



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	3

### Assignment - 3

Question	Mark Split Up	Mark Score	Total Marks
Question 1	Code Implementation Session Data Accuracy Efficiency, clarity Explanation	(8) (5) (3) (4)	20
Question 2	Scenario Explanation Function library Custom functions clarity and Organization	(6) (5) (5) (4)	20
Question 3	Script functioning UI Design Code Efficiency Explanation	(8) (5) (4) (2)	20
Question -4	Understanding WSDL Client Code Generation Error handling clarity & Depth	(6) (6) (4) (4)	20

① Why?

actions

JD

API

Achie

four

com

20

### ASSIGNMENT -3

- Implementing a feature in a web application that tracks the number of accesses by a client within a single session using Java servlets using HttpSession object to manage and monitor session data.

Servlet Code:

```
import java.io.IOException; import javax.servlet.ServletException; import java.io.PrintWriter; import javax.annotation.WebServlet; import javax.servlet.http.HttpServlet; import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse  
@WebServlet("/Session Tracker")  
public class Session Tracker extends HttpServlet {  
    protected void doGet(HttpServletRequest request,  
                         HttpServletResponse response) throws ServletException,  
    response.setContentType("text/html");  
    PrintWriter out = response.getWriter();  
    HttpSession session = request.getSession(true);  
    String sessionId = session.getId();  
    long creationTime = session.getCreationTime();  
    long lastAccessedTime = session.getLastAccessedTime();  
    long lastModifiedTime = session.getLastModifiedTime();  
    int count = (int) session.getAttribute("count");  
    count++;  
    session.setAttribute("count", count);  
    out.println("Session ID: " + sessionId);  
    out.println("Creation Time: " + creationTime);  
    out.println("Last Accessed Time: " + lastAccessedTime);  
    out.println("Last Modified Time: " + lastModifiedTime);  
    out.println("Count: " + count);  
}
```

```
2. Integer visitCount = (Integer) session.getAttribute("visitCount");  
if (visitCount == null) {  
    visitCount = 0; visitCount++; // Visit count  
    session.setAttribute("visitCount", visitCount);  
    out.println("<html><body>");  
    out.println("<h1>Session Tracking </h1>");  
    out.println("<p> Session created : " + new Date());
```

```
out.println("<p> No. of accesses in this  
session : " + visitCount);  
out.println("</body></html>");
```

}

**Output :-**

Session Tracking Example

Session Id : 123456789 (short duration - 30s. session)

Session created : Tue Sep 10 12:00:00 IST 2024

Last Accessed : Tue Sep 10 12:01:05 IST 2024

No. of access in this session : 1

Elaborate a more complex program to elaborate the create session.

JSP Code

<%@ page

<%@ fac

<html>

<head>

</head>

<body>

00:00

00:00

Lec

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.

### JSP Code Using JSTL:-

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core"
   prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions"
   prefix="fn"%>

<html>
<head>
    <title> Order Management </title>
</head>
<body>
    <h2> Orders </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="Delivered"> Delivered </option>
        </select>
        <input type="submit" value="Filter" />
    </form>
    <table border="1">
```

JSTL

Create

Table

```

</thead>
<tbody>
    <c:forEach var="order" items="${orders}">
        <c:choose>
            <c:when test="param.status == 'All' || order.status == param.status">
                <c:choose>
                    <c:when test="order.status == param.status">
                        <c:forEach var="item" items="order.items">
                            <tr>
                                <td>${item.name}</td>
                                <td>${item.quantity}</td>
                                <td>${item.price}</td>
                                <td>${item.total}</td>
                            </tr>
                        </c:forEach>
                    <c:otherwise>
                        <tr>
                            <td colspan="4" style="text-align: center;">No orders found!
                        </tr>
                    </c:otherwise>
                </c:choose>
            <c:otherwise>
                <tr>
                    <td colspan="4" style="text-align: center;">No orders found!
                </tr>
            </c:otherwise>
        </c:choose>
    </c:forEach>
</tbody>

```

Output:-

Order List:

Order ID	Date	Status	Amount
1002	2024-09-08	Pending	150:00
1003	2024-09-09	Delivered	300:00

## JSTL Function Library (fn)

Creating custom functions in JSTL:

```
<taglib xml:ns="http://java.sun.com/xml/ns/javaee"  
version="2.1">  
<tlib-version>1.0</tlib-version>  
<short-name>custom</short-name>  
<uri>http://example.com/custom</uri>  
<function>  
<name>reverseString</name>  
<function-class>com.example.custom.Functions  
</function-class>  
<function-signature>javax.lang.String  
reverseString(javax.lang.String)</function-signature>  
</function>  
</functions>  
</taglib>
```

Output:-

custom function Example:

Original: Hello World

Reversed: dlrow olleH

8. To implement the described functionality for refreshing a stock market quotes page every five minutes, with a confirmation dialog appearing 20 seconds before the refresh.

JavaScript Code Snippet:

```
<!DOCTYPE html>
<html>
<head>
    <title>Stock Market</title>
<script>
    function refreshPage() {
        location.reload();
        SetTimeout(20000);
        const confirmRefresh = confirm("The
            page will refresh in 20 seconds");
        if (!confirm Refresh) {
            refreshPage();
        } else {
            alert("Page refresh cancelled");
        }
    }
    setTimeout(refreshPage, 280000);
</script>
<head>
<body>
    <h1> Stock Market Quotes </h1>
</body>
</html>
```

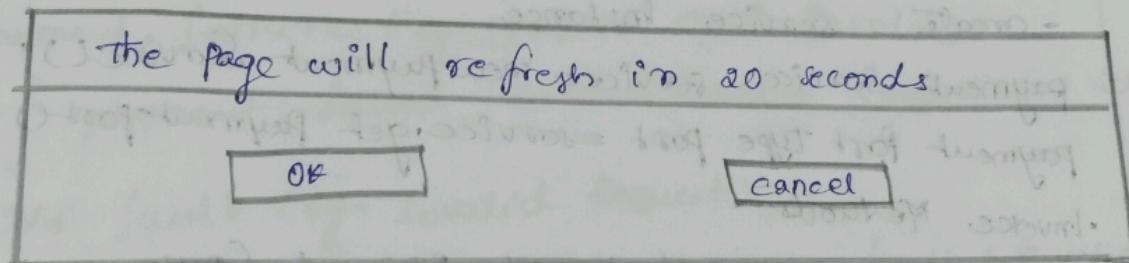
Output  
Page

Output :-

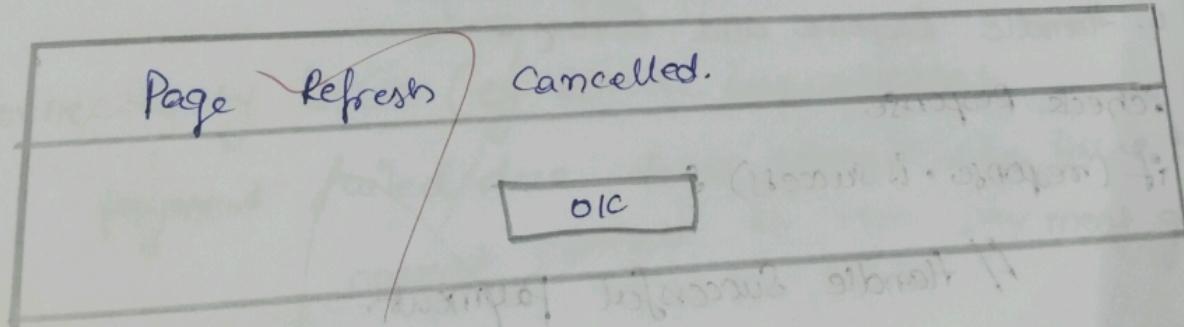
Page Display:

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

Confirmation Dialog:



Alert:



Page Refresh:

Stock Market Quotes	
- Apple Inc (AAPL)	: \$ 152.00
- Microsoft Corp. (MSFT)	: \$ 250.00
- Alphabet Inc (GOOGL)	: \$ 2850.00

4. To integrate an external payment gateway service into your e-commerce application using a WSDL file.

1. Generate client code from WSDL -

```
wsimport -keep -s src -d bin -p com.example  
payment -verbose http://example.com/  
payment_gateway?wsdl
```

2. Integrate generated code into application -

- Include Generated code
- Configure Service Endpoint

3. Invoke the payment Service -

- Create Service Instance

```
PaymentService service = new PaymentService();
```

```
PaymentPortType port = service.getPaymentPort();
```

- Invoke Methods

```
PaymentResponse response = port.processPayment(PaymentRequest);
```

4. Handle Response and Errors -

- Check Response

```
if (response.isSuccess()) {
```

// Handle Successful Payment.

```
} else {
```

// Handle Payment Failure

- Exception Handling

try {

    paymentResponse = post("processPayment");  
} catch (SoapFault e) {

    // handle SOAP faults (e.g.; invalid request)

} catch (WebServiceException e) {

    // handle connectivity or configuration errors

}

**Output :-**

successful payment :

Payment successful. Transaction Id: 1928346210589.

Payment failure (e.g. Invalid Card Details):

Payment failed. Error: Invalid credit card details.

SOAP fault (e.g. Invalid request)

Payment failed due to SOAP fault & Invalid  
Request Format.

Connectivity Issue (e.g. Service Unavailable)

Payment failed due to a connectivity issue:

cannot connect to the payment gateway.