

JSP- DAY 1_2 ASSIGNMENT

Harshit Kushmakar | 16896

JSP Day 1 ASSIGNMENT

1. Do the following with only JSPs (and no servlets) and using JSP scripting elements:

- a. Create a method that take two int numbers and return their sum as int.
- b. Invoke the method from within template text using JSP elements (example o/p - Sum of 3 and 4 is 7.)
- c. Print above line to console as well using JSP elements.

Deploy the project into tomcat, restart server and compare the source jsp with the generated servlet.

```
<%@ page language="java" %>
<%
int sum(int a, int b) {
return a + b;
}
%>
<%
int num1 = 3;
int num2 = 4;
int result = sum(num1, num2);
out.println("Sum of " + num1 + " and " + num2 + " is " + result + ".");
System.out.println("Sum of " + num1 + " and " + num2 + " is " + result +
".");
%>
```

2. Write a program in JSP that prints the current date and time using JSP expressions.

```
<html>
<head>
<title>Current Date and Time</title>
</head>
<body>
<p>The current date and time is: <%= new java.util.Date() %></p>
</body>
</html>
```

3. Write a program in JSP that prints n numbers and their factorials in a tabular form. Also, get the value of n from the user using the GET method.

```
<%  
int n = Integer.parseInt(request.getParameter("n"));  
%>  
<html>  
<head>  
<title>Factorials</title>  
</head>  
<body>  
<h1>Factorials of Numbers from 1 to <%= n %></h1>  
<table>  
<tr>  
<th>Number</th>  
<th>Factorial</th>  
</tr>  
<%  
long factorial = 1;  
for (int i = 1; i <= n; i++) {  
factorial *= i;  
%>  
<tr>  
<td><%= i %></td>  
<td><%= factorial %></td>  
</tr>  
<%  
}  
%>  
</table>  
</body>  
</html>
```

4. Write a JSP that takes the user's name and age from a form.

a. Echo back the name and age along with a message stating the price of movie tickets.

b. The price is determined by the age passed to the JSP.

c. If the age is greater than 62, the movie ticket price is \$7.00.

d. If the user is less than 10 years old, the price is \$5.00.

e. For everyone else, the price is \$9.50.

```
<%  
int n = Integer.parseInt(request.getParameter("n"));  
%>  
<html>  
<head>  
<title>Factorials</title>
```

```

</head>
<body>
<h1>Factorials of Numbers from 1 to <%= n %></h1>
<table>
<tr>
<th>Number</th>
<th>Factorial</th>
</tr>
<%
long factorial = 1;
for (int i = 1; i <= n; i++) {
factorial *= i;
%>
<tr>
<td><%= i %></td>
<td><%= factorial %></td>
</tr>
<%
}
%>
</table>
</body>
</html>

```

JSP – Day 2:

6. Create a form to accept product details and store in a Collection. If the product is already inserted, display the appropriate message to insert another product. If the

product is inserted, show all the products available in the JSP page using:

- a. Scriptlets**
- b. Java Bean**
- c. Expression Language**
- d. JSTL tags**

→ Document the difference between all 4 mechanisms.

a. Scriptlets: In this mechanism, Java code is written directly in JSP using scriptlets. The code is embedded between tags. However, this mechanism is not recommended as it can lead to code clutter, decreased readability, and reduced reusability.

b. Java Bean: In this mechanism, a Java class with getter and setter methods is created to store the data entered by the user in the form. The JSP accesses the bean using the `useBean` action and uses the `getProperty` action to retrieve the data. This mechanism improves code reusability, modularity, and readability.

c. Expression Language: In this mechanism, expressions are used to retrieve the data stored in the bean. The expressions are embedded in the JSP using the `{}` syntax. This mechanism makes the code more concise and improves readability.

d. JSTL tags: In this mechanism, JSTL tags are used to access the data stored in the bean. The tags are embedded in the JSP using the `<jstl:tag>` syntax. This mechanism provides a high level of abstraction, makes the code more modular and reusable, and improves readability.

Overall, the use of JSTL tags is considered the best practice as it provides a high level of abstraction, makes the code more modular and reusable, and improves readability. Expression Language is also a good alternative as it makes the code more concise and readable. However, the use of scriptlets is not recommended due to its drawbacks.

