
Date 4/4/2023

CSM3023

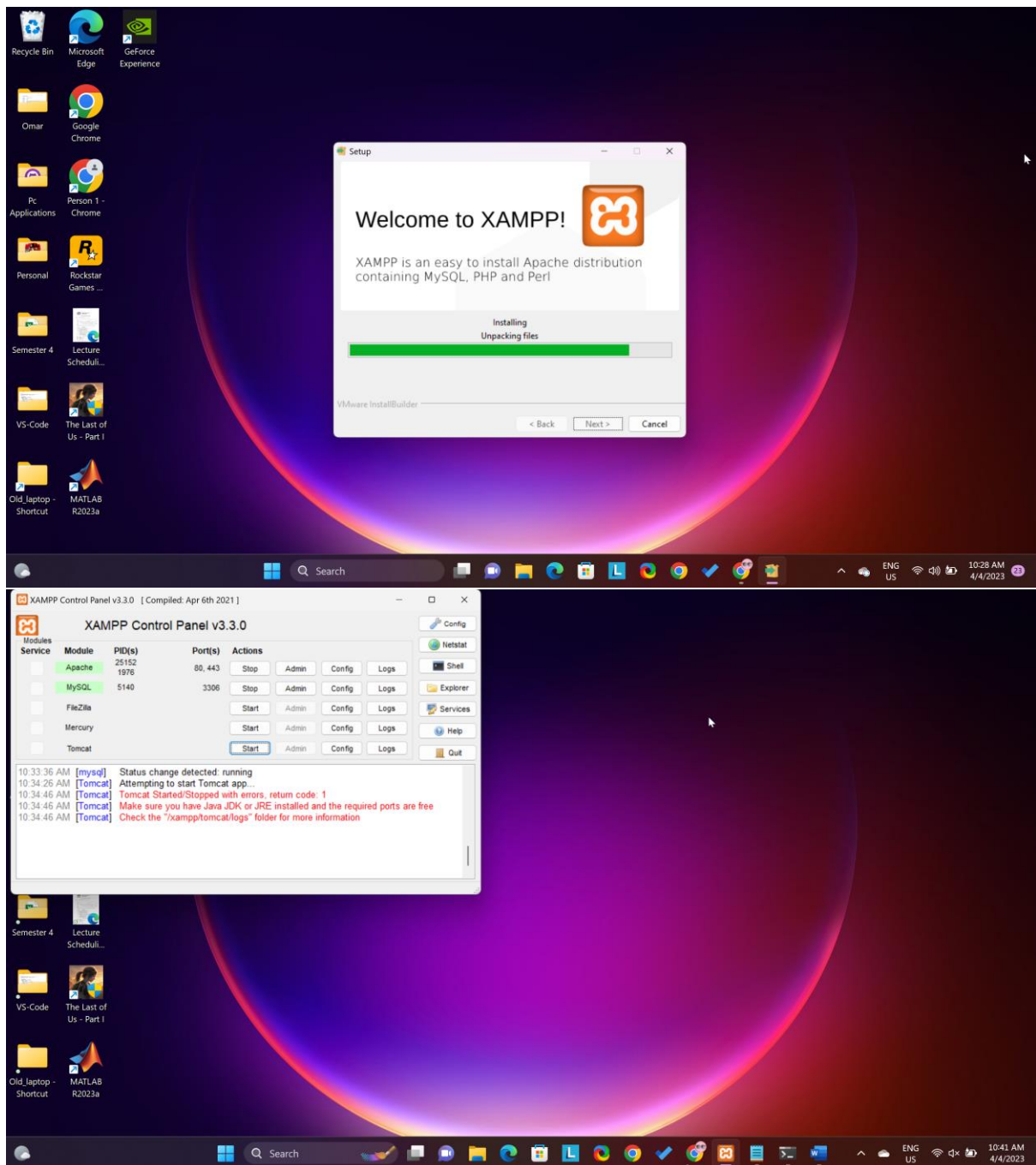
WEB BASED APPLICATION DEVELOPMENT

DR FAIZAH BINTI APLOP APLOP

OMAR ISMAIL ABDJALEEL ALOMORY

LAB 1 (MP2)

XAMPP installation

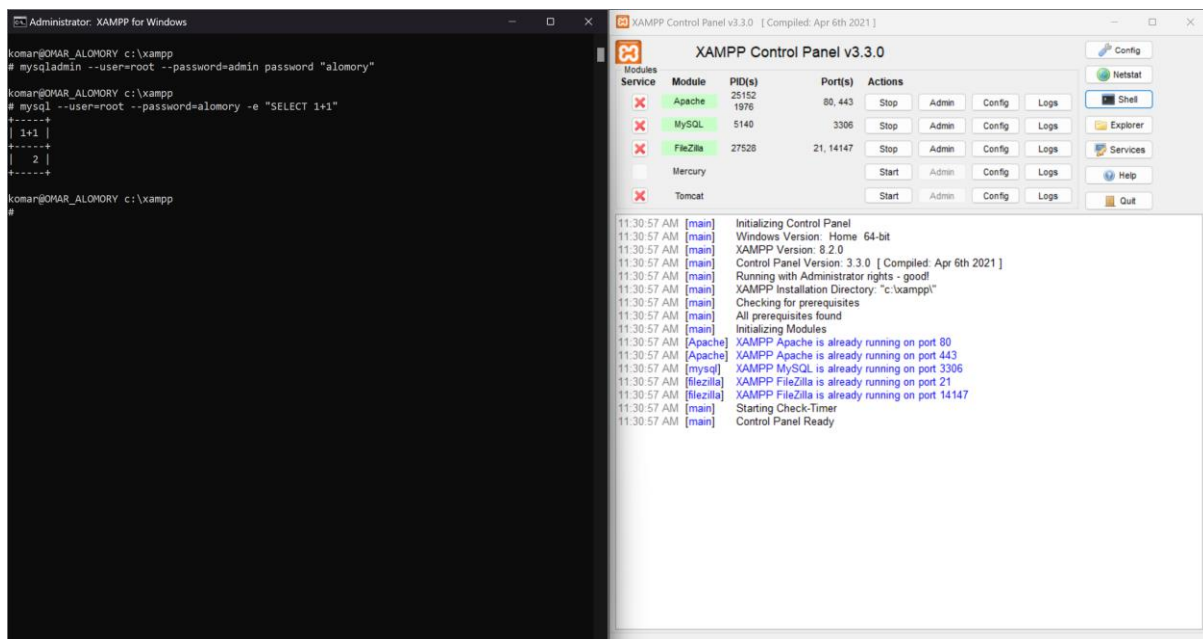
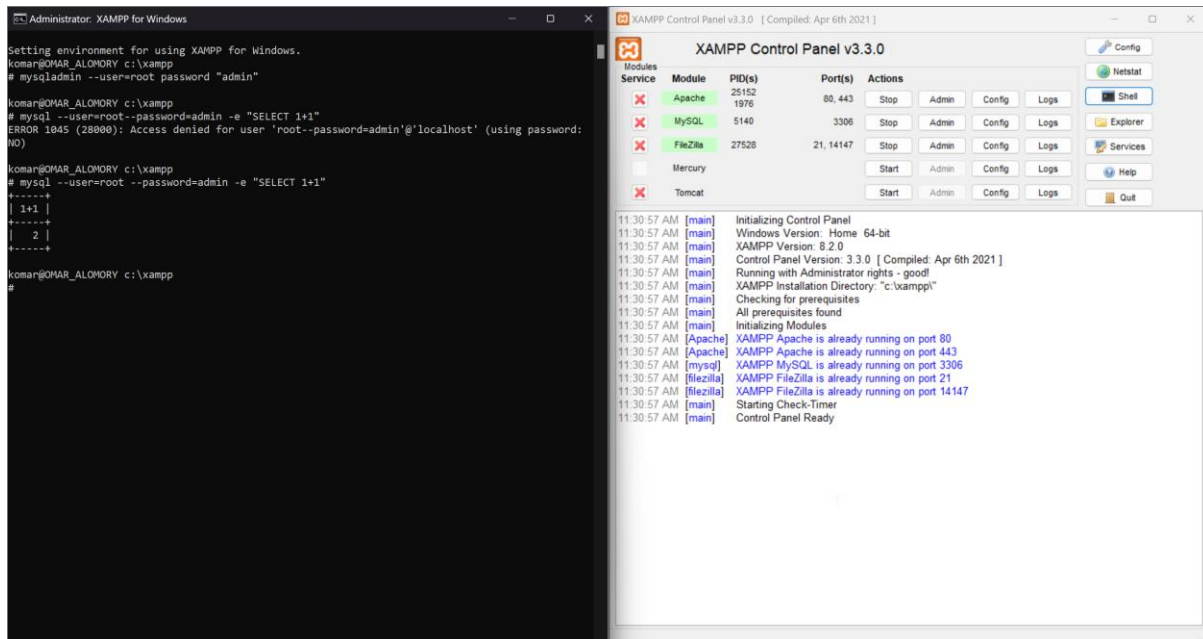


Reflection

What have you learnt from this exercise?

We learned how to install, configure xampp and how to change the port number. It appears that the port no 3036 was used by mySql workbench so eventually we stopped it from running mysql in xampp worked as expected.

Changing the default root password of MySQL Database

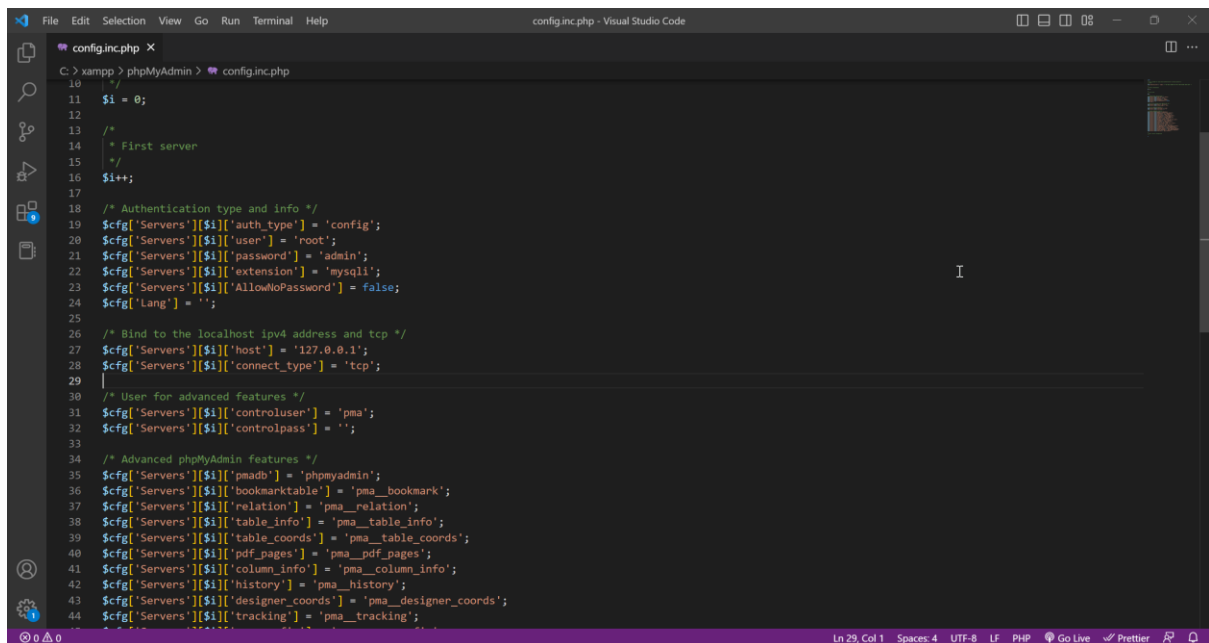


Reflection

What have you learned from this exercise?

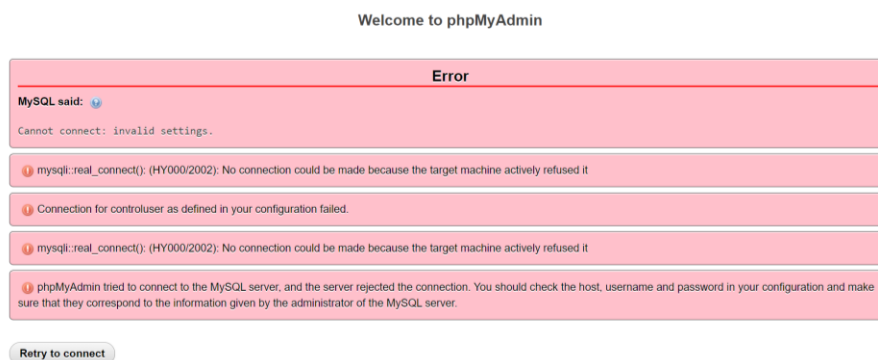
We have learnt how to create password to mysql through shell in xampp, and how to change it and lastly how to confirm if the password has changed or not.

Troubleshooting Notes: Fixing phpMyAdmin Error

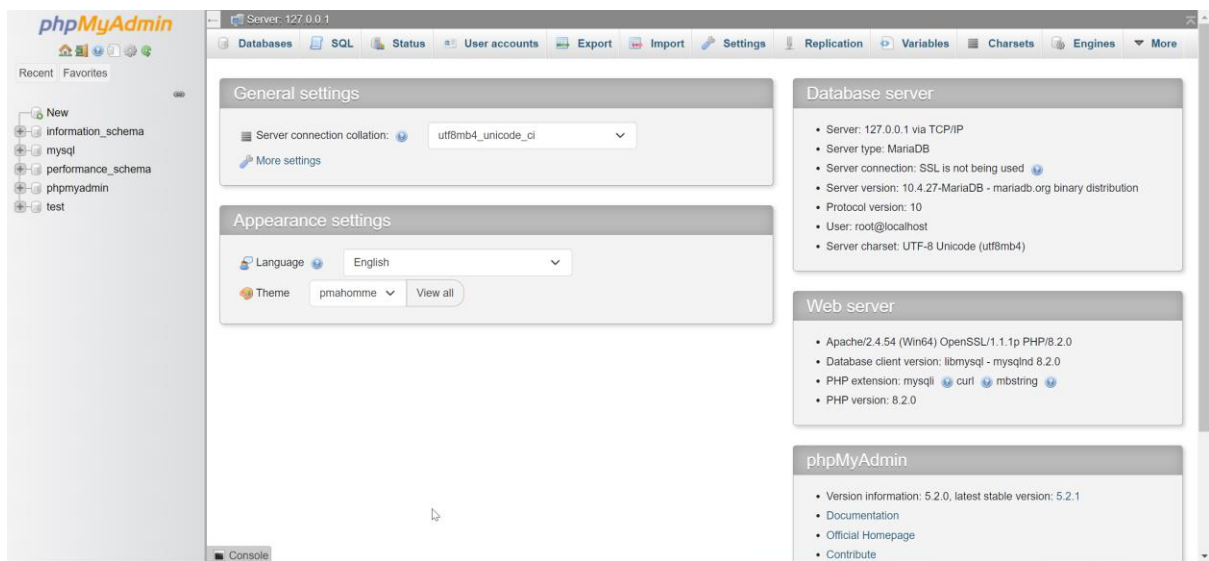


```
10 /*
11 $i = 0;
12
13 /*
14 * First server
15 */
16 $i++;
17
18 /* Authentication type and info */
19 $cfg['Servers'][$i]['auth_type'] = 'config';
20 $cfg['Servers'][$i]['user'] = 'root';
21 $cfg['Servers'][$i]['password'] = 'admin';
22 $cfg['Servers'][$i]['extension'] = 'mysql';
23 $cfg['Servers'][$i]['AllowNoPassword'] = false;
24 $cfg['Lang'] = '';
25
26 /* Bind to the localhost ipv4 address and tcp */
27 $cfg['Servers'][$i]['host'] = '127.0.0.1';
28 $cfg['Servers'][$i]['connect_type'] = 'tcp';
29
30 /* User for advanced features */
31 $cfg['Servers'][$i]['controluser'] = 'pma';
32 $cfg['Servers'][$i]['controlpass'] = '';
33
34 /* Advanced phpMyAdmin features */
35 $cfg['Servers'][$i]['pmadb'] = 'phpmyadmin';
36 $cfg['Servers'][$i]['bookmarktable'] = 'pma_bookmark';
37 $cfg['Servers'][$i]['relation'] = 'pma_relation';
38 $cfg['Servers'][$i]['table_info'] = 'pma_table_info';
39 $cfg['Servers'][$i]['table_coords'] = 'pma_table_coords';
40 $cfg['Servers'][$i]['pdf_pages'] = 'pma_pdf_pages';
41 $cfg['Servers'][$i]['column_info'] = 'pma_column_info';
42 $cfg['Servers'][$i]['history'] = 'pma_history';
43 $cfg['Servers'][$i]['designer_coords'] = 'pma_designer_coords';
44 $cfg['Servers'][$i]['tracking'] = 'pma_tracking';
```

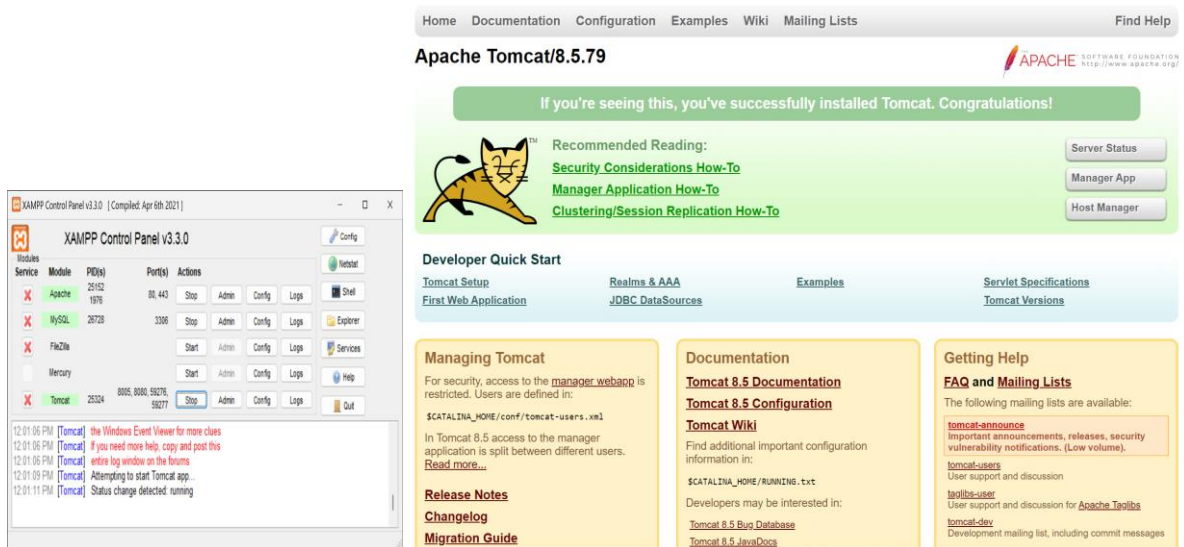
Changing the user and password configuration from the config.inc to remove the error shown in phpMyAdmin



Now once we click reconnect it should work.



Managing Apache Tomcat



The image shows two side-by-side screenshots. On the left is the XAMPP Control Panel v3.3.0 window, which displays a table of services including Apache, MySQL, FileZilla, Mercury, and Tomcat. The Tomcat service is highlighted in red and has a 'Start' button. Below the table is a log window showing messages from 12:01:06 PM to 12:01:11 PM, indicating that Tomcat is starting successfully. On the right is the Apache Tomcat 8.5.79 homepage. It features a navigation bar with links like Home, Documentation, Configuration, Examples, Wiki, and Mailing Lists. A green banner at the top says 'If you're seeing this, you've successfully installed Tomcat. Congratulations!'. Below this is a 'Recommended Reading' section with links to 'Security Considerations How-To', 'Manager Application How-To', and 'Clustering/Session Replication How-To'. There are also buttons for 'Server Status', 'Manager App', and 'Host Manager'. The 'Developer Quick Start' section includes links for 'Tomcat Setup', 'Realms & AAA', 'Examples', 'Servlet Specifications', and 'Tomcat Versions'. The 'Managing Tomcat' section provides instructions on accessing the manager webapp and lists links for 'Release Notes', 'Changelog', and 'Migration Guide'. The 'Documentation' section links to 'Tomcat 8.5 Documentation', 'Tomcat 8.5 Configuration', and 'Tomcat Wiki'. The 'Getting Help' section lists 'FAQ and Mailing Lists' and provides links to various mailing lists like 'tomcat-announce', 'tomcat-users', 'tomcat-dev', and 'tomcat-tasks'.

Tomcat is running and the homepage is working.

401 Unauthorized

You are not authorized to view this page. If you have not changed any configuration files, please examine the file `conf/tomcat-users.xml` in your installation. That file must contain the credentials to let you use this webapp.

For example, to add the `manager-gui` role to a user named `tomcat` with a password of `s3cret`, add the following to the config file listed above.

```
<role rolename="manager-gui"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
```

Note that for Tomcat 7 onwards, the roles required to use the manager application were changed from the single `manager` role to the following four roles. You will need to assign the role(s) required for the functionality you wish to access.

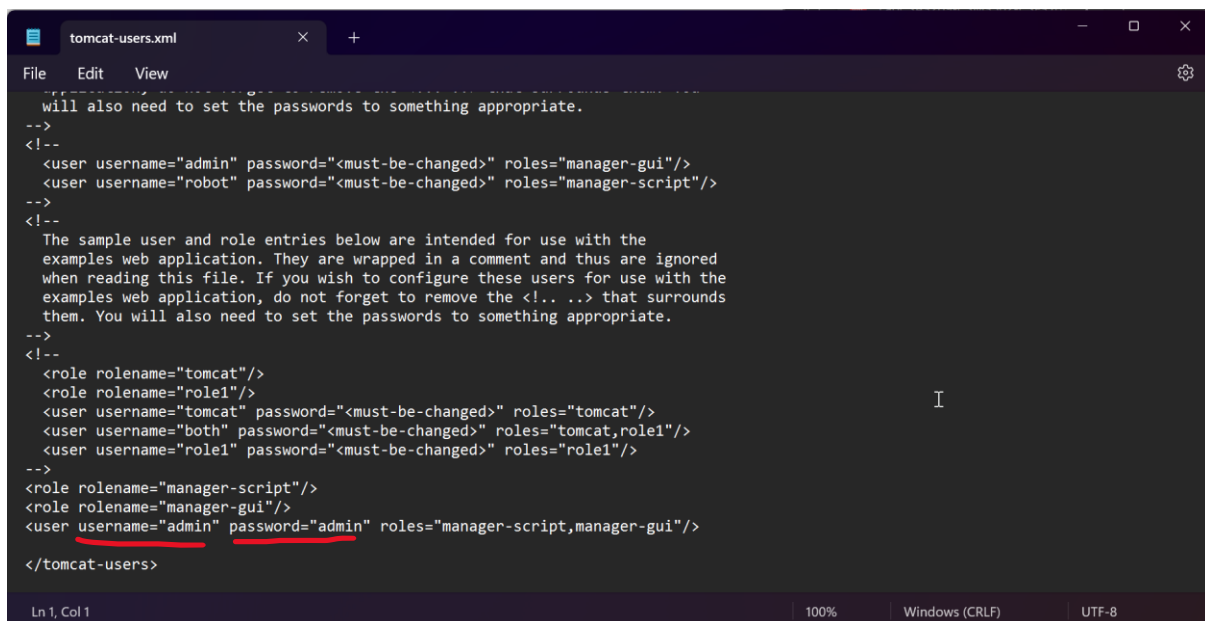
- `manager-gui` - allows access to the HTML GUI and the status pages
- `manager-script` - allows access to the text interface and the status pages
- `manager-jmx` - allows access to the JMX proxy and the status pages
- `manager-status` - allows access to the status pages only

The HTML interface is protected against CSRF but the text and JMX interfaces are not. To maintain the CSRF protection:

- Users with the `manager-gui` role should not be granted either the `manager-script` or `manager-jmx` roles.
- If the text or jmx interfaces are accessed through a browser (e.g. for testing since these interfaces are intended for tools not humans) then the browser must be closed afterwards to terminate the session.

For more information - please see the [Manager App How-To](#).

Since we don't have user yet this page is shown, in the next steps we will create new user.



The image shows a text editor window titled 'tomcat-users.xml'. The editor contains XML code for defining roles and users. The code is as follows:

```
<!--
will also need to set the passwords to something appropriate.
-->
<!--
<user username="admin" password="<must-be-changed>" roles="manager-gui"/>
<user username="robot" password="<must-be-changed>" roles="manager-script"/>
-->
<!--
The sample user and role entries below are intended for use with the
examples web application. They are wrapped in a comment and thus are ignored
when reading this file. If you wish to configure these users for use with the
examples web application, do not forget to remove the <!-- ... --> that surrounds
them. You will also need to set the passwords to something appropriate.
-->
<!--
<role rolename="tomcat"/>
<role rolename="role1"/>
<user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
<user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
<user username="role1" password="<must-be-changed>" roles="role1"/>
-->
<role rolename="manager-script"/>
<role rolename="manager-gui"/>
<user username="admin" password="admin" roles="manager-script,manager-gui"/>
-->
</tomcat-users>
```

The status bar at the bottom of the editor shows 'Ln 1, Col 1', '100%', 'Windows (CRLF)', and 'UTF-8'.



Tomcat Web Application Manager

Message:

OK

Manager

List ApplicationsHTML Manager HelpManager HelpServer Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	<div>Start Stop Reload Undeploy</div> <div>Expire sessions with idle > 30 minutes</div>
/docs	None specified	Tomcat Documentation	true	0	<div>Start Stop Reload Undeploy</div> <div>Expire sessions with idle > 30 minutes</div>
/examples	None specified	Servlet and JSP Examples	true	0	<div>Start Stop Reload Undeploy</div> <div>Expire sessions with idle > 30 minutes</div>
/host-manager	None specified	Tomcat Host Manager Application	true	0	<div>Start Stop Reload Undeploy</div> <div>Expire sessions with idle > 30 minutes</div>
/manager	None specified	Tomcat Manager Application	true	1	<div>Start Stop Reload Undeploy</div> <div>Expire sessions with idle > 30 minutes</div>

Deploy

Deploy directory or WAR file located on server

Context Path (required)

XML Configuration file path

WAR or Directory path

Deploy

WAR file to deploy

Select WAR file to upload

Choose File

No file chosen

Deploy

Configuration

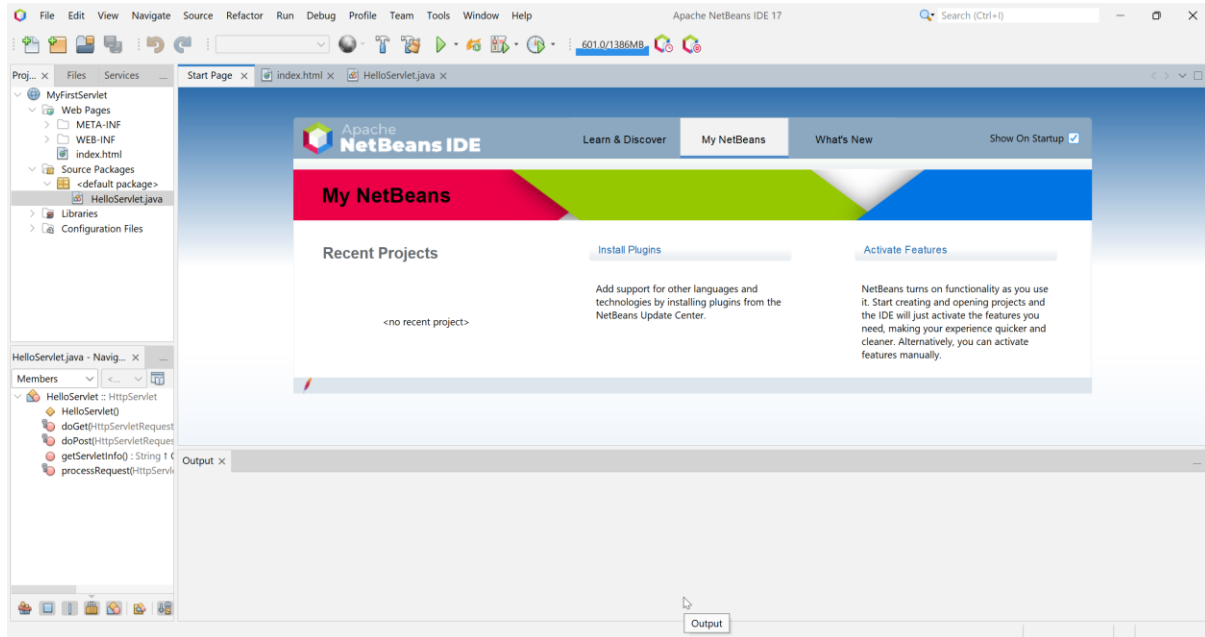
Re-read TLS configuration files

TLS (Secret engine: DefaultSunJSSE)

Everything seems to work perfectly now.

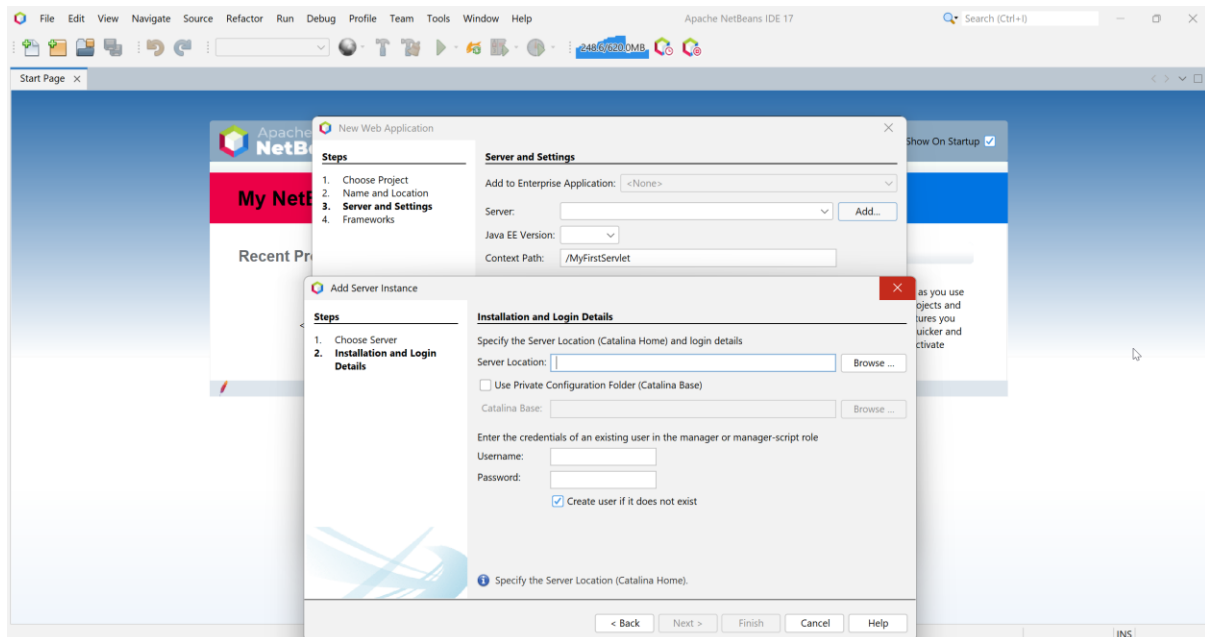
NetBeans 12.3 IDE Installation.

Sorry I could not put the installation process because I already have NetBeans in my laptop.

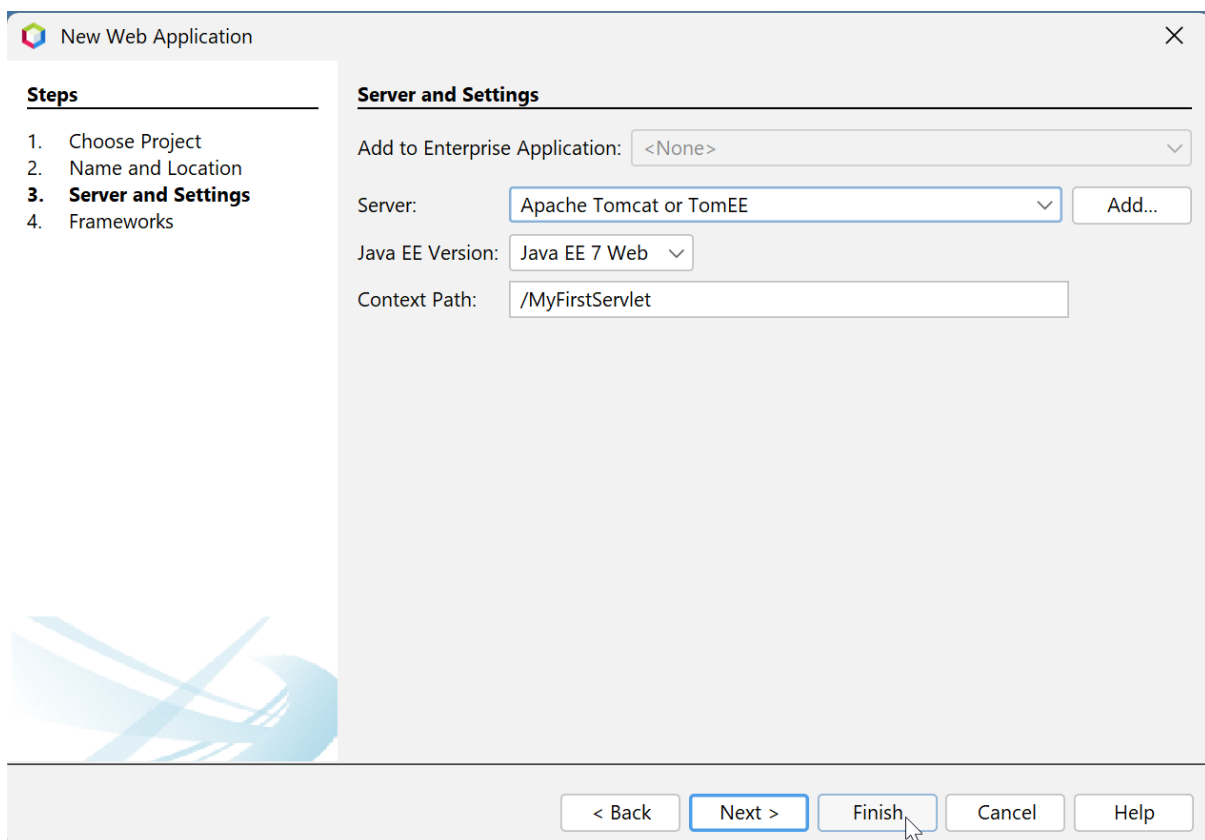


Task 5: Linking NetBeans to Apache Tomcat and Writing a Simple Java Servlet

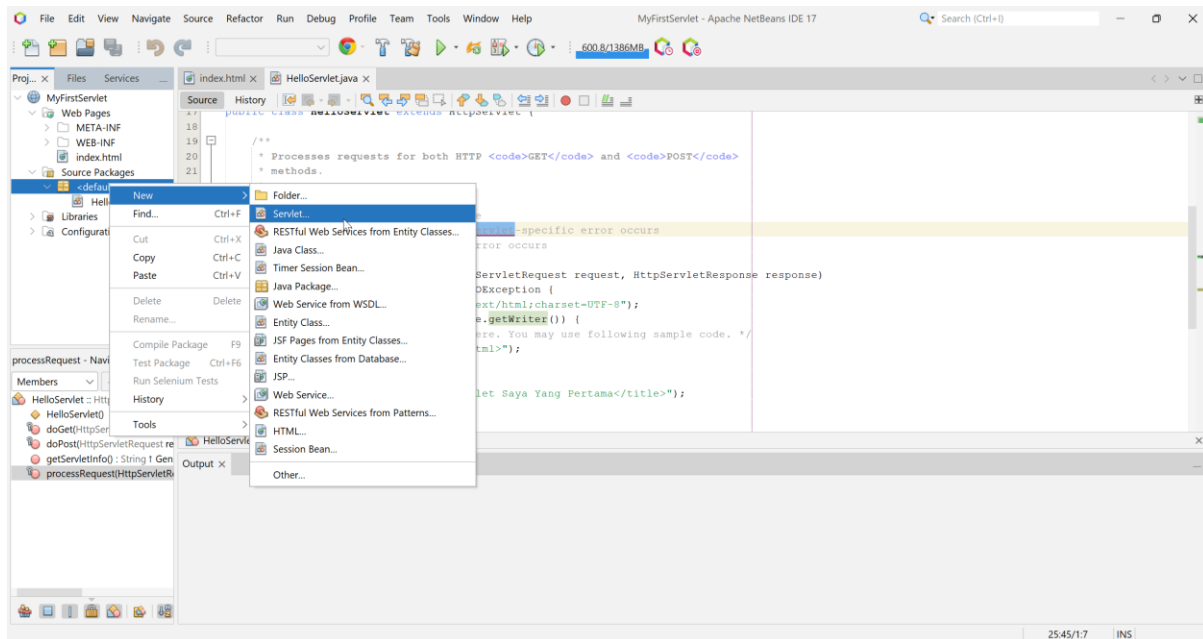
Linking tomcat server to the project



After choosing the folder for the project we need to choose server too. After clicking add we choice tomcat tomEE



Now, we have to create servlet file.



New Servlet

Steps

1. Choose File Type
2. Name and Location
3. **Configure Servlet Deployment**

Configure Servlet Deployment

Register the Servlet with the application by giving the Servlet an internal name (Servlet Name). Then specify patterns that identify the URLs that invoke the Servlet. Separate multiple patterns with commas.

☒ Add information to deployment descriptor (web.xml)

Class Name:

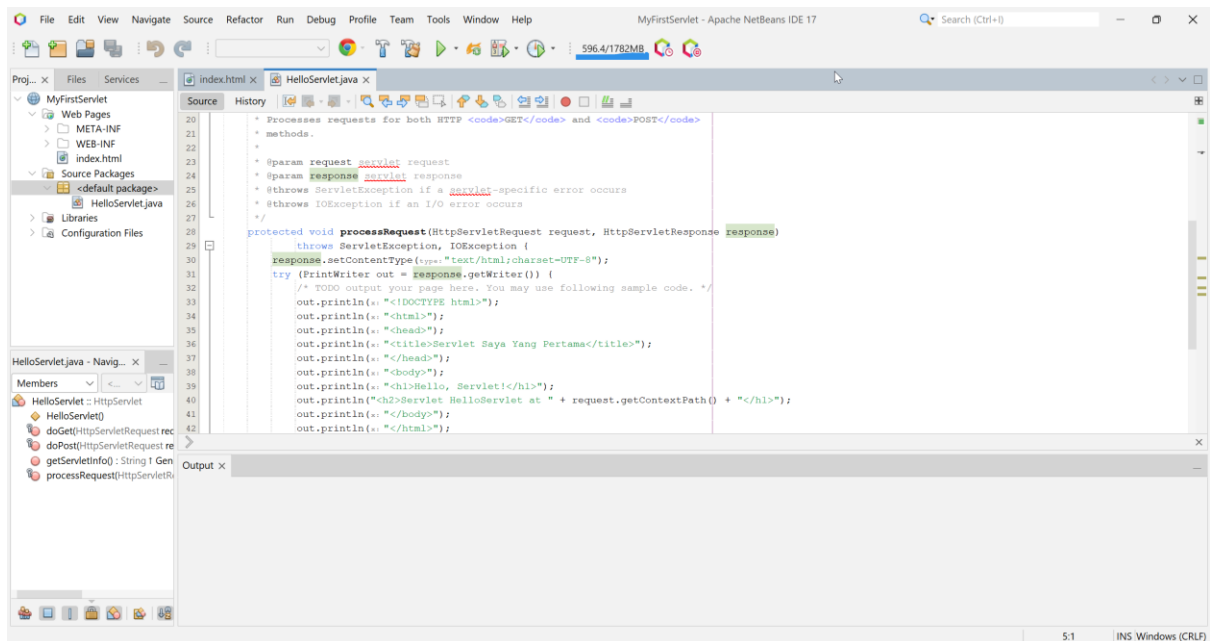
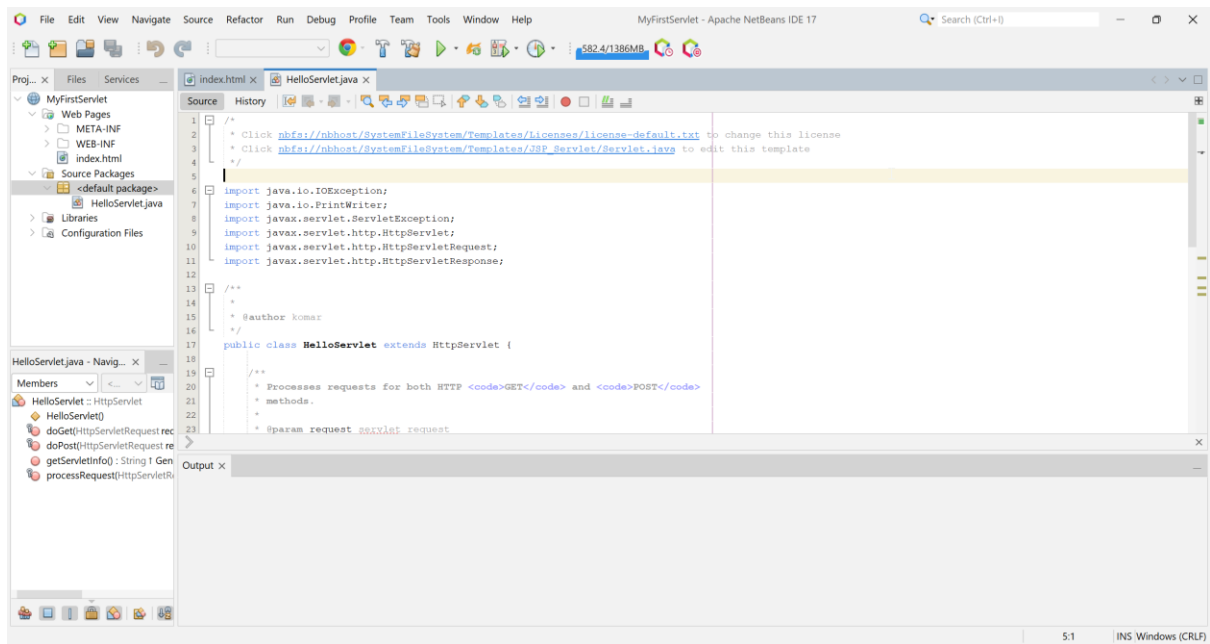
Servlet Name:

URL Pattern(s):

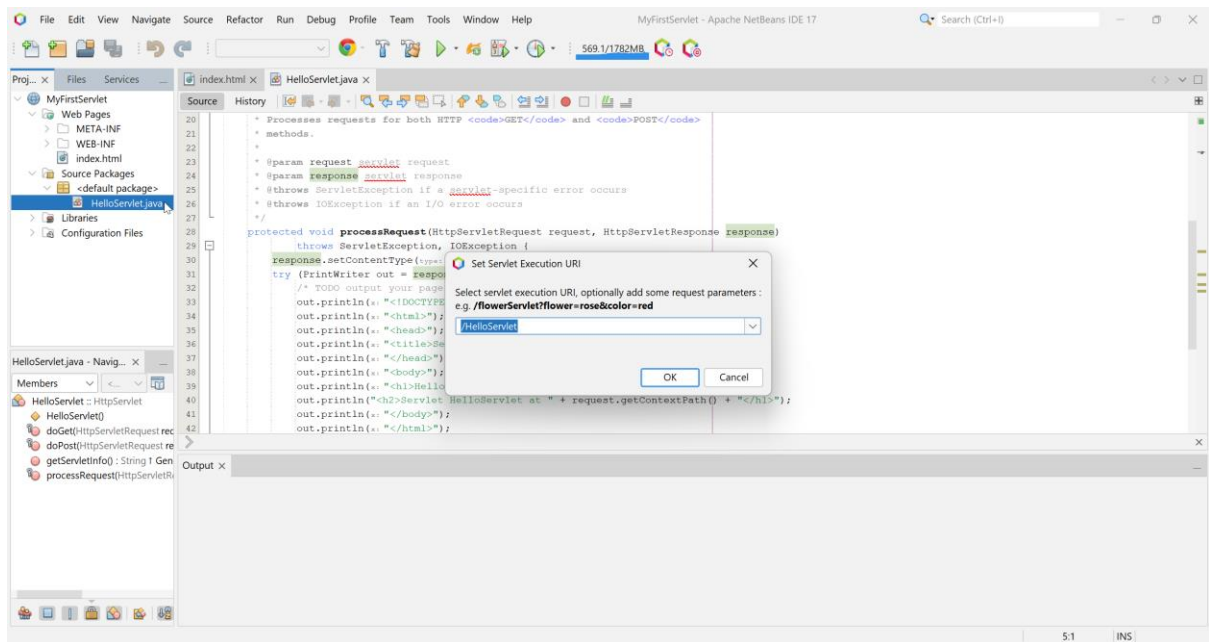
Initialization Parameters:

Name	Value
------	-------

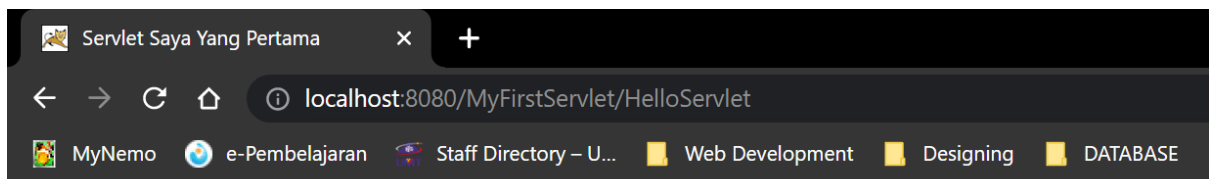
HelloServlet.Java



Run the file.



Output



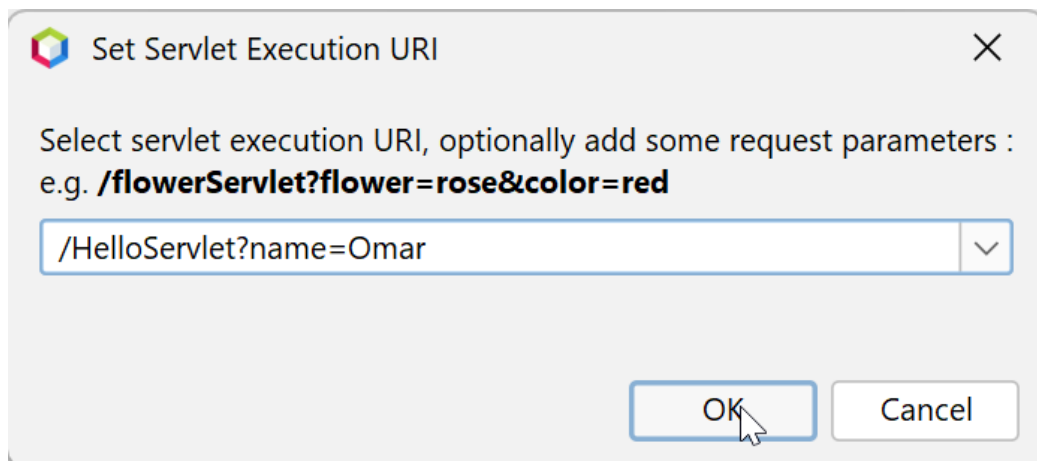
Hello, Servlet!

Servlet HelloServlet at /MyFirstServlet

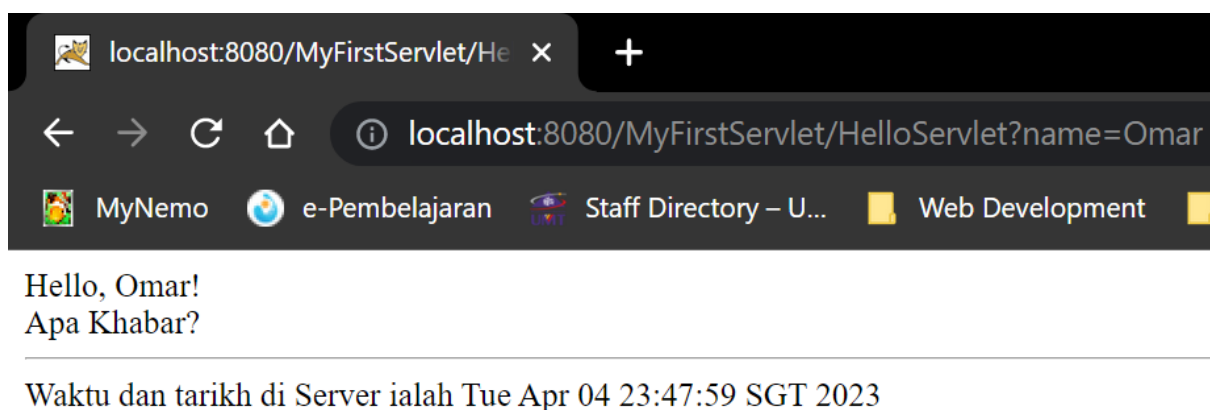
Modification to our servlet file so that it accept parameter from user's request along with using `java.util.Date()` class to display the date and time.

```
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
//    processRequest(request, response);
    response.setContentType(type: "text/html");
    PrintWriter out = response.getWriter();
    String name = request.getParameter(name: "name");
    out.println(x: "<html><body>");
    out.println("Hello, "+name+"!<br>");
    out.println(x: "Apa Khabar? <hr>");
    out.println("Waktu dan tarikh di Server ialah "+new java.util.Date());
    out.println(x: "</body></html>");
}
```

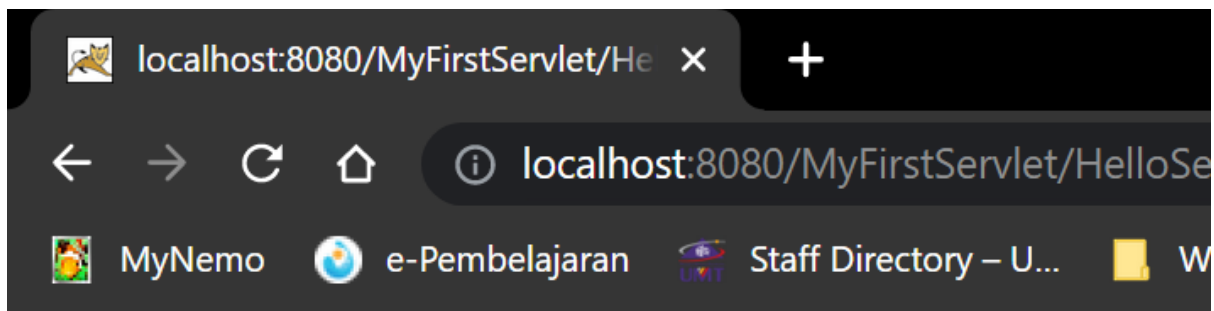
Passed parameter = Omar 😊



Output



If name have no value



Hello, null!
Apa Khabar?

Waktu dan tarikh di Server ialah Tue Apr 04 23:52:22 SGT 2023

If there is no value passed to the parameter we can write if statement to check if name is equal to no value then, we can process anything we want. For example we will processRequest() method if no value is passed.

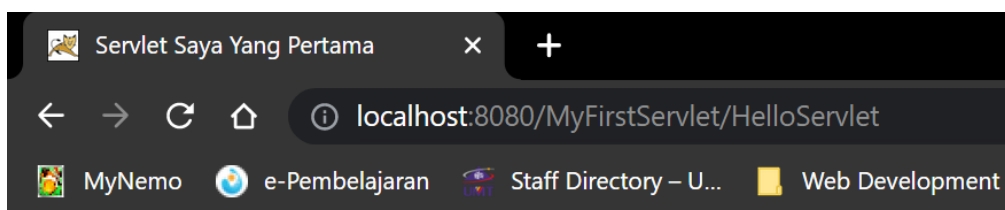
```
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    response.setContentType(type: "text/html");
    PrintWriter out = response.getWriter();
    String name = request.getParameter(name: "name");

    if(name == null)processRequest(request, response);

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

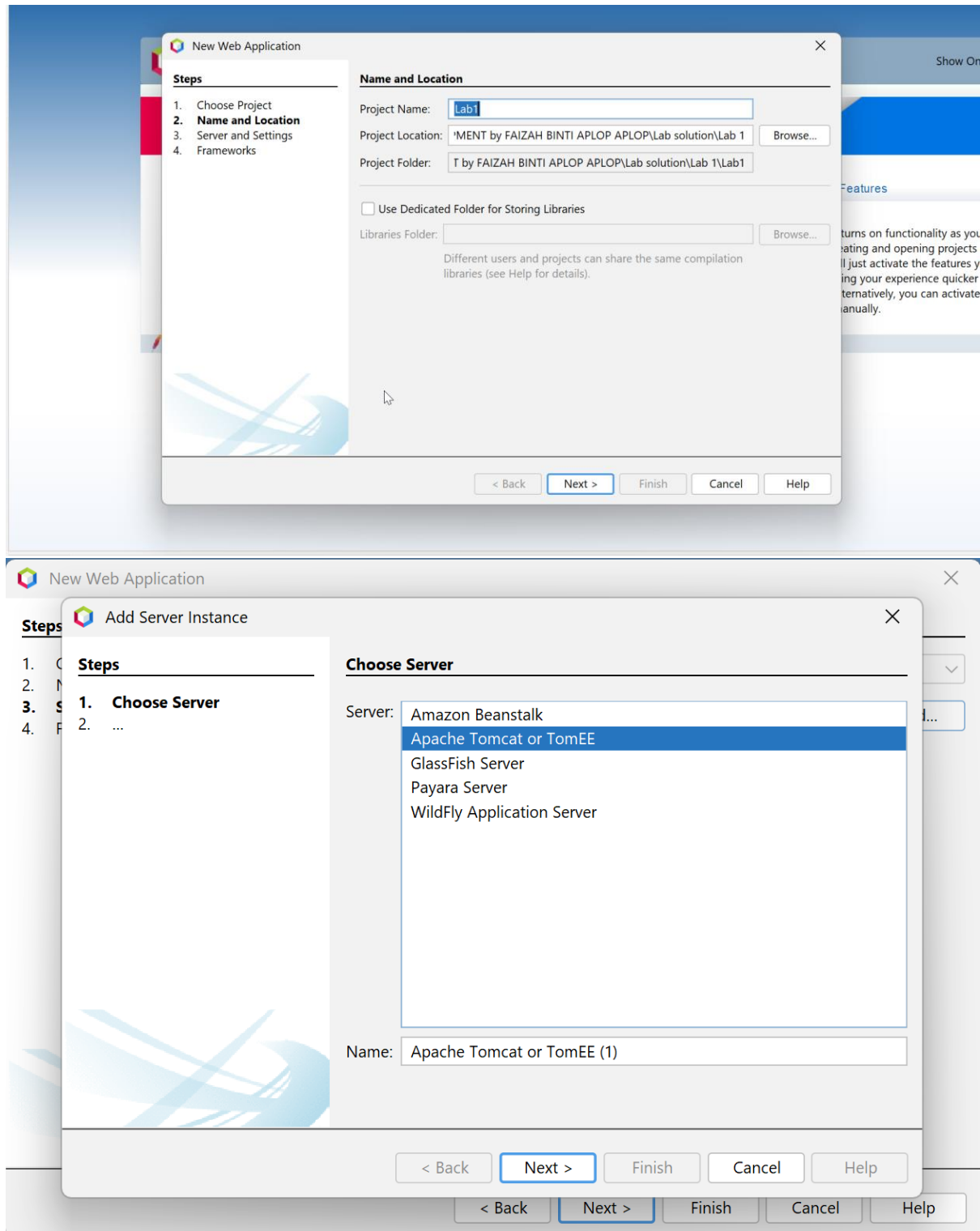
Output

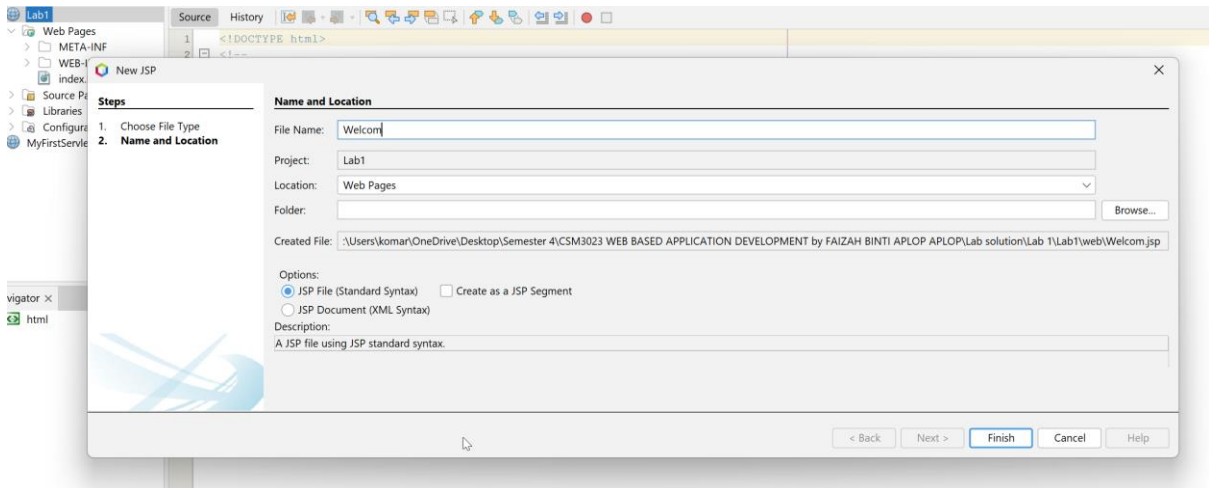
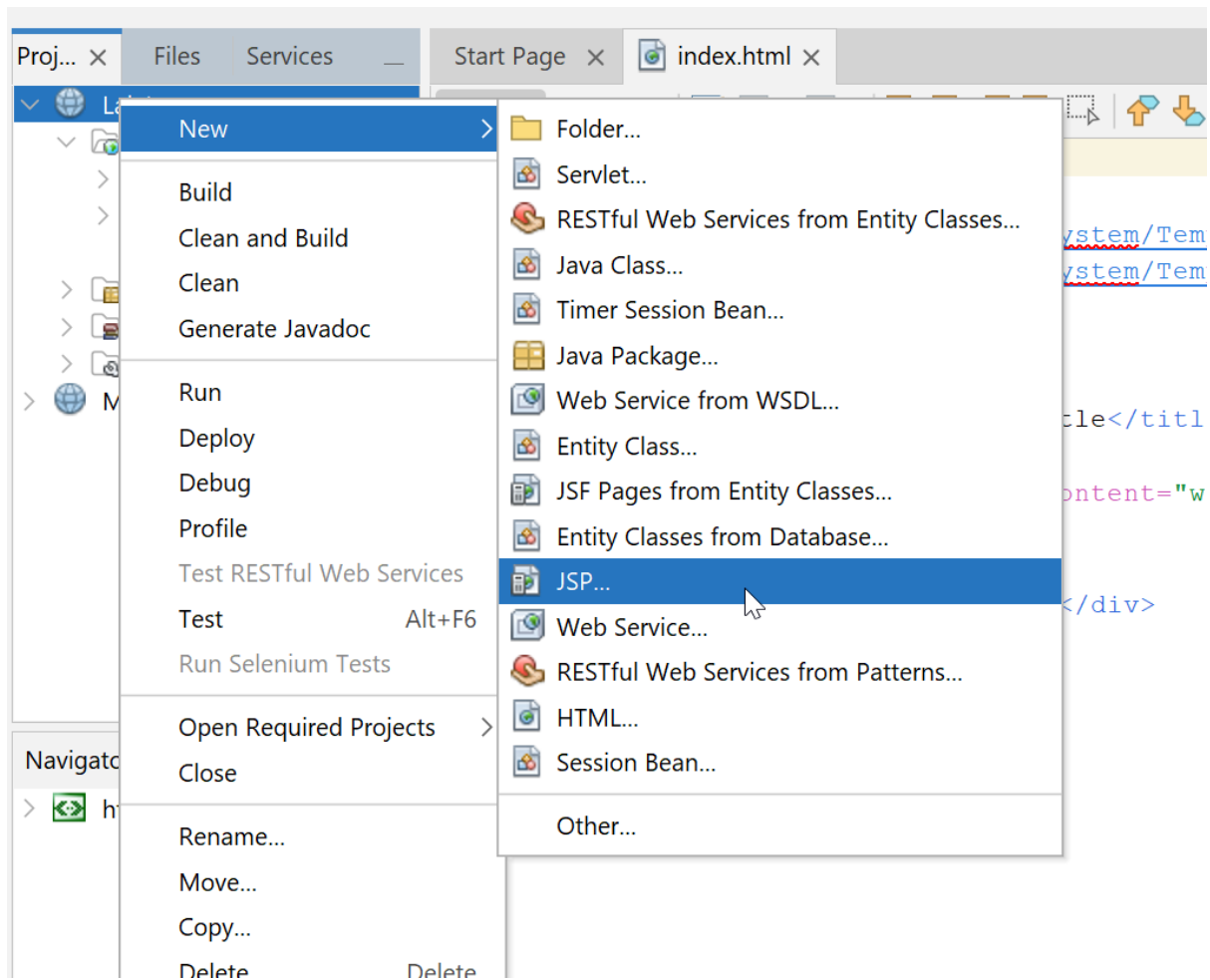


Hello, Servlet!

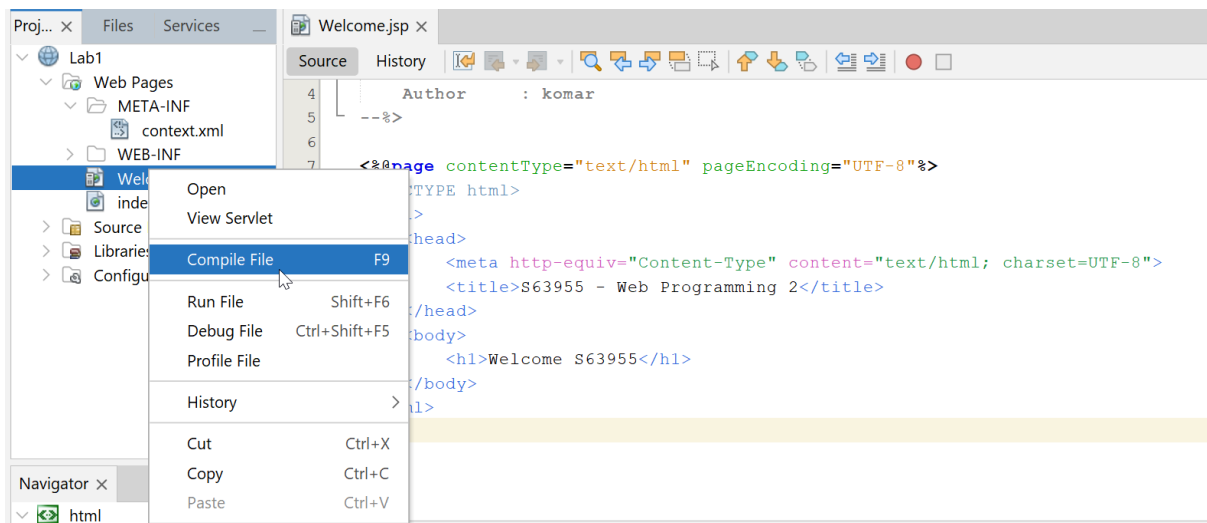
Servlet HelloServlet at /MyFirstServlet

Task 6: Writing a Simple JSP Program





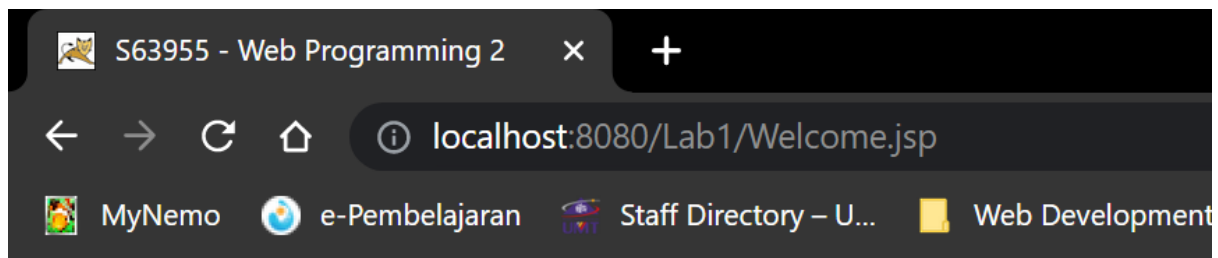
Compile the welcome.jsp file



Compiled successfully.



Output



Welcome S63955

Reflection

1. What have you learned from this exercise?

I learned how to create dynamic HTML web page using JSP, its easier in term of writing and going sequentially with the code.

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

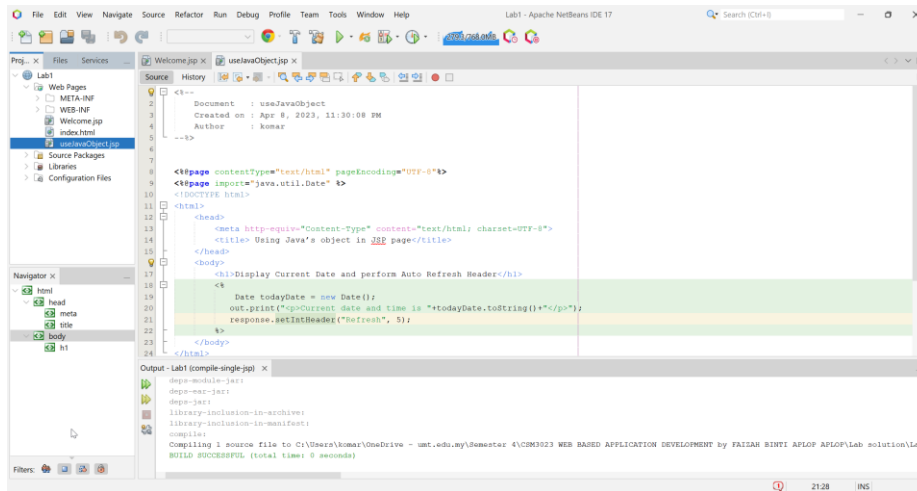
JSP file is compiled into a Java Servlet, which generates dynamic content based on Java code and data retrieved from other sources. The resulting HTML content is sent back to the user's browser as an HTTP response, which is then rendered and displayed to the user.

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?

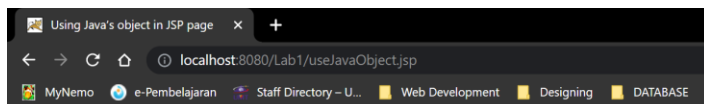
The differences I found in task 1 and 4 is that servlet are Java classes that handle HTTP requests and responses, while JSPs are text-based documents that are compiled into Servlets. Servlets use Java code to generate dynamic content, while JSPs use a combination of HTML and Java code. Servlets are more efficient than JSPs but are less accessible to non-Java developers and web designers. At the end Servlets and JSPs are both technologies used for creating dynamic web applications in Java.

Task 7

File compiled successfully.



Output



Display Current Date and perform Auto Refresh Header

Current date and time is Sat Apr 08 23:42:52 SGT 2023

Reflection

1. What have you learnt from this exercise?

in this task I have learnt how use scriptlet to refresh the header automatically using java objects response to refresh the page every 5 second, with each refresh the time and date will be generated by the object Date(). Then all of this will be displayed in HTML file.

2. What is Java Scriptlet?

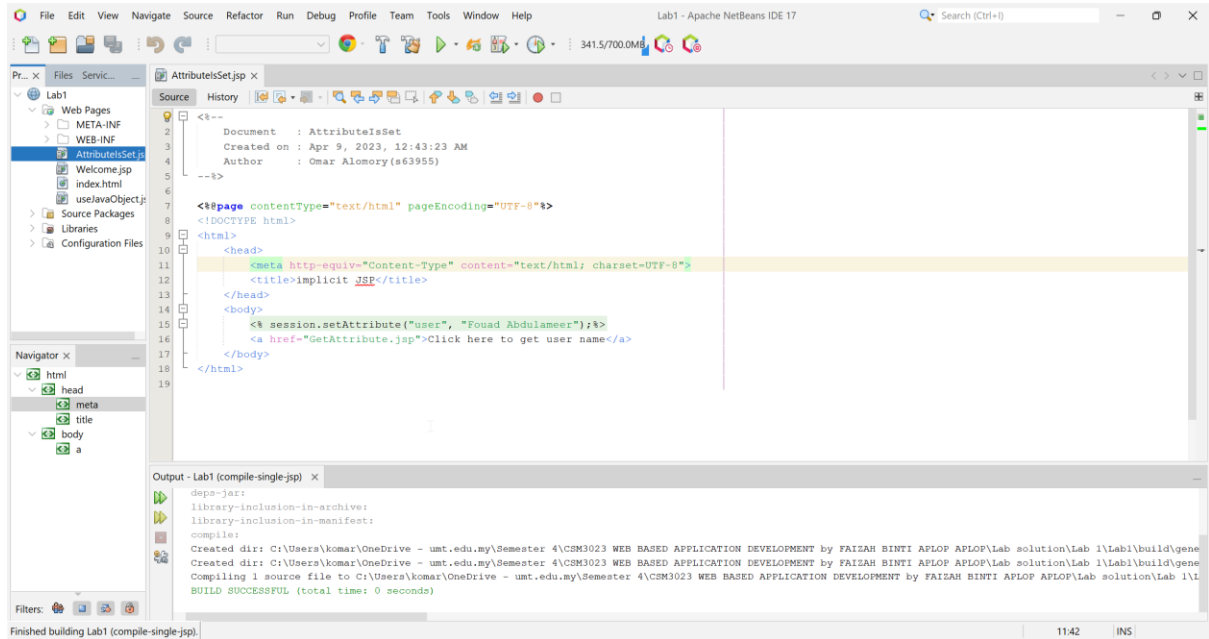
A scriptlet is a piece of software code that is used by a native Web page scripting language to perform a specific function or process. Scriptlets are primarily implemented in JSP and include variables, expressions or statements that are used only when requested by a certain client or process.

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

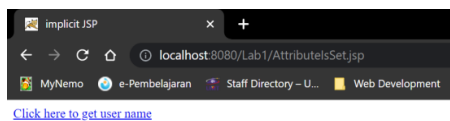
We can use it inside the scriptlet `<%code fragment%>`, if we want to import any libraries, we can import it inside `<%@ page ... %>`.

Task 8

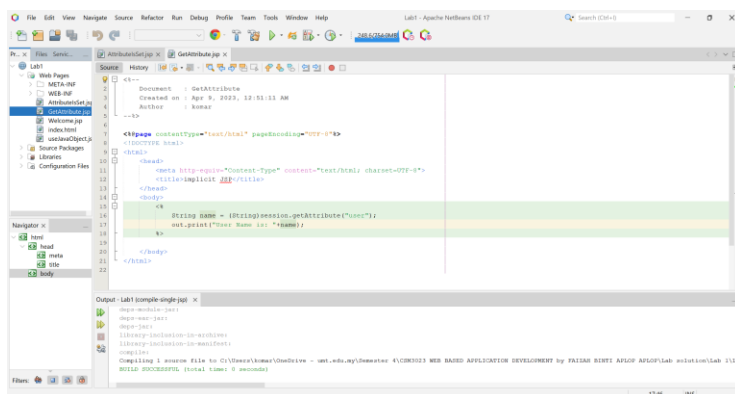
Compiled successfully(AttributelsSet.jsp).



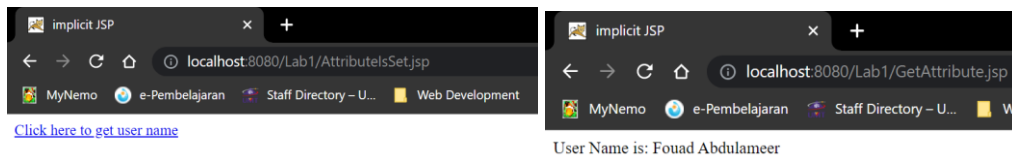
Output (AttributelsSet.jsp)



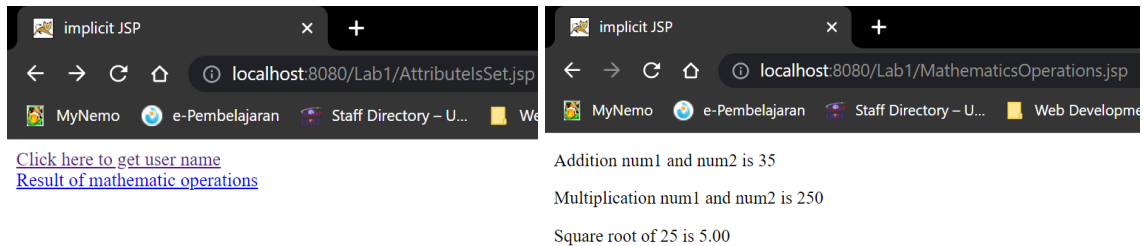
Compiled successfully(GetAttribute.jsp)



Output(AttributelsSet.jsp + GetAttribute.jsp)



Output(AttributelsSet.jsp + MathematicsOperations.jsp)



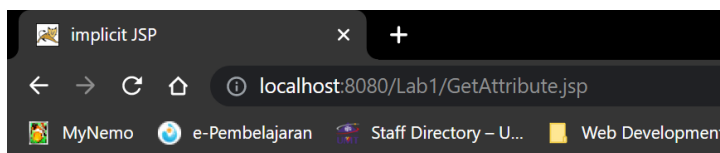
Reflection

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

We can use the HTTP GET or POST method to submit the form data, with GET method appending the data to the URL while POST method sends data in the HTTP request body. We can also use hidden form fields or session variables to store specific information that needs to be passed from one form to the next. The approach you choose will depend on your specific requirements and the nature of the data you need to pass.

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?

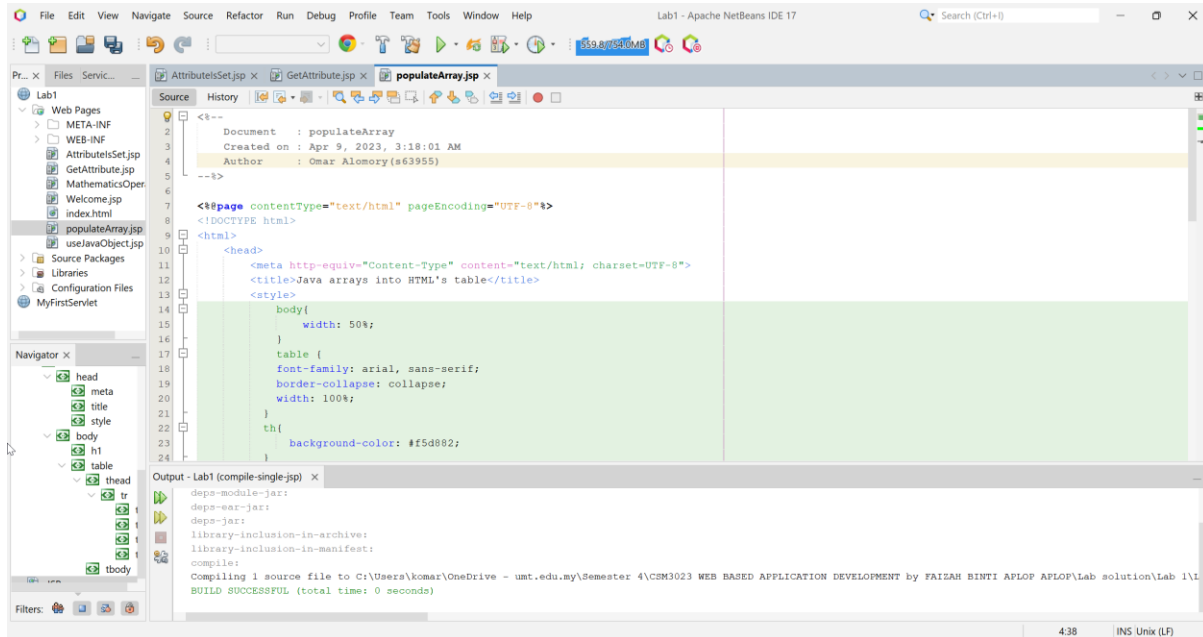
```
<%  
    String name = (String)session.getAttribute("somethingElse");  
    out.print("User Name is: "+name);  
%>
```



As we can see, if the field name is different, the value would be null because it can not recognize the attribute name we want to pass.

Task 9

Compiled successfully.



Source code:

```
<!--
Document   : populateArray
Created on : Apr 9, 2023, 3:18:01 AM
Author    : Omar Alomory(s63955)
-->

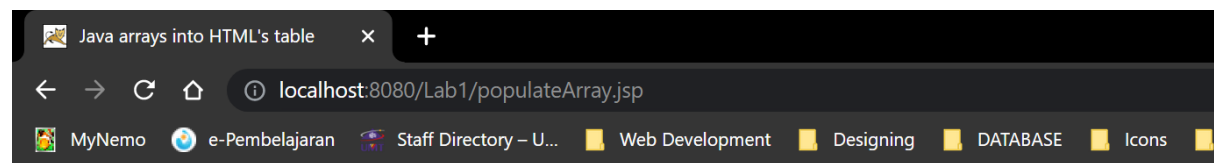
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Java arrays into HTML's table</title>
    <style>
      body{
        width: 50%;
      }
      table {
        font-family: arial, sans-serif;
        border-collapse: collapse;
        width: 100%;
      }
      th{
        background-color: #f5d882;
      }
      td{
        background-color: #f8f1df;
      }
      td, th {
        border: 1px solid #dddddd;
        text-align: center;
        padding: 8px;
      }
    </style>
  </head>
```

```

</head>
<body>
    <h1>Read Java array and populate it into HTML's table</h1>
    <table>
        <thead>
            <tr>
                <th>Salesmen</th>
                <th>Jan</th>
                <th>Feb</th>
                <th>Mac</th>
            </tr>
        </thead>
        <tbody>
            <%
                //initializing the 2D array
                int data[][] = {
                    {2500,2100,2200},
                    {2000,1900,2400},
                    {1800,2200,2450}
                };
                // populating 2D array in HTML using java for loops and HTML's tags
                for(int i = 0; i< data.length; i++){
                    out.print("<tr><td>Salesmen"+(i+1)+"</td>");
                    for(int j = 0; j< data[i].length; j++){
                        out.print("<td>" + data[i][j] + "</td>");
                    }
                    out.print("</tr>");
                }
            %>
        </tbody>
    </table>
</body>
</html>

```

Output



Read Java array and populate it into HTML's table

Salesmen	Jan	Feb	Mac
Salesmen1	2500	2100	2200
Salesmen2	2000	1900	2400
Salesmen3	1800	2200	2450

Reflection

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

datatype variableName [][] = new datatype[noOfRows][noOfColumns(optional)];

or second way is initialization and assigning values directly.

datatype variableName [][] = {

{value1, value2, value3}, {value4, etc}

};

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

After initialization the 2D array and assign the required values into it, then we write scriptlet to read and display the values in HTML table as follow:

1. Create nested for loop to read the values from 2D array.
2. In the first loop we will have to display the salesmen name so we will use both java object and HTML tags, `out.print("<tr><td>Salesmen"+(i+1)+"</td>");` . since the salesmen name are combined with number we will use the variable from the first loop "i +1" (i+1 because we start from index 0). `<tr>` and `<td>` tags to display it in HTML table and lastly `</td>` to close the tag.
3. In the second loop we will use the same steps as 1, but we will display only `<td>` only, because we want to show from column 2 to 4, don't forget to close the `</td>` at the same line.
4. Just after going from the second loop close `</tr>` tag, because we finished one row.
5. Loops will be repeated till it finish row*column.

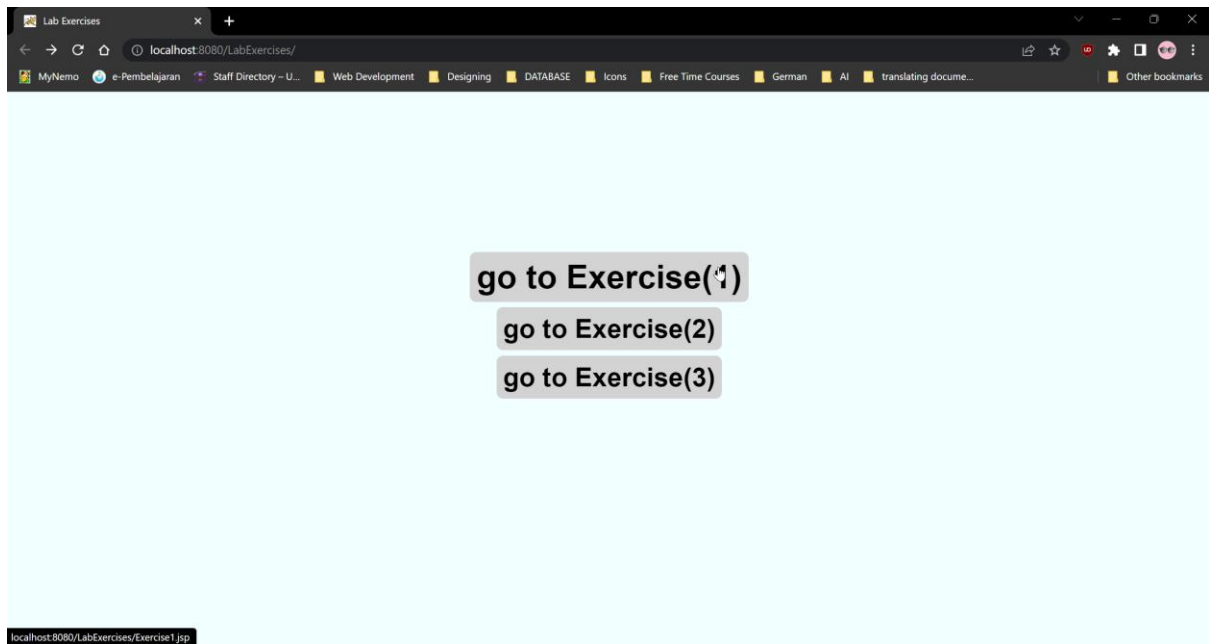
```
<%
//initializing the 2D array
int data[][] = {
    {2500,2100,2200},
    {2000,1900,2400},
    {1800,2200,2450}
};
// populating 2D array in HTML using java for loops and HTML's tags
for(int i = 0; i< data.length; i++){
    out.print("<tr><td>Salesmen"+(i+1)+"</td>");
    for(int j = 0; j< data[i].length; j++){
        out.print("<td>"+data[i][j]+"</td>");
    }
    out.print("</tr>");
}
%>
```

Figure 2.1 show sequence on how to accomplish Task 7

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

HTML is a markup language used for creating static web pages, while JSP is a server-side technology used to create dynamic web pages. The main difference between them is that HTML pages are static, while JSP pages are dynamic and can contain Java code for server-side processing tasks such as interacting with databases or processing user input.

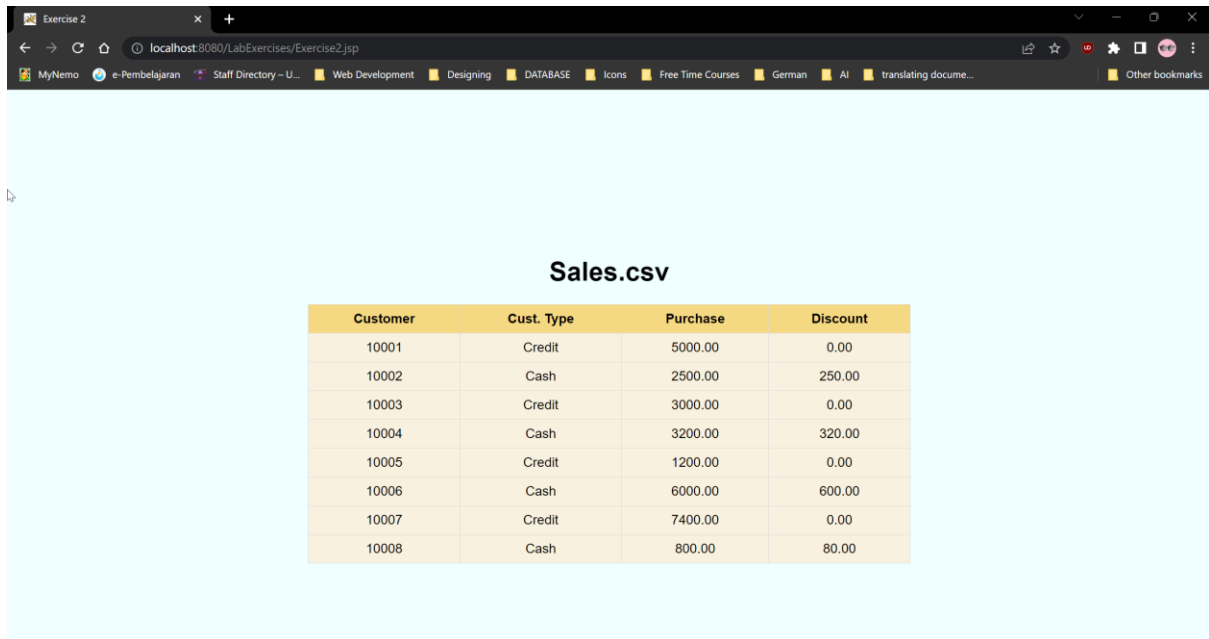
Lab Exercises (Home)



Exercise 1



Exercise 2



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/LabExercises/Exercise2.jsp'. The browser's bookmark bar is visible at the top, containing various links like 'MyNemo', 'e-Pembelajaran', 'Staff Directory', 'Web Development', 'Designing', 'DATABASE', 'Icons', 'Free Time Courses', 'German', 'AI', 'translating docume...', and 'Other bookmarks'. The main content area of the browser displays a table titled 'Sales.csv' on a light blue background. The table has four columns: 'Customer', 'Cust. Type', 'Purchase', and 'Discount'. It contains eight rows of data, with customer IDs ranging from 10001 to 10008. The 'Purchase' and 'Discount' columns contain numerical values representing currency.

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

END OF LAB 1