```

```

Connection con = null;
try {
    Class.forName("com.mysql.jdbc.Driver");
    con = DriverManager.getConnection("jdbc:mysql://localhost:3306/mrec", "root", "mysql");
} catch (Exception e) {
    System.out.println(e);
}
return con;
}

```

```

public static int save(UserBean u) {
    int status = 0;
    try {
        Connection con = getConnection();
        PreparedStatement ps = con.prepareStatement
            ("insert into register(username,password,email) values(?,?,?)");
        ps.setString(1, u.getUsername());
        ps.setString(2, u.getPassword());
        ps.setString(3, u.getEmail());
        status = ps.executeUpdate();
    } catch (Exception e) {
        System.out.println(e);
    }
    return status;
}

```

```

public static int update(UserBean u) {
    int status = 0;
    try {
        Connection con = getConnection();
        PreparedStatement ps = con.prepareStatement
            ("update register set username=?,password=?,email=? where userid=?");
        ps.setString(1, u.getUsername());
        ps.setString(2, u.getPassword());
        ps.setString(3, u.getEmail());
        ps.setInt(4, u.getUserid());
        status = ps.executeUpdate();
    } catch (Exception e) {
        System.out.println(e);
    }
    return status;
}

```

```

public static int delete(UserBean u) {
    int status = 0;
    try {
        Connection con = getConnection();
        PreparedStatement ps = con.prepareStatement("delete from register where userid=?");
        ps.setInt(1, u.getUserid());
        status = ps.executeUpdate();
    }
}

```

```

    } catch (Exception e) {
        System.out.println(e);
    }
    return status;
}

```

```

public static List<UserBean> getAllRecords() {
    List<UserBean> list = new ArrayList<UserBean>();
    try {
        Connection con = getConnection();
        PreparedStatement ps = con.prepareStatement("select * from register");
        ResultSet rs = ps.executeQuery();
        while (rs.next()) {
            UserBean u = new UserBean();
            u.setUserid(rs.getInt("userid"));
            u.setUsername(rs.getString("username"));
            u.setPassword(rs.getString("password"));
            u.setEmail(rs.getString("email"));
            list.add(u);
        }
    } catch (Exception e) {
        System.out.println(e);
    }
    return list;
}

```

```

public static UserBean getRecordById(int userid) {
    UserBean u = null;
    try {
        Connection con = getConnection();
        PreparedStatement ps = con.prepareStatement("select * from register where userid=?");
        ps.setInt(1, userid);
        ResultSet rs = ps.executeQuery();
        while (rs.next()) {
            u = new UserBean();
            u.setUserid(rs.getInt("userid"));
            u.setUsername(rs.getString("username"));
            u.setPassword(rs.getString("password"));
            u.setEmail(rs.getString("email"));
        }
    } catch (Exception e) {
        System.out.println(e);
    }
    return u;
}

```

```

public static List<UserBean> getRecords(int start, int total) {
    List<UserBean> list = new ArrayList<UserBean>();
    try {
        Connection con = getConnection();

```

```

PreparedStatement ps = con.prepareStatement(
    "select * from register limit " + (start - 1) + "," + total);
ResultSet rs = ps.executeQuery();
while (rs.next()) {
    UserBean e = new UserBean();
    e.setUserid(rs.getInt(1));
    e.setUsername(rs.getString(2));
    list.add(e);
}
con.close();
} catch (Exception e) {
    System.out.println(e);
}
return list;
}
}

```

7. Create the following web pages in the folder JSP-MVC-CRUD\SRC\MAIN\WEB

- | | |
|-------------------------------|-----------------------------|
| a) index.jsp | f) adduser-error.jsp |
| b) adduserform.jsp | g) viewusers.jsp |
| c) userform.html | h) editform.jsp |
| d) adduser.jsp | i) edituser.jsp |
| e) adduser-success.jsp | j) deleteuser.jsp |

a) index.jsp

```

<html> <body>
<h1>JSP MVC CRUD Example</h1>
<a href="adduserform.jsp">Add User</a>
<a href="viewusers.jsp">View Users</a>
</body> </html>

```

b) adduserform.jsp

```

<html> <body>
<jsp:include page="userform.html"></jsp:include>
</body> </html>

```

c) userform.html

```

<html> <body>
<a href="viewusers.jsp">View All Records</a><br/>

<h1>Add New User</h1>
<form action="adduser.jsp" method="post">
<table>
<tr><td>Name:</td><td><input type="text" name="username"></td></tr>
<tr><td>Password:</td><td><input type="password" name="password"></td></tr>
<tr><td>Email:</td><td><input type="text" name="email"></td></tr>

<tr><td><input type="submit" value="Add User"></td></tr>
</table>
</form> </body> </html>

```

d) adduser.jsp

```
<% @page import="dao.UserDao"%>
<jsp:useBean id="u" class="model.UserBean"></jsp:useBean>
<jsp:setProperty property="*" name="u" />

<%
int i = UserDao.save(u);
if (i> 0) {      response.sendRedirect("adduser-success.jsp");
} else {        response.sendRedirect("adduser-error.jsp");
}
%>
```

e) adduser-success.jsp

```
<html> <body>

<p>Record successfully saved!</p>
<jsp:include page="userform.html"></jsp:include>
</body> </html>
```

f) adduser-error.jsp

```
<html> <body>

<p>Sorry, an error occurred!</p>
<jsp:include page="userform.html"></jsp:include>
</body> </html>
```

g) viewusers.jsp

```
<html> <body>

<% @page import="dao.UserDao" %>
<% @page import="model.UserBean" %>
<% @page import="java.util.*"%>
<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<h1>Users List</h1>
<%
List<UserBean> list2=UserDao.getAllRecords();
request.setAttribute("list",list2);
%>

<table>
<tr><th>Id</th><th>Name</th><th>Password</th><th>Email</th>
<th>Edit</th><th>Delete</th></tr>

<c:forEach items="${list}" var="u">
<tr><td>${u.getUserid()}</td><td>${u.getUsername()}</td>
<td>${u.getPassword()}</td><td>${u.getEmail()}</td>

<td><a href="editform.jsp?id=${u.getUserid()}">Edit</a></td>
<td><a href="deleteuser.jsp?id=${u.getUserid()}">Delete</a></td></tr>
```

```

</c:forEach>
</table>

<br/><a href="adduserform.jsp">Add New User</a>
</body> </html>

```

h) editform.jsp

```

<html> <body>
<% @page import="model.UserBean"%>
<% @page import="dao.UserDao"%>

<%
String userid=request.getParameter("userid");
UserBean u=UserDao.getRecordById(Integer.parseInt(userid));
%>

<h1>Edit Form</h1>
<form action="edituser.jsp" method="post">
<input type="hidden" name="id" value="<%=u.getUserid() %>">

<table>
<tr><td>Name:</td><td>
<input type="text" name="username" value="<%= u.getUsername() %>"></td></tr>
<tr><td>Password:</td><td>
<input type="password" name="password" value="<%= u.getPassword() %>"></td></tr>
<tr><td>Email:</td><td>
<input type="email" name="email" value="<%= u.getEmail() %>"></td></tr>
<tr><td colspan="2"><input type="submit" value="Edit User"></td></tr>
</table>
</form>
</body> </html>

```

i) edituser.jsp

```

<% @page import="dao.UserDao"%>
<jsp:useBean id="u" class="model.UserBean"/>
<jsp:setProperty property="*" name="u"/>
<%
int i=UserDao.update(u);
response.sendRedirect("viewusers.jsp");
%>

```

j) deleteuser.jsp.

```

% @page import="dao.UserDao"%>
<jsp:useBean id="u" class="model.UserBean"/>
<jsp:setProperty property="*" name="u"/>
<%
int i=UserDao.delete(u);
response.sendRedirect("viewusers.jsp");
%>

```


8. Deploying and Testing the Application demo.

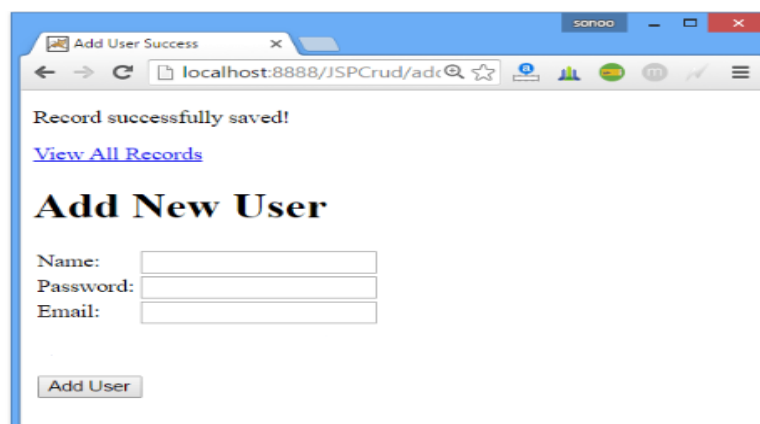
Step a) Once all the functionalities are done then run the application using index.jsp in Eclipse IDE.



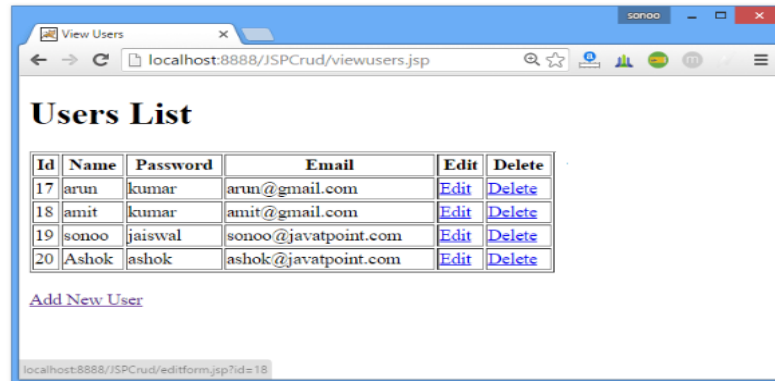
Step b) Click on Add user link and register the user details.



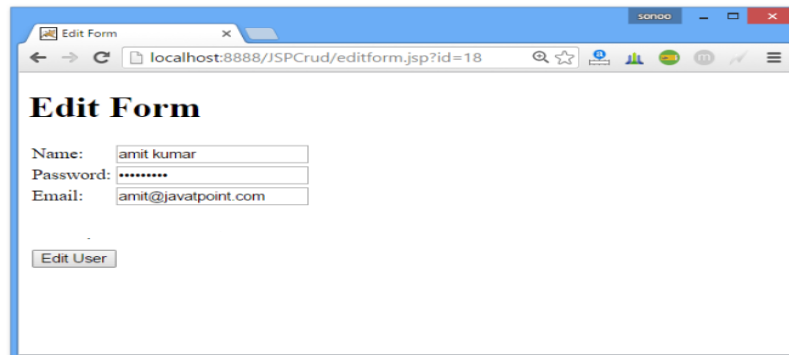
Step c) Once the registration is done then shown below message and we can add new user or view the existing users.



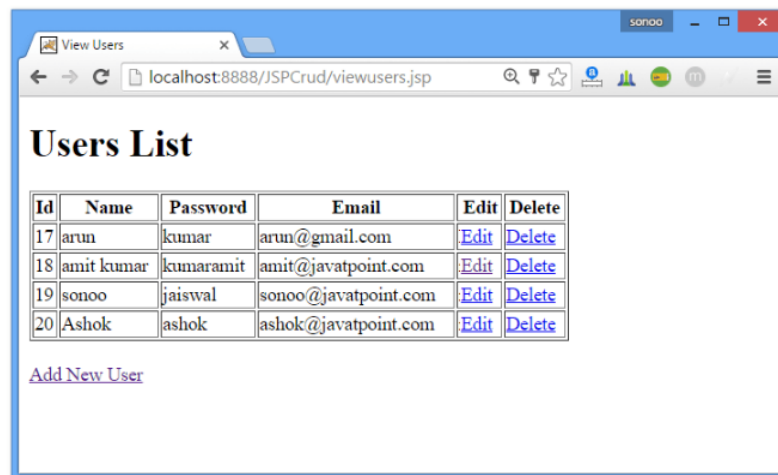
Step d) We can view all the user details once we click on “View All Records “



Step e) We can edit the existing user details once we click on particular user edit link.



Step f) We can see the particular user updated details once we click on View All Records link.



Step g) We can delete the particular user details once we click on delete link of that user.

