



**UNIVERSITI MALAYSIA TERENGGANU**

---

**CSM 3023 WEB BASED APPLICATION DEVELOPMENT (K1)**

**LAB REPORT 1**

**SEMESTER II 2023/2024**

---

**Prepared for:**

DR. MOHAMAD NOR BIN HASSAN

SIR MOHD ARIZAL SHAMSUL BIN MAT RIFIN

**Prepared by:**

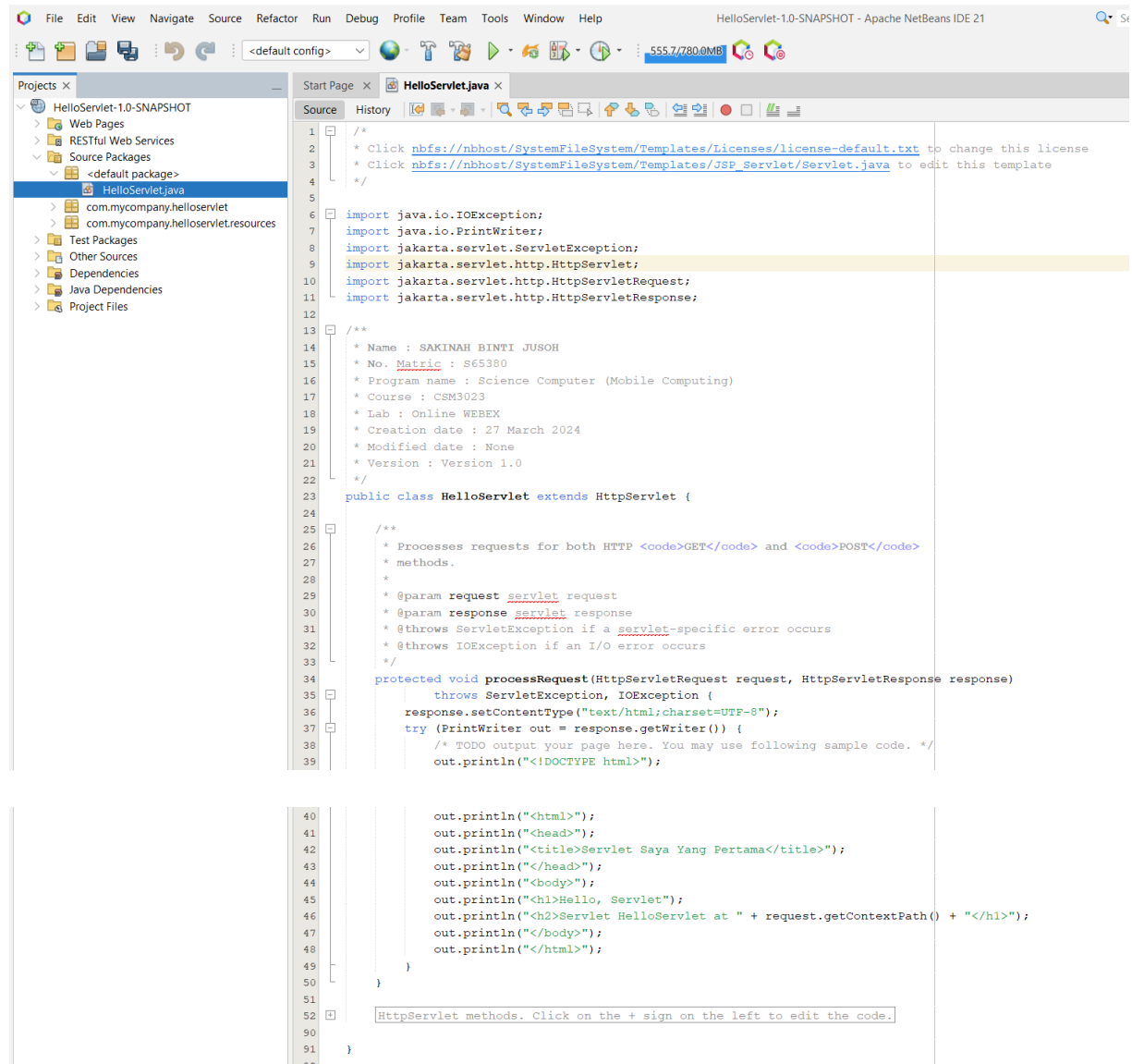
SAKINAH BINTI JUSOH (S65380)

## Task 5: Linking Netbeans to Apache Tomcat and Writing a Simple Java Servlet

Objective: Link Netbeans to Apache Tomcat and Writing a Simple Java Servlet

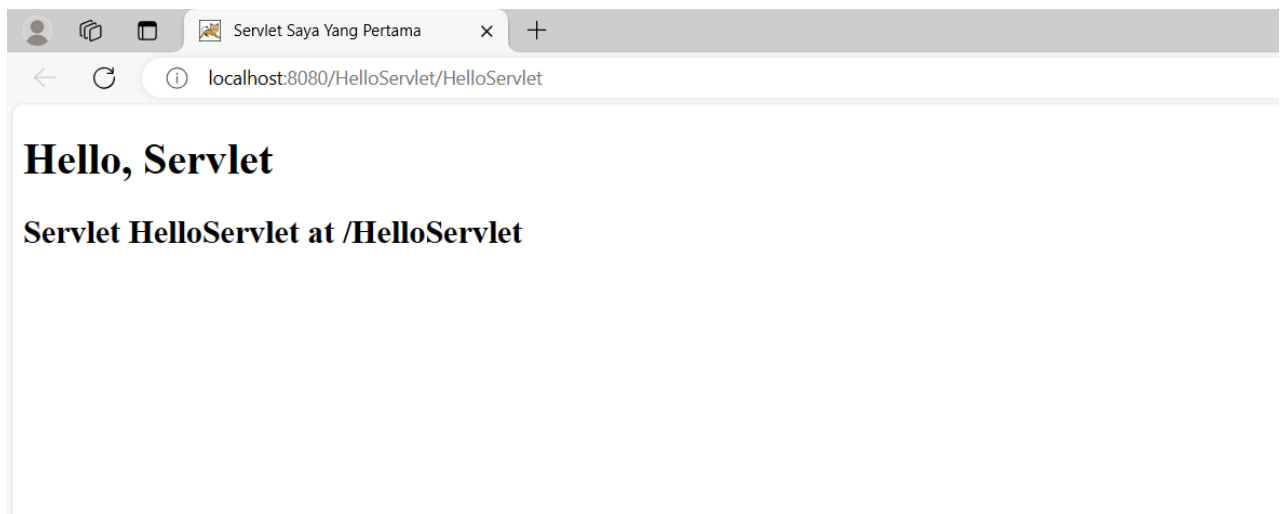
Problem description: To write simple Java Servlet

### Source code

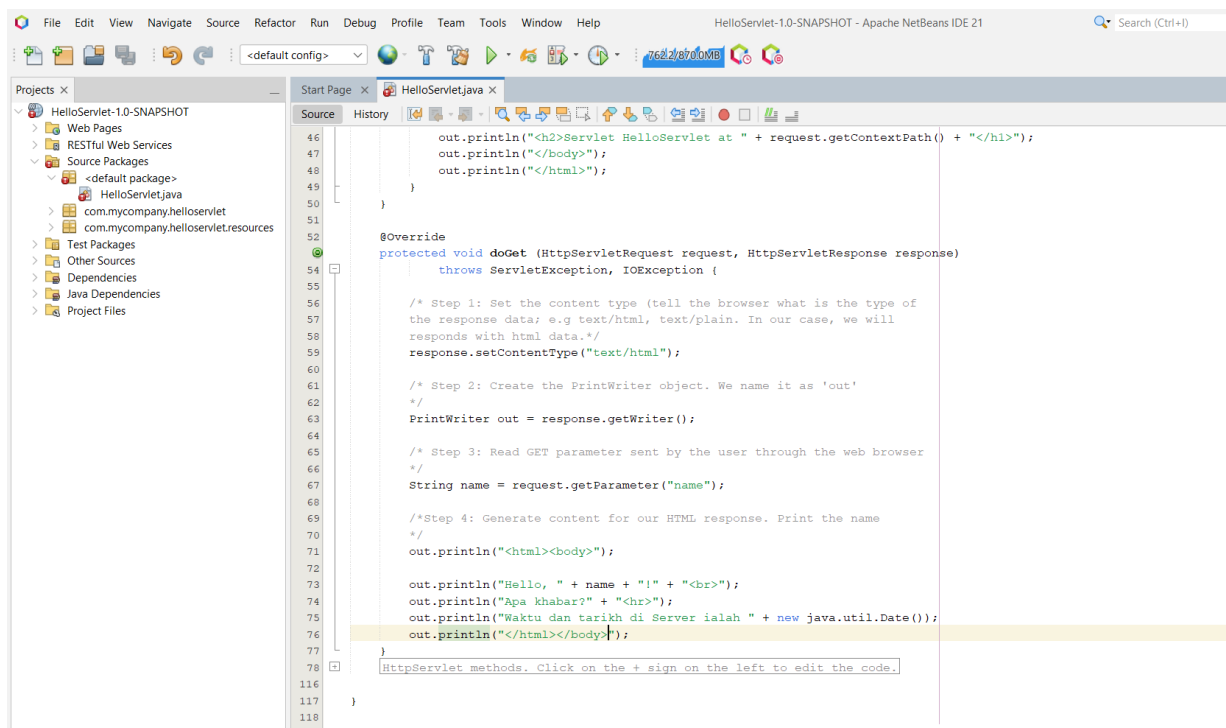


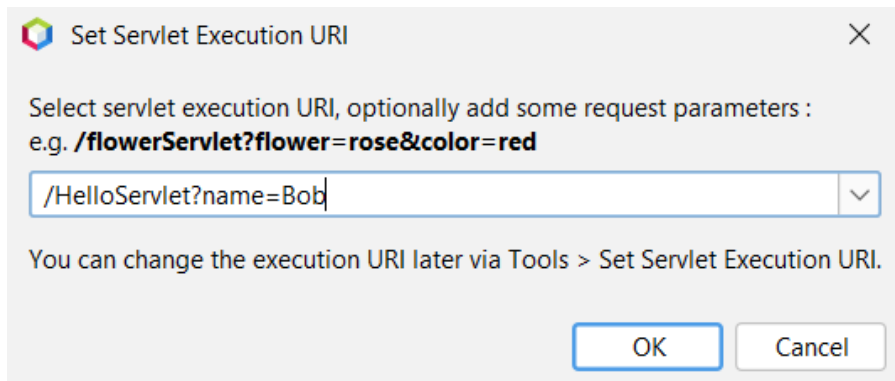
```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/JSP\_Servlet/Servlet.java to edit this template
4  */
5
6  import java.io.IOException;
7  import java.io.PrintWriter;
8  import jakarta.servlet.ServletException;
9  import jakarta.servlet.http.HttpServlet;
10 import jakarta.servlet.http.HttpServletRequest;
11 import jakarta.servlet.http.HttpServletResponse;
12
13 /**
14  * Name : SAKINAH BINTI JUSOH
15  * No. Matric : S65380
16  * Program name : Science Computer (Mobile Computing)
17  * Course : CSM3023
18  * Lab : Online WEBEX
19  * Creation date : 27 March 2024
20  * Modified date : None
21  * Version : Version 1.0
22  */
23 public class HelloServlet extends HttpServlet {
24
25     /**
26      * Processes requests for both HTTP GET and POST
27      * methods.
28      *
29      * @param request servlet request
30      * @param response servlet response
31      * @throws ServletException if a servlet-specific error occurs
32      * @throws IOException if an I/O error occurs
33      */
34     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
35         throws ServletException, IOException {
36         response.setContentType("text/html;charset=UTF-8");
37         try (PrintWriter out = response.getWriter()) {
38             /* TODO output your page here. You may use following sample code. */
39             out.println("<!DOCTYPE html>");
40
41             out.println("<html>");
42             out.println("<head>");
43             out.println("<title>Servlet Saya Yang Pertama</title>");
44             out.println("</head>");
45             out.println("<body>");
46             out.println("<h1>Hello, Servlet");
47             out.println("<h2>Servlet HelloServlet at " + request.getContextPath() + "</h1>");
48             out.println("</body>");
49             out.println("</html>");
50         }
51     }
52
53     HttpServlet methods. Click on the + sign on the left to edit the code.
54
55 }
```

## Output

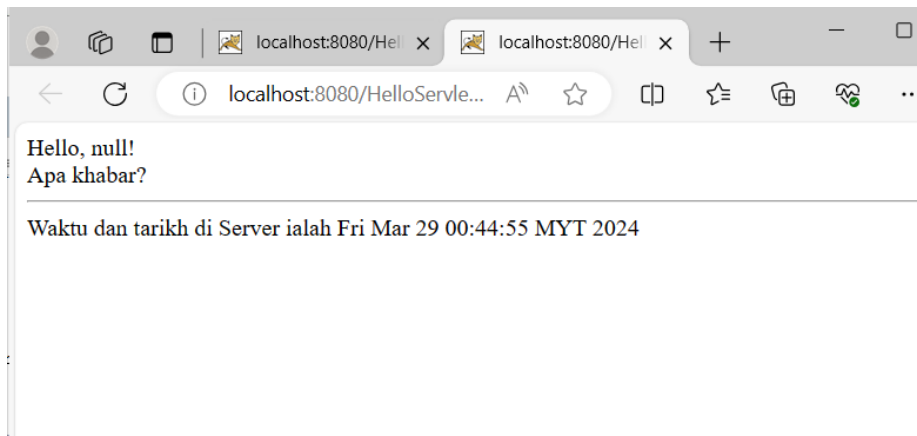
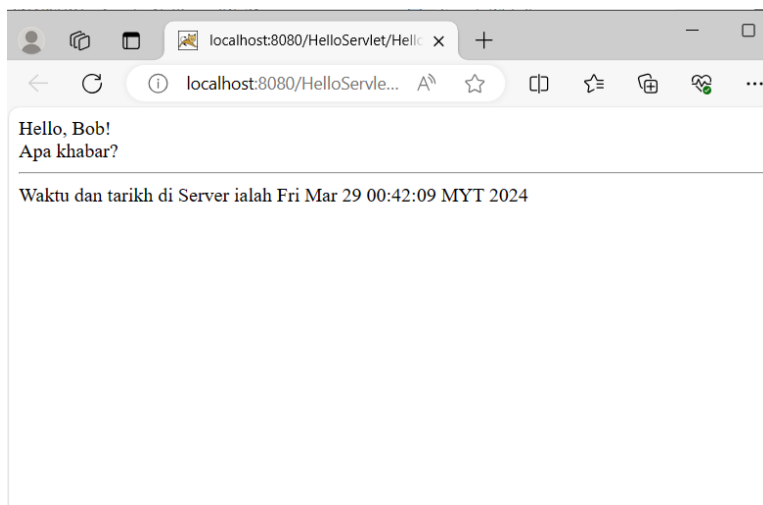


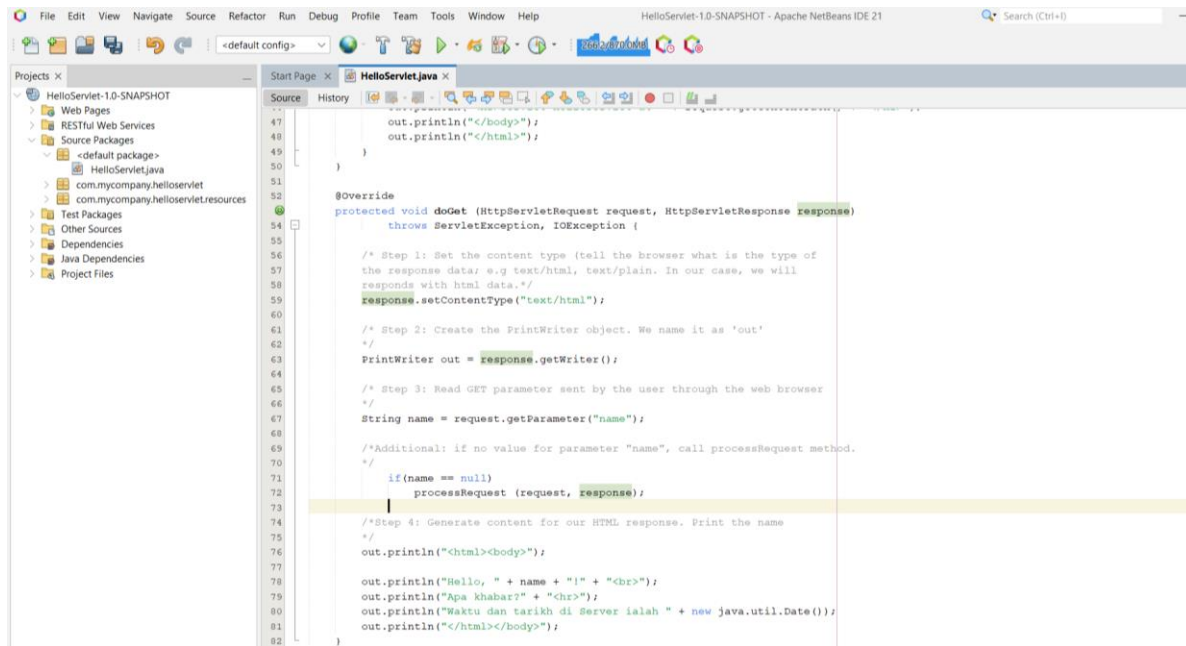
## Source code



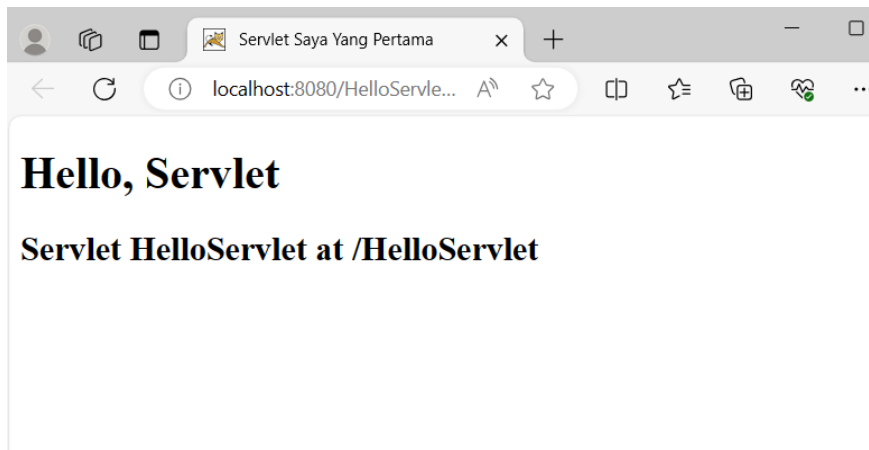


## Output





```
1..
47 out.println("</body>");
48 out.println("</html>");
49 }
50
51
52
53
54 @Override
55 protected void doGet (HttpServletRequest request, HttpServletResponse response)
56     throws ServletException, IOException {
57
58     /* Step 1: Set the content type (tell the browser what is the type of
59     the response data; e.g text/html, text/plain. In our case, we will
60     responds with html data.*/
61     response.setContentType("text/html");
62
63     /* Step 2: Create the PrintWriter object. We name it as 'out'
64     */
65     PrintWriter out = response.getWriter();
66
67     /* Step 3: Read GET parameter sent by the user through the web browser
68     */
69     String name = request.getParameter("name");
70
71     /*Additional: if no value for parameter "name", call processRequest method.
72     */
73     if(name == null)
74         processRequest (request, response);
75
76     /*Step 4: Generate content for our HTML response. Print the name
77     */
78     out.println("<html><body>");
79
80     out.println("Hello, " + name + "!" + "<br>");
81     out.println("Apa khabar?" + "<br>");
82     out.println("Waktu dan tarikh di Server ialah " + new java.util.Date());
83     out.println("</html></body>");
84 }
```

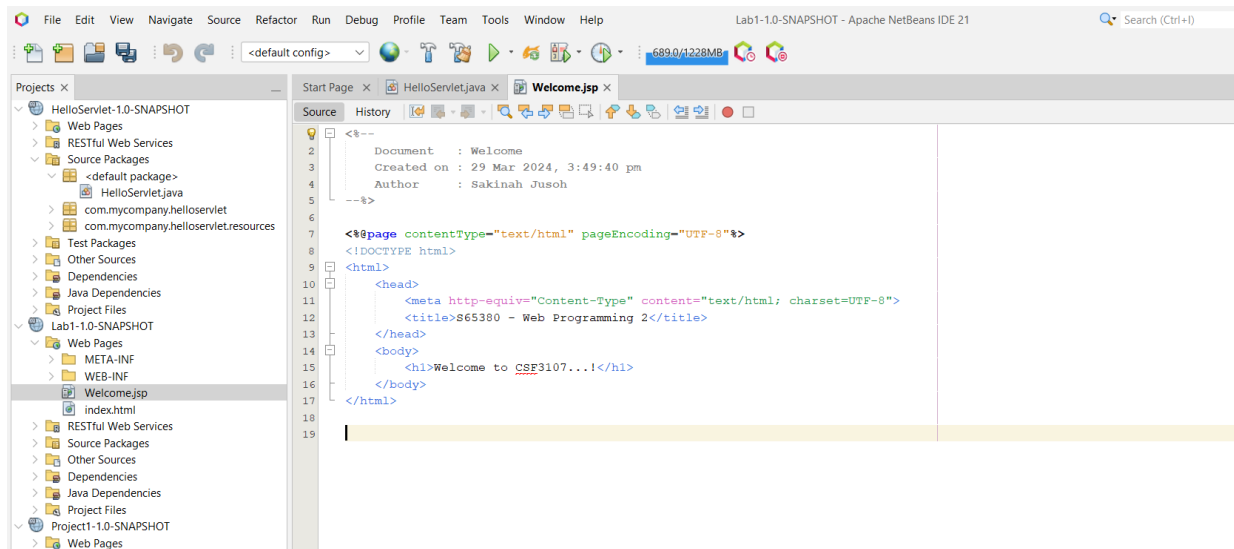


## Task 6: Writing a simple JSP Program

Objective: Writing a simple plain JSP Program

Problem description: Write a simple plain JSP Program to display “Welcome to [MATRICNUMBER]..!”

## Source code



```
<!--
2  Document   : Welcome
3  Created on : 29 Mar 2024, 3:49:40 pm
4  Author    : Sakinah Jusoh
5  -->
6
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>
8  <DOCTYPE html>
9  <html>
10 <head>
11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 <title>S65380 - Web Programming 2</title>
13 </head>
14 <body>
15 <h1>Welcome to CSF3107...!</h1>
16 </body>
17 </html>
18
19
```

## Output



## Reflection

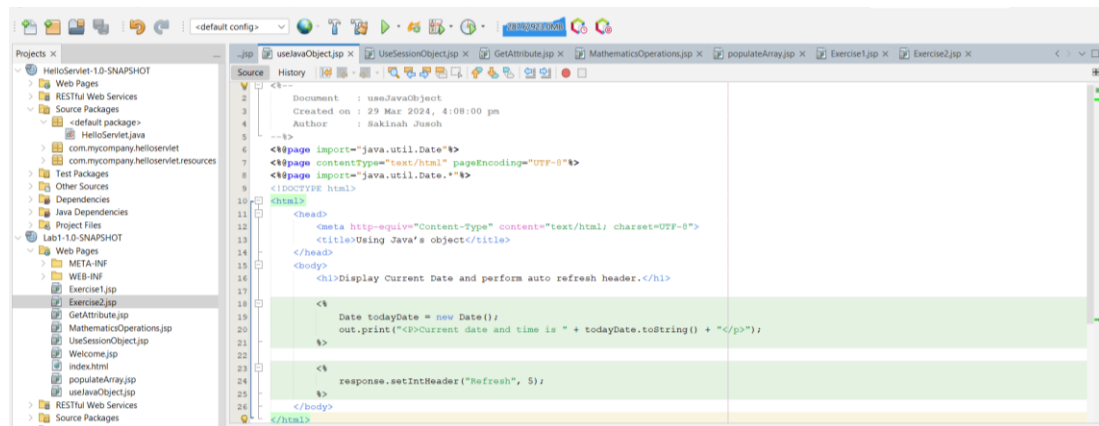
1. What have you learned from this exercise?  
**I have learnt about how to create and run JSP file in NetBeans IDE.**
2. Explain the general concept of how the JSP's file work?  
**A JSP container works with the web server to provide the runtime environment and other services a JSP needs.**
3. Based on your observation of the previous tasks (Task 3 and Task 4), what are the different you can find between servlet and JSP?  
**JSP is easier to code than servlet.**

## Task 7: Use Java Reference Datatype/Class Wrapper in JSP

Objective: Using Java's object in JSP page.

Problem: Display the current date, perform auto refresh header in JSP's page

Source code

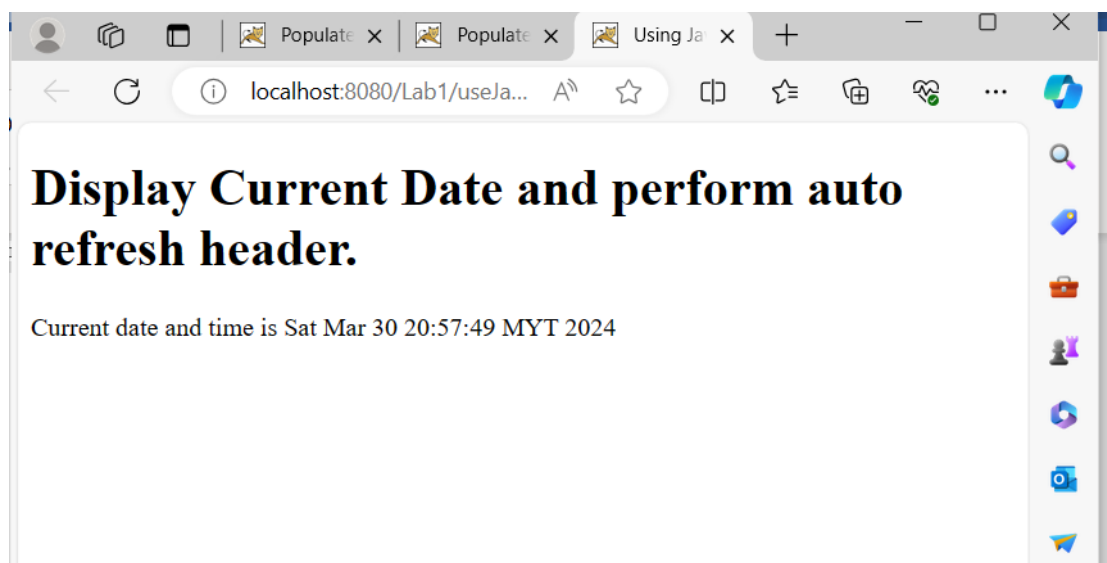


```
<!--
Document : useJavaObject.jsp
Created on : 29 Mar 2024, 4:08:00 pm
Author : Sakinah Jusoh
-->
<%page import="java.util.Date"%>
<%page contentType="text/html" pageEncoding="UTF-8"%>
<%page import="java.util.Date"%>
<DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Using Java's object</title>
</head>
<body>
<h1>Display Current Date and perform auto refresh header.</h1>

<%
Date todayDate = new Date();
out.print("<P>Current date and time is " + todayDate.toString() + "</p>");
%>

<%
response.setHeader("Refresh", 5);
%>
</body>
</html>
```

output



Reflection

1. What have you learnt from this exercise?

**I have learnt about how to call method date.**

2. What is Java Scriptlet?

**A scriptlet is a piece of software code that is used by a native web page scripting language to perform a specific function or process. Scriptlets are primarily**

implemented in JavaServer Pages (JSP) and include variables, expressions or statements that are used only when requested by a certain client or process.

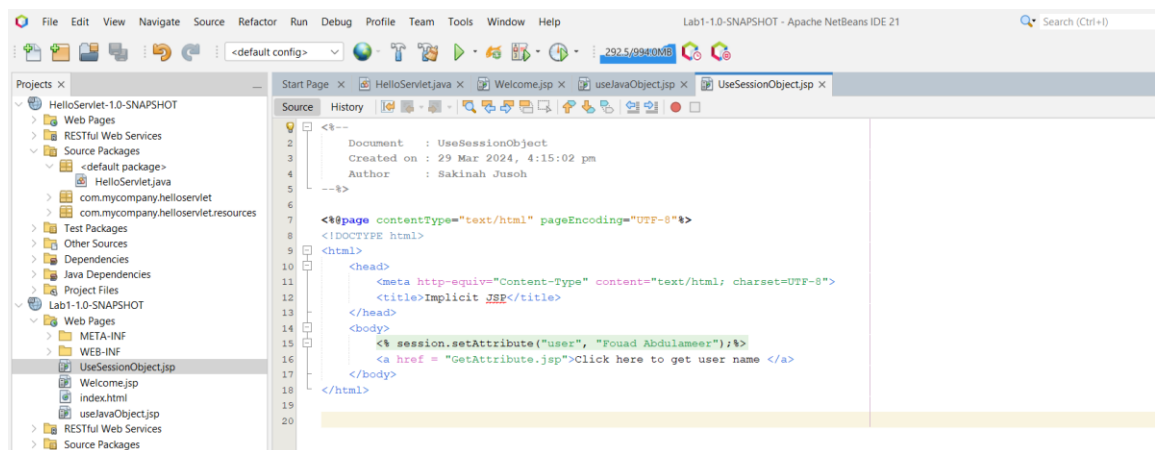
3. How to use Java code in your JSP's page?  
**Use tags and put the Java code in between the tags.**

### Task 8: Using JSP Implicit object in JSP page.

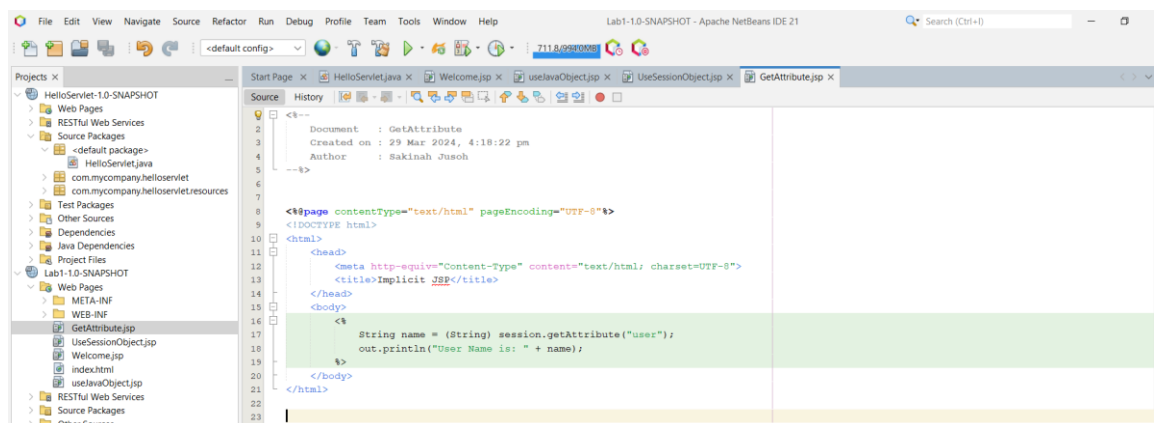
Objective: Using JSP Implicit object (session) in JSP page.

Problem description: Using Session object, perform simple Mathematics operations in JSP's page.

### Source code



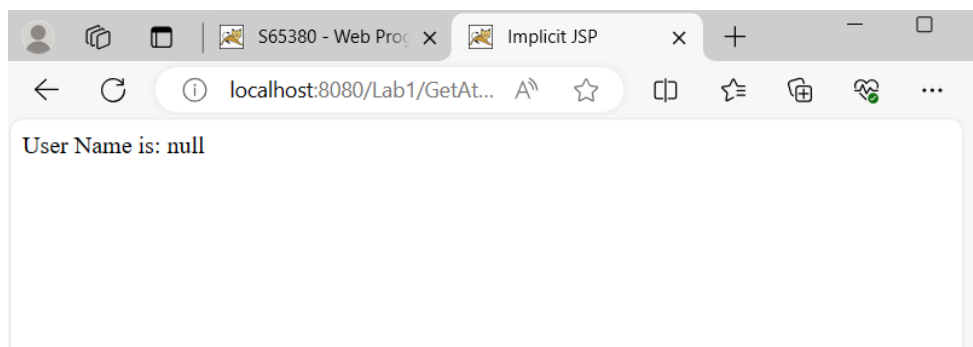
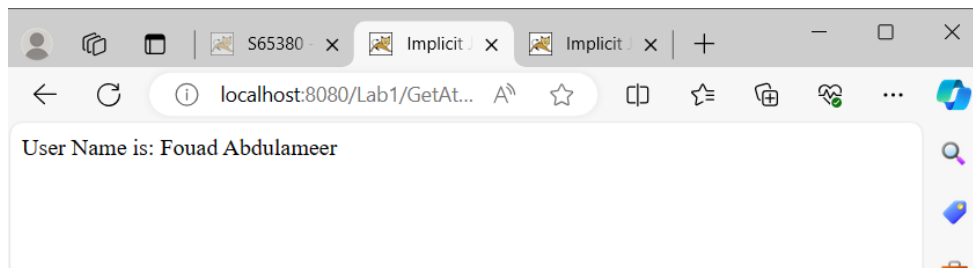
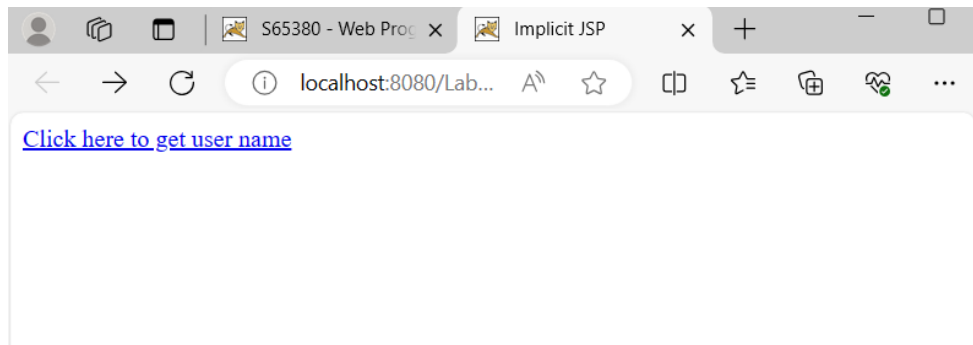
```
<!--
2  Document : UseSessionObject
3  Created on : 29 Mar 2024, 4:15:02 pm
4  Author : Sakinah Jusoh
5  -->
6
7  <@page contentType="text/html" pageEncoding="UTF-8"%>
8  <!DOCTYPE html>
9  <html>
10 <head>
11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 <title>Implicit JSP</title>
13 </head>
14 <body>
15 <% session.setAttribute("user", "Fouad Abdulameer");%>
16 <a href = "GetAttribute.jsp">Click here to get user name </a>
17 </body>
18 </html>
19
20
```



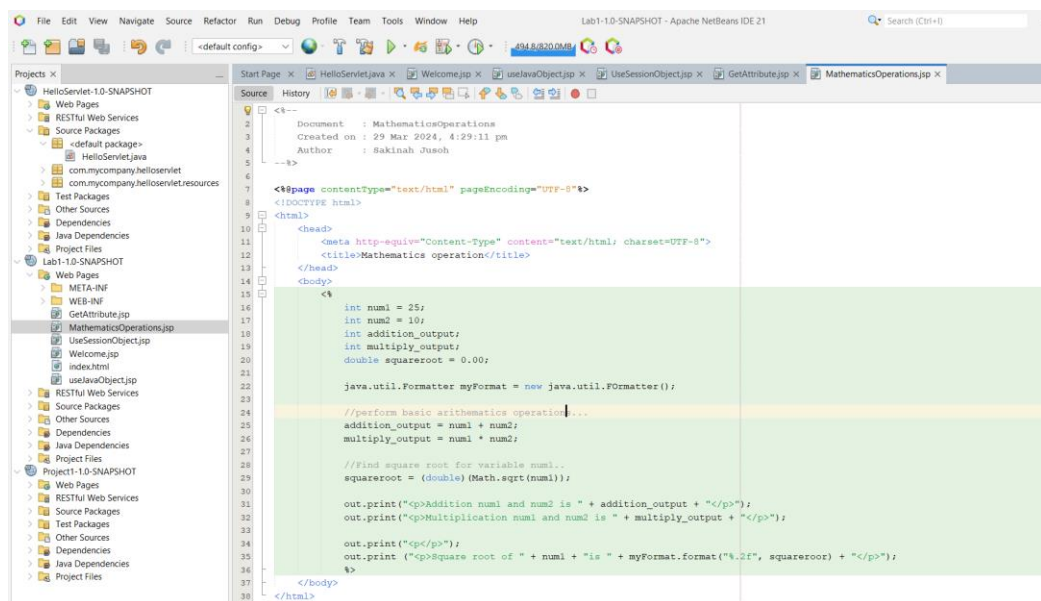
```
<!--
2  Document : GetAttribute
3  Created on : 29 Mar 2024, 4:18:22 pm
4  Author : Sakinah Jusoh
5  -->
6
7  <@page contentType="text/html" pageEncoding="UTF-8"%>
8  <!DOCTYPE html>
9  <html>
10 <head>
11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 <title>Implicit JSP</title>
13 </head>
14 <body>
15 <%
16 String name = (String) session.getAttribute("user");
17 out.println("User Name is: " + name);
18 %>
19 </body>
20 </html>
21
22
23
```



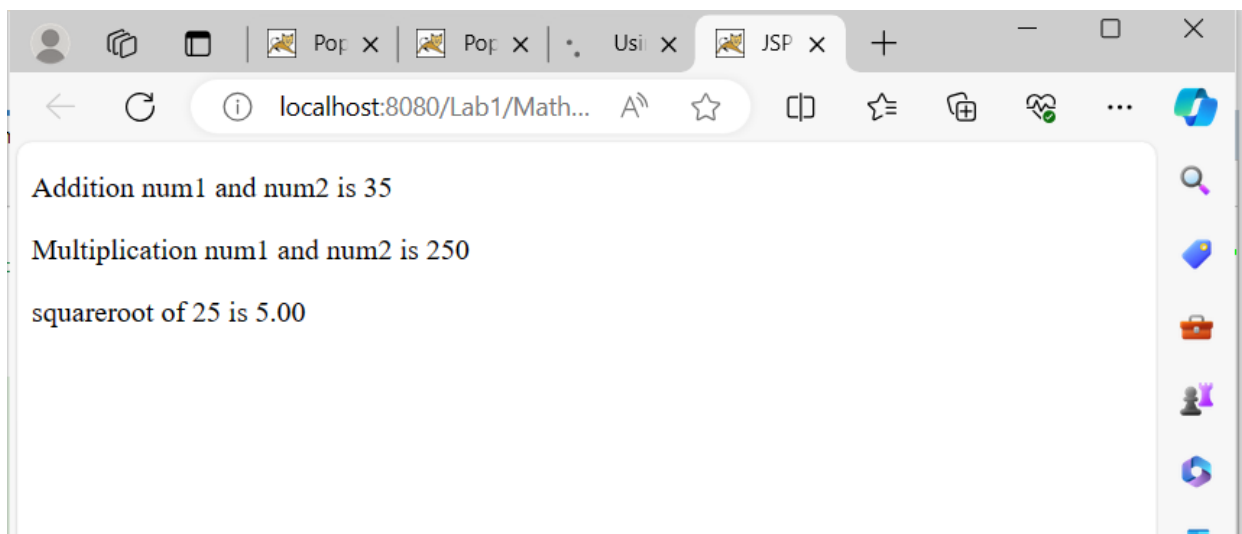
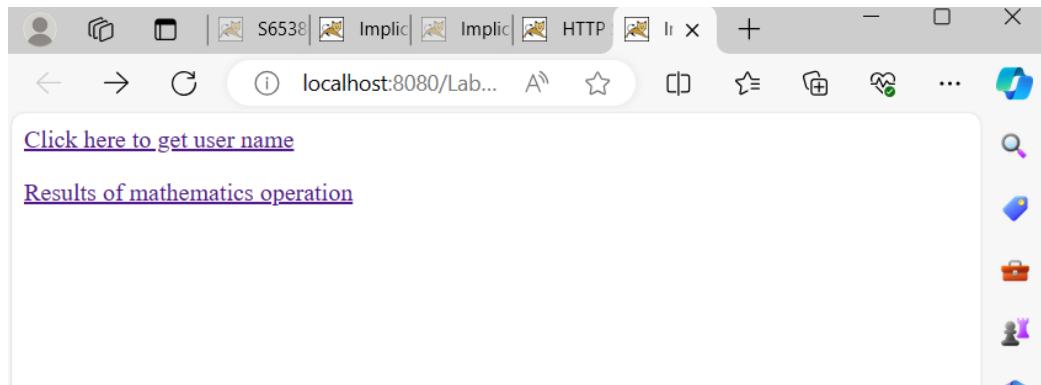
## Output



## Source code



## Output



## Reflection

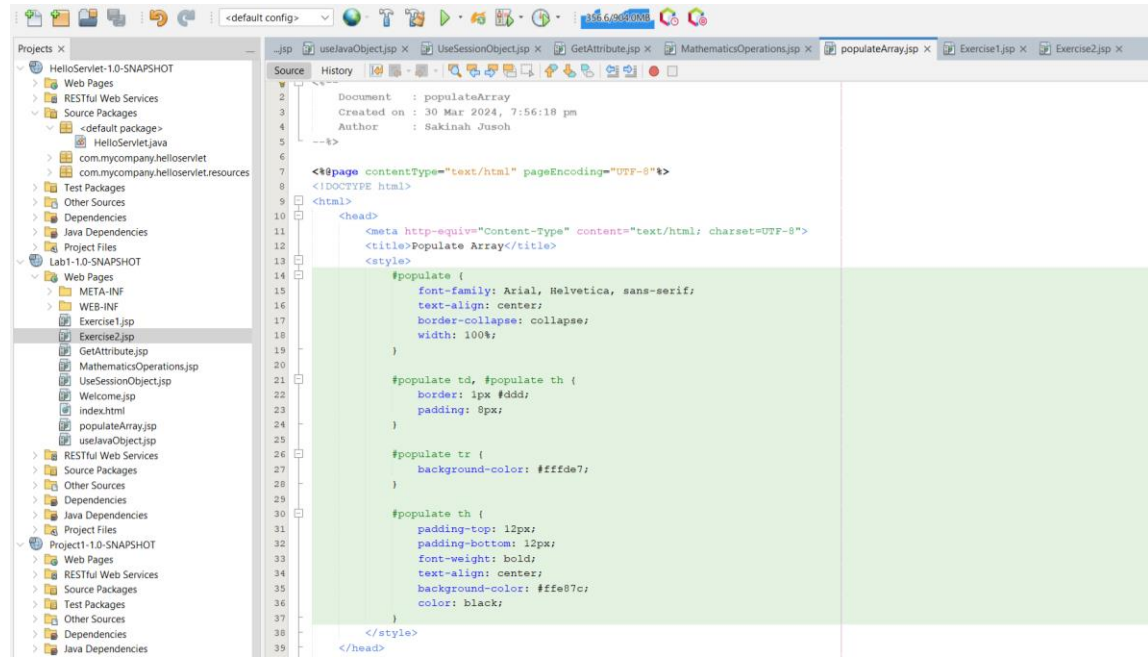
1. How do you want to submit specific information from one form to next form?  
**Use `getAttribute` method and `request.getParameter`.**
2. What happened if the field name you specify in `request.getParameter` ("field\_name") in the second page is different from the field name you defined in the first page?  
**Error because the parameter cannot be passed.**

## Task 9: Populate array values into HTML's Table

Objective: Read Java array and populate it into HTML's Table

Problem description: i. Create a 2D array that store sales data  
ii. Then, read an array and populate into HTML's table.

### Source code



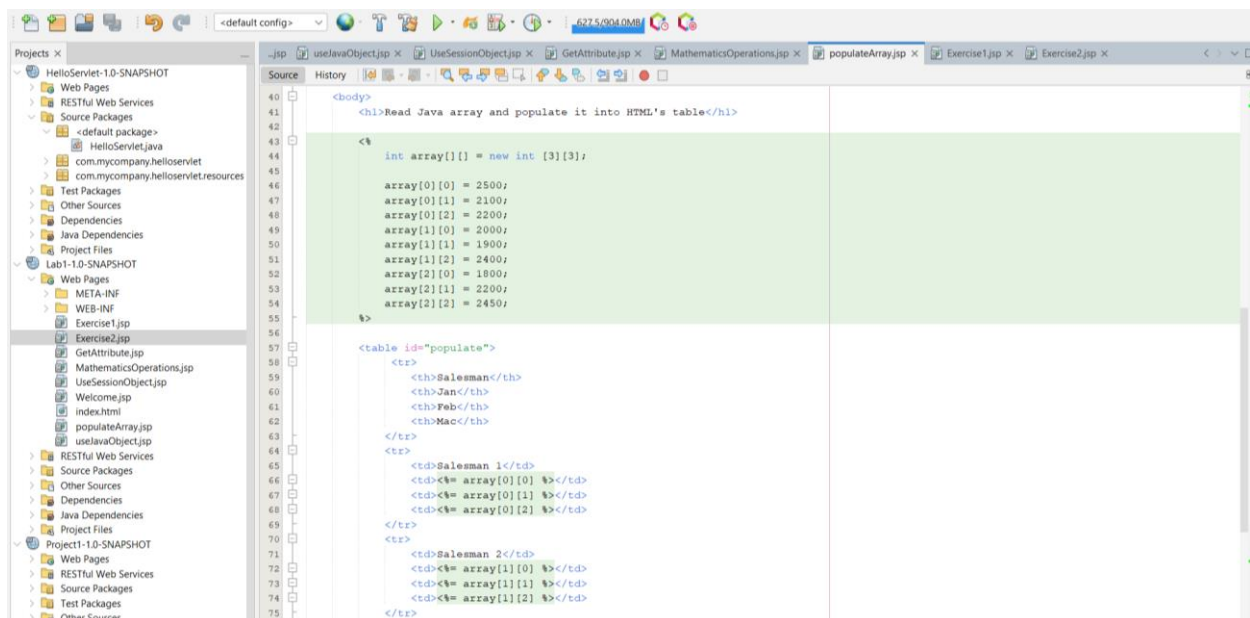
```
Document : populateArray
Created on : 30 Mar 2024, 7:56:18 pm
Author : Sakinah Jusoh

<!-->
<!--page contentType="text/html" pageEncoding="UTF-8">
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Populate Array</title>
<style>
#populate {
font-family: Arial, Helvetica, sans-serif;
text-align: center;
border-collapse: collapse;
width: 100%;
}

#populate td, #populate th {
border: 1px #ddd;
padding: 8px;
}

#populate tr {
background-color: #ffde7;
}

#populate th {
padding-top: 12px;
padding-bottom: 12px;
font-weight: bold;
text-align: center;
background-color: #ffe87c;
color: black;
}
</style>
</head>
```

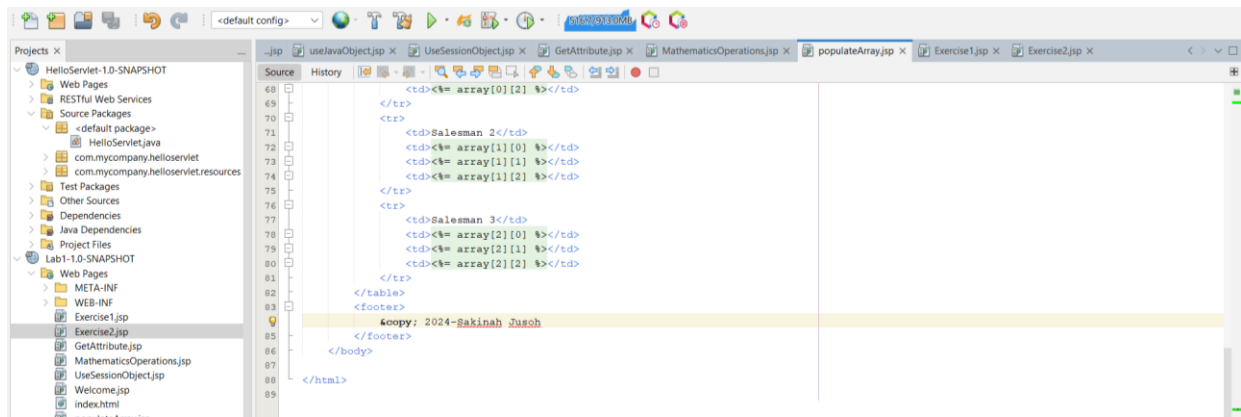


```
<body>
<h1>Read Java array and populate it into HTML's table</h1>

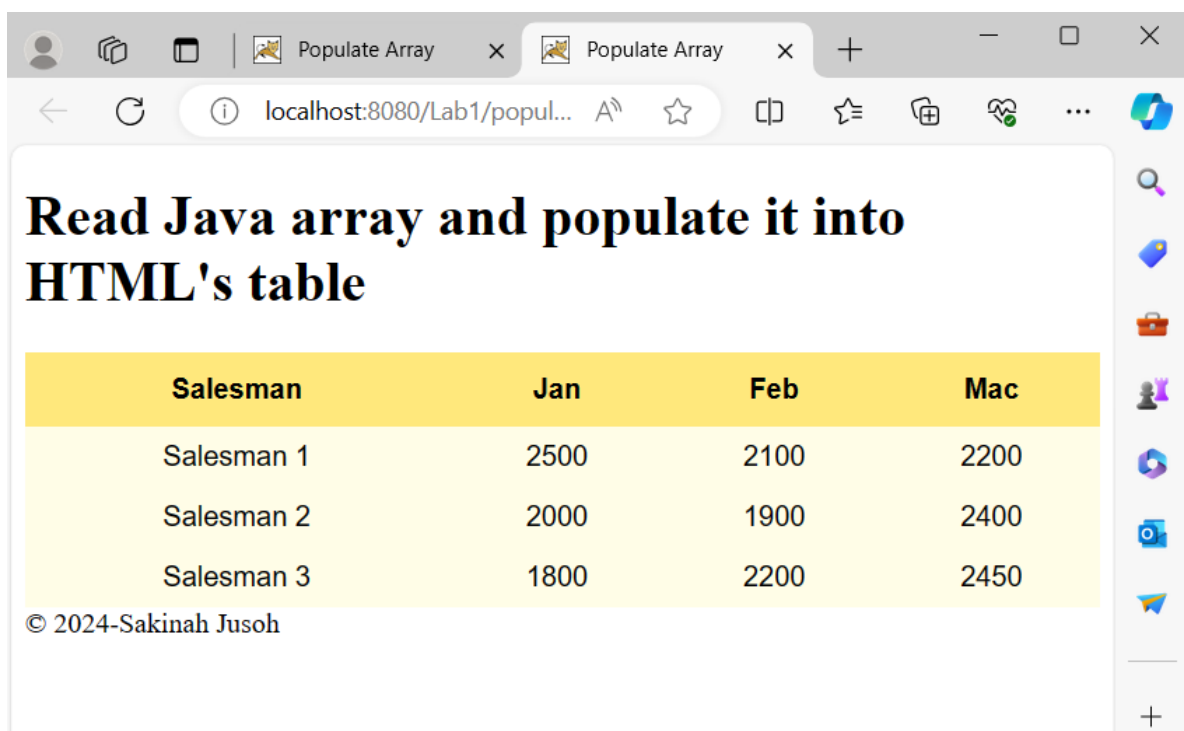
<!--
int array[][] = new int [3][3];

array[0][0] = 2500;
array[0][1] = 2100;
array[0][2] = 2200;
array[1][0] = 2000;
array[1][1] = 1900;
array[1][2] = 2400;
array[2][0] = 1800;
array[2][1] = 2200;
array[2][2] = 2450;
-->

<table id="populate">
<tr>
<th>Salesman</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
</tr>
<tr>
<td>Salesman 1</td>
<td>array[0][0] </td>
<td>array[0][1] </td>
<td>array[0][2] </td>
</tr>
<tr>
<td>Salesman 2</td>
<td>array[1][0] </td>
<td>array[1][1] </td>
<td>array[1][2] </td>
</tr>
</table>
```



## Output



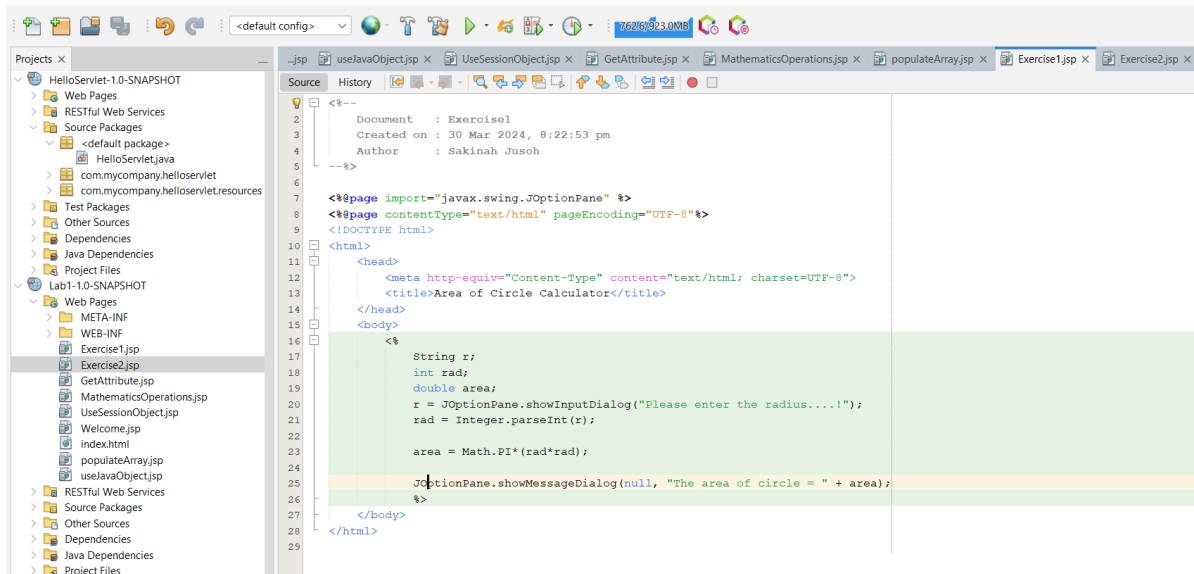
## Reflection

- Write a sample syntax to declare 2D java array.  
**int array [] [] = new int [3][3];**
- Define a sequence of steps on how you accomplish Task 7.
  - Add the java util package for Date
  - Use data type to create Date's object by code **Date todayDate = new Date();**. Then use the out.print command to display the date and time.
  - Set the refresh and sutoload it in 5 seconds
- What is the difference between HTML's page and JSP's page?

- a. HTML can only build static web pages but JSP can generate dynamic web pages.
- b. JSP must communicate with the Web Server, HTML can load faster.
- c. HTML is a client-side scripting language whereas JSP is a server-side scripting language.

## Exercise 1

### Source code



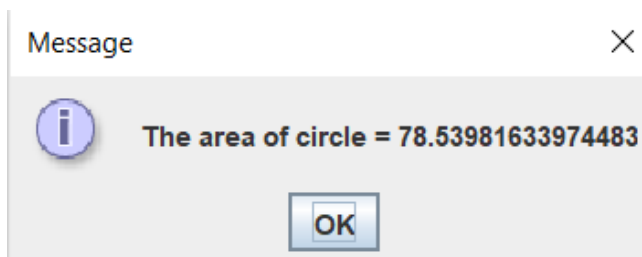
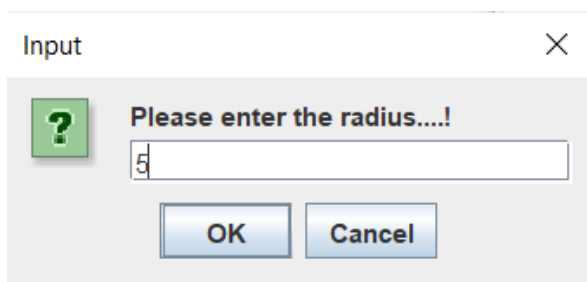
```
<!--
Document : Exercisel
Created on : 30 Mar 2024, 8:22:53 pm
Author : Sakinah Jusoh
-->

<%page import="javax.swing.JOptionPane" %>
<%page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Area of Circle Calculator</title>
</head>
<body>
<%
String r;
int rad;
double area;
r = JOptionPane.showInputDialog("Please enter the radius....!");
rad = Integer.parseInt(r);

area = Math.PI*(rad*rad);

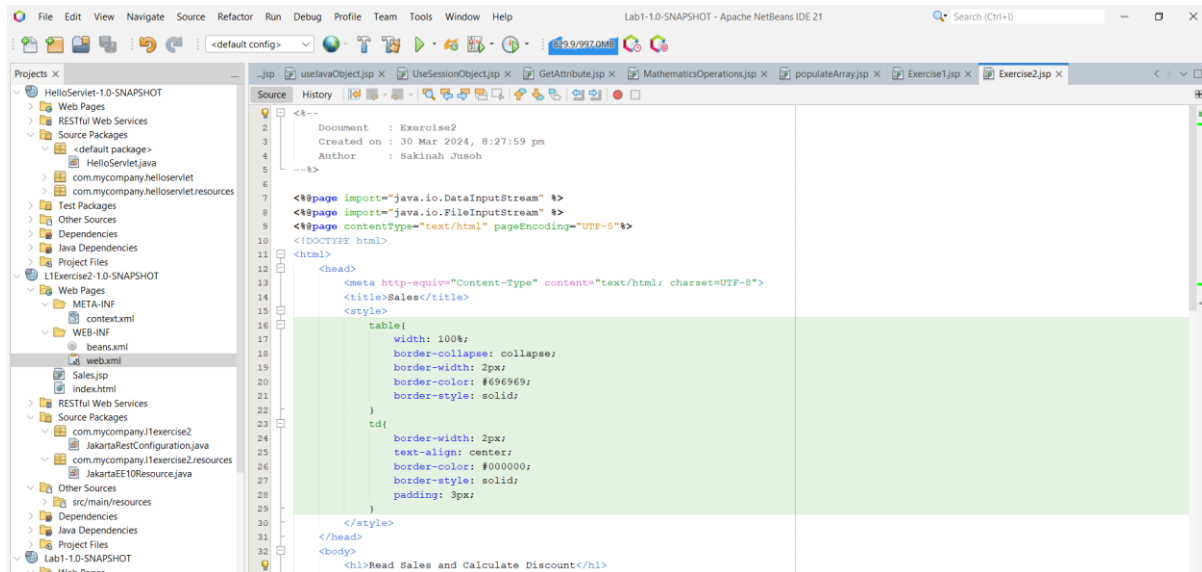
JOptionPane.showMessageDialog(null, "The area of circle = " + area);
%>
</body>
</html>
```

### Output



## Exercise 2

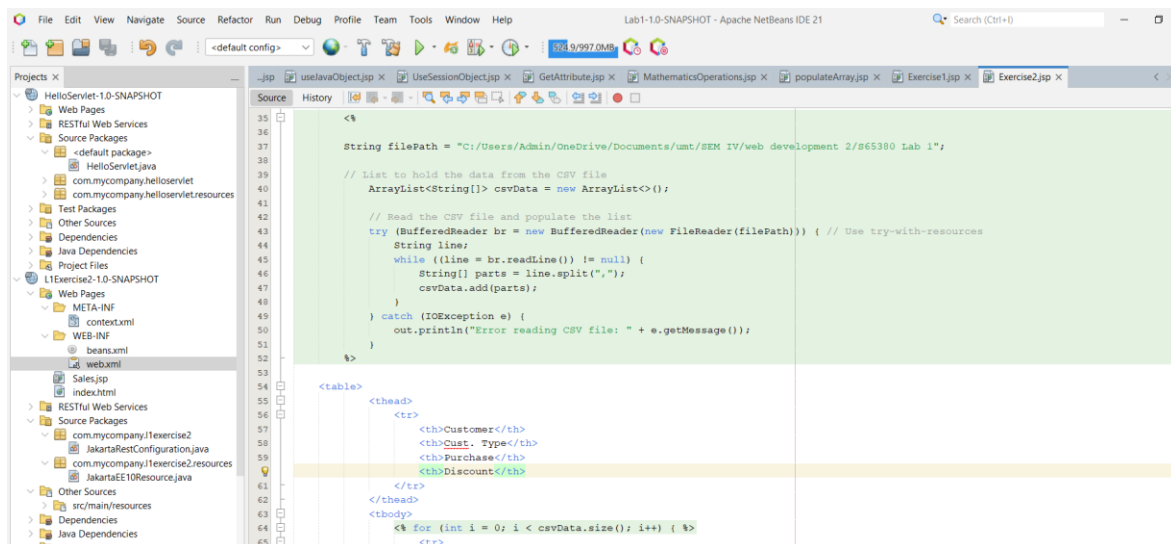
### Source code



```
<!--
Document : Exercise2
Created on : 30 Mar 2024, 8:27:59 pm
Author : Sakinah Jusoh
-->

<%page import="java.io.DataInputStream" %>
<%page import="java.io.FileInputStream" %>
<%page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Sales</title>
<style>
table{
width: 100%;
border-collapse: collapse;
border-width: 2px;
border-color: #666969;
border-style: solid;
}
td{
border-width: 2px;
text-align: center;
border-color: #000000;
border-style: solid;
padding: 5px;
}
</style>
</head>
<body>
<table>
<tr>
 Read Sales and Calculate Discount</td> |
```



```
<%
String filePath = "C:/Users/Admin/OneDrive/Documents/unt/SEM IV/web development 2/865380 Lab 1";

// List to hold the data from the CSV file
ArrayList<String[]> csvData = new ArrayList<>();

// Read the CSV file and populate the list
try (BufferedReader br = new BufferedReader(new FileReader(filePath))) { // Use try-with-resources
String line;
while ((line = br.readLine()) != null) {
String[] parts = line.split(",");
csvData.add(parts);
}
} catch (IOException e) {
out.println("Error reading CSV file: " + e.getMessage());
}

<table>
<thead>
<tr>
<th>Customer</th>
<th>Cust. Type</th>
<th>Purchase</th>
<th>Discount</th>
</tr>
</thead>
<tbody>
<% for (int i = 0; i < csvData.size(); i++) { %>
<tr>
```

