**MVC Using Servlet and JSP: (Model View and Controller)**

Note: When we create a jsp it will directly convert to servlet

Never write service logic inside JSP.

Overview:

When client send the request to controller so here controller is servlet and that controller will call the view here view is jsp pages and to display some data on jsp pages view will call the model where model is treated as Pojo class (Entity Class).

Note: we write always business logic inside the controller.

But if we want to perform any database operation then for that we should have service layer where we will write the database connection steps.

**JSTL:** (JSP Standard Tag Library)

For using jstl we have to download jstl jar file and paste in lib folder in our project

And add the tag in jsp pages

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

Now you can use any tag but that tag should be prefixed with c: suppose

<c:out value="ansari masood"/>

**JSTL:SQL**

First we have to add the mysql connector jar files and after that we have to add the taglib in jsp file

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %>

Then write the jdbc connection in jsp file.

<sql:setDataSource var="db" driver="com.mysq.jdbc.cj.Driver" url="jdbc:mysql://localhost:3306/stud" user="root" password="root"/>

<sql:query var="rs" dataSource="${db}">select \* from student</sql:query>

Program in jsp:

<sql:setDataSource var="db" driver="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost:3306/stud" user="root" password="root"/>

<sql:query var="rs" dataSource="${db}">select \* from student</sql:query>

<c:forEach items="${rs.rows}" var="quer">

<br><c:out value="${quer.id}"></c:out>:<c:out value="${quer.name}"></c:out>:<c:out value="${quer.email}"></c:out>

</c:forEach>

**Jstl Function Tags:**

If we want to find the length of the string then we have to add one more tag

<%@ taglib prefix="fn" uri="http://java.sun.com//jsp/jstl/functions" %>

<c:set var="str" value="Masood Ansari From Ayodhya UP 272130"></c:set>

Length: ${fn:length(str)}<br>

**Login Using Servlet and JSP:**

**Login.java**

**Note: If we want to logout the session then we have to use HttpSession**

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String username = request.getParameter("uname");

String password = request.getParameter("upass");

//without using database

if(username.equals("ansari")&&password.equals("masood")) {

HttpSession httpSession = request.getSession();

httpSession.setAttribute("username", username);

response.sendRedirect("welcome.jsp");

}

else {

response.sendRedirect("Login.jsp");

}

}

**Login.jsp:**

<body>

<form action="Login" method="post">

Enter Username: <input type="text" name="uname"><br>

Enter Password: <input type="password" name="upass"><br>

<input type="submit" value="login">

</form>

</body>

**Logout.java**

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

HttpSession httpSession = request.getSession();

httpSession.removeAttribute("username");

httpSession.invalidate();

}

**Welcome.jsp**

<body>

<%

response.setHeader("Cache-Control","no-cache, no store, must-revalidate");

response.setHeader("Expires", "0");

if(session.getAttribute("username")==null){

response.sendRedirect("Login.jsp");

}

%>

<h3>Welcome to Bangalore</h3>

<a href="videos.jsp">Watch Videos</a>

<form action="Logout">

<input type="submit" value="logout">

</form>

</body>

**How to Use HttpSession in Login page:**

HttpSession httpSession = request.getSession();

httpSession.setAttribute("username", username);

response.sendRedirect("welcome.jsp");

**How to prevent BackButton after Logout:**

Response.setHeader("Cache-Control","no-cache","no-store,must-revalidate");

Response.setHeader("Expires","0");