ASSIGNMENT - 3

1. Tracking Session Accesses using Java Servlets:

Servlet Configuration:

⇒ Create Java servlet that uses Http Session to Manage the Session.

⇒ Each time client accesses the servlet, check if the session is new or existing

Session Data Tracking:

Session ID: use Session.getId () to retrieve the unique ID for session.

Creation time: use Session.getCreationTime ().

Last accessed time use: Session.getLastAccessedTime () to check most recent time.

Sample Code:

```
HttpSession Session = request.getSession ();
Integer accessCount = (Integer) Session.getAttribute ("accessCount")
if (access count == null) {
    accessCount = 1;
} else {
    access count += 1;
}
```

Output: Session. ID
Creation time
Last Accessed time

2. Using JSTL to Solve Complex problem :

Steps to use JSTL :
Core JSTL library Include JSTL in the JSP file using :

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

Using JSTL for Data Manipulation :

⇒ Iterate over a list using <c: foreach >.
⇒ Conditional rendering <c: if > or <c: choose>.
⇒ format dates and numbers with <fmt. format Date > and <fmt: format Number >.

⇒ JSTL function library :

The JSTL function library (fn:) allows for String Manipulation like Substring, checking for Containment, etc

Sample Code :

```
<c: forEach var = "product" items = "${productlist}">
  <div> ${product.name} - ${product.price}.
```

Output :

```
List of product = "Apple" - 500
                  "Orange" - 100.
                  "Banana" 50
```

3. Auto - refreshing page with user Conformation:

The task is to create a page that refreshes every five minutes, but 20 seconds before a refresh.

Steps:

1) page Auto - refresh logic: use Java scripts SetTimeout () to trigger page refresh every 5 Minutes.

Confirmation dialog:

use Confirm () to display a dialoge box 20 Seconds before the page refresh.

Sample Code: output:

```
Set Timeout ( function() {
var refresh= Confirm ("Do you need more time?");
if (! refresh) {
location .reload ();
}}, (5 * 60- 20) * 1000);
```

This ensures the page automatically refreshes unless the user cancles the refresh via the dialog box.

output:

Do you need more time ?

4. Integrating with an External payment gateway using WSDL:

1) Generating Client code from WSDL:

→ use wsimport (available in Jdk) to generate Java Client classes.

bash
```
wsimport -keep -p com.example.payment http://example.
    com/payment?wsdl
```

This command will generate client side Stubs in specified package.

Web Service output:

```
paymentService service = new paymentService();
payment port = Service.getpaymentport();
String response = port.processpayment(paymentdetails);
```

Error handling

Handle potential network errors or invalid response using try-catch blocks.

Eg:
```
try {
    paymentresponse response = port.processpayment(payDetai
    System.out.println("payment Successful" + response.
                                    getStatus());
```

Output:

```
Payment Successful.
```