Assignment - ③

1) Implementing a feature in a web application that tracks the no of access by a client with in a single session using java servlets using HTTP session object to manage and manitor session data

Servlet Code

```
import java.io.IoException;
import java.io.printWriter;
import java.util.Date;
import javax.servlet.servletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet("/SessionTracker")
public classsessionTracker (HttpServlet
request Request HttpS
-servedtResponse response) throws
    servletException IoException {
Response.setContentType(text/html");
```

```
print writer & : responses . get write ();
HP session session : request. get session (true
String session Id = session. get Id ();
Long creation time = session . get (Id);
long Last Accessed Time = session. getLast
Accessed Time ();
Integer visit count = (Integer) session get
Attribute ("visit Count");
if (visit Count == null) {
out . println ("<body><html>");
}
}
```

output

Session Tracking Example

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 the session: 1

2) write a scenario where you had to use JSTL to slove a complex problem & how you went about it Also, Elabroate 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, "http://java sun com/jsp/jstl/function" prefix = "fn" %>

<Html>
<head>
<title> order management </title>
</head>
<body>

<h2> order list </h2>
<form method = "GET" action = "order.jsp">
<label for = "status" id = "status">
<select name = "status" > filter by status:
<label>.
<option value = "All"> All </option>
<option value = "Delivered"> delievered
</option>

```
</select>
<input type="submit" value="Filter">
</form>
<table border="1">
 <head>
  <tr>
<th>order ID</th>
<th>Date</th>
<th>status</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>${order.id}</td>
<td>${order.date}</td>
<td>${order.status}</td>
<td>${order.amount}</td>
</tr>
</c:when>
<c:choose>
</c for choose>
```

</c:forEach>

</tbody>
</table>
</body>
</html>

output

Order List

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

JSTL Function Library (fn)

* creating custom functions in JSTC

<taglib xmlns = "http://java.sun.com/xml/ns/javacc"

version = "2.1">

<Hib.version>1.0</Hib.version>

<short-name> customc </short-name>

<uri> http://example.com/customc/uri</uri>

<function>

<name> Reverse String </name>

<function-class> com.Example.CustomFunction</function-class>

<function signature> java.lang.string
Reversing (java.lang.string)
</function signature>

</function>
</taglib>

Output:

Custom function example
Original : Helloworld
Reversal dlrowolleH

3) 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 lang = "en">
<head>
<meta charset = "UTF-8">
<title> stock market quotes </title>
<script>

function Refresh page ();
location Reload ();
3, 28,000);
</script>
<body>
<h> stock market quotes </h>
</body>
</html>

output: Page display

Stock Market quates

- Apple Inc (APP2) : $150.000
- Micrsoft Comp (MS.FT) - 250.000
- Alphabet - Inc (google) - $ 28000

Confirmation dialog:
The page will be Refresh in 20 mins

OK Cancel

Alert

Page Refresh Cancelled

OK

Page Refresh

The page Reloads, displays, updated content

- Apple Inc (ARPL) : $152.000
- Microsoft Corp (MSET) : $250.000
- Alphabet Inc (Google) $ 2850.00

Question - ④

To integrate an External payment gateway Service into your e-commerce application using a NSDL File

① generate client code for WSDC

wsimport -keep -s src -d bin -p Com.Example Payment -verbose http://example.com/payment gateaway?wsdl

② Integrate generated code into Application

- Include Generate Code
- Configure Service Endpoint

③ Involve the payment service

- create service Instance

Payment service service = new payment service();

PaymentPortType port = service.getPaymentPort();

• Involve methods:

PaymentResponse response = port.processPayment(PaymentRequest);

Output

PaymentResponse: Payment successful for amount 100.00.

Assignment - ④

① Why JDBC is Essential in Building Database-driven Applications

JDBC is essential because it provides a standard API for java applications to interface with database.

Achieving JDBC Connection pooling using JDBC.

Configuration DataSource

```
<Resource name = "jdbc/MyDB"
          auth = "Container"
          type = "javax.sql.DataSource"
          Max total = "20"
          max Idle = "10"
          Max wait Mills = "10000"
          username = "dbuser"
          password = "dbpassword"
          driverClassName = "class.mysql.
                            jdbc.driver"
url = "jdbc:mysql://localhost:3306/
       mydatabase" />
```

Lookup datasource in java code using JDBC

```
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.sql.DataSource;
import java.sql.Connection;
public class DatabaseUtil {
    public static Connection getConnection()
            throws exception {
        Context initContext = new InitialContext();
        DataSource ds = (DataSource) initContext.lookup("java:/comp/env/jdbc/myDB");
        Return ds.getConnection();
    }
}
```

Executing sql queries using JDBC Statement

① using a statement

```
try(Connection conn = DatabaseUtil.getConnection();
    Statement stmt = conn.createStatement()) {
```

```
String query = "select * FROM Users";
Result set rs = stmt.Executequery (query)
while (rs.next()){
System.out.println ("user ID:" + rs.getInt
    ("id") + " + Name :" + rs.getString
("name");
}
}
```

Output

Employee ID : Name : John
Employee ID2 : Name : Mary
Rows updated;

② using a prepared statment Employee Name : John

```
try (Connection) conn = database util.get
Connection(); prepared statment pstmt =
Conn.prepare statment
("SELECT * FROM users WHERE id = ?"))
{pstmt.setINT (1, 1);
Result set rs = pstmt.ExecutiQuery ();
while (rs.next()) {
System.out.println ("user ID:" + rs.getINT
("id") + ", Name:" + rs.getString
("name"));
}
}
```

using a callable statement for stored procedure:

```
try (Connection conn = database.util.
getConnection (callable statement
cstmt = conn.prepare call (call get user
By Id(?)}"")) {
  cstmt.setInt(1,1);
  ResultSet rs = cstmt.Execute query();
while (rs.next()){
  system.out.println("user ID": get Int
      +1", Name :" + rs.getsstring ("clame");
}
}
```

Output :

Statiment Example output.

user ID : 1, Name : Swaina

user ID : 2, Name : Radha

Prepared statiments

user ID : 1, Name : Swaina

callabel statiment

User ID : 1, Name : Swaina

④ PHP applications to Extract Data and store in XML

Steps:

① Read Content from a text file

② Extract patterns using Regular

③ Create & store Results in an XML file

④ Define XML schema

PHP Code

```
<?php
$filename = "input.txt";
$content = file_get_content ($filename);
preg_match_all ("/[a-zA-Z0-9.-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]/", $content, $matches);
$xml = new SimpleXMLElement ("<data></data>");
foreach ($matches[0] as $email) {
    $email Element -> addChild ("email", $email);
}
{
```

phone Element = $xml -> add child
("phones");
foreach ($phones[0] as $phone) {
$phone Element -> add child ("phone",
$phone)
->asXML ("output.xml");
"Data Extracted and saved to output

Output:

<data>
<E-mail:>
<E-mail> Example.1@Example.com
</E-mail>
<E-mail> Example.2@Example.com
</E-mail>
<phones>
<phone> +123-456-7890 </phone>
<phone> 987-654-8210 </phone>
</phone>
</data>

Assignment - 3

Rubrics	Split up	Marks Obtained	Total marks
Code implementation	8M		
Session Data accuracy	5M		
Efficiency and clarity	3M		
Explaination	4M		
Senario Explanation	6M		
function library Explanation	5M		
custom functions	5M		
Clarity & Org-anisation	4M		
Script functionality	8M		
User Interaction	5M		
Code Efficiency	4M		

Explaination	3M		
understanding	6M		
client code generation	6M		
Error handling	4M		
clarity & depth	4M		

→ Assignment - ④

Rubrics	split up	Marks obtained	Total
Explination of JOBS	5M		
Connection pooling	6M		
SQL queries	5M		
Statement types	4M		
Embarcing Java Code	5M		
Life cycle phase explaination	6M		

Advantages & disadvantages	5M	
clarity & depth	4M	
Code Implementation	8M	
HTML Table structure	5M	
Alternate colours logic	4M	
Explaination	3M	
Code Implementation	8M	
Pattern Extraction	5M	
XML file generation	4M	
DTD VS XML Schema Comparition.	3M	