UNIVERSITI MALAYSIA TERENGGANU

Faculty of Computer Sciences and Mathematics

Web Based Application Development

CSM3023

Lab Report 1

Prepared for:

Sir. Mohd Arizal Shamsil bin Mat Rifin

Prepared by:

Ahmad Afif Syahmi bin Ahmad Rozali (S65526)

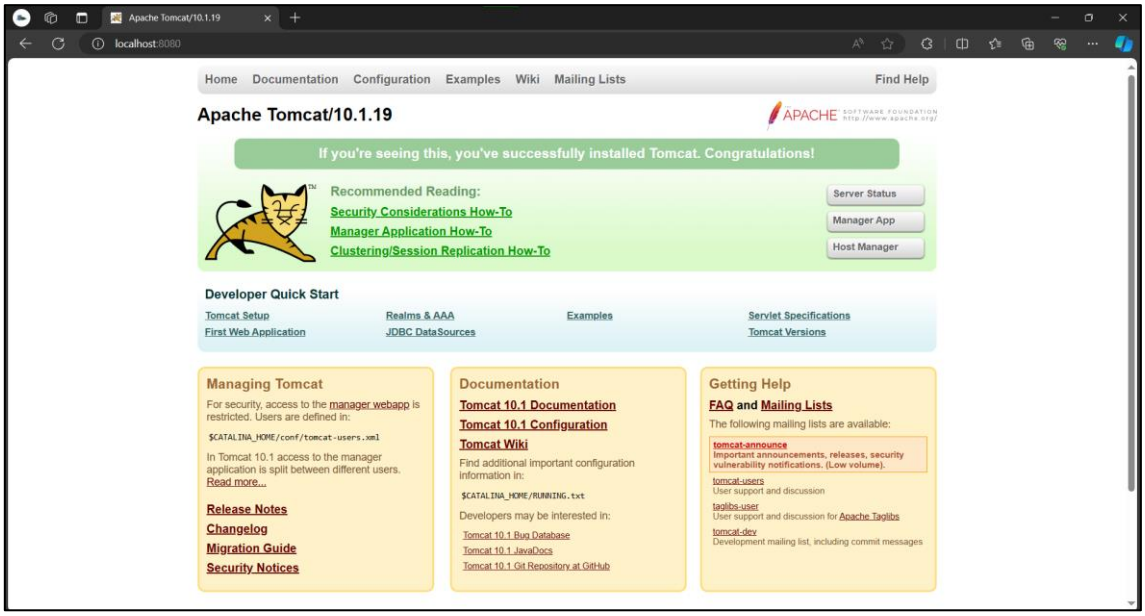
27th March 2024

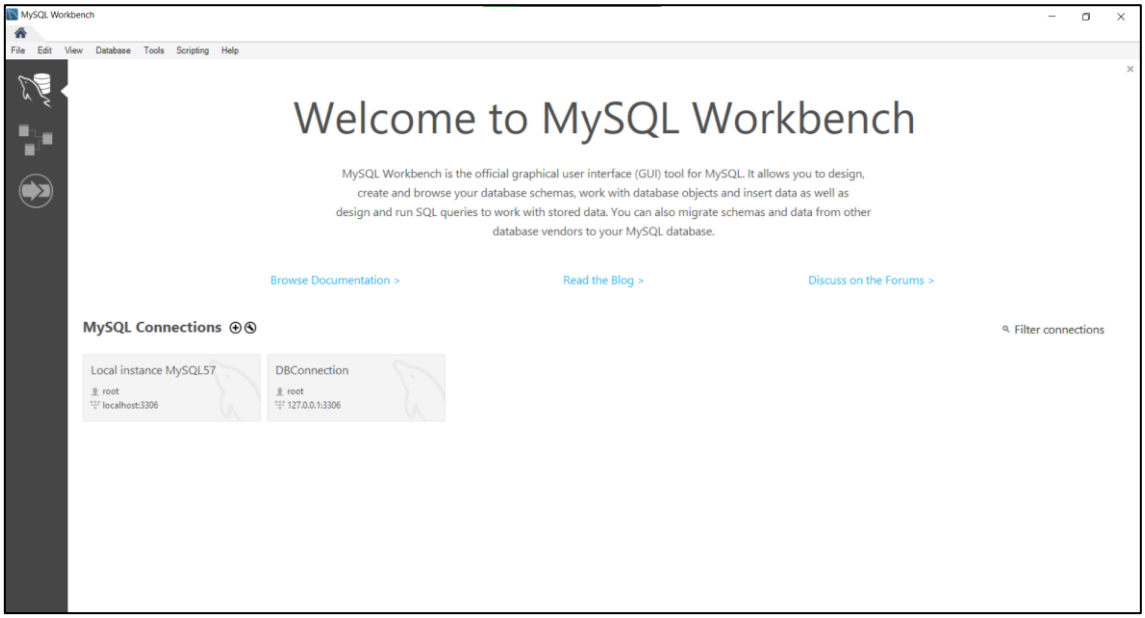
Bachelor of Computer Science (Mobile Computing) with Honors

Semester II 2023/2024

Task 1 : Apache Tomcat and MySQL Installation

- Objective: Installation of Apache Tomcat and MySQL
- Output:





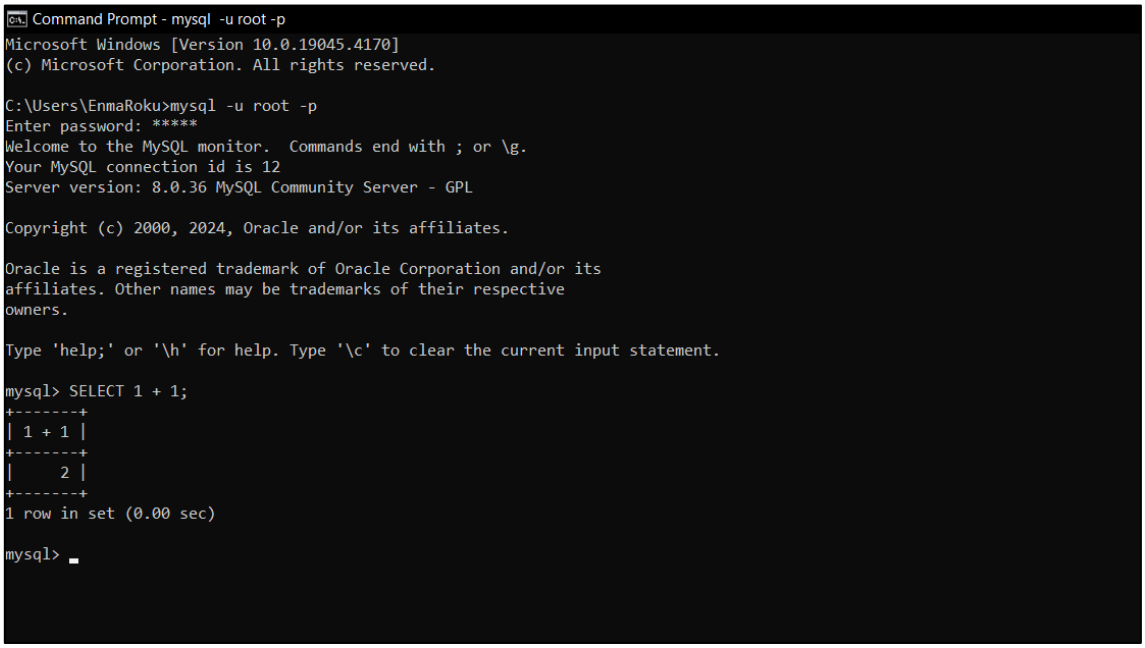
- Reflection:

From this exercise, I learnt how to install Apache Tomcat and MySQL in my device. Apache Tomcat is a free and open-source software that runs Java-based web applications. It's essentially a platform specifically designed to handle Java code within a web server

environment. Tomcat is popular because it's lightweight, efficient, and easy to use. Next, MySQL is an open-source relational database management system (RDBMS). This means it stores information in a structured way using tables with rows and columns, and allows you to access and manipulate that data using a query language called SQL. MySQL is popular for its ease of use, speed, reliability, and affordability.

Task 2 : Change the Default Root Password of MySQL Database

- Objective: Change the root password of MySQL Database
- Output:



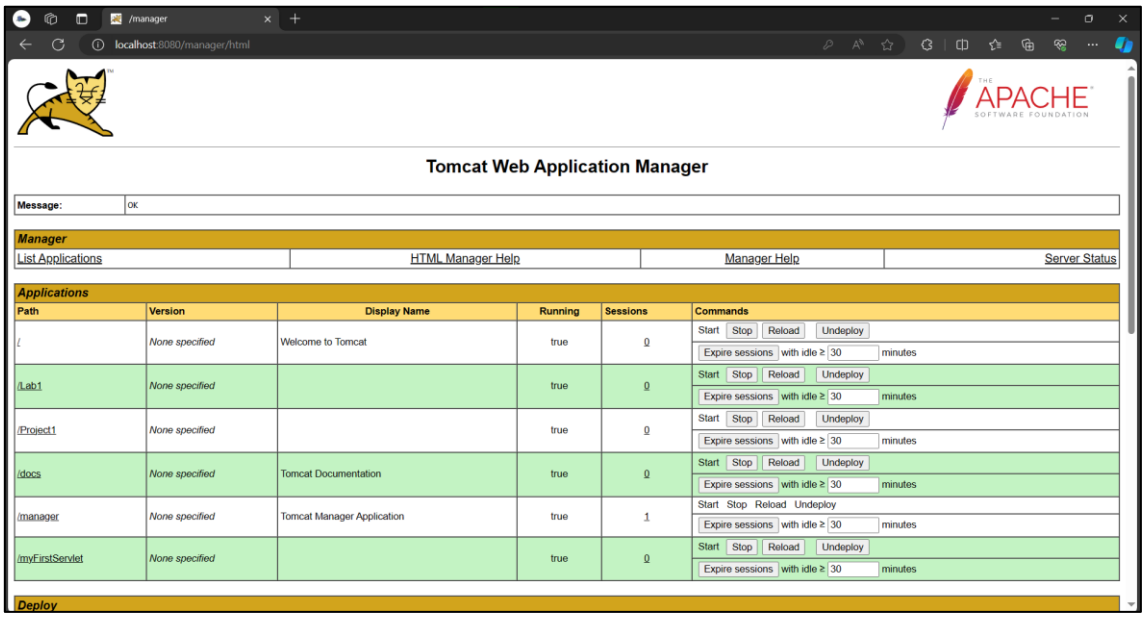
- Reflection:

From this exercise, I learnt how to access MySQL using command prompt. While graphical user interfaces (GUIs) exist for managing MySQL, knowing how to access it through the command prompt offers several advantages. It allows for quicker tasks and scripting for automation. For server administration, the command line is often the only way to directly interact with the database, especially for troubleshooting or remote

connections. Even with GUIs, understanding command-line prompts empowers you to better understand what's happening behind the scenes.

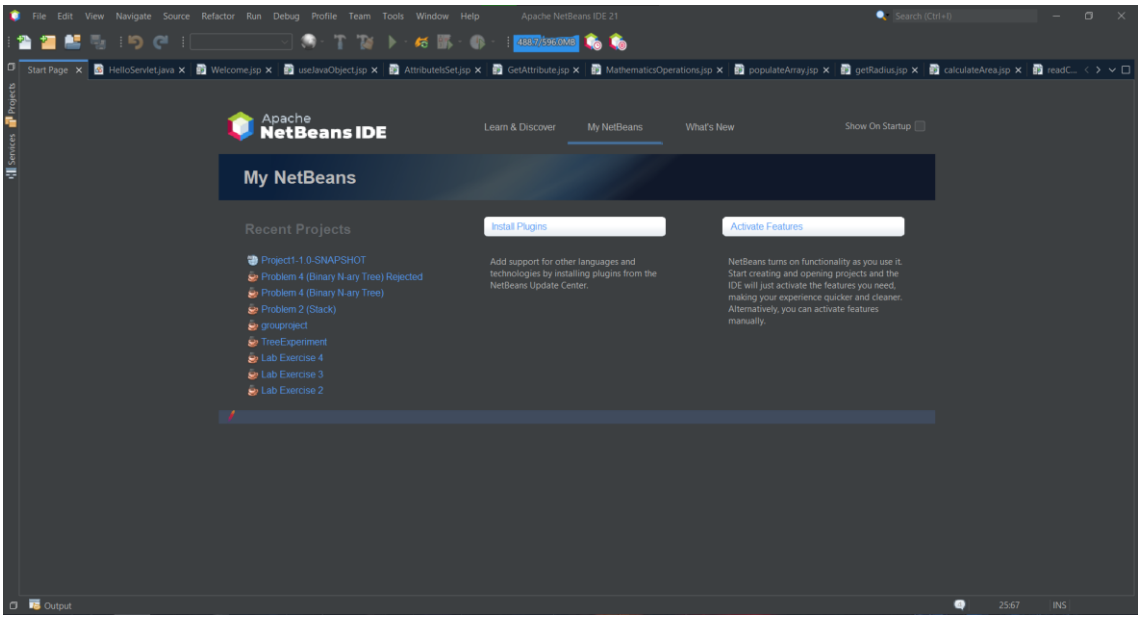
Task 3 : Managing Apache Tomcat

- Objective: Testing the access and add a new user to Apache Tomcat
- Output:



Task 4 : NetBeans IDE Installation

- Objective: To setup a proper environment for Java Web Application development.
- Output:



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

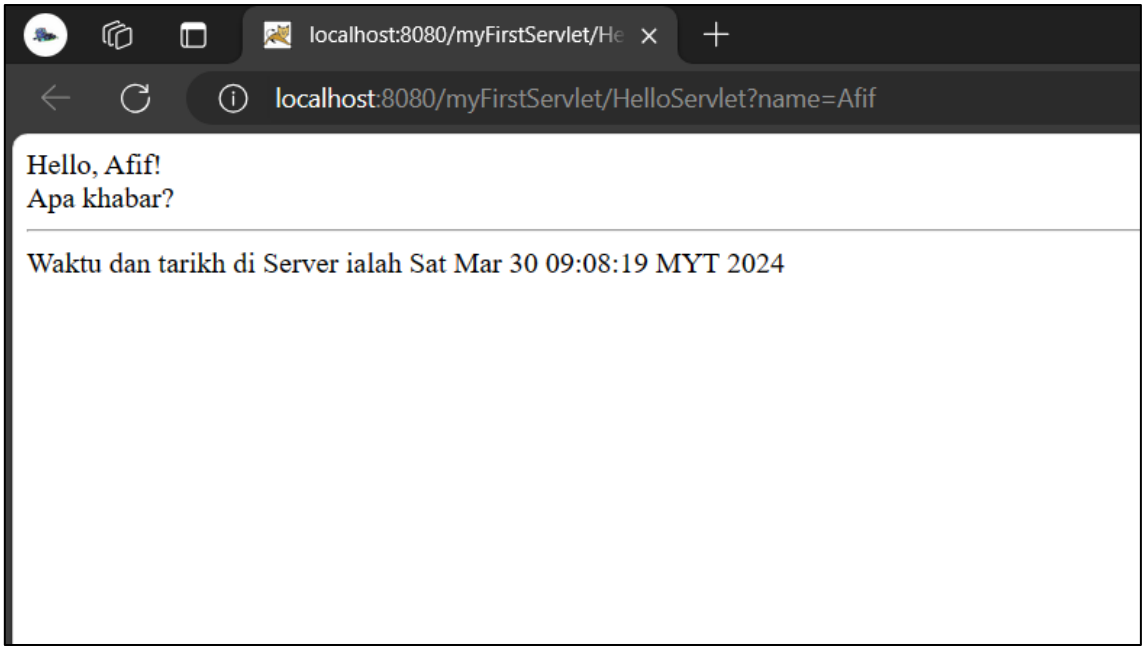
- Objective: To write a simple Java Servlet.
- Code:

```
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

/**
 *
 * @author Ahmad Afif Syahmi bin Ahmad Rozali
 */
public class HelloServlet extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following
sample code. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet Saya Yang Pertama</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Hello, Servlet!</h1>");
            out.println("<h2>Servlet HelloServlet at " +
request.getContextPath() + "</h2>");
            out.println("</body>");
            out.println("</html>");
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet
methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
```

```
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {

        /*
            Step 1: Set the context type (tell the browser what is the
type of the response data; e.g text/html, text/plain). In our case, we
will responds with html data.
        */
        response.setContentType("text/html");

        /*
            Step 2: Create the PrintWriter object. We name it as 'out'
        */
        PrintWriter out = response.getWriter();

        /*
            Step 3: Read GET parameter sent by the user through the
web browser
        */
        String name = request.getParameter("name");

        /*
            Additional: if no value for parameter "name", call
processRequest method
        */
        if (name==null)
            processRequest(request, response);

        /*
            Step 4: Generate content for our HTML response. Print the
name
        */
        out.println("<html><body>");

        out.println("Hello, " + name + "!<br>");
        out.println("Apa khabar?<hr>");
        out.println("Waktu dan tarikh di Server ialah " + new
java.util.Date());
        out.println("</body></html>");
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
        processRequest(request, response);
    }
```

```
    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

- Output:



- Reflection:

  It is the same output as can be seen in Step 18. Why?

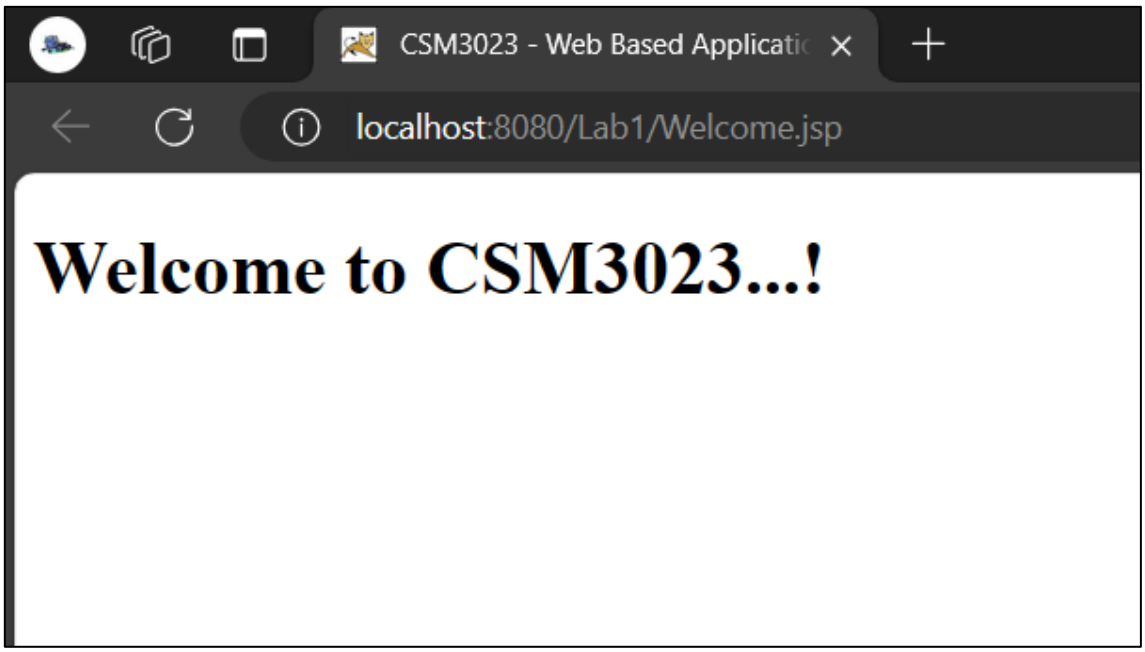
  It is the same output because the same method processRequest() was recalled.

Task 6 : Writing a Simple JSP Program

- Objective: Writing a simple plain JSP program.
- Code:

```
<%--
    Document   : Welcome
    Created on : 27 Mar 2024, 3:34:29 pm
    Author     : Ahmad Afif Syahmi bin Ahmad Rozali
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>CSM3023 - Web Based Application Development</title>
    </head>
    <body>
        <h1>Welcome to CSM3023...!</h1>
    </body>
</html>
```

- Output:



- Reflection:

1. What have you learned from this exercise?

   From this exercise I learnt how to set up a JSP program in NetBeans using Apache Tomcat server.

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

   JSP files, or Java Server Pages, are the secret sauce behind dynamic web content. These files mix regular HTML with special JSP tags. When a JSP file is requested, the server doesn't just send it raw. Instead, it translates the JSP into a special Java program called a servlet. This servlet can interact with databases, other programs, or anything else on the server. Finally, the servlet generates a standard HTML page that gets sent back to the user's browser. This way, JSP pages can create different content for each user, based on their requests or information stored on the server.

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?

   Servlets and JSPs are both the fundamental blocks building a web application, but they have different functions. Servlets can be equated to backstage workers who are written in a pure language (Java) and are responsible for the heavy lifting tasks such as data processing and communication with databases. The distinct aspect of JSPs is the emphasis on the presentation. They blend HTML with JSP tags and are of easy writing than classic versions as they develop don HTML familiarity. JSPs are turned into servlets before executing, which JSPs do not have the other servlets' ability to deal with non-HTTP requests or to directly override core works. In general, a servlet processes the application logic, and a JSP page then is responsible for how the user sees it.

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

- Objective: Display the current date, perform auto refresh header in JSP's page.
- Code:

```
<%--
    Document   : useJavaObject
    Created on : 28 Mar 2024, 7:34:02 am
    Author     : Ahmad Afif Syahmi bin Ahmad Rozali
--%>

<%@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 in JSP page</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>");

            // Set refresh, autoload time as seconds
            response.setIntHeader("Refresh", 5);
        %>

    </body>
</html>
```

- Output:

- Reflection:

1. What have you learnt from this exercise?

   From this exercise I learnt how to use the <%%> symbol which is scriptlet tag used to embed Java code within the HTML content of a JSP file. We can insert Java code in HTML by using scriptlet tag.

2. What is Java Scriptlet?

   A Java Scriptlet is a block of Java code slipped directly between speech tags in a JSP page. These become `<% %>` encoded. With its use of scriptlets, you are able to inser dynamic logic and get swing to the server-side resources that can take place right within your JSP file. This enables you to accomplish such as assignments of the variables to the received data, open the databases, or do everything with calculations. Though mini-scripts are helpful, they can complicate code base and deteriorate it dealing with security.

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

   We can use scriptlets, expressions, declarations, and directives to accomplish it. Chunks of Java code which is enclosed in `<% %>` tags are called scriptlets. The scriptlets allow operations like data processing, decision making, and the looping. Things in angels brackets will be used to evaluate and output Java expressions being directly rendered into the HTML response. Declarations, denoted by `<%! <code></code> are utilized to define both variables and methods as well as classes that are needed throughout the JSP page.

Task 8 : Using JSP Implicit object in JSP page

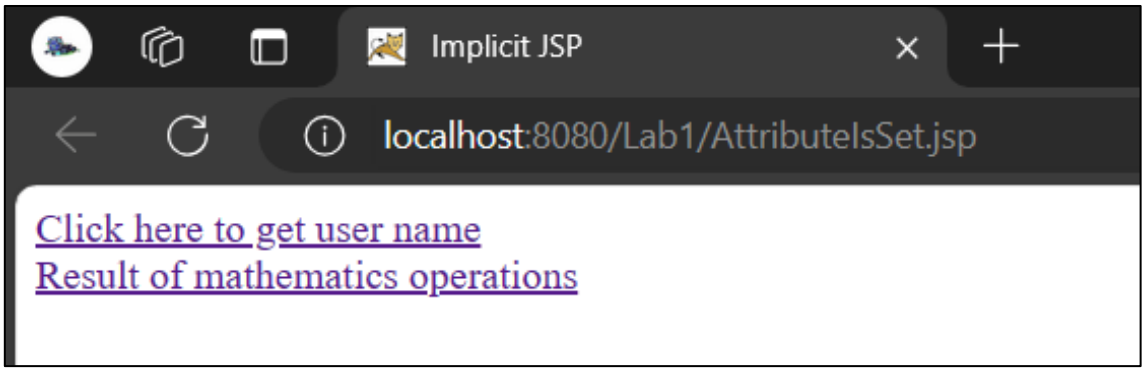
- Objective: Using Session object, perform simple Mathematics operations in JSP's page.
- Code:

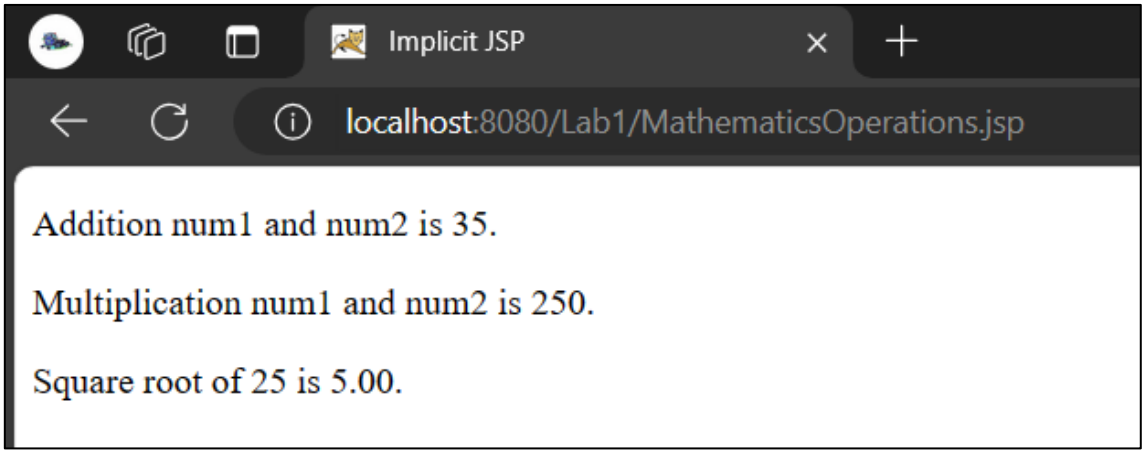
```
<%--
    Document   : AttributeIsSet
    Created on : 29 Mar 2024, 7:23:18 am
    Author     : Ahmad Afif Syahmi bin Ahmad Rozali
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>Implicit JSP</title>
    </head>
    <body>

        <% session.setAttribute("user", "Ahmad Afif Syahmi"); %>
        <a href="GetAttribute.jsp">Click here to get user name</a>
        <br>
        <a href="MathematicsOperations.jsp">Result of mathematics
operations</a>

    </body>
</html>
```

```
<%--
    Document   : GetAttribute
    Created on : 29 Mar 2024, 7:29:15 am
    Author     : Ahmad Afif Syahmi bin Ahmad Rozali
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>Implicit JSP</title>
    </head>
    <body>

        <%
            String name = (String) session.getAttribute("user");
            out.println("User name is: " + name);
        %>

    </body>
</html>
```

```
<%--
    Document   : MathematicsOperations
    Created on : 29 Mar 2024, 7:33:10 am
    Author     : Ahmad Afif Syahmi bin Ahmad Rozali
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.math.*" %>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>Implicit JSP</title>
    </head>
    <body>

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

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

            // Perform basic arithmetics operations
            addition_output = num1 + num2;
            multiply_output = num1 * num2;

            // Find square root for variable num1
            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>Square root of " + num1 + " is " +
myFormat.format("%.2f", squareroot) + ".</p>");

        %>

    </body>
</html>
```

- Output:





- Reflection:

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

   By using href attributes. It is because href attribute specifies the URL of the page that the link goes to or the page next destination, like hyperlink.

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?

   It will show null.

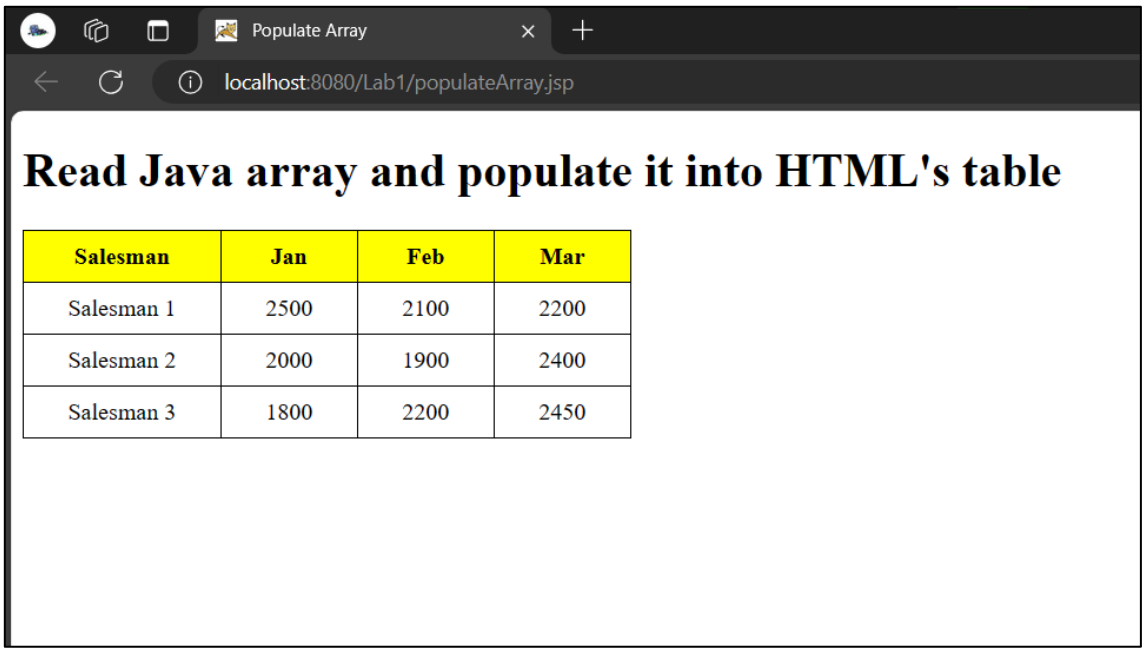
Task 9 : Populate Array values into HTML's Table

- Objective: Read Java array and populate it into HTML's table.
- Code:

```
<%--
    Document   : populateArray
    Created on : 29 Mar 2024, 7:48:21 am
    Author     : Ahmad Afif Syahmi bin Ahmad Rozali
--%>

<%@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>
            table, th, td {
                border: 1px solid black;
                border-collapse: collapse;
                padding: 8px 30px;
                text-align: center;
            }

            th {
                background-color: yellow;
            }
        </style>
    </head>
    <body>

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

        <%
            String[][] sales  = {
                {"Salesman", "Jan", "Feb", "Mar"},
                {"Salesman 1", "2500", "2100", "2200"},
                {"Salesman 2", "2000", "1900", "2400"},
                {"Salesman 3", "1800", "2200", "2450"}
            };
        %>
```

```
        <table border="1">
            <thead>
                <tr>
                    <% for (int j = 0; j < sales[0].length; j++) { %>
                        <th><%= sales[0][j] %></th>
                    <% } %>
                </tr>
            </thead>
            <tbody>
                <% for (int i = 1; i < sales.length; i++) { %>
                    <tr>
                        <% for (int j = 0; j < sales[i].length; j++) {
%>
                            <td><%= sales[i][j] %></td>
                        <% } %>
                    </tr>
                <% } %>
            </tbody>
        </table>

    </body>
</html>
```

- Output:



- Reflection:

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

```
datatype[][] arrayName = new dataType[row][colums];
```

2. Define a sequence of steps on how you accomplish Task 7.
    - Create a new JSP file and name it as useJavaObject.
    - Import java.util.Date.
    - Create Date object and display the current date and time.
    - Use setIntHeader() to trigger refresh function and set the autoload time to 5 seconds.
    - Compile and run the program.

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

    HTML standing for Hypertext Markup Language is a kind of webpage where the content and structure are static, which is often perfect or fast to display regular content on the web. Furthermore, the improvement of JSP (JavaServer Pages) pages is dynamic web pages, which can contain both HTML as well as Java program. In contrast to JSP pages that embed java code inside of an HTML markup, HTML pages allow only a static presentation of content that can be generated using HTML code alone. JSP pages firstly go through the run-time processing from one of the most widely used server-side engines, in which the Java code is interpreted and then executed to create the dynamic HTML content based on the results of the calculation. This enables the development of interactive and data-driven web apps with the capability to fetch data from different sources and tailor the content based on the requested information. Apart from it, it offers the functionalities that are not restricted to the custom tags, but it provides easier interface with Java servlets as well as other backend technology and helps better separate the presentations and business logic.

Exercise 1

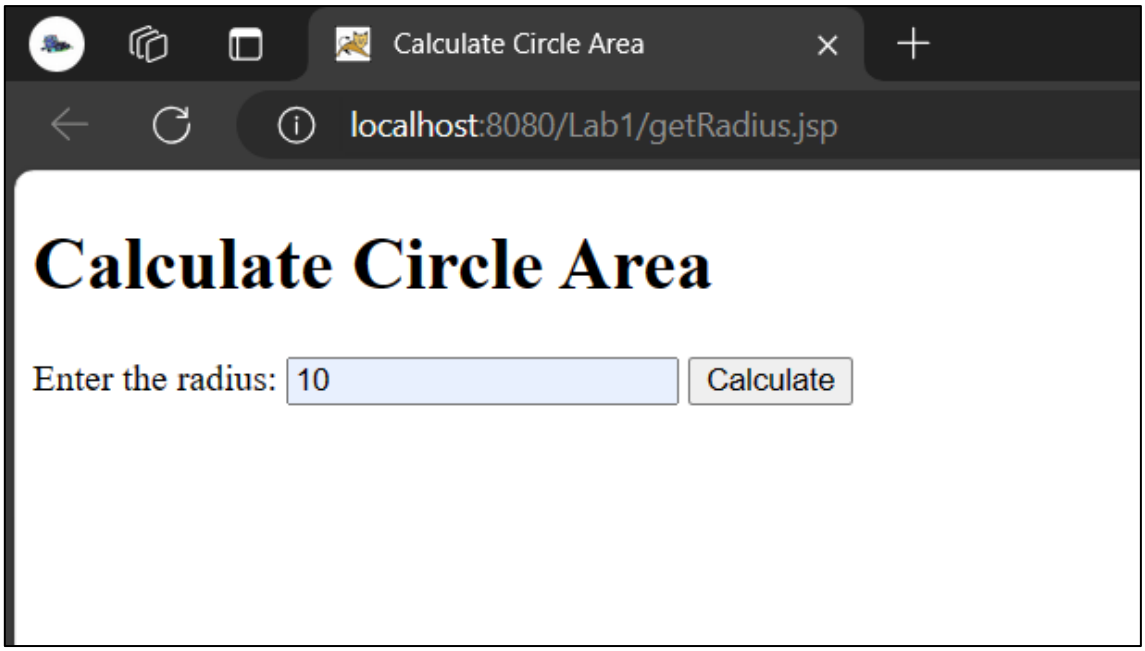
- Objective: To convert a Java program into Web-Based application.
- Code:

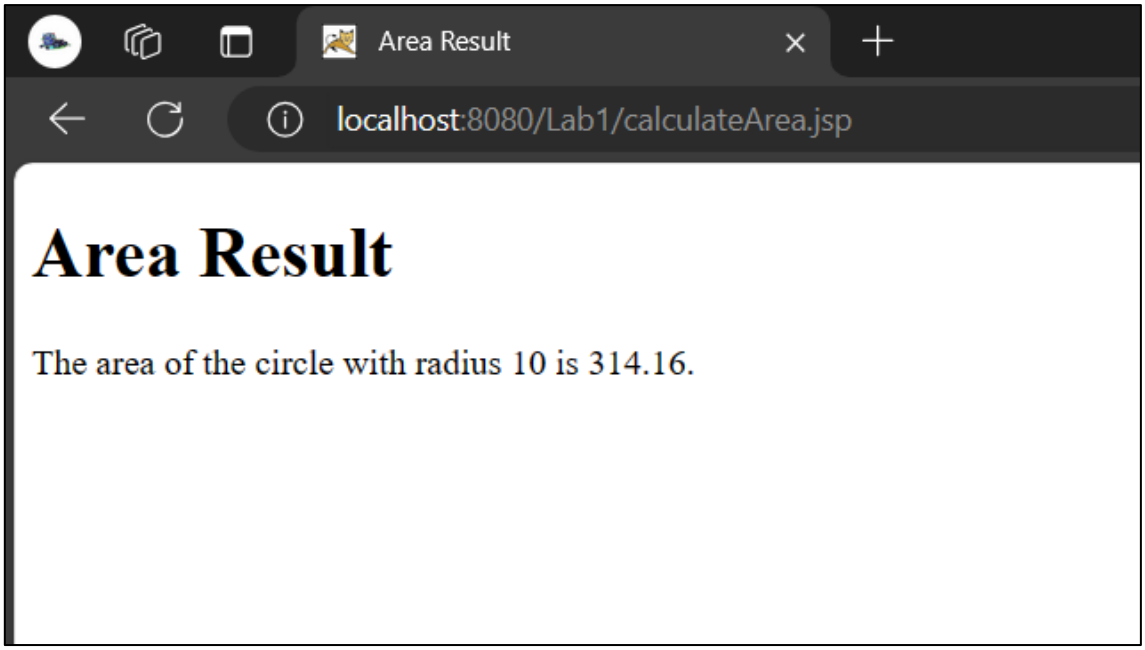
```
<%--
    Document   : getRadius
    Created on : 29 Mar 2024, 8:43:51 am
    Author     : Ahmad Afif Syahmi bin Ahmad Rozali
--%>

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Calculate Circle Area</title>
    </head>
    <body>
        <h1>Calculate Circle Area</h1>
        <form method="post" action="calculateArea.jsp">
            <label for="radius">Enter the radius:</label>
            <input type="text" id="radius" name="radius" pattern="[0-
9]+" title="Please enter a number" required>
            <button type="submit">Calculate</button>
        </form>
    </body>
</html>
```

```
<%--
    Document   : calculateArea
    Created on : 29 Mar 2024, 8:47:27 am
    Author     : Ahmad Afif Syahmi bin Ahmad Rozali
--%>

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ page import="java.util.Scanner" %>
<%@ page import="java.text.DecimalFormat" %>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Area Result</title>
    </head>
    <body>
        <h1>Area Result</h1>
        <%-- Retrieve the radius value from the form --%>
        <% String radiusStr = request.getParameter("radius"); %>

        <%-- Check if the radius value is provided and is a valid
number --%>
```

```
        <% if (radiusStr != null && radiusStr.matches("\\d+")) { %>
            <%-- Convert the radius string to an integer --%>
            <% int radius = Integer.parseInt(radiusStr); %>

            <%-- Calculate the area --%>
            <% double area = Math.PI * radius * radius; %>

            <%-- Format the area value --%>
            <% DecimalFormat df = new DecimalFormat("#.##"); %>

            <%-- Display the result --%>
            <p>The area of the circle with radius <%= radius %> is <%=
df.format(area) %>.</p>
        <% } else { %>
            <%-- Display an error message if the radius value is
missing or not a valid number --%>
            <p>Please enter a valid number for the radius.</p>
        <% } %>
    </body>
</html>
```

- Output:





Exercise 2

- Objective: To create a simple JSP's page to read CSV file.
- Code:

```
<%--
    Document   : readCSVFile
    Created on : 29 Mar 2024, 9:17:24 am
    Author     : Ahmad Afif Syahmi bin Ahmad Rozali
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="java.io.*, java.util.*" %>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>Read CSV File</title>
        <style>
            table, th, td {
                border: 1px solid black;
                border-collapse: collapse;
                padding: 8px 30px;
                text-align: center;
            }

            th {
                background-color: lightgray;
            }
```

```
        </style>
    </head>
    <body>

        <h1>Read CSV files and populate it into HTML's table</h1>

        <%
            // Path to the CSV file
            String filePath = "D:/UMT/Sarjana Muda/SEM 4/Pembangunan
Aplikasi Berasaskan Web/Lab 1/Sales.csv";

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

            // Read the CSV file and populate the list
            try {
                BufferedReader br = new BufferedReader(new
FileReader(filePath));
                String line;
                while ((line = br.readLine()) != null) {
                    String[] parts = line.split(",");
                    csvData.add(parts);
                }
                br.close();
            } 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>
                        <%
                            String[] rowData = csvData.get(i);
                            String customer = rowData[0];
                            String custType = rowData[1];
                            int purchase =
Integer.parseInt(rowData[2]);
                            double discount = 0.0;
                            if (custType.equalsIgnoreCase("Cash")) {
                                discount = purchase * 0.1; // 10%
discount for cash customers
                            }
                        %>
                        <td><%= customer %></td>
                        <td><%= custType %></td>
                        <td><%= purchase %></td>
                        <td><%= String.format("%.2f", discount)
%></td>
```

```
                    </tr>
                <% } %>
            </tbody>
        </table>

    </body>
</html>
```

	A	B	C
1	10001	Credit	5000
2	10002	Cash	2500
3	10003	Credit	3000
4	10004	Cash	3200
5	10005	Credit	1200
6	10006	Cash	6000
7	10007	Credit	7400
8	10008	Cash	800

Sales.csv

- Output: