Assignment - 3

① Servlet Code:

```
import java.IoException;
import java.printWriter;
import java.util.Date;
import java.Servelt.ServeltException;
import javax.servelt.annotation.WebServelt;
import javax.Servelt.HttpServelt;
import javax.servelt.http.HttpServletRequest;
import javax.servelt.http.HttpSession;
@WebServlet("/SessionTracker")
Public Class SessionTracker extends HttpServelt {
    Protected void doGet(HttpServletRequest request,
            HttpServletResponse response) throws
            ServletExpection IOExpections {
        response.SetContentType("type/html");
        printWriter out = response.getWriter();
        HttpSession Session = request.getSession(true);
        String SessionId = SessionId();
        long CreationTime = Session.getCreationTime();
        long LastAccessedTime = Session.getLast
            AccessedTime();
    Integer VisitCount = (Integer) Session.get
        Attribute("visit count");
    if (Visit count == null) {
        Visit Count = 0;
    }
    Vist Count++;
    Session.SetAttribute("visit Count", Visit Count);
    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 : 12345ABCDE

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 Managment </title>
</head>
<body>
    <h2> order </h2>
    <form method="GET" action="orders.jsp">
    <label for="status"> Filter by staus: </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">
 <thead>
    <tr>
      <th> Order ID </th>
      <th> Date </th>
      <th> Status </th>
      <th> Amount </th>
    </tr>
 </thead>
 <tbody>
    <c:forEach var="order" items="${orders}">
    <c:choose>
      <c:when test="${param.status == 'All' || order.status == param.status}">
        <tr>
          <td>${order.id}</td>
          <td>${order.date}</td>
          <td>${order.status}</td>
          <td>${order.status}</td>
        </tr>
      </c:when>
    </c:forEach>
  </tbody>
</table>
</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>
    <function>
        <name>reverse_string</name>
        <function-class>com.example.customFunction
        </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();
        }
        setTimeout(() => {
            const confirmRefresh = confirm("The Page will
```

```
    refresh in 20 Sec ");
    if (! confirm Refresh) {
        refresh page();
    } else {
        alert ("page refresh Canceled");
    }
    }, 20000);
</script>
<body>
    <h1> Stock Market Quotes </h1>
</body>
</html>
```

Output :

Page Display

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

Confirmation Dialog

The page will refresh in 20 Sec
OK Cancle

Page refresh
(The page reload, displaying updated Content

Stock Market Quotes
- Apple Inc. (APPL) : $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 Generate 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 respons = port.ProcessPayment
        (paymentRequest);
```

④ Handle Response and Errors:

- Check Response

```
if (response.isSuccess()) {
        // Handle Successful payment
} else {
        // Handle payment failure
}
```

- Exception Handling

```
try {
    PaymentResponse response = port.processPayment(
            (paymentRequest)
} Catch (SOAPFaultException e) {
        // Handle SOAP faults (eg; Invalid request)
} Catch (WebServiceException e) {
        // Handle Connectivity or Configuration errors
}
```

Output:

Successful payment:

Payment Successful. Transaction ID: 12897653158031

Payment failed (eg: Invalid Card Details):

Payment failed. Error: Invalid Credit Card Details

SOAP Fault (eg: Invalid Request)

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