

### Assignment - 3

- ① Implement a feature in a web application that tracks the number of accesses by a client with a single session using Java services using HttpSession object to manage and monitor 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.HttpServletResponse;
```

```
@WebServlet("/Session Tracker")
public class Session Tracker extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html");
        HttpSession session = request.getSession(true);
    }
}
```

```
long retainTime = session.getCreateTime();
```

```
long lastAccessedTime = session.getLastAccessedTime();
```

```
if (visitRecord == null) {
```

```
out.println("<html><body>"),  
out.println("<p> Version created  
" + new Date (creation time) +  
"</p>").
```

```
out.println("<p> no of access in  
this version: " + visit count + "</p>")
```

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

gathering history info through  
getters & setters, DAOs, program  
method calls

### Output:

Version Tracking example.

VERSION ID : 1245 ABDE

VERSION CREATED : MON SEP 09 12:00:00  
IST 2024

VERSION USED : MON SEP 09 12:01:05  
IST 2024

NO: OF ACCESS IN THIS VERSION : 1

- 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  
customer fu

### JSP code

```
<!@ taglib o  
com/jsp/1
```

```
<!@ tagl  
complint/fc
```

```
<html>  
<hea
```

```
<ht>c
```

```
<from /
```

```
<option
```

```
<option
```

```
<opti
```

```
</s
```

```
<input
```

library in JSTL and how to create  
custom functions.

### JSP code using JSTL:

```
<@ taglib uri="http://java.sun.com  
    com/jsp/jstl/core" prefix="c" />  
<@ taglib uri="http://java.sun.com  
    jstl/functions" prefix="fn" />  
<html>  
  <head>  
    <title>Order List</title>  
<form method="GET" action="Order.  
        jsp">  
  <option value="All">"All"</option>  
  <option value="Delivered">  
    Delivered</option>  
  <option value="Pending">  
    Pending</option>  
</select>  
<input type="submit" value="filter"/>  
</form>  
<table border="1">  
  <thead>  
    <tr>
```

<tr>  
</thead>  
</body>

<tr>  
<td> & {order.id} & <td>  
<td> & {order.date} & {td}>  
<td> & {order.status} & {td}>  
<td> & {order.amount} & {td}>

</tr>

<lc:when>

<lc:choose>

<lc:for choose>

  <for each>

<lt body>

</body>

& <table>

</body>

</html>

## Output:

Order ID	Date	Status
1002	2024-09-07	Pending
1003	2024-09-09	Pending

② creating customer function in JUML

```
<oglibxmlnr: "http://java.sun.com/xml/ns/javaee"  
version="2.1">
```

<hibVersion>1.0</hibVersion>

Output:

Custom function example

Original : Hello World

Reversed : dleW olleH

③ To implement the described functionality  
for refreshing stock market annals.  
Pages few min. with a confirmation  
dialog appearing 20 seconds before  
refresh.

Java Script code Script:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

```
    <meta charset="UTF-8">
```

```
    <title>Stock market annals
```

```
    <title>
```

```
    <script>
```

function refreshPage() {

```
location.reload();  
}  
SetTimeout(0) = {  
    var confirmation = confirm  
    ("The page will refresh")  
    if (!confirmation.refresh)  
        alert("page refresh canceled")  
    else  
        alert("page refresh canceled")  
        location.reload();  
        setTimeout(function () {  
            location.reload();  
        }, 20000);  
    }  
</script>  
<body>  
    <h1>Stock Market Quotes</h1>  
    </body>  
</html>
```

Output:

Page Display:

Stock Market Quotes.

- Apple Inc (AAPL): \$150.00
- Microsoft Corp (MSFT): \$280.00
- Alphabet Inc (GOOGL), \$2800

Confirmation Dialog:

The page will refresh in 20

Seconds.

Alert

page

OK

Page Re

(The Pa  
cont'd)

- APPLE

- Micro

- Alph

④ To integra  
service is  
using a

① Genera  
W3Impo  
exam

② Inte

app

→ incl

→ cor

③ In

OK

cancel

reject

page

refresh

cancelled

OK

Page Refresh

(The page reloads, display updated  
content)

- 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.

WSDLImport -keep -src -d bin p-com-  
example-payment -verbose http://

② Integrate generated code into application.

→ include generated code

→ configure Service endpoint

③ Invoke the payment

## Output:

Successful payment

Payment successful

Payment failure

Payment failed due to  
fault

SOAP fault

Payment failed due to

SOAP fault

①

Why JDB  
Database  
configu

L Resou

## Assignment - 9

① Why JDBC is essential in building Database - Driver applications

Configure Data Source :-

```
<Resource name = "jdbc/MyDB"
    auth = "Container"
    type = "java.sql.DataSource"
    Max Total = "20"
    Max Idle = "10"
    username = "dbuser"
    driver class name = "com.mysql.jdbc.Driver"
```

Lookup datasource in Java code using JDBC

```
import javax.naming.Context;
import javax.naming.InitialContext;
import java.sql.DataSource
import java.sql.Connection
public class DatabaseUtil {
    Context initContext = new InitialContext()
    Connection db = (Connection) lookup("java/com/
        myn/jdbc/MyDB");
```

## P- Transpiration

executing SQL queries using  
JDBC statement

- ① using a statement:

```
try{connection conn = DatabaseUtil.
```

```
getConnection();
```

```
Statement st = conn.createStatement();
```

```
ResultSet rs;
```

```
while(rs.next()) {
```

```
System.out.println("Cursor ID "+
```

```
"rs.getInt("
```

```
cursorID") + " Name" + rs.getString("name"));
```

②

- using a callable statement for  
stored procedure.

```
try{connection conn = DatabaseUtil.
```

```
Util.getConnectionCallable();
```

```
Statement st = conn.createStatement();
```

```
while(st.setInt(1, 1);
```

```
System.out.println("Cursor  
ID" + rs.getInt("cursorID"));
```

Output

page

or

Prepa

v

calla

v

②

Lifecyc

① Tra

② C

③

④

⑤

embed

<P>

Outpu

### Output

Statement example Output:

User ID:1, Name:Vietnam

User ID:2, Name:Oman

Prepared Statement

User ID:1, Name:Vietnam

Callable Statement

User ID:, Name:Vietnam

### ② Lifecycle phases of JSP Page:

① Translation phase

② compilation phase

③ Initialization phase

④ Request processing phase

⑤ Destruction phase

### Embedding Java code in JSP

<P> The sum is <% =sum%>

Output: The sum is 1.5

### Scriptlets:

Advantages: easy to use for embedding simple Java logic

Disadvantages: Leads to messy code, difficult to maintain.

### Declarations:

Advantages: useful to easy use for embedding simple Java logic

Disadvantages: can clutter JSP with Java code, leading to poor separation of concerns.

### Expressions:

Advantages: simplifies outputting dynamic content directly in JSP

Disadvantages: limited to expressions

### PHP code:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Hello World</title>
```

PHP code:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <style>
      table {
        width: 30px;
        height: 30px;
      }
    </style>
  </head>
  <body>
    <table>
      <tr>
        <td>
          <?php
            <?LOOP for $row
              for ($row2 = 0; $row2 < 8; $row2++) {
                <? echo "<td>";
                for ($col = 0; $col < 8; $col++) {
                  <? echo "<td>";
                }
              }
            </?>
          </td>
        </td>
      </tr>
    </table>
  </body>
</html>
```

Output:

[ ] [H] [ ] [H] [ ] [H] [ ] [H]  
[ ] [H] [ ] [H] [ ] [H] [ ] [H]  
[ ] [H] [ ] [H] [ ] [H] [ ] [H]  
[ ] [H] [ ] [H] [ ] [H] [ ] [H]  
[ ] [H] [ ] [H] [ ] [H] [ ] [H]  
[ ] [H] [ ] [H] [ ] [H] [ ] [H]  
[ ] [H] [ ] [H] [ ] [H] [ ] [H]  
[ ] [H] [ ] [H] [ ] [H] [ ] [H]  
[ ] [H] [ ] [H] [ ] [H] [ ] [H]

④

You are developing a PHP application that reads content from a text file and uses regular expression to extract specific pattern, such as email id & phone num. Describe the steps & provide code for the application.

<?PHP

\$text\_file\_path = "input.txt";

\$xml\_file\_path = "output.xml";

```
$ textContent = "[a-zA-Z0-9-/-]+"]  
+ @ [a-zA-Z0-9-+JP]  
[a-zA-Z]{
```

\$ XML-new

```
$ implements Element<'<data'>);  
Simple XML element<'<data'>);
```

echo "Data extracted and  
Saved successfully;

Output:

Email: p@email.com

phone no: 1234567890

### Assignment - 3

| Rubrics                  | Split up | Mark | Total Marks |
|--------------------------|----------|------|-------------|
| code implementation      | 8M       |      |             |
| session data accuracy    | 5M       |      |             |
| efficiency & clarity     | 3M       |      |             |
| explanation              | 4M       |      |             |
| selection explain        | 6M       |      |             |
| function library         | 5M       |      |             |
| custom functions         | 5M       |      |             |
| clarity and organization | 4M       |      |             |
| script functionality     | 8M       |      |             |
| User Interface           | 5M       |      |             |
| code efficiency          | 4M       |      |             |
| Explanation              | 3M       |      |             |

understanding

client code generation  
error handling  
clarity & depth

Assignment

Rubrics  
Explanation  
of JDBC

connection pooling

SQL  
Query  
Statement  
types

embedding  
Java

lifecycle  
Explanation

Advantages  
disadvantages

clarity

code implementation

Alternating colors

| understanding              | 6M     |                |             |
|----------------------------|--------|----------------|-------------|
| client code generation     | 6M     |                |             |
| error handling             | 4M     |                |             |
| clarity & depth.           | 4M     |                |             |
| Assignment + 4             |        |                |             |
| Rubric                     | Slipup | Marks Obtained | Total Marks |
| Explanation of JDBC        | 5M     |                |             |
| connection pooling         | 6M     |                |             |
| SQL Queries                | 5M     |                |             |
| Statement types            | 4M     |                |             |
| embedding Java             | 5M     |                |             |
| lifecycle explanation      | 6M     |                |             |
| Advantages & disadvantages | 5M     |                |             |
| Clarity                    | 4M     |                |             |
| code implement             | 8M     |                |             |
| Alternate colour           | 4M     |                |             |

|                       |    |
|-----------------------|----|
| Explanation           | 3M |
| code implementation   | 8M |
| Pattern extraction    | 5M |
| XML file Generation   | 4M |
| XMI Schema Comparison | 3M |

|       |     |
|-------|-----|
| total | 22M |
| M1    | 1M  |
| M2    | 1M  |
| M3    | 1M  |
| M4    | 1M  |
| M5    | 1M  |
| M6    | 1M  |
| M7    | 1M  |
| M8    | 1M  |
| M9    | 1M  |
| M10   | 1M  |
| M11   | 1M  |
| M12   | 1M  |
| M13   | 1M  |
| M14   | 1M  |
| M15   | 1M  |
| M16   | 1M  |
| M17   | 1M  |
| M18   | 1M  |