##### JSP L1 Assignments

##### User needs to create a dynamic web application “JSPL1Assignments” in eclipse using steps mentioned as below:

##### Open eclipse in Java EE perspective:

##### 

##### Create a dynamic web application “JSPL1Assignments”

##### Click File 🡪New🡪Dynamic Web Project

##### 

##### Enter web project name as “JSPL1Assignments:

##### 

##### Click next :

##### 

##### Again click next:

##### 

##### Check the check box “ Generate web.xml deployment descriptor”. And click finish button.

##### Dynamic web application “JSPL1Assignments” get created and it’s structure is as shown in the below screen shot :

##### 

##### Create a folder “jsp” under Web Content as shown below:

##### 

##### Now user can add all the jsp files which get created while executing below mentioned demos in “jsp” folder.

##### User need to run the JSP on server (Tomcat) to test .

##### User can submit the project by exporting it as .war file .

##### Demo 1: A JSP file that displays the date

Solution

<%@page contentType=*"text/html"* import=*"java.util.\*"* %>

<html>

<head><title>First JSP</title></head>

<body>

<h2>Here is today's date</h2>

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

</body>

</html>

**Demo 2 : To collect form data, with UI fields "username", "email" and "age"  and display the same**

Step 1 :

<HTML>

<BODY>

<FORM METHOD=POST ACTION="SaveName.jsp">

What's your name? <INPUT TYPE=TEXT NAME=username SIZE=20><BR>

What's your e-mail address? <INPUT TYPE=TEXT NAME=email SIZE=20><BR>

What's your age? <INPUT TYPE=TEXT NAME=age SIZE=4>

<P><INPUT TYPE=SUBMIT>

</FORM>

</BODY>

</HTML>

Step 2 : Define a Java class with fields "username", "email" and "age" with getter and setter methods

package user;

 public class UserData {

    String username;

    String email;

    int age;

//getter and setter

Step 3 :  "SaveName.jsp" to use a bean to collect the data.

<jsp:useBean id="user" class="user.UserData" scope="session"/>

<jsp:setProperty name="user" property="\*"/>

<HTML>

<BODY>

<A HREF="NextPage.jsp">Continue</A>

</BODY>

</HTML>

d) Implementing Sessions in JSP

Step 1 : GetName.html

<HTML>

<BODY>

<FORM METHOD=POST ACTION="SaveName.jsp">

What's your name? <INPUT TYPE=TEXT NAME=username SIZE=20>

<P><INPUT TYPE=SUBMIT>

</FORM>

</BODY>

</HTML>

Step 2: "SaveName.jsp", which saves the user's name in the session.

<%

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

   session.setAttribute( "theName", name );

%>

<HTML>

<BODY>

<A HREF="NextPage.jsp">Continue</A>

</BODY>

</HTML>

Step 3 : NextPage.jsp shows how to retrieve the saved name

<HTML>

<BODY>

Hello, <%= session.getAttribute( "theName" ) %>

</BODY>

</HTML>

**Demo 3 :**

Develop a JSP Page allows the user to select more than one option from multiple checkbox and also print the user-defined message that specifies the 'languages' option chosen by the user on submitting the button.

**Create**  **jspCheckBox.jsp with following code**

<html>  
<h2>Select Languages:</h2>  
  
<form ACTION="jspCheckBox.jsp">  
<input type="checkbox" name="id" value="Java"> Java<BR>  
<input type="checkbox" name="id" value=".NET"> .NET<BR>  
<input type="checkbox" name="id" value="PHP"> PHP<BR>  
<input type="checkbox" name="id" value="C/C++"> C/C++<BR>  
<input type="checkbox" name="id" value="PERL"> PERL <BR>  
<input type="submit" value="Submit">  
</form>

<%  
  
String select[] = request.getParameterValues("id");   
if (select != null && select.length != 0) {  
out.println("You have selected: ");  
for (int i = 0; i < select.length; i++) {  
out.println(select[i]);   
}  
}  
%>  
</html>

**Demo 4 :**

create the validation on the login form using Javascript method in JSP page

**AlertMessage.jsp**

<html>  
<head>  
<script>  
function validLogin(){  
if (document.form.userName.value == ""){  
alert ( "Please enter Login Name." );  
document.loginform.userName.focus();  
return false;  
}  
if (document.form.password.value == ""){  
alert ( "Please enter password." );  
document.userform.password.focus();  
return false;  
}  
alert ( "Welcome User" );  
return true;  
}  
</script>  
</head>  
<body>  
<form name="form" method="post" onsubmit="return validLogin();">  
<table width="250px" border=0 style="background-color:ffeeff;">  
<tr><td colspan=2 align="center" style="font-weight:bold;font-size:20pt;" align="center">User Login</td>  
</tr>  
<tr><td colspan=2>&nbsp;</td>  
</tr>  
<tr><td style="font-size:12pt;" align="center">Login Name:</td>  
<td><input type="text" name="userName" value=""></td>  
</tr>  
<tr><td style="font-size:12pt;" align="center">Password:</td>  
<td><input type="password" name="password" value=""></td>  
</tr>  
<tr>  
<td></td>  
<td><input type="submit" name="Submit" value="Login"></td>  
</tr>  
</table>  
</form>  
</body>  
</html>

Demo 5: jsp Page is Uploading The Image into Database using Multipart Request class FileSystem concepts and ByteArray .

<!-- uploaddb.jsp -->

<%@ page import="java.sql.\*,java.io.\*,java.util.\*,com.oreilly.servlet.MultipartRequest, com.mysql.jdbc.Driver;"%>

<%

/\* The Following Code is Used To Insert An Image Into Database \*/

String filename="";

try

{

//Download com.oreilly package

MultipartRequest multi= new MultipartRequest(request,".",5\*1024\*1024);

Enumeration files=multi.getFileNames();

File f=null;

while(files.hasMoreElements())

{

String name=(String)files.nextElement();

filename=multi.getFilesystemName(name);

String type=multi.getContentType(name);

f=multi.getFile(name);

System.out.println("The File is "+f);

}

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/<databasename>”,"username","password");

Statement stmt = con.createStatement();

InputStream is = new FileInputStream(f);

byte b[]=new byte[is.available()];

is.read(b);

String sql = "INSERT into photo\_test (\"Photo\") values('" + b + "')";

stmt.execute(sql);

stmt.close();

}catch(Exception e)

{

System.out.println(e);

}

out.println("The Image is Added into Database");

**Demo 6:** Uploading a file to a server using JSP

<!-- uploadserver.jsp -->  
<%@ page import="java.io.\*" %>  
  
<%  
String contentType = request.getContentType();  
System.out.println("Content type is :: " +contentType);  
if ((contentType != null) && (contentType.indexOf("multipart/form-data") >= 0)) {  
DataInputStream in = new DataInputStream(request.getInputStream());  
int formDataLength = request.getContentLength();  
  
byte dataBytes[] = new byte[formDataLength];  
int byteRead = 0;  
int totalBytesRead = 0;  
while (totalBytesRead < formDataLength) {  
byteRead = in.read(dataBytes, totalBytesRead, formDataLength);  
totalBytesRead += byteRead;  
}  
  
String file = new String(dataBytes);  
String saveFile = file.substring(file.indexOf("filename=\"") + 10);  
saveFile = saveFile.substring(0, saveFile.indexOf("\n"));  
saveFile = saveFile.substring(saveFile.lastIndexOf("\\") + 1,saveFile.indexOf("\""));  
  
//out.print(dataBytes);  
  
int lastIndex = contentType.lastIndexOf("=");  
String boundary = contentType.substring(lastIndex + 1,contentType.length());  
//out.println(boundary);  
int pos;  
pos = file.indexOf("filename=\"");  
  
pos = file.indexOf("\n", pos) + 1;  
  
pos = file.indexOf("\n", pos) + 1;  
  
pos = file.indexOf("\n", pos) + 1;  
  
  
int boundaryLocation = file.indexOf(boundary, pos) - 4;  
int startPos = ((file.substring(0, pos)).getBytes()).length;  
int endPos = ((file.substring(0, boundaryLocation)).getBytes()).length;  
  
FileOutputStream fileOut = new FileOutputStream(saveFile);  
  
  
//fileOut.write(dataBytes);  
fileOut.write(dataBytes, startPos, (endPos - startPos));  
fileOut.flush();  
fileOut.close();  
  
out.println("File saved as " +saveFile);  
  
}  
%>

Following code is for the error page, user can use this page to handle error conditions:

<%@ page language="java" isErrorPage="true" %>

<html>

<body>

<!-- This displays the fully-qualified name of the exception

and its message-->

<%= exception.toString() %>

<br>

<!-- This displays the exception's descriptive message -->

<%= exception.getMessage() %>

</body>

</html>