**JAVA SERVER PAGE EXERCISE**

1. **Problem Statement:**

Create a JSP named syntax.jsp which contains a textbox and a submit button named ‘OK’. After entering a string into the textbox and on submitting the jsp the control should be passed on to a new jsp named result.jsp, which should display the string entered in the textbox of syntax.jsp.Do this by using JSP expression.

**Deliverables Expected:**

A war file named syntax1.war containing two jsp pages ‘syntax.jsp’ and ‘result.jsp’ and a Web.xml file.

**Tips:**

1. Form submitting
2. Use JSP expression:
3. Use JSP implicit object request
4. **Problem Statement:**

Create a JSP named syntax.jsp, which contains a textbox and a submit button named ‘OK’. After entering a string into the textbox and on submitting the jsp the control should be passed on to a new jsp named result.jsp which should display the string entered in the textbox of syntax.jsp. Do this by using JSP scriplet.

**Deliverables Expected:**

A war file named syntax2.war containing two jsp pages ‘syntax.jsp’ and ‘result.jsp’ and a Web.xml file.

**Tips:**

1. Form submitting
2. Use JSP scriplet:
3. Use JSP implicit object request

**3.Problem Statement:**

Create a JSP named syntax.jsp, which prints the following line:

“You are visiting the page for: 1 time **“**when the page is loaded. On each subsequent refreshing of the page the no should be increased by one. For instance, if you are refreshing it for the 10th time then it should display “You are visiting the page for: 10 time”

**Deliverables Expected:**

A war file named syntax3.war containing one jsp pages ‘syntax.jsp’ and a Web.xml file.

**Tips:**

1. Use JSP declaration:

**4.Problem Statement:**

Create a Java Class com.cts.PerformTask. Containing a method doTask(), which returns a String “hello world”. Call this function from result.JSP page using scriplet. Full package name should not be used inside the scriplet while calling the method. Do this using import attribute of page directive.

**Deliverables Expected:**

A war file named page1.war containing one JSP page ‘result.JSP’ , com.cts.PerformTask and a Web.xml file.

**Tips:**

Use

**5.Problem Statement:**

Create a JSP named page.JSP. Set an attribute named ‘age’ with value ‘22’ in the session scope using scriplet within page.JSP. Try retrieving and printing the same value using session implicit variable. Now set the session attribute of page directive to false and study the difference

**Deliverables Expected:**

A war file named page2.war containing one JSP page ‘page.JSP’ and a web.xml file.

**Tips:**

1. Use
2. Use JSP implicit object session

**6.Problem Statement:**

Create a JSP named page.JSP which generates a divide by zero (For example: 10/0) error using scriplet. On executing the scriplet due to the the divide by zero error the control should pass to a new file ‘error.JSP’ which should display ‘’ **java.lang.ArithmeticException: / by zero”**

**Deliverables Expected:**

A war file named page3.war containing two JSP pages ‘page.JSP’ ,error.JSP and a web.xml file.

**Tips:**

1. Use
2. Use
3. Use exception implicit variable

**7.Problem Statement:**

Age processing application retrieves name and age from the user. It submits the age as number to another JSP, which prints the age in words.

**Deliverables Expected:**

Two JSP pages

**Tips:**

Apply JSP XML declarations

**8. Problem Statement:**

Car selling application has the logo and the organization’s name in the top of the page. It is common for all the JSPs used in the application. Provide a suitable solution for it.

**Deliverables Expected:**

A web application that has two or more JSPs

**Tips:**

Use JSP XML include directive

**9.Problem Statement:**

A photo gallery application needs to track the number of hits on a particular page (say “wonders.JSP”). Other parts of the application need access to the number of hits. A hits counter view page should be able to print the number of hits to the wonder.JSP page.

Deliverables Expected:

A Web application with two JSPs and one Java class (not a servlet)

Tips:

Use JSP XML page directive to import a Java class that has a static counter method.

**10. Problem Statement:**

A Web application has an index page, which asks the users for “user name”. Once it is submitted it calls another JSP which has to print that “user name”

Deliverables Expected:

A Web application with two JSPs

Tips:

Use JSP XML Expressions

**11. Problem Statement:**

A Web application has an index page, which asks the users for “user name”. Once it is submitted it calls another JSP, which connects to a database and fetches the age of the user from the table and displays it in the same.

**Deliverables Expected:**

A Web application with two JSPs

**Tips:**

Use JSP XML Expressions

**12. Problem Statement**:

A Web application needs to retrieve the contact email configured as an init param for whole of the application.

Deliverables Expected:

A Web application with one or more JSP page.

Tips:

Use JSP Implicit Object – application.

**13. Problem Statement:**

A Web application needs to retrieve the department name configured as an init param for that JSP alone.

Deliverables Expected:

A Web application that has one or more JSPs.

Tips:

Use JSP Implicit Object - Config

**14.Problem Statement:**

A Photo gallery application has an index page, which is designed just to throw a checked exception. The error should be captured in another JSP and the message has to be printed in the JSP that captures the exception.

Deliverables Expected:

A Web application with two or more JSPs.

Tips:

Use JSP Implicit Object – Exception, page directive.

**15.Problem Statement:**

A Web application has an index page, which takes username from the user. If submitted, it has to call a servlet, which then calls another JSP. This JSP has to print the username attached with the request.

Deliverables Expected:

A Web application with two JSP page and a Servlet.

Tips:

Use JSP Implicit Object – request.

**16. Problem Statement:**

A Web application needs to retrieve the contact email configured as an init param for whole of the application.

Deliverables Expected:

A Web application that has one JSP.

Tips:

Use JSP Page Scope – application.

**17. Problem Statement:**

A Web application needs to retrieve the time for which a user has logged in.

Deliverables Expected:

A Web application with two or more JSPs.

Tips:

Use JSP Page Scope – session.

**18. Problem Statement:**

A Web application has a header JSP used by other pages. Develop the application in such a way the Header page inclusion is always static

Deliverables Expected:

A Web application with two or more JSP pages

Tips:

Use JSP include directive

**19. Problem Statement:**

A Web application has a header JSP used by other pages. Develop the application in such a way the Header page inclusion is always dynamic.

Deliverables Expected:

A Web application with two or more JSP pages.

Tips:

Use JSP include action .

**20. Problem Statement:**

Create a Java Bean Class com.cts.Employee with int code, String name and float ‘salary’ as accessor methods. Create index.JSP containg three text boxes labeled as ‘Name’, ‘Employee Code’, and ‘Salary’. Index.JSP should also contain a submit button.

On submitting the inputs through the text areas of index.JSP the control should pass to another JSP file ‘result.JSP’. Result.JSP file should display the values entered in the previous page as ‘Name’, ‘Employee Code’ and ‘Salary’.

Deliverables Expected:

A war file named page.war containing two JSP pages index.JSP, result.JSP and com.cts.Employee and a Web.xml file.

Tips:

Use ,, and < JSP:setProperty >

**21. Problem Statement:**

Create a Java Bean Class com.cts.Employee with int code, String name and float ‘salary’ as accessor methods. Create index.jsp containg three textboxes labeled as ‘Name’, ‘Employee Code’ and ‘Salary’.

Index.jsp should also contain a submit button. On submitting the inputs through the text areas of index.jsp the control should pass to another jsp file ‘result.jsp’. Result.jsp file should display the values entered in the previous page as ‘Name’, ‘Employee Code’ and ‘Salary’.

Deliverables Expected:

A war file named page.war containing two JSP pages index.jsp, result.jsp and com.cts.Employee and a Web.xml file.

Tips:

1. Use Scriplets
2. Use implicit Object request in result page to retrieve the value from index.jsp

22.

Create a Welcome HTML page that accepts the first name and the last name. Clicking

the Submit button, values should be parsed in a JSP and the accepted values should be

displayed on the HTML page.

Additional Information:

Please note the following:

Title for the Html page: HomePage.

The name of the elements in the html page should be exactly same as following

o FirstName (textbox)

o LastName (textbox)

o Submit (button)

The JSP page should be created with following specifications.

o Title: GreetingJSP

o The page should display the text in the following format:

Welcome

Include the html page in the welcome list of web.xml file

Example: Suppose the name of the html page is HomePage.html

HomePage.html

23. Exercise 4.2 (Required)

(10 Marks)

Create a JSP called EmployeeSearch.jsp that accepts the Employee ID and

connects to the database, provided to you, to display the details of that person.

The table structure will be as follows:

Field Name DataType

EmpId Numeric

Name String

Age Numeric

Gender String

Deptname String

Basic Numeric

Grade String

You should have at least 5 records in that table.

If the queried row is not available, then forward the control to a JSP page, which

displays an appropriate error message.

If the row is available, then dispatch the request to another JSP called

RegisteredUser.jsp by including all the messages/details in the

EmployeeSearch.jsp and also display the details of the employee in

RegisteredUser.jsp in a tabular format.

24. Complete the following assignment:

Create a Javabean according to the bean naming conventions (Have a setter

i.

and a getter method for instance members “name” (smallcase) (String) and

“salary” (smallcase)(Float)).

Call it as MyJavaBean.java

ii.

The class name should be MyJavaBean

iii.

The package name should be com.cts.j2ee4\_5.

iv.

Create an HTML page with 2 text elements for accepting name and salary and

v.

1 submit button and call it as Call.html. It should be in a folder called

Ex4\_5:

Name the text element for name as name

Name the text element for salary as salary

Name the submit button as submit

When the user clicks the Submit button of the Call.html page, it should

vi.

invoke a JSP page called LearnJavabean.jsp. It should also be in a folder

called Ex4\_5 (Same as Call.html)

LearnJavaBean.jsp page will retrieve parameters name and salary from

vii.

Call.html and set the bean members using the setter methods of the bean.

Have a link in the LearnJavaBean.jsp called “GetDetails”(Case Sensitive)

viii.

By Clicking the “GetDetails” link will access the details, that is, name and

ix.

salary from the bean and display it in another page called DisplayBean.jsp.

It should also be in a folder Ex4\_5.

Thanks & Regards,

T. Senthil Kumar

Mobile: 9894088066