



**UNIVERSITI MALAYSIA TERENGGANU**

---

**CSM3023 WEB BASED APPLICATION DEVELOPMENT (K1)**

**BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONORS**

**LAB REPORTS**

**LAB 1-INTRODUCTION TO SERVLET, JSP AND MYSQL DATABASE**

**SEMESTER II 2023/2024**

---

**Prepared for:**

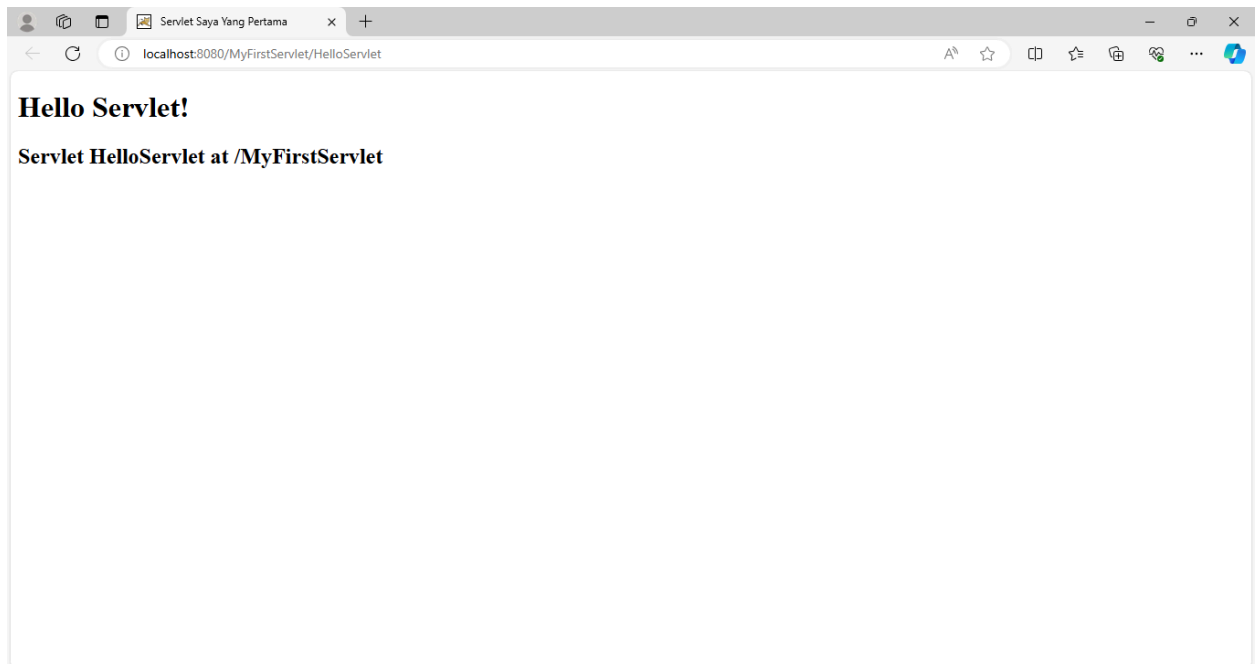
**Dr MOHAMAD NOR BIN HASAN**

**Prepared by:**

**IZZUL ILYAS BIN ROSLAN (S65343)**

## Task 5

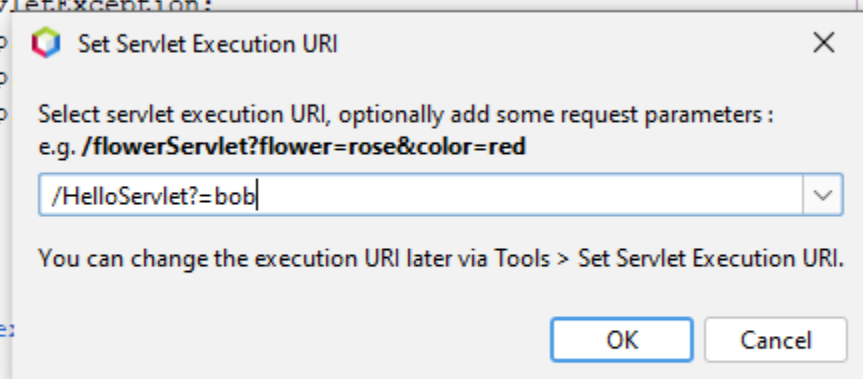
### Linking Netbeans to Apache Tomcat and Writing a Simple Java Servlet

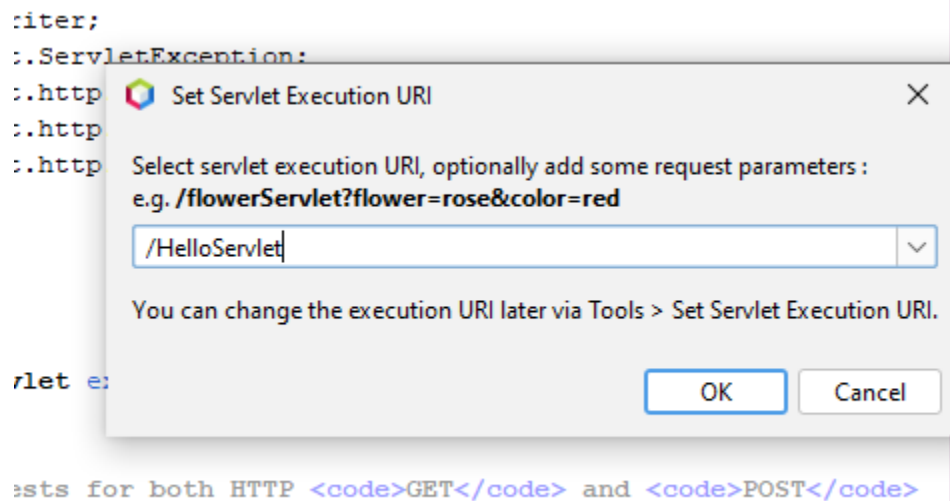
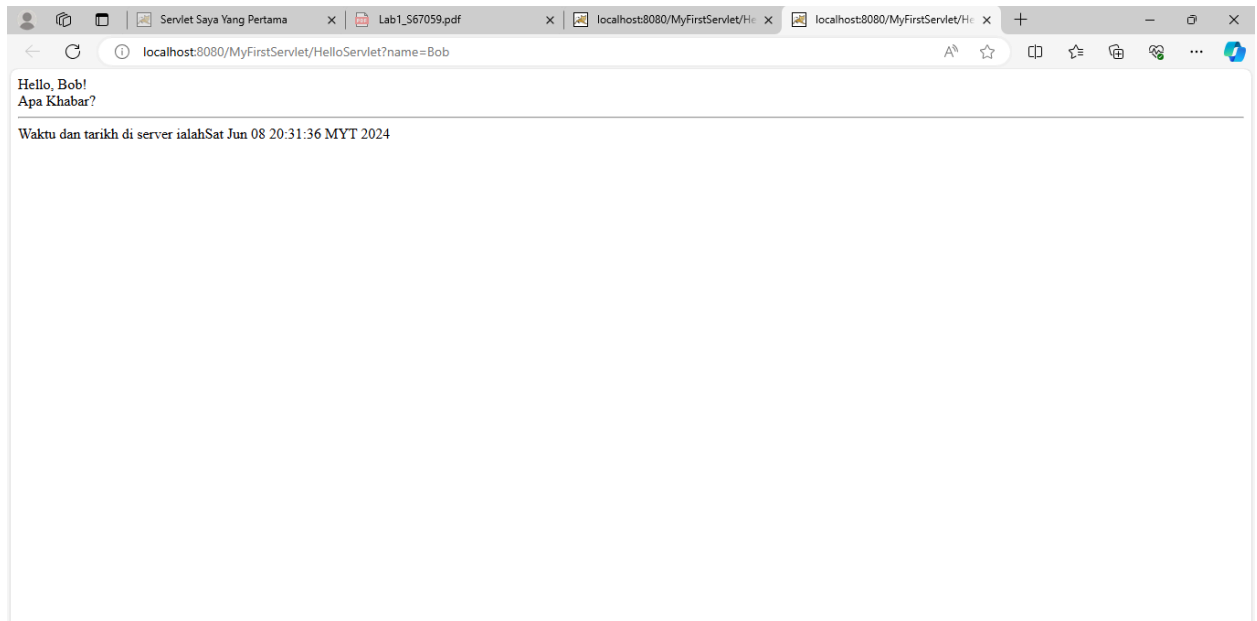


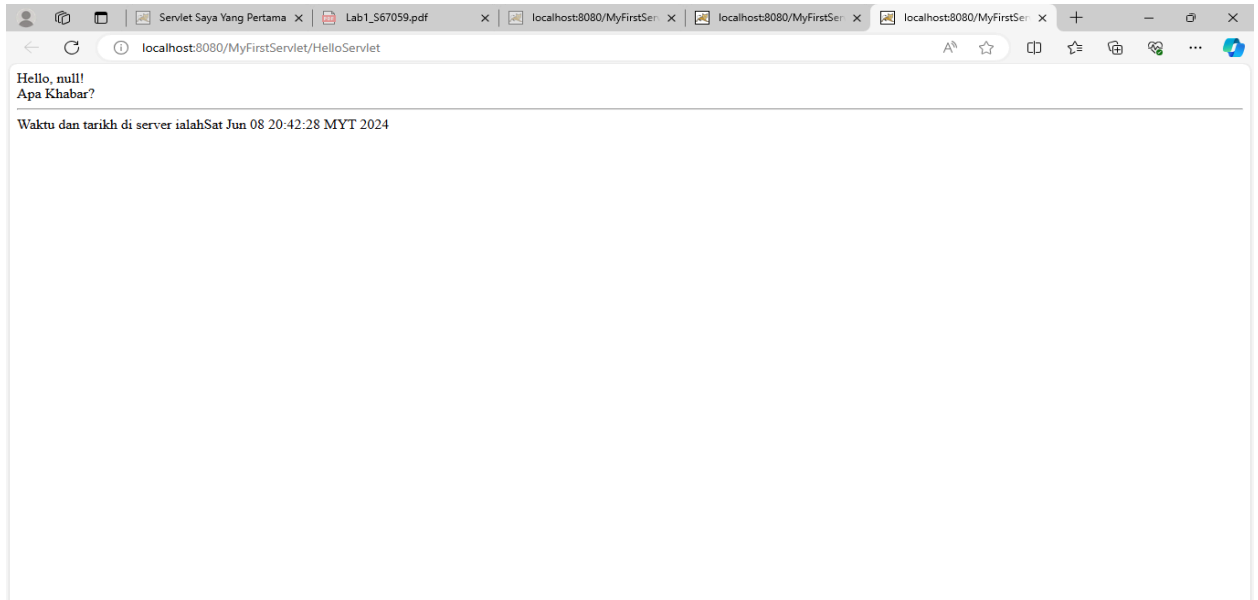
```
exception;  
PrintWriter;  
Servlet.ServletException:  
Servlet.http  
Servlet.http  
Servlet.http
```

Servlet es

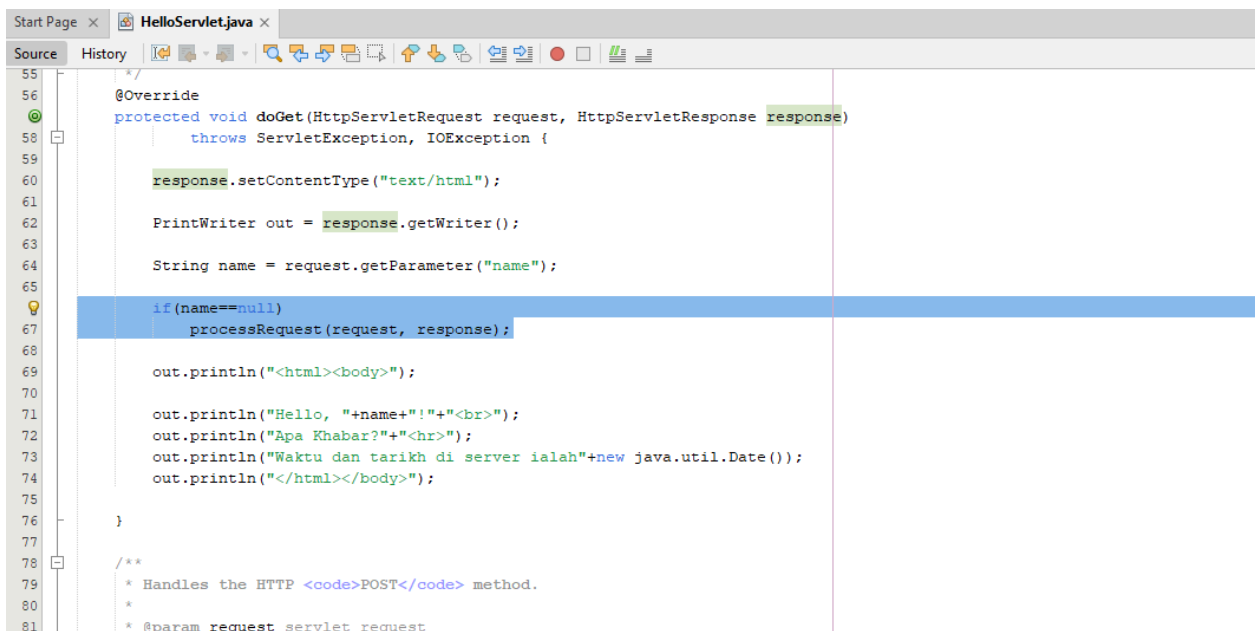
requests for both HTTP `GET` and `POST`

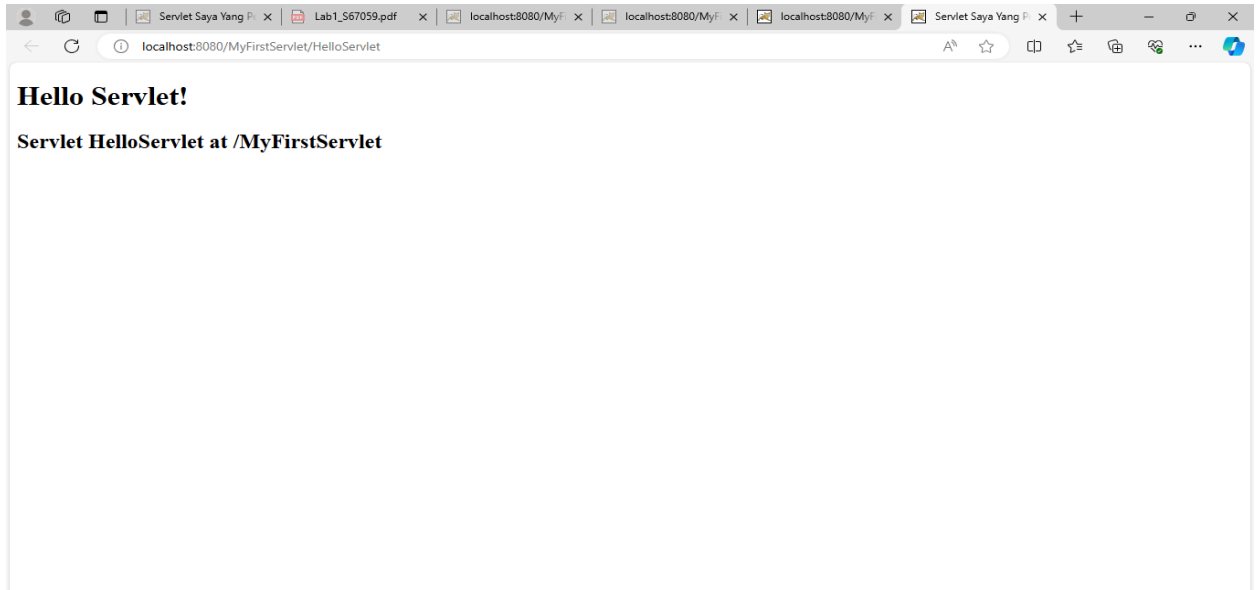






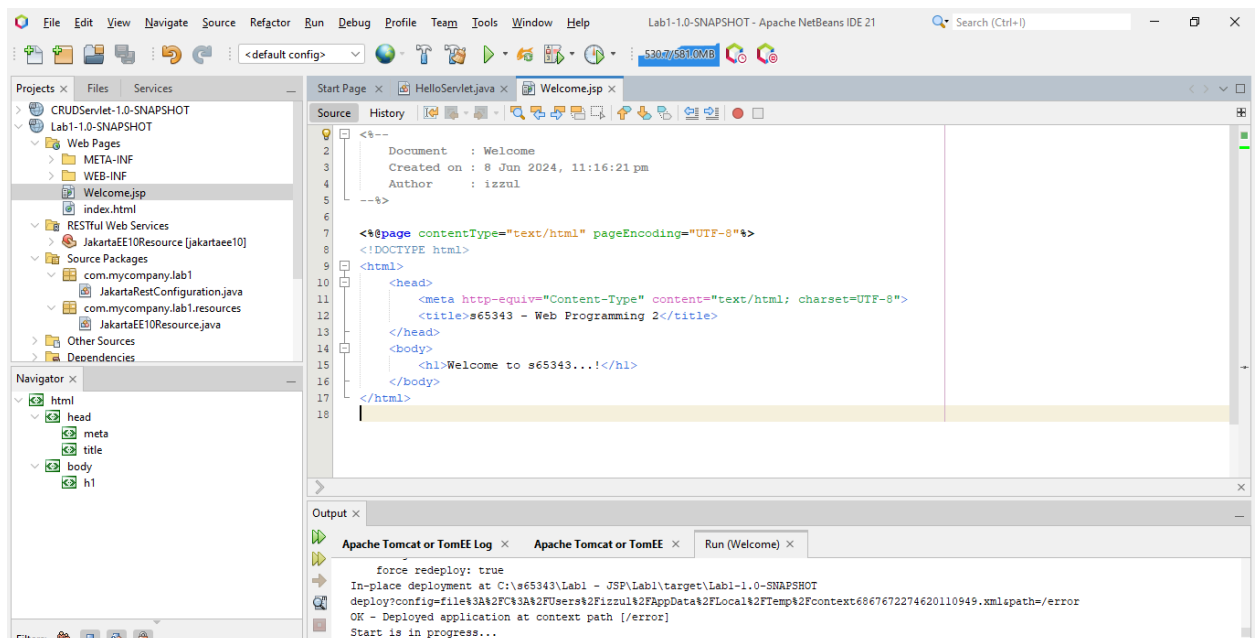
Upgrade your code in HelloServlet.java by putting the following codes into it. By doing this, if no value is supplied to the parameter name, the request will be passed to processRequest() method, and this will avoid the null value appearing on the browser.

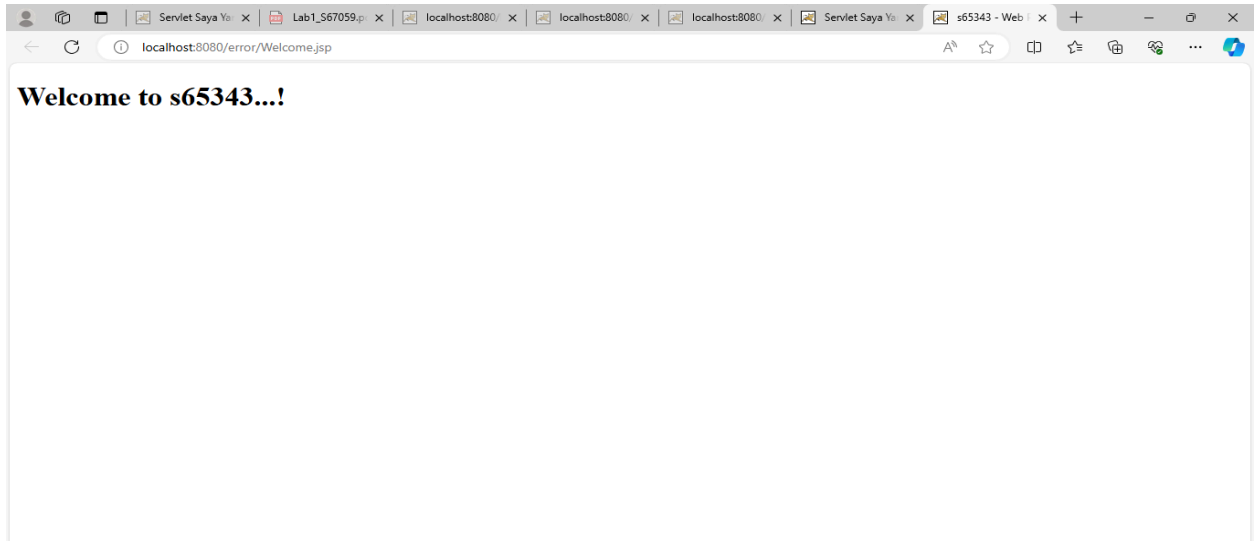




## Task 6

### Writing a Simple JSP Program





### Reflection

1. What have you learned from this exercise?

Writing a simple Java Servlet JSP file.

2. Explain the general concept of how the JSP's file works?

Embed Java code directly into HTML pages.

3. Based on your observation of the previous tasks (Task 3 and Task 4), what are the differences you can find between servlet and JSP?

Differences between JSP and Servlet is Servlet for backend operation and JSP is for creating the view layer of the web.

## Task 7

### Use Java Reference Datatype/Class Wrapper in JSP

The image shows a JSP file in an IDE and its rendered output in a web browser. The JSP file, named `useJavaObject.jsp`, contains the following code:

```
Created on : 8 Jun 2024, 11:40:37 pm
Author    : izzul

-->

<%@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.setIntHeader("Refresh", 5);
%>

</body>
</html>
```

The browser output shows the rendered HTML page at `localhost:8080/error/useJavaObject.jsp`. The page title is "Using Java's object" and the main heading is "Display Current Date and perform auto refresh header". The page content displays the current date and time: "Current date and time is Sun Jun 09 00:19:04 MYT 2024". The page is set to auto-refresh every 5 seconds.

## Reflection

### 1. What have you learnt from this exercise?

Using Java's object in JSP page

### 2. What is Java Scriptlet?

A Java scriptlet, also known simply as a scriptlet, is a block of Java code embedded directly into a JavaServer Pages (JSP) file. It allows developers to include Java code within the HTML markup of a JSP file, enabling dynamic content generation and server-side processing.

### 3. How to use Java code in your JSP's page?

Use scriptlets, Scriptlets allow you to embed Java code directly within the HTML markup of a JSP page.



## Task 8

### Using JSP Implicit object in JSP page

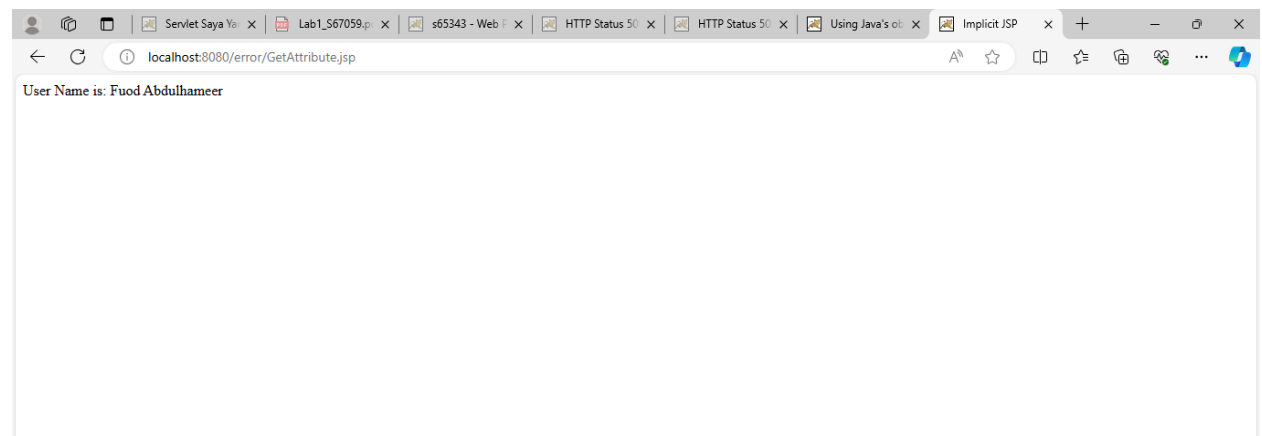


The image displays a development environment with two windows. The top window is a code editor showing a JSP file named `AttributelsSet.jsp`. The code includes a comment block with document details, a page directive for UTF-8 encoding, and an HTML structure with a head section containing a meta tag and a title, and a body section containing a session attribute setting and a link.

```
1 <!--
2     Document    : AttributelsSet
3     Created on  : 9 Jun 2024, 12:27:30 am
4     Author     : izzul
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     <% session.setAttribute("user", "Fuod Abdulhameer");%>
17     <a href="GetAttribute.jsp"> Click here to get user name </a>
18
19 </body>
20 </html>
```

The bottom window is a web browser showing the rendered output of the JSP page. The address bar indicates the URL `localhost:8080/error/AttributelsSet.jsp`. The page content shows the rendered link: [Click here to get user name](#).

```
1  <%--
2      Document    : GetAttribute
3      Created on  : 9 Jun 2024, 12:39:34 am
4      Author     : izzul
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         <%
17             String name = (String) session.getAttribute("user");
18             out.println("User Name is: " + name);
19         %>
20
21     </body>
22 </html>
23
```



```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Implicit JSP</title>
</head>
<body>

    <%
        int num1 = 25;
        int num2 = 10;
        int addition_output;
        int multiply_output;
        double squareroot = 0.00;

        java.util.Formatter myFormat = new java.util.Formatter();

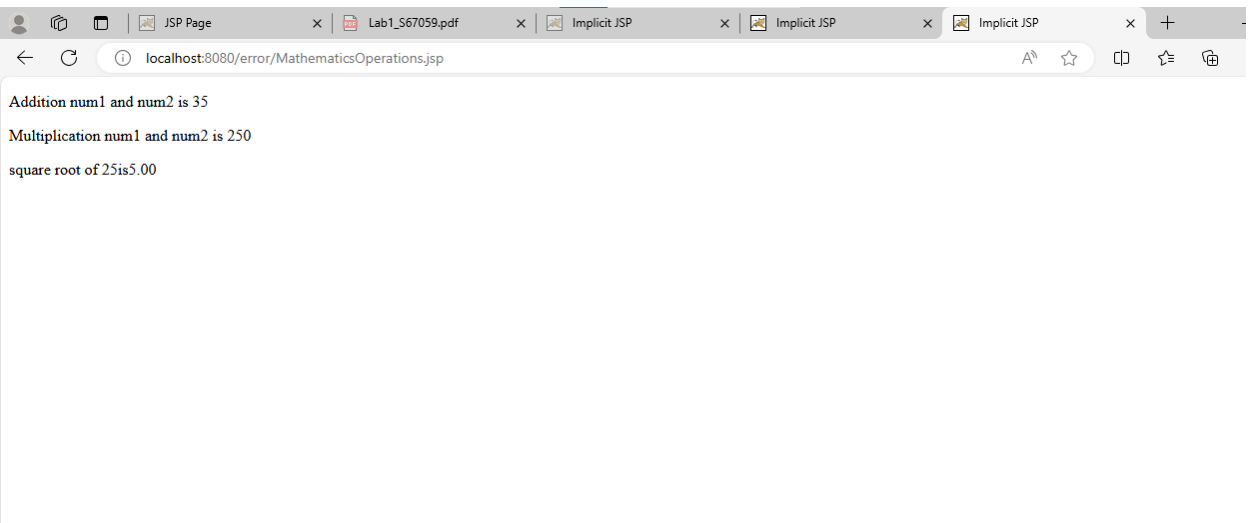
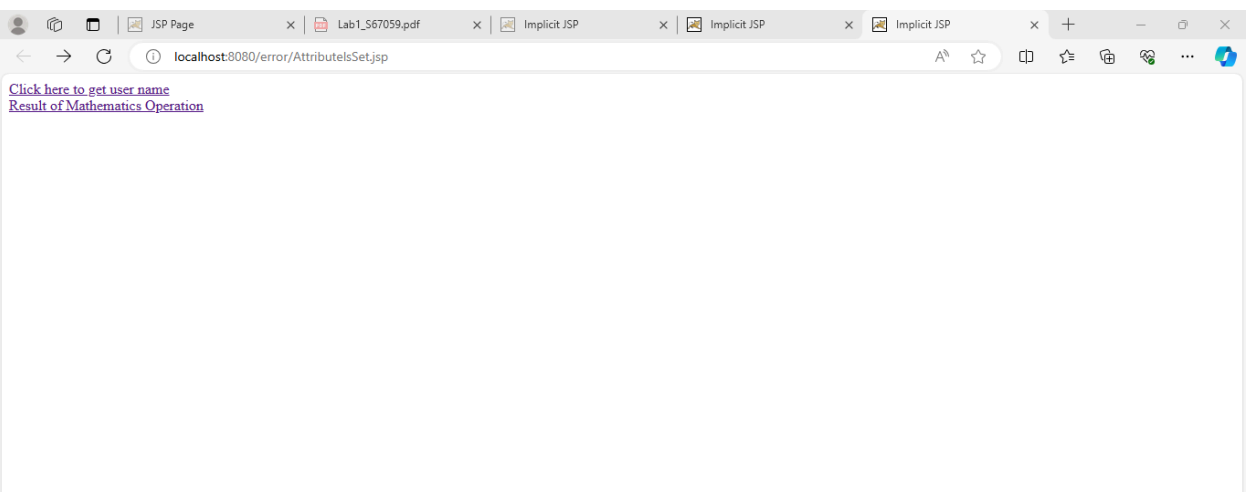
        addition_output = num1 + num2;
        multiply_output = num1 * num2;

        squareroot = (double) (Math.sqrt(num1));

        out.print("<p>Addition num1 and num2 is " + addition_output + "</p>");
        out.print("<p>Multiplication num1 and num2 is " + multiply_output + "</p>");

        out.print("<p></p>");
        out.print("<p>square root of " + num1 + "is" + myFormat.format("%.2f", squareroot) + "</p>");

    %>
</body>
</html>
```



## Reflection

1. How do you want to submit specific information from one form to next form?
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?

## Task9

### Populate Array values into HTML's Table

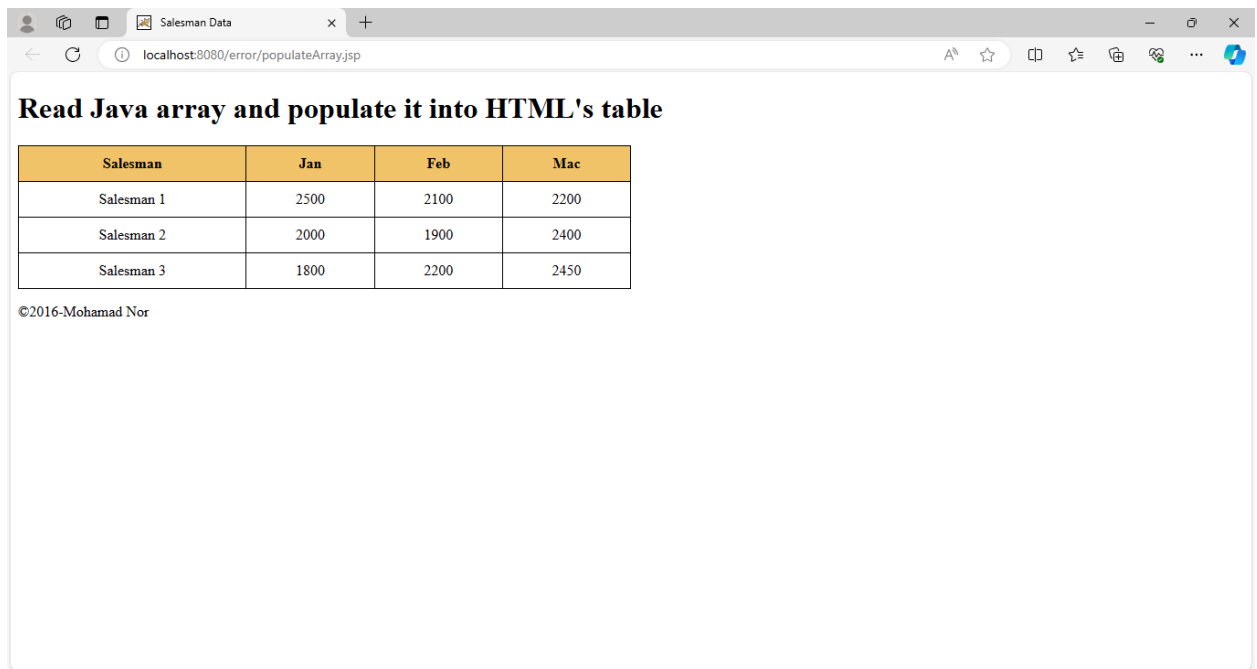
The image displays two screenshots of the Apache NetBeans IDE, showing the development of a JSP page named `populateArray.jsp`.

**Top Screenshot:** Shows the initial structure of the JSP page. It includes a document header with metadata (Document: populateArray, Created on: 9 Jun 2024, 4:01:40 pm, Author: izzul) and a page declaration: `<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>`. The page is wrapped in `<html>` and `<head>` tags. A `<meta charset="UTF-8">` tag is present. The `<title>Salesman Data</title>` is set. A `<style>` block defines the table's appearance: width 50%, border-collapse: collapse, border: 1px solid black, padding: 10px, text-align: center, and background-color: #f2c66d.

**Bottom Screenshot:** Shows the implementation of the table population. It starts with a comment `// Define the salesman data` and defines two arrays: `String[] salesmen = { "Salesman 1", "Salesman 2", "Salesman 3" };` and `int[][] sales = { { 2500, 2100, 2200 }, { 2000, 1900, 2400 }, { 1800, 2200, 2450 } };`. The table is constructed using `<table>` and `<tr>` tags. The first row contains the names of the salesmen: `<th>Salesman</th>`, `<th>Jan</th>`, `<th>Feb</th>`, and `<th>Mar</th>`. The subsequent rows are populated using a loop: `for (int i = 0; i < salesmen.length; i++) {`. Each row contains the name and the sales data for January, February, and March: `<td><%= salesmen[i] %></td>`, `<td><%= sales[i][0] %></td>`, `<td><%= sales[i][1] %></td>`, and `<td><%= sales[i][2] %></td>`. The table is closed with `</table>`. The page footer contains the text `<p>©2016-Mohamad Nor</p>`.

```
43 <tn><Salesman</tn>
44 <th>Jan</th>
45 <th>Feb</th>
46 <th>Mac</th>
47 </tr>
48 <tr>
49 <td><%= salesmen[i] %></td>
50 <td><%= sales[i][0] %></td>
51 <td><%= sales[i][1] %></td>
52 <td><%= sales[i][2] %></td>
53 </tr>
54 </table>
55 <p><%= salesmen[i] %></p>
56 </tr>
57 </table>
58 </tr>
59 </table>
60 </table>
61 <p><%= salesmen[i] %></p>
62 </body>
63 </html>
64
```

Output



## Reflection

1. Write a sample syntax to declare 2D Java array.

```
-// Syntax to declare a 2D Java array dataType[][] arrayName = new  
dataType[rows][columns];
```

2. Define a sequence of steps on how you accomplish Task 7.

3. What is the difference between HTML's page and JSP's page?

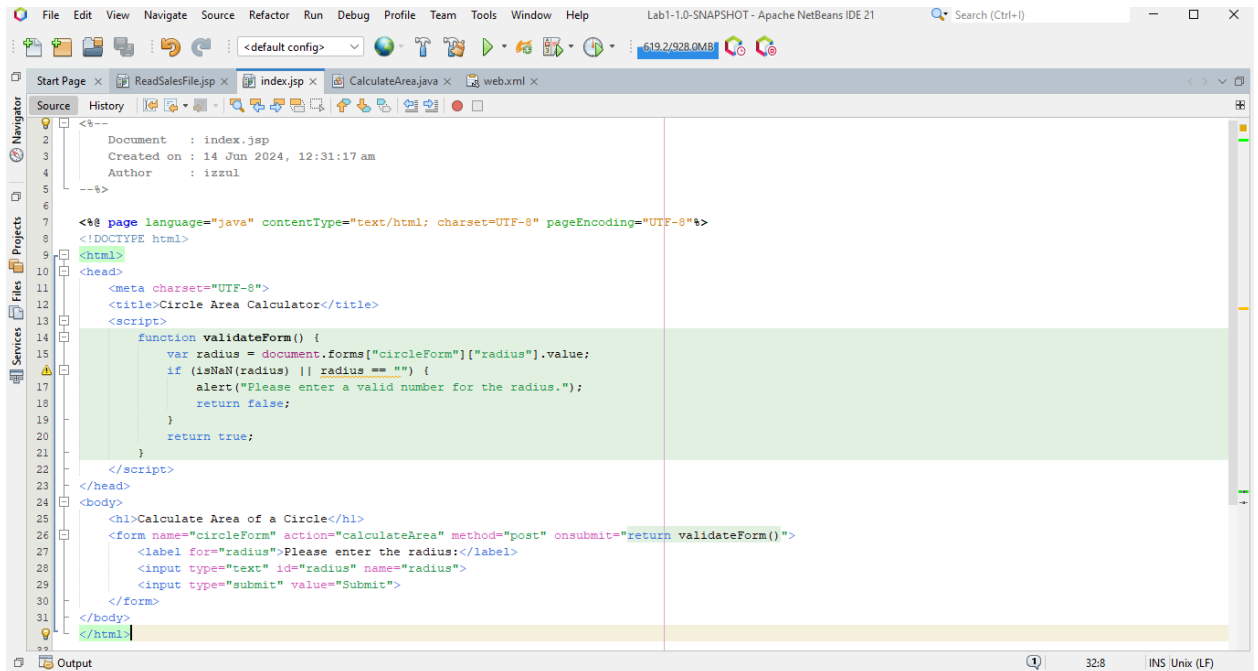
HTML pages are static and consist of markup languages such as HTML, CSS, and JavaScript. They are primarily used for displaying static content and structuring the layout of web pages.

JSP pages are dynamic and allow embedding Java code within HTML markup.

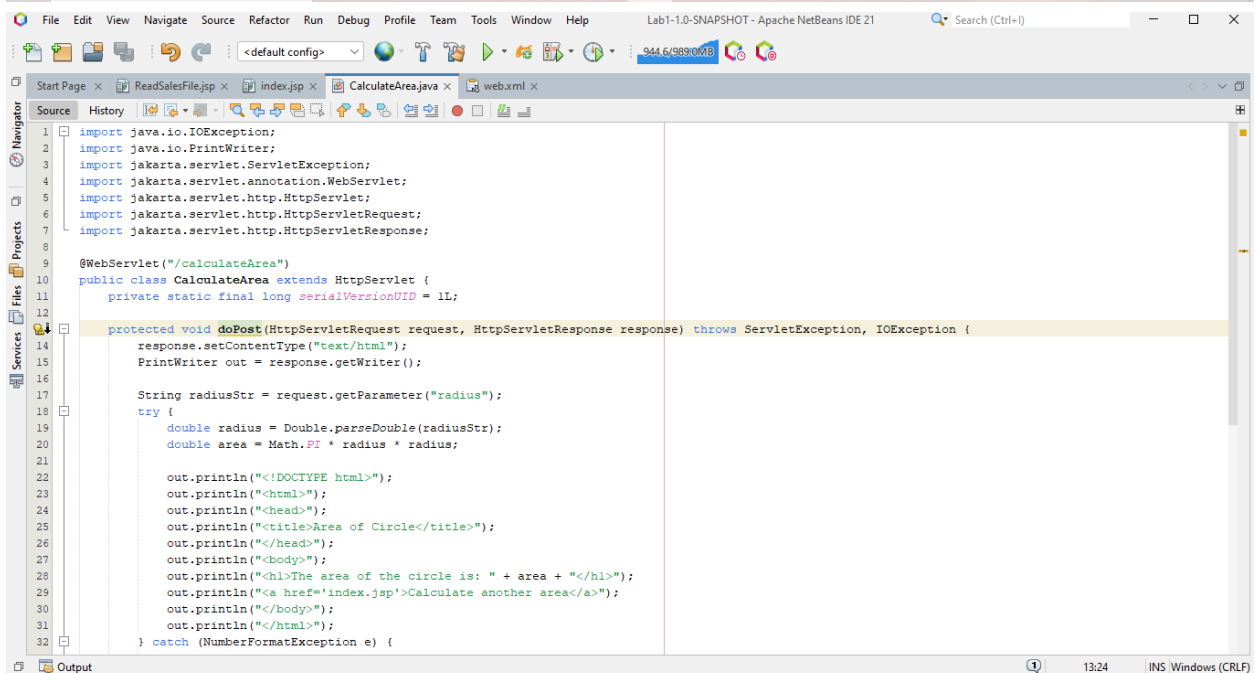
## Lab Exercise

### 1. Convert the program

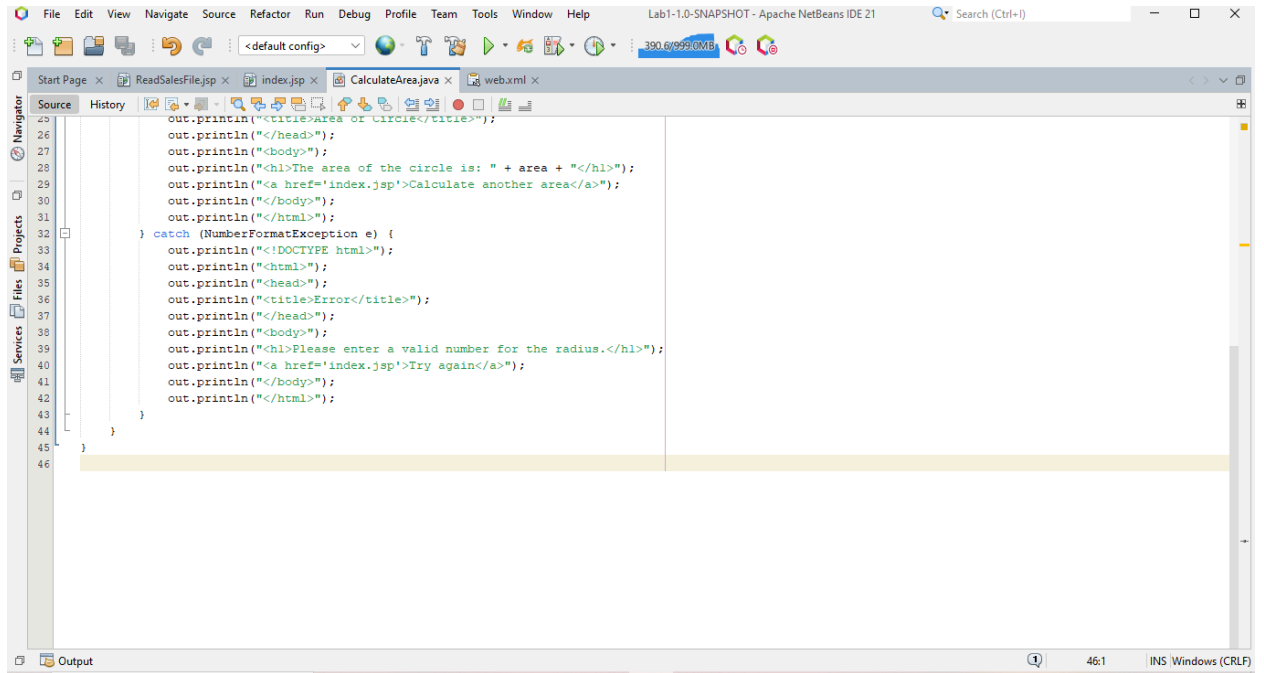
Code:



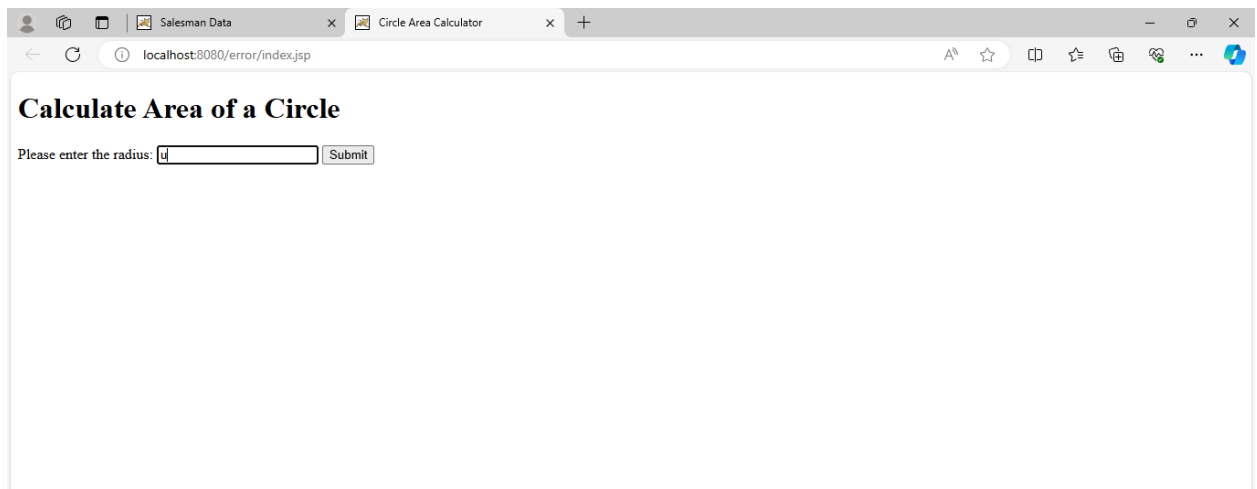
```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Lab1-1.0-SNAPSHOT - Apache NetBeans IDE 21 Search (Ctrl+I)
<default config> 619.2/928.0MB
Start Page x ReadSalesFile.jsp x index.jsp x CalculateArea.java x web.xml x
Source History
Document : index.jsp
Created on : 14 Jun 2024, 12:31:17 am
Author : izzul
-->
<% page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Circle Area Calculator</title>
<script>
function validateForm() {
    var radius = document.forms["circleForm"]["radius"].value;
    if (isNaN(radius) || radius == "") {
        alert("Please enter a valid number for the radius.");
        return false;
    }
    return true;
}
</script>
</head>
<body>
<h1>Calculate Area of a Circle</h1>
<form name="circleForm" action="calculateArea" method="post" onsubmit="return validateForm()">
    <label for="radius">Please enter the radius:</label>
    <input type="text" id="radius" name="radius">
    <input type="submit" value="Submit">
</form>
</body>
</html>
Output 32.8 INS Unix (LF)
```



```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Lab1-1.0-SNAPSHOT - Apache NetBeans IDE 21 Search (Ctrl+I)
<default config> 944.6/989.0MB
Start Page x ReadSalesFile.jsp x index.jsp x CalculateArea.java x web.xml x
Source History
1 import java.io.IOException;
2 import java.io.PrintWriter;
3 import jakarta.servlet.ServletException;
4 import jakarta.servlet.annotation.WebServlet;
5 import jakarta.servlet.http.HttpServlet;
6 import jakarta.servlet.http.HttpServletRequest;
7 import jakarta.servlet.http.HttpServletResponse;
8
9 @WebServlet("/calculateArea")
10 public class CalculateArea extends HttpServlet {
11     private static final long serialVersionUID = 1L;
12
13     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
14         response.setContentType("text/html");
15         PrintWriter out = response.getWriter();
16
17         String radiusStr = request.getParameter("radius");
18         try {
19             double radius = Double.parseDouble(radiusStr);
20             double area = Math.PI * radius * radius;
21
22             out.println("<!DOCTYPE html>");
23             out.println("<html>");
24             out.println("<head>");
25             out.println("<title>Area of Circle</title>");
26             out.println("</head>");
27             out.println("<body>");
28             out.println("<h1>The area of the circle is: " + area + "</h1>");
29             out.println("<a href='index.jsp'>Calculate another area</a>");
30             out.println("</body>");
31             out.println("</html>");
32         } catch (NumberFormatException e) {
33
34         }
35     }
36 }
Output 13.24 INS Windows (CRLF)
```



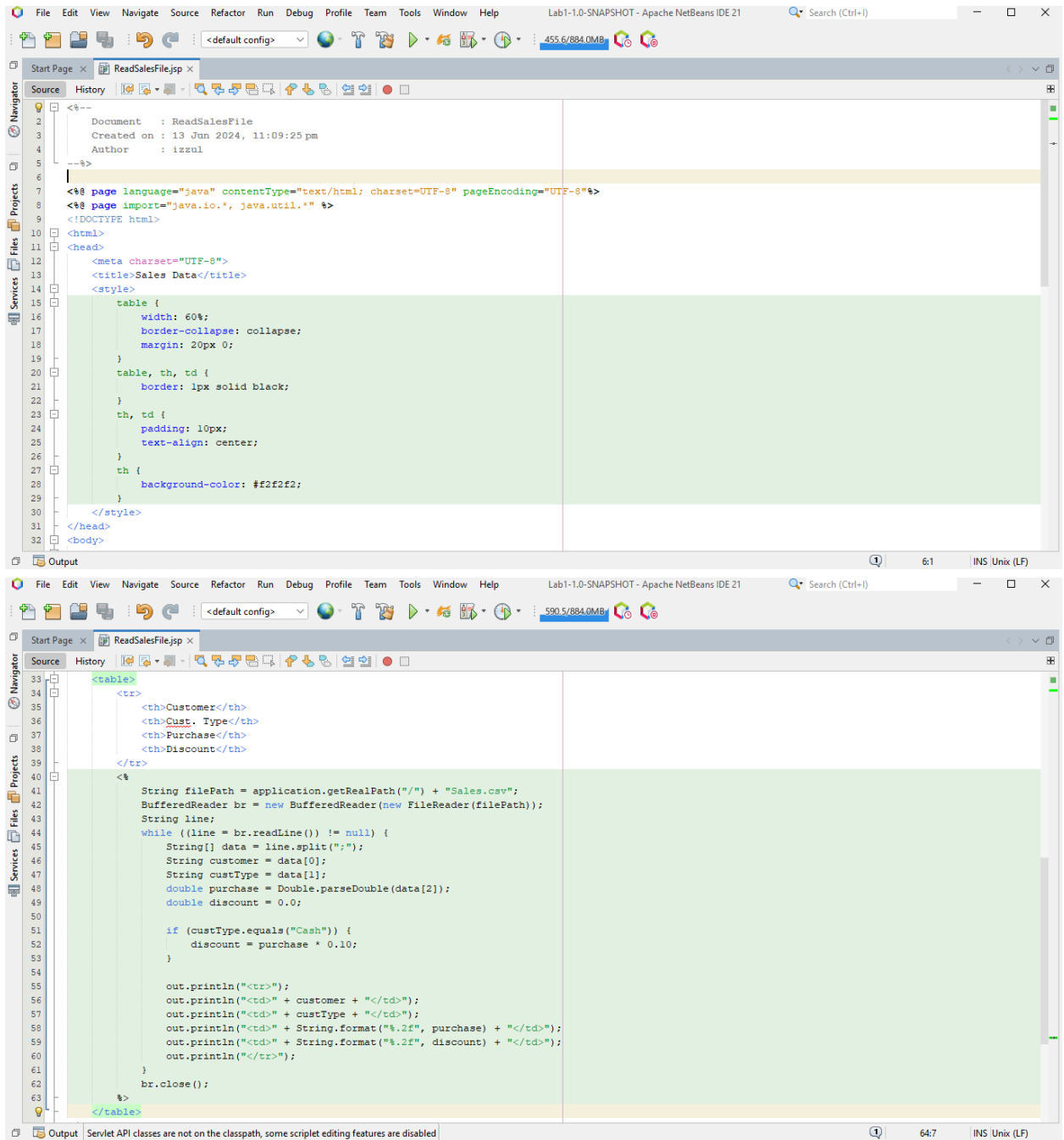
Output:





## 2. Sales.csv

Code:



The image displays two screenshots of the Apache NetBeans IDE, showing the development of a Java web application. The top screenshot shows the HTML file 'ReadSalesFile.jsp' with a table structure and CSS styling. The bottom screenshot shows the same file with the Java code that reads a CSV file and populates the table.

**Top Screenshot (HTML File):**

```
<!--
Document   : ReadSalesFile
Created on : 13 Jun 2024, 11:09:25 pm
Author    : izzul
-->

<%% page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%% page import="java.io.*, java.util.*"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Sales Data</title>
<style>
table {
width: 60%;
border-collapse: collapse;
margin: 20px 0;
}
table, th, td {
border: 1px solid black;
}
th, td {
padding: 10px;
text-align: center;
}
th {
background-color: #f2f2f2;
}
</style>
</head>
<body>
```

**Bottom Screenshot (Java Code):**

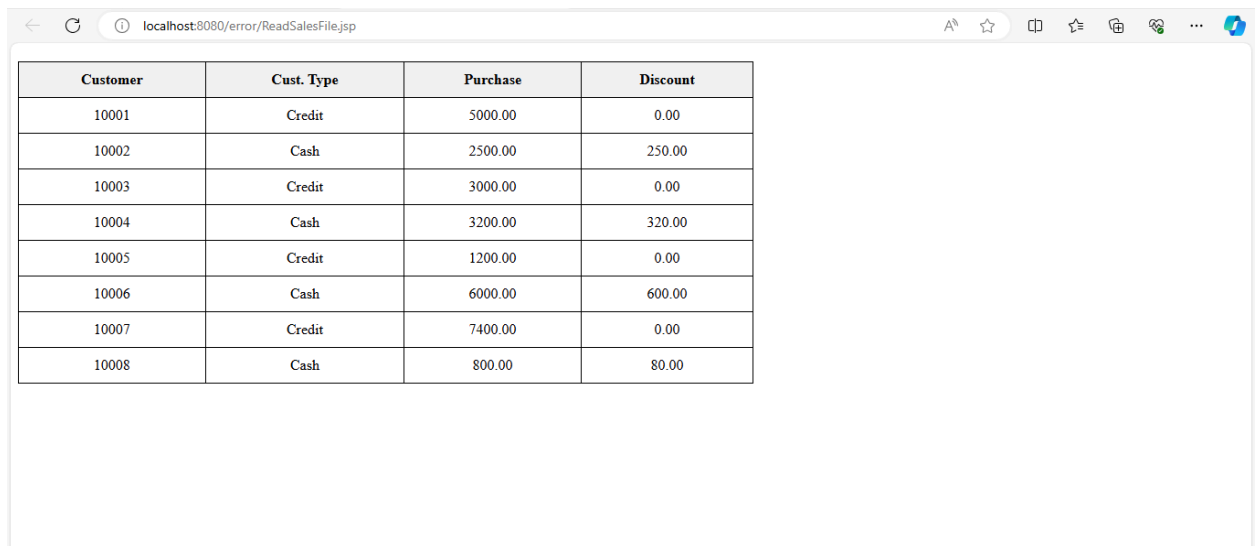
```
<table>
<tr>
<th>Customer</th>
<th>Cust. Type</th>
<th>Purchase</th>
<th>Discount</th>
</tr>
<%
String filePath = application.getRealPath("/") + "Sales.csv";
BufferedReader br = new BufferedReader(new FileReader(filePath));
String line;
while ((line = br.readLine()) != null) {
String[] data = line.split(",");
String customer = data[0];
String custType = data[1];
double purchase = Double.parseDouble(data[2]);
double discount = 0.0;

if (custType.equals("Cash")) {
discount = purchase * 0.10;
}

out.println("<tr>");
out.println("<td>" + customer + "</td>");
out.println("<td>" + custType + "</td>");
out.println("<td>" + String.format("%.2f", purchase) + "</td>");
out.println("<td>" + String.format("%.2f", discount) + "</td>");
out.println("</tr>");
}
br.close();
%>
</table>
```

Output: Servlet API classes are not on the classpath, some scriptlet editing features are disabled

Output :



A screenshot of a web browser window showing a table of sales data. The browser's address bar displays 'localhost:8080/error/ReadSalesFile.jsp'. The table has four columns: 'Customer', 'Cust. Type', 'Purchase', and 'Discount'. It contains eight rows of data, alternating between 'Credit' and 'Cash' customer types. The 'Purchase' and 'Discount' values are numerical amounts.

Customer	Cust. Type	Purchase	Discount
10001	Credit	5000.00	0.00
10002	Cash	2500.00	250.00
10003	Credit	3000.00	0.00
10004	Cash	3200.00	320.00
10005	Credit	1200.00	0.00
10006	Cash	6000.00	600.00
10007	Credit	7400.00	0.00
10008	Cash	800.00	80.00