

# Assignment - 3

## ① Servlet Code:

```

import java.io.IOException;
import java.printWriter;
import java.util.Date;
import java.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.HttpServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

@WebServlet("/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(true);
        String sessionId = session.getId();
        long creationTime = session.getCreationTime();
        long lastAccessedTime = session.getLast
        AccessedTime();
        Integer visitCount = (Integer) session.get
        Attribute("visit count");
        if (visitCount == null) {
            visitCount = 0;
        }
        visitCount++;
        session.setAttribute("visit count", visitCount);
        out.println("<html><body>");
    }
}

```

```
out.println("<p> Session ID: " + session.getId() + "</p>");  
out.println("<p> Session Created: " + new Date(session.getCreationTime()) + "</p>");
```

```
out.println("<p> No. of access in this session: Visit  
Count++ </p>");
```

```
out.println("</body></html>");
```

## Output:

Session ID: 12345ABCDEHGFEDCBA

Session Created: Mon Sep 09 12:00:00 IST 2024

Last Accessed: Mon Sep 09 12:01:05 IST 2024

No. of accesses in 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/functions" prefix="fn"%>
```

```
<html>
```

```
<head>
```

```
    <title> Order Management </title>
```

```
</head>
```

```
<body>
```

```
    <h2> Order </h2>
```

```
    <form method="GET" action="orders.jsp">
```

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

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

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

```
            <option value="pending"> pending </option>
```

```
        </select>
```

```


<form>
<table border="1">
<thead>
<tr>
<th> Order ID </th>
<th> Date </th>
<th> Status </th>
<th> Amount </th>
</tr>
</thead>
</body>
<c:for each var="order" items="${orders}">
<c:choose>
<c:when test="${param.status='All' || orders
•status=param.status}">
<tr>
<td>${order.id}</td>
<td>${order.date}</td>
<td>${order.status}</td>
<td>${order.amount}</td>
</tr>
</c:when>
<c:when>
<c:for each>
</t body>
</body>
</html>

```

Output:

Order ID	Date	Status	Amount
1002	2024-09-08	Pending	150.00
1003	2024-09-09	Pending	300.00

## \* Creating Custom Functions in JSTL:

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

### Output:

Custom function Example

Original : Hello World

Reversed : dlrow olleH

## ③ Javascript code Snippet

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <title> Stock Market Quotes </title>  
<script>  
  function refreshPage() {  
    location.reload();  
  }</script>
```

Set Timeout (c) => {

const confirmRefresh = confirm ("The Page will

```
refresh in 20 Sec ");  
if (!confirm("Refresh")) {  
    refreshPage();  
}  
else {  
    alert("Page refresh Canceled");  
}  
(280000);  
</script>
```

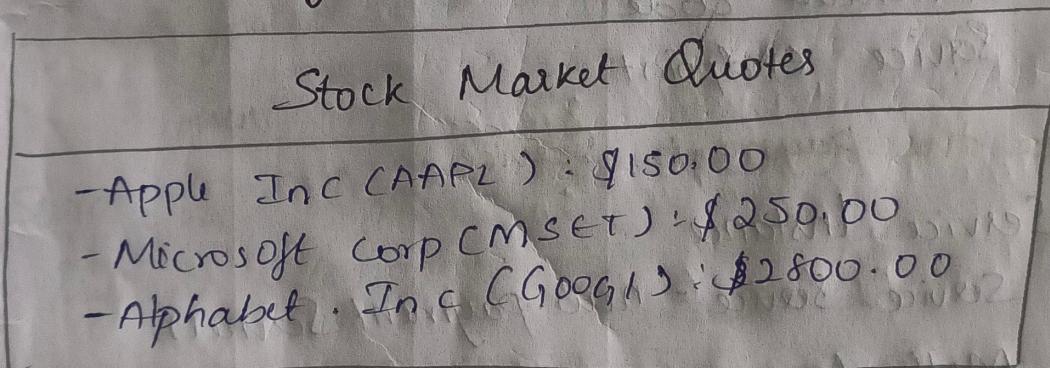
```
<body>  
    <h1> Stock Market Quotes </h1>
```

```
</body>
```

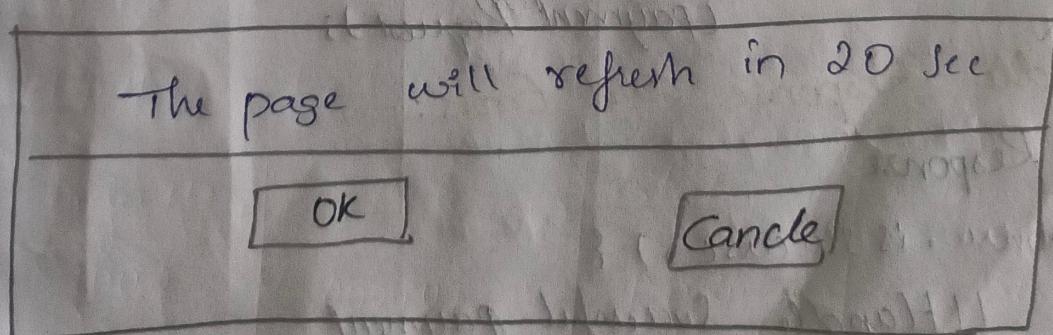
```
</html>
```

Output :

Page Display



Confirmation Dialog



Page refresh

(the page reload, displaying updated Content)

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 Invoice the payment Service

- Create Service Instance
- Payment Service Service = new PaymentService();
- Invoke Methods
- Payment Response response = port.processPayment(paymentRequest);

④ Handle Response and Errors:

- Check Response

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

// Handle Successful payment

else {

// Handle payment failure

- Exception Handling

try {  
Payment Response response = port.processPayment(  
(Payment Request))

} Catch { SOAPFaultException } {

// Handle SOAP faults (e.g.; Invalid request)

} Catch CWebServiceException {

// Handle Connectivity or Configuration errors

}

## Output:

Successful Payment:

Payment successful. Transaction ID: 1289763215803

Payment failed (e.g.; invalid card details):

Payment failed. Error: Invalid Credit Card Details

SOAP Fault (e.g.; Invalid Request)

Payment failed. due to a SOAP fault : Invalid Request format