**AIM:** Programs based on JSP.

**THEORY:**

* ***JSP:-***
* Java Server Pages (JSP) is a server-side programming technology that enables the creation of dynamic, platform-independent method for building Web-based applications.
* JSP is a technology for developing web pages that support dynamic content which helps developers insert java code in HTML pages by making use of special JSP tags.
* A Java Server Pages component is a type of Java servlet that is designed to fulfill the role of a user interface for a Java web application.
* Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands.
* Using JSP, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.
* JSP tags can be used for a variety of purposes, such as retrieving information from a database or registering user preferences, accessing JavaBeans components, passing control between pages and sharing information between requests, pages etc.
* ***JSP Lifecycle:-***
* A JSP life cycle can be defined as the entire process from its creation till the destruction which is similar to a servlet life cycle with an additional step which is required to compile a JSP into servlet.
* The following are the paths followed by a JSP
* Compilation
* Initialization
* Execution
* Cleanup
* The four major phases of JSP life cycle are very similar to Servlet Life Cycle and they are as follows:



* ***The Scriptlet:-***
* A scriptlet can contain any number of JAVA language statements, variable or method declarations, or expressions that are valid in the page scripting language.
* Following is the syntax of Scriptlet:

<% code fragment %>

* ***JSP Declarations:-***
* A declaration declares one or more variables or methods that you can use in Java code later in the JSP file. You must declare the variable or method before you use it in the JSP file.
* Following is the syntax of JSP Declarations:

<%! declaration; [ declaration; ]+ ... %>

* ***JSP Expression:-***
* A JSP expression element contains a scripting language expression that is evaluated, converted to a String, and inserted where the expression appears in the JSP file.
* Because the value of an expression is converted to a String, you can use an expression within a line of text, whether or not it is tagged with HTML, in a JSP file.
* The expression element can contain any expression that is valid according to the Java Language Specification but you cannot use a semicolon to end an expression.
* Following is the syntax of JSP Expression:

<%= expression %>

* ***JSP Comments:-***
* JSP comment marks text or statements that the JSP container should ignore. A JSP comment is useful when you want to hide or "comment out" part of your JSP page.
* Following is the syntax of JSP comments:

<%-- This is JSP comment --%>

* ***JSP Directives:-***
* A JSP directive affects the overall structure of the servlet class. It usually has the following form:

<%@ directive attribute="value" %>

* There are three types of directive tag:

|  |  |
| --- | --- |
| **Directives** | **Description** |
| <%@ page ... %> | Defines page-dependent attributes, such as scripting language, error page, and buffering requirements. |
| <%@ include ... %> | Includes a file during the translation phase. |
| <%@ taglib ... %> | Declares a tag library, containing custom actions, used in the page. |

* ***The page Directive:-***
* The page directive is used to provide instructions to the container that pertain to the current JSP page. You may code page directives anywhere in your JSP page. By convention, page directives are coded at the top of the JSP page.
* Following is the basic syntax of page directive:

<%@ page attribute="value" %>

* Following is the list of attributes associated with page directive:

|  |  |
| --- | --- |
| **Attribute** | **Purpose** |
| buffer | Specifies a buffering model for the output stream. |
| autoFlush | Controls the behavior of the servlet output buffer. |
| contentType | Defines the character encoding scheme. |
| errorPage | Defines the URL of another JSP that reports on Java unchecked runtime exceptions. |
| isErrorPage | Indicates if this JSP page is a URL specified by another JSP page's errorPage attribute. |
| extends | Specifies a superclass that the generated servlet must extend |
| import | Specifies a list of packages or classes for use in the JSP as the Java import statement does for Java classes. |
| info | Defines a string that can be accessed with the servlet's getServletInfo() method. |
| isThreadSafe | Defines the threading model for the generated servlet. |
| language | Defines the programming language used in the JSP page. |
| session | Specifies whether or not the JSP page participates in HTTP sessions |

* ***The include Directive:-***
* The include directive is used to includes a file during the translation phase.
* This directive tells the container to merge the content of other external files with the current JSP during the translation phase.
* You may code include directives anywhere in your JSP page.
* The general usage form of this directive is as follows:

<%@ include file="relative url" >

* The filename in the include directive is actually a relative URL. If you just specify a filename with no associated path, the JSP compiler assumes that the file is in the same directory as your JSP.
* ***The taglib Directive:-***
* The JavaServer Pages API allows you to define custom JSP tags that look like HTML or XML tags and a tag library is a set of user-defined tags that implement custom behavior.
* The taglib directive declares that your JSP page uses a set of custom tags, identifies the location of the library, and provides a means for identifying the custom tags in your JSP page.
* The taglib directive follows the following syntax:

<%@ taglib uri="uri" prefix="prefixOfTag" >

* Where the uri attribute value resolves to a location the container understands and the prefix attribute informs a container what bits of markup are custom actions.
* ***JSP - Actions:-***
* JSP actions use constructs in XML syntax to control the behavior of the servlet engine.
* You can dynamically insert a file, reuse JavaBeans components, forward the user to another page, or generate HTML for the Java plugin.
* There is only one syntax for the Action element, as it conforms to the XML standard:

<jsp:action\_name attribute="value" />

* Action elements are basically predefined functions and there are following JSP actions available:

|  |  |
| --- | --- |
| **Syntax** | **Purpose** |
| jsp:include | Includes a file at the time the page is requested |
| jsp:useBean | Finds or instantiates a JavaBean |
| jsp:setProperty | Sets the property of a JavaBean |
| jsp:getProperty | Inserts the property of a JavaBean into the output |
| jsp:forward | Forwards the requester to a new page |
| jsp:plugin | Generates browser-specific code that makes an OBJECT or EMBED tag for the Java plugin |
| jsp:element | Defines XML elements dynamically. |
| jsp:attribute | Defines dynamically defined XML element's attribute. |
| jsp:body | Defines dynamically defined XML element's body. |
| jsp:text | Use to write template text in JSP pages and documents. |

**CONCLUSION:**

In this practical we learned how to create jsp pages and use various scripting elements and directives of it. We also learned to use the use bean tag and how session can be managed in jsp.

***Write a program to design a form and demonstrate the use of JSP Scripting Elements and JSP Directives.***

**CODE:**

**index.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Calculator</title>

</head>

<body>

<div align="center">

<h1> DIVISION OPERATION</h1><br>

<form action="Division.jsp">

<table>

<tr>

<td>Enter First Number:</td>

<td><input type="text" name="number1"></td>

</tr>

<tr>

<td>Enter Second Number:</td>

<td><input type="text" name="number2"></td>

</tr>

<tr>

<td colspan="2"><center><input type="submit" value="Divide"></center></td>

</tr>

</table>

</form>

</div>

</body>

</html>

**Division.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Division Operation</title>

</head>

<body>

<%@page errorPage="Error.jsp"%>

<%

int x = Integer.parseInt(request.getParameter("number1"));

int y = Integer.parseInt(request.getParameter("number2"));

int z = (x / y);

%>

<%= x +"/"+y+" = " +z%>

<%@include file="Footer.jsp" %>

</body>

</html>

**Error.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Error Page</title>

</head>

<body>

<%@page isErrorPage="true" %>

<h3> Sorry!!Some error occurred.</h3>

<%= exception %>

</body>

</html>

**Footer.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Division Operation</title>

</head>

<body>

<hr>

<center>

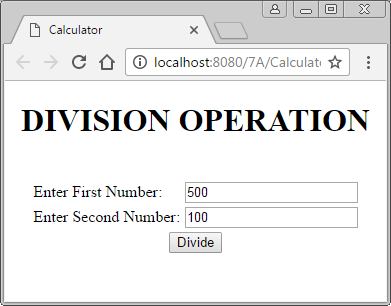
Developed by Mac <%= new java.util.Date()%>

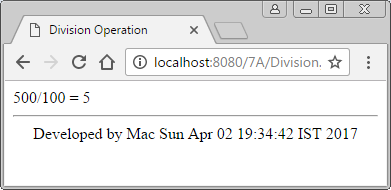
</center>

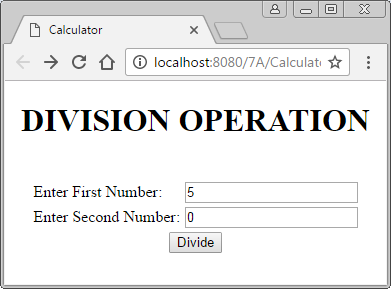
</body>

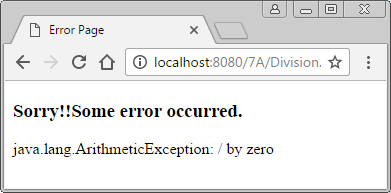
</html>

**OUTPUT:**

****

****

****

****

***Write a program to create a bean class and access it in JSP using useBean tag.***

**CODE:**

**index.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<jsp:useBean id="cal" scope="session" class="calclass.calclass">

<h1>Time:<jsp:getProperty name="cal" property="time" /><br>

Hour:<jsp:getProperty name="cal" property="hour" /><br>

Minute:<jsp:getProperty name="cal" property="minute" /><br>

Second:<jsp:getProperty name="cal" property="second" /></h1>

</jsp:useBean>

</body>

</html>

**calclass.java**

package calclass;

import java.util.\*;

public class calclass

{

private Calendar c;

public calclass()

{

c=Calendar.getInstance();

}

public Date getTime()

{

return c.getTime();

}

public int getHour()

{

return c.get(c.HOUR\_OF\_DAY);

}

public int getMinute()

{

return c.get(c.MINUTE);

}

public int getSecond()

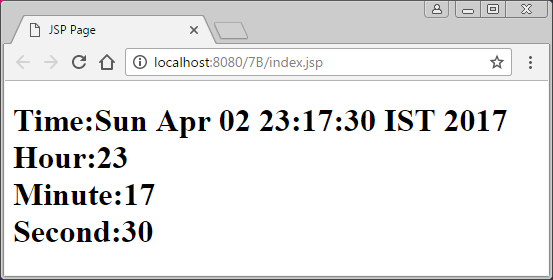
{

return c.get(c.SECOND);

}

}

**OUTPUT:**

****

***Write a program to create a visitor log that reports the IP Address of each user, and the time they visited the page.***

**CODE:**

**index.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Home</title>

</head>

<body>

<div align="center">

<form action="visitor.jsp" method="post">

<table>

<tr>

<td>Enter Name:</td>

<td><input type="text" name="Name"></td>

</tr>

<tr>

<td colspan="2" align="center"><input type="submit" value="Submit"></td>

</tr>

</table>

</form>

</div>

</body>

</html>

**visitor.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Visitor </title>

</head>

<body>

<div id="div2" align="center">

<%

String name = request.getParameter("Name");

%>

<b>User Logged In As : </b> <%= name%><br>

<b>IP Address of Client : </b>

<%

out.print(request.getRemoteAddr());

%>

<br>

<b>Time of Login is : </b>

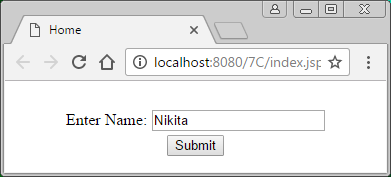
<%= new java.util.Date()%>

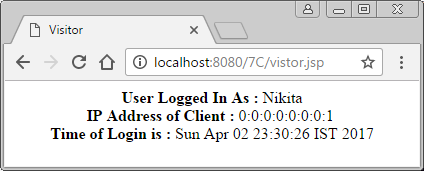
</div>

</body>

</html>

**OUTPUT:**

****

****

***Write a program for session management in JSP for shopping cart application.***

**CODE:**

**Login.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Login</title>

</head>

<body>

<div id ="div1" align="center">

<form action="List.jsp" method="post">

<h1>Login:</h1>

<table>

<tr>

<td>UserName:</td>

<td><input type="text" name="Name"></td>

</tr>

<tr>

<td>Password:</td>

<td><input type="password" name="Password"></td>

</tr>

<tr>

<td colspan="2" align="center"><input type="submit" value="Submit"></td>

</tr>

</table>

</form>

</div>

</body>

</html>

**List.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Online Cart</title>

</head>

<body>

<div id="div2" align="center">

<form action="List2.jsp" method="post">

<h1> Select Mobile :</h1>

<input type="checkbox" name="mobile" value="Motorola">Motorola

<input type="checkbox" name="mobile" value="Samsung">Samsung<br><br>

<input type="checkbox" name="mobile" value="Apple">Apple

<input type="checkbox" name="mobile" value="Oppo">Oppo<br><br>

<% session.setAttribute("uname", request.getParameter("Name"));%>

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

</form>

</div>

</body>

</html>

**List2.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Online Cart</title>

</head>

<body>

<div id="div3" align="center">

<form action="Item.jsp">

<h1>Select Laptop : </h1>

<input type="checkbox" name="Laptop" value="Dell">Dell

<input type="checkbox" name="Laptop" value="Lenevo">Lenevo<br><br>

<input type="checkbox" name="Laptop" value="HP">HP

<input type="checkbox" name="Laptop" value="ACER">Acer<br><br>

<%

String select[] = request.getParameterValues("mobile");

String items="";

if (select != null && select.length != 0)

{

for (int i = 0; i < select.length; i++)

{

items=items+select[i]+"<br>";

}

}

session.setAttribute("mobile",items);

%>

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

<h1>Items In Cart:<br><%=items%></h1>

</form>

</div>

</body>

</html>

**Item.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Online Cart</title>

</head>

<body>

<div id="div4" align="center">

<h1>Welcome, <%=session.getAttribute("uname")%> </h1>

<%

String select[] = request.getParameterValues("Laptop");

String items = "";

if (select != null && select.length != 0)

{

for (int i = 0; i < select.length; i++)

{

items = items + select[i] + "<br>";

}

}

session.setAttribute("Laptop", items);

%>

<h1>Selected Items are :<br><%=session.getAttribute("mobile")%>

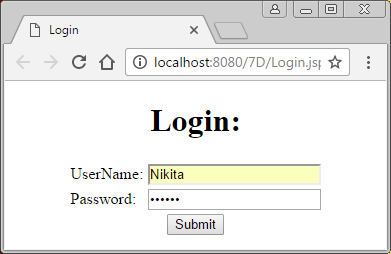
<%=session.getAttribute("Laptop")%></h1>

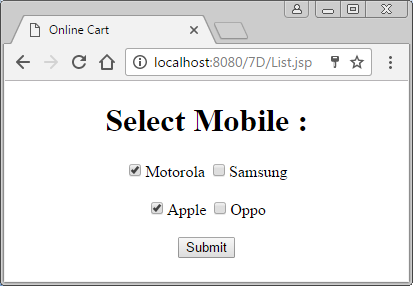
</div>

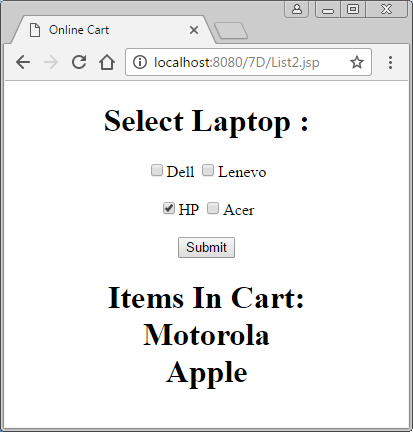
</body>

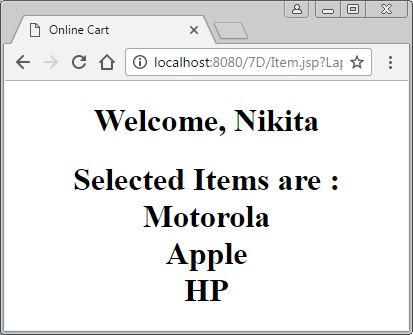
</html>

**OUTPUT:**

****

****

****

****