



Course Code /Title: CSA4399 – Internet Programming
Programme : Computer Science and Engineering

ASSIGNMENT 3 QUESTIONS

SET -5

S.No	Questions	Marks	CO	BTL
1	Implementing a feature in a web application that tracks the number of accesses by a client within a single session. You need to use Java Servlets to manage and monitor session data. The application should count how many times the client accesses the application during their session and retrieve information about the session, such as the session ID, creation time, and last accessed time.	20	CO4	3
2	Write a scenario where you had to use JSTL to solve a complex problem and how you went about it. Also, elaborate the function library in JSTL and how to create custom functions.	20	CO4	2
3	A page of stock market quotes uses script to refresh the page every five minutes in order to ensure the latest statistics remain available. 20 seconds before the five minute period expires, a confirm dialog appears asking if the user needs more time before the page refreshes. This allows the user to be aware of the impending refresh and to avoid it if desired.	20	CO5	3
4	You are developing an e-commerce application that needs to integrate with an external payment gateway service. This service is described using a WSDL file. How would you use the WSDL file to integrate the payment gateway service into your e-commerce application? Describe the steps involved in generating the client code, invoking the service, and handling any potential errors	20	CO6	

Assignment - 3

1) Servlet Code

```
import java.io.*;
import java.print.*;
import java.util.*;
import java.servlet.*;
import javax.servlet.*;
import javax.servlet.annotation.*;
import javax.servlet.http.*;
```

a) Web Servlet ("session Tracker")

```
public class SessionTracker extends HttpServlet {
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws
        ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        HttpSession session = request.getSession();
        String sessionID = session.getId();
        long creationTime = session.getCreationTime();
        long lastAccessedTime = session.getLastAccessedTime();
        Integer visitCount = (Integer) session.getAttribute("visitCount");
        if (visitCount == null)
            visitCount = 0;
        {
            visitCount++;
            session.setAttribute("visitCount", visitCount);
            out.print("<html><body>");
        }
    }
}
```

out.println ("Session ID: " + session.getId() + "
");
 out.println ("Session created: " + new Date (Time) + "
");
 out.println ("</body> </html>");

Output:

Session ID: 1234567890ABCDEF
 Session created: Mon Sep 09 12:00:00 IST 2024
 Last Accessed: Mon Sep 09 12:01:05 IST 2024
 No. of Accesses on this Session: 1

2) JSP Code using JSTL:

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

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

</html>

<head>

<title> Order Management </title>

</head>

<body>

<h2> Order </h2>

<form method = "GET" action = "Order.jsp">

<label for = "status"> Filter by status: </label>

<select name = "status" id = "status">

<option value = "All"> All </option>

<option value = "Pending"> Pending </option>

</select>


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

</form>

<table border = "1">

2007 04 20 11:00

4th Date 11/15



412 Amount 2142

40

2.1.1.1.1

11/14/20

20 (done?)

Coculation test: + Paxam statum = All Holstein
statu: Paxam statum

ホフ

207 10/10/90 21/1/97
21/1/97 21/1/97

Lagerplatz (H)

24178 1000. station 2 cited?

4b) \$ \{ \text{order statistic } (1 \text{st}) \}

452

20 Feb 2017

$\langle 1 \text{ body} \rangle$

2 body

21h 2m 17s

output

Order ID	Date	Status	Amount
1002	2024-09-08	Pending	156.00
1003	2024-09-09	Pending	300.01

* Creating Custom Functions in JSTL:

2 tags/xmlns: "http://java.sun.com/xml/ns/java:version"

$$\text{Hilb-version} > 1.0 < \text{Hilb-version}$$

$\langle \text{short-name} \rangle$ Custom $\langle \text{short-name} \rangle$

<url> http://example.com/customClass

(function)

← Function class → Com. Example. Custom function

← Function class → Com. Example. Custom function

6. Author Signature gave. Lan. ~~for~~

Handwritten Signature

17/10/17

Output

Custom	Function	Example

Original : Hello world.

Reversed : 1/80w 6/14

8 Java Script Code Snippet:

<!DOCTYPE html>

<html lang="en">

<head>

meta charset="UTF-8">

Article > stock market Quotes <Title>

(Script)

function xefu Page 1/1

Location selected (),

2

Set Timeout (c) \Rightarrow d.

Cont Conform Ref: Conform (The Page will separate)

{chr}
 alert ("Page - refresh Canceled"),

3

{ 236000 }

(1) Script

$\langle \text{body} \rangle$

- ΔI > Stock Market Quota (Chic)

</body>
</html>

Output

Page Display

Stock Market Quote

- Apple Inc (AAPL): \$150.00
- Microsoft Corp (MSFT): \$250.00
- Alphabet Inc (GOOGL): \$2800.00

Confirmation Dialog

The page will refresh in 20 Sec.

Page refresh

(The page reload, displays, update content)

Stock Market Quote

- Apple Inc. (AAPL): \$152.00
- Microsoft Corp. (MSFT): \$256.00
- Alphabet Inc. (GOOGL): \$2850.00

4) To integrate an External Payment gateway source into your e-commerce application using a wsdl file.

5) Generate client code from wsdl

wsimport - keep - s - src - cl bin - p com. Example

Payment - verbose http:// Example

com/payment gateway <url>

2) ~~Interact~~ Generate Code into Application

- Include Generated Code.
- Configure Service Endpoint

3) Invoice the payment Service

- ~~Create~~ Service Instance

Payment Service Service = new payment

Invoke Method

Payment Response Response = Post. process ~~payment~~

4) Handle Response & Error:

- check Response

if (Response.IsSuccess)

|| Handle Successful payment.

else

|| Handle Payment failure.

↳ Exception Handling

↳ Catch (Soap or Exception)

↳ Catch Web Server Exception

|| Handle Connectivity & Configuration errors.

},

Output

Successful payment.

payment Successful. Transaction ID: 12897632183D3142C

Payment failed

Payment failed. Error: Invalid Credit Detail

SOAP Fault (e.g. Invalid Request)

Payment failed due to SOAP Fault: Invalid Request

Question no.	Rubrics.	Marks	Marks obtained.
Q1.	Explanation of JDBC.	5 Marks	
	Connection pooling	6 Marks	
	SQL Queries.	5 Marks	
	Statement Types.	4 Marks	
	Life cycle phase.	6 Marks.	
Q2.	Embedding Java Code.	5 Marks	
	Advantages / Disadvantages	5 Marks	
	Clarity & Depth	4 Marks.	
	PHP Code Implementation	8 Marks.	
Q3.	HTML table structure	5 Marks.	
	Alternating Colour logic	4 Marks.	
	Explanation	3 Marks.	
	Code Implementation.	8 Marks.	
Q4.	Pattern Extraction	5 Marks.	
	XML File Generation	4 Marks	
	DTD vs XML Schema.	8 Marks.	