



UNIVERSITI MALAYSIA TERENGGANU

CSM3023 WEB-BASED APPLICATION DEVELOPMENT (K1)

BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONORS

LAB 1 - INTRODUCTION TO SERVLET, JSP AND MYSQL DATABASE

SEMESTER II 2023/2024

Prepared for:

DR. MOHAMAD NOR HASSAN

Prepared by:

LUQMAN HAKIM BIN AZIZ

(S66292)



Week 1

Introduction to Servlet, JSP and MySQL Database

Web Based Application
Development



Lecturers

PUSAT PENGAJIAN INFORMATIK DAN MATEMATIK GUNAAN
(PPIMG), UNIVERSITI MALAYSIA TERENGGANU (UMT)

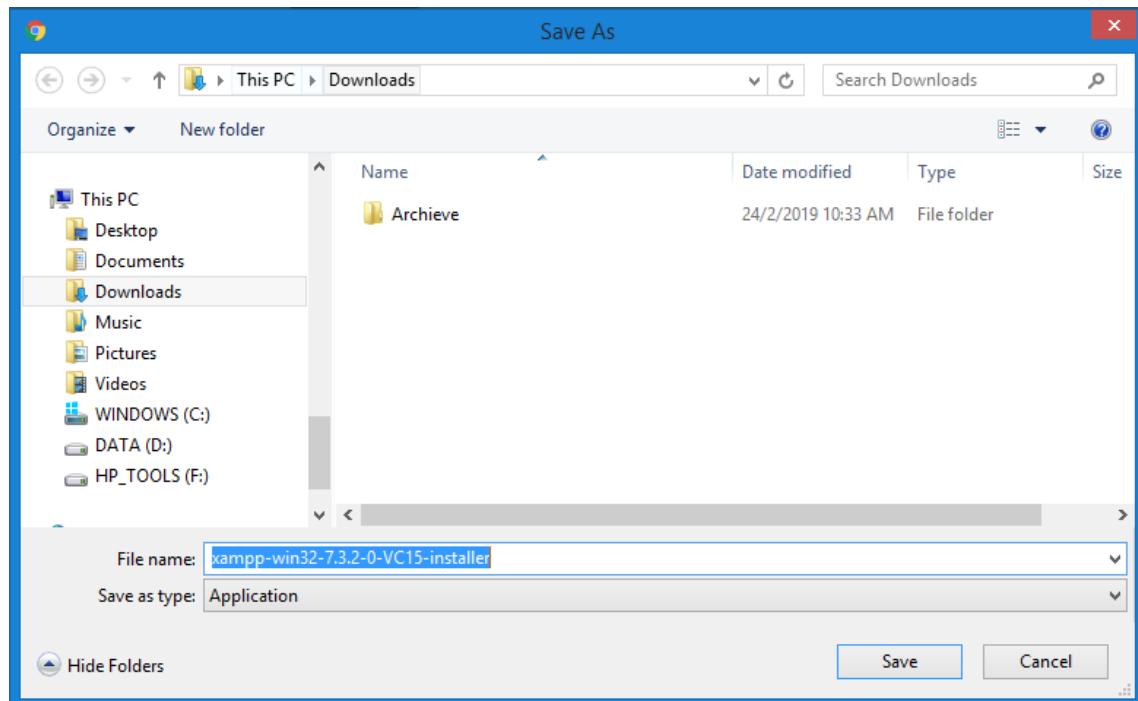
Task 1: Apache Tomcat and MySQL Installation Using XAMPP

Objective	: Installation of Apache Tomcat and MySQL
Problem	: To install Apache Tomcat and MySQL
Description	
Estimated time	: 25 minutes

1. Go to the browser and type URL
<http://www.apachefriends.org/en/xampp.html>
2. Click to XAMPP for Windows.



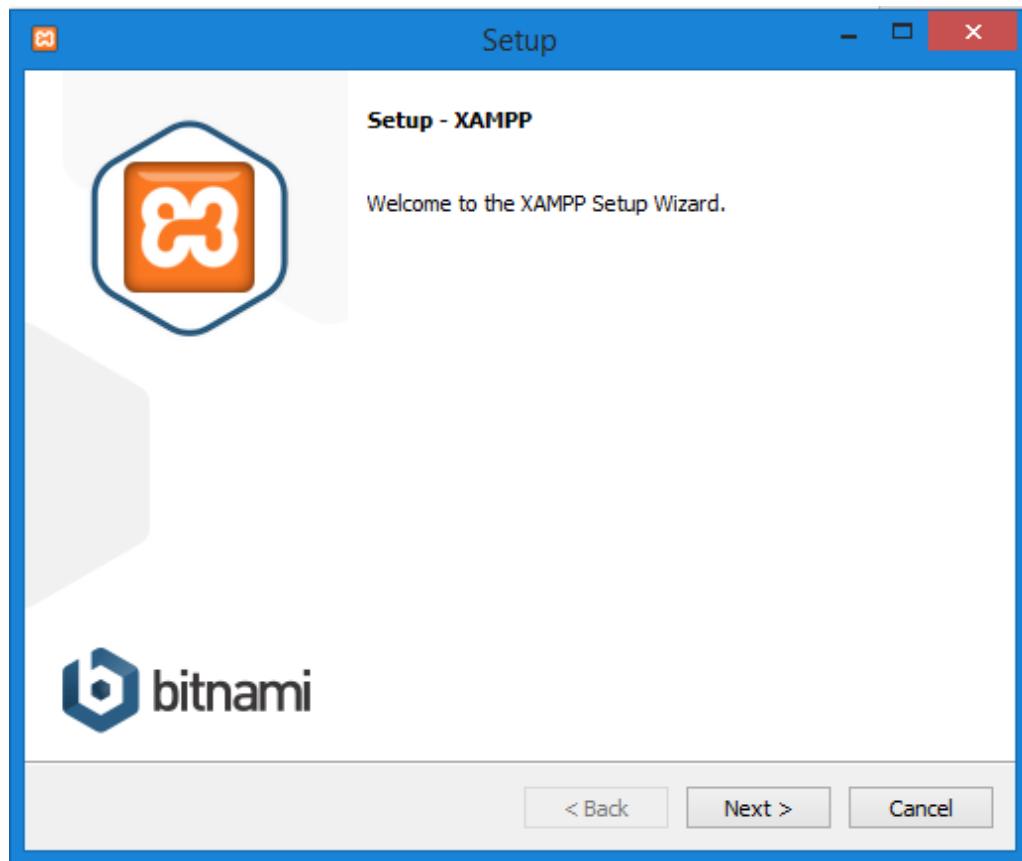
3. Save to a specific directory – for example, the Downloads folder.



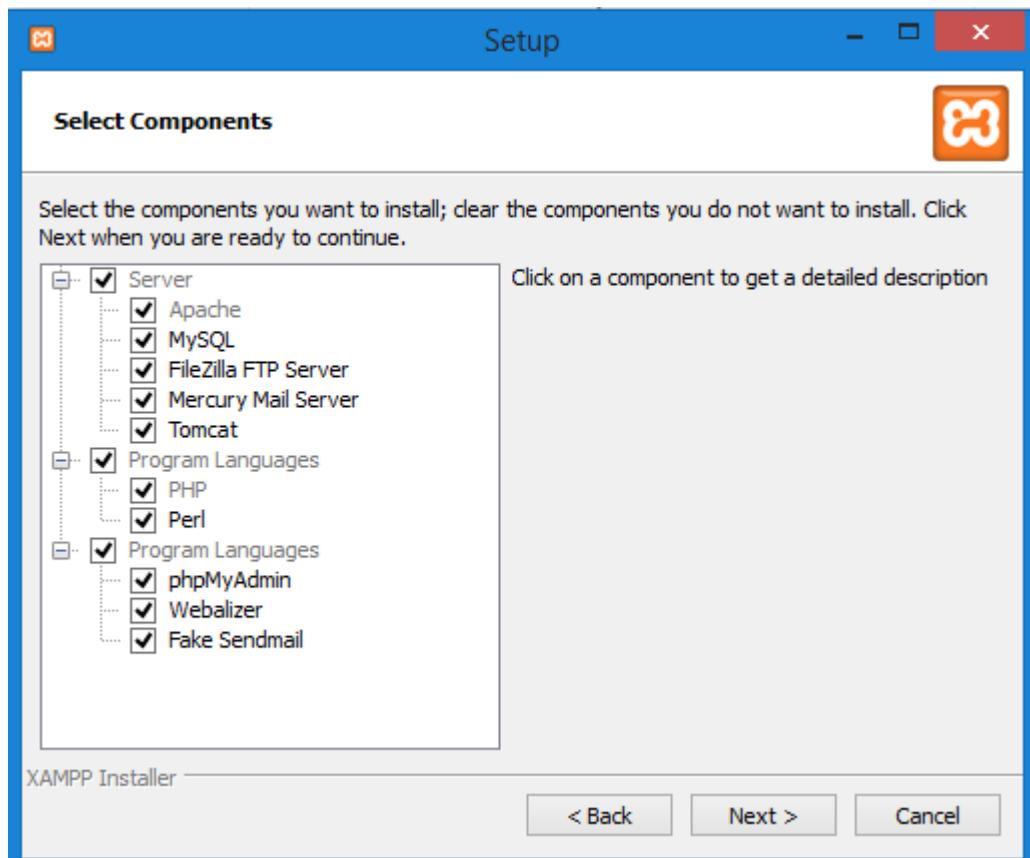
5. Run file xampp-win32-[X.X.X-VCXX]-installer.exe

Note: [X.X.X-VCXX] refers to the current version of the installer. It might differ with the version you see on the XAMPP website.

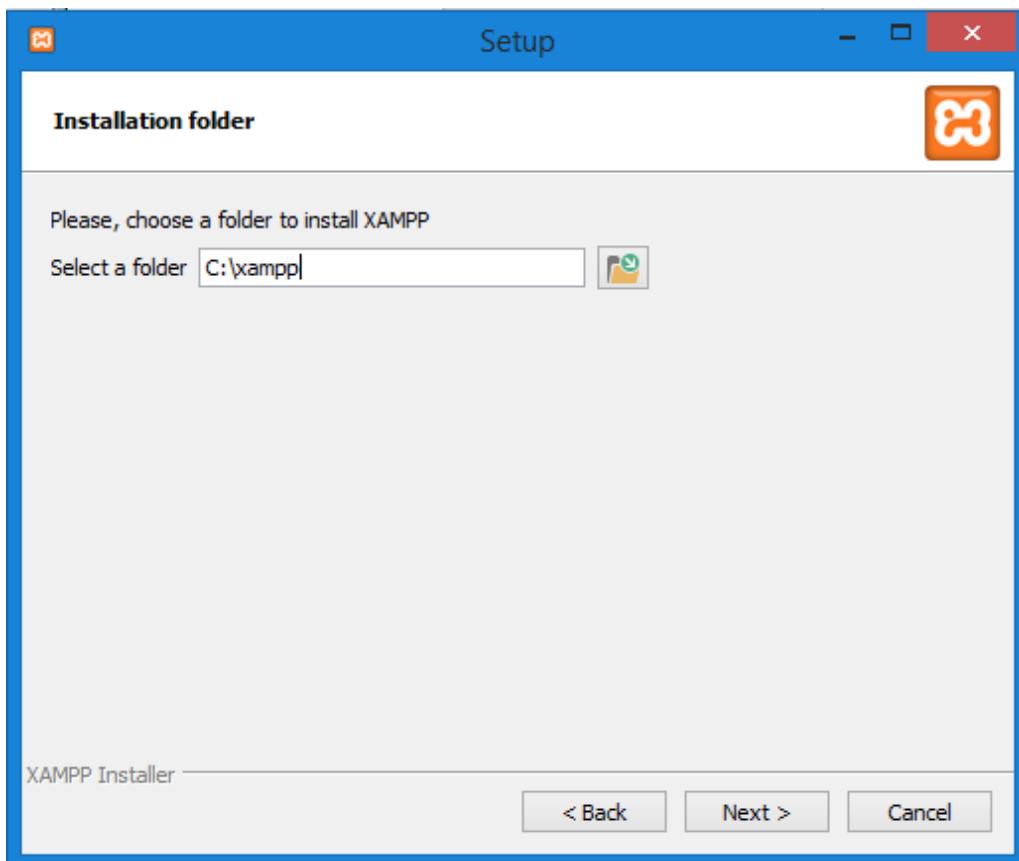
6. Click the Next button.



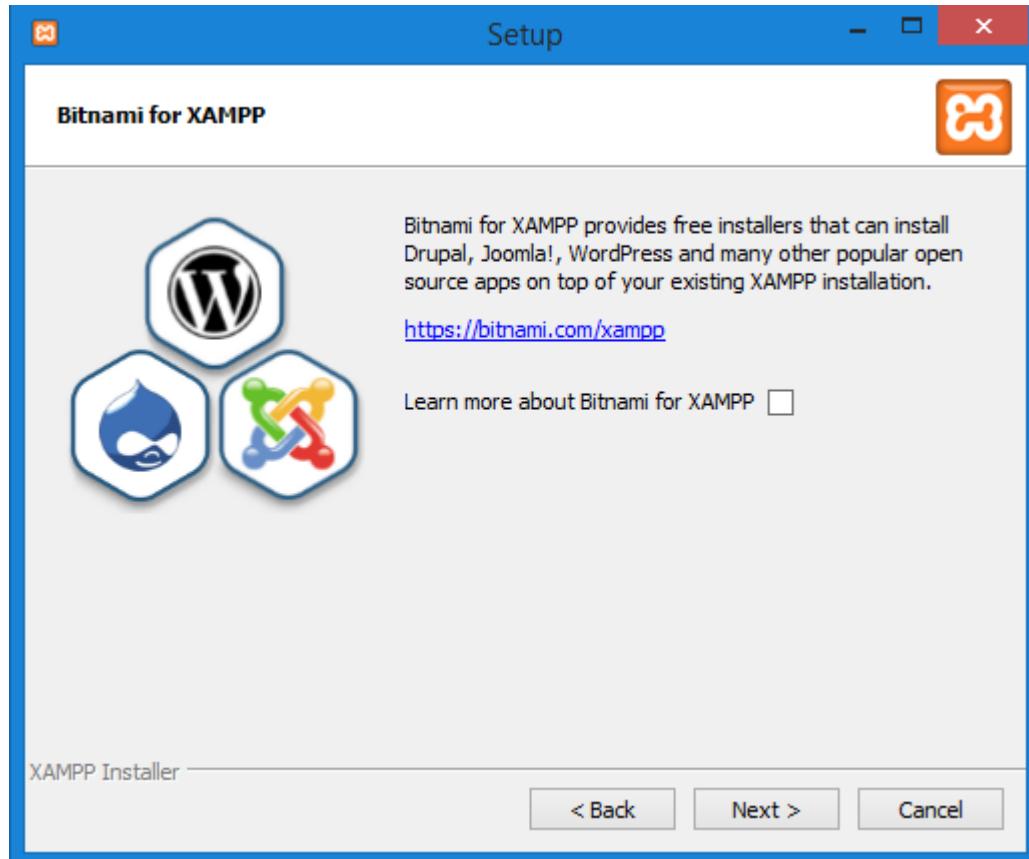
7. Please ensure you choose Tomcat and MySQL.



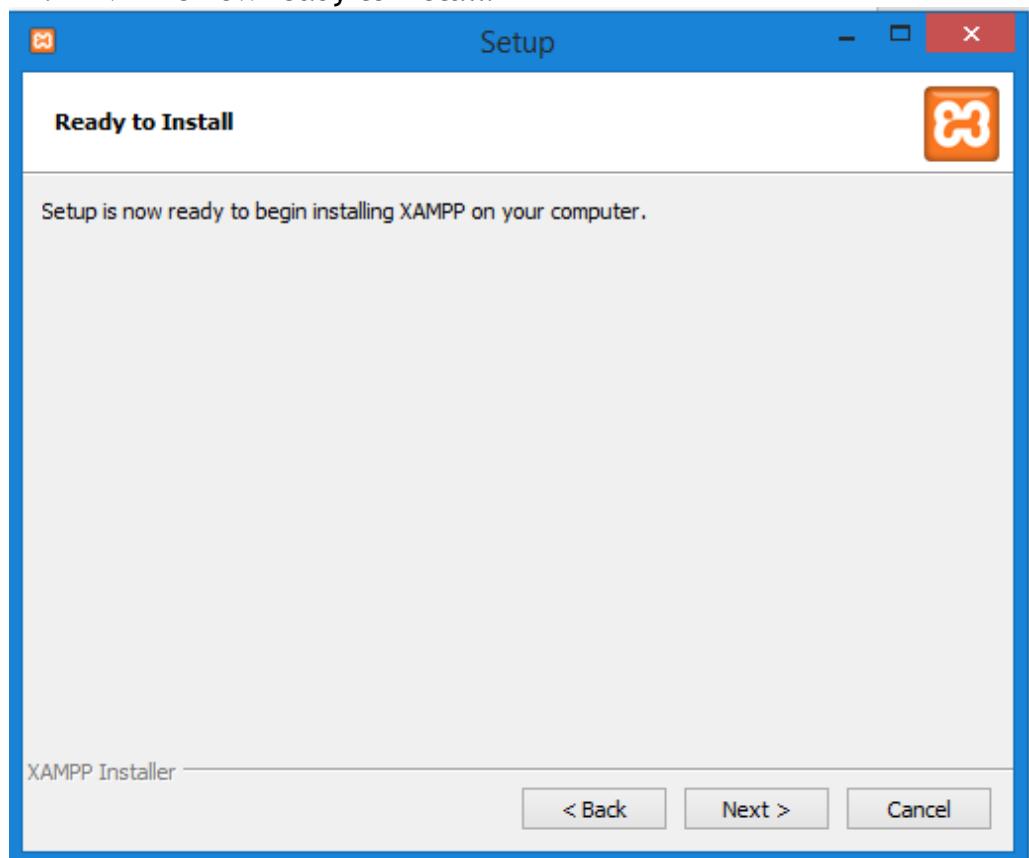
8. Choose a folder as C: \xampp and click the Next > button.



10. Untick “Learn more about Bitnami for XAMPP” and click Next button.



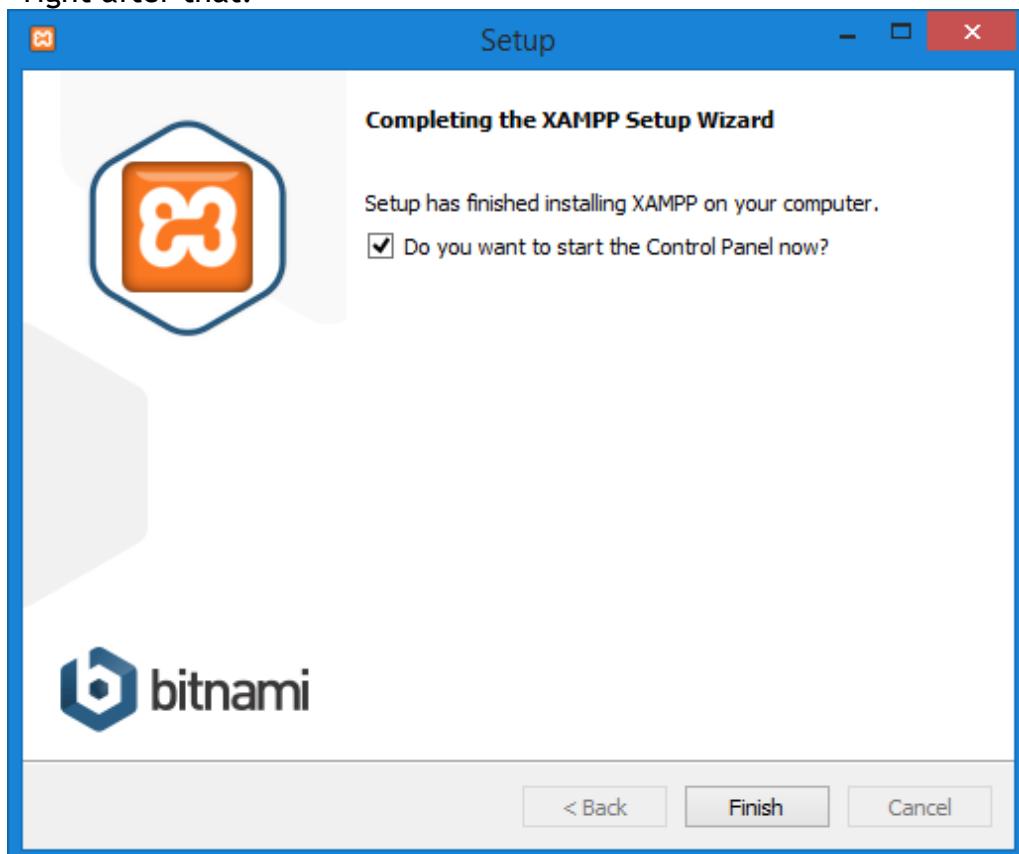
11. XAMPP is now ready to install.



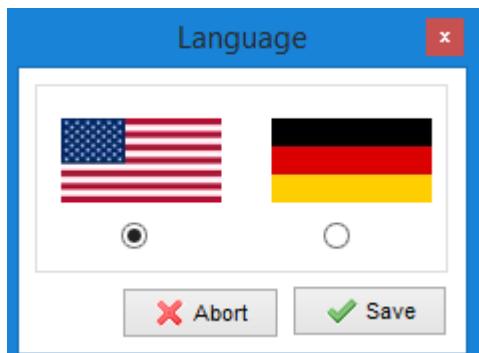
12. Wait for the installation process to complete.



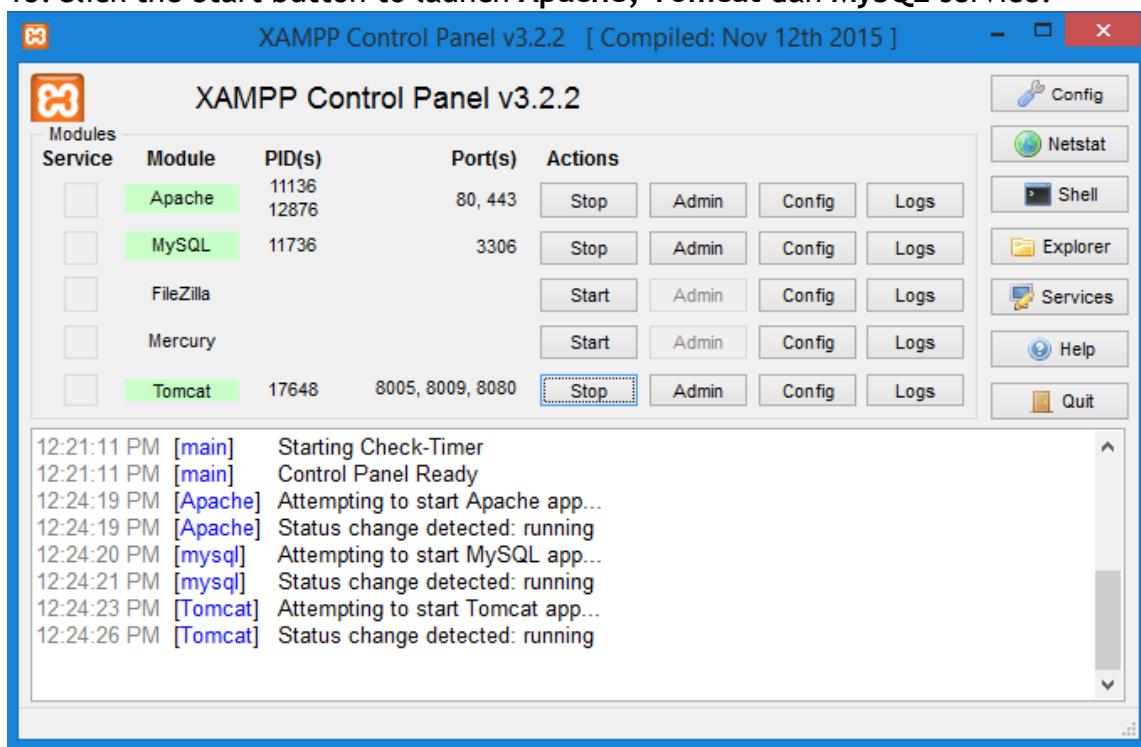
13. Click Finish button once completed and the Control Panel should start right after that.



14. Only for first time launching, select English as the default language for XAMPP. Click the Save button.



15. Click the Start button to launch Apache, Tomcat dan MySQL service.



Note: Why do we need Apache? We will use phpmyadmin as a tool to manage our database in the upcoming tasks. You may use other tools such as MySQL Workbench, Netbeans Database Manager etc. Do not confuse with the functions of various instruments used during the JSP development. Try to understand the context of the usage of each of them.

Reflection

What have you learnt from this exercise?

- Apache refers to the Apache HTTP Server, which is one of the most widely used web servers in the world.

Task 2: Change the Default Root Password of MySQL Database

Objective:

Change the root password of MySQL Database

Problem

To reset a new root password for MySQL Database

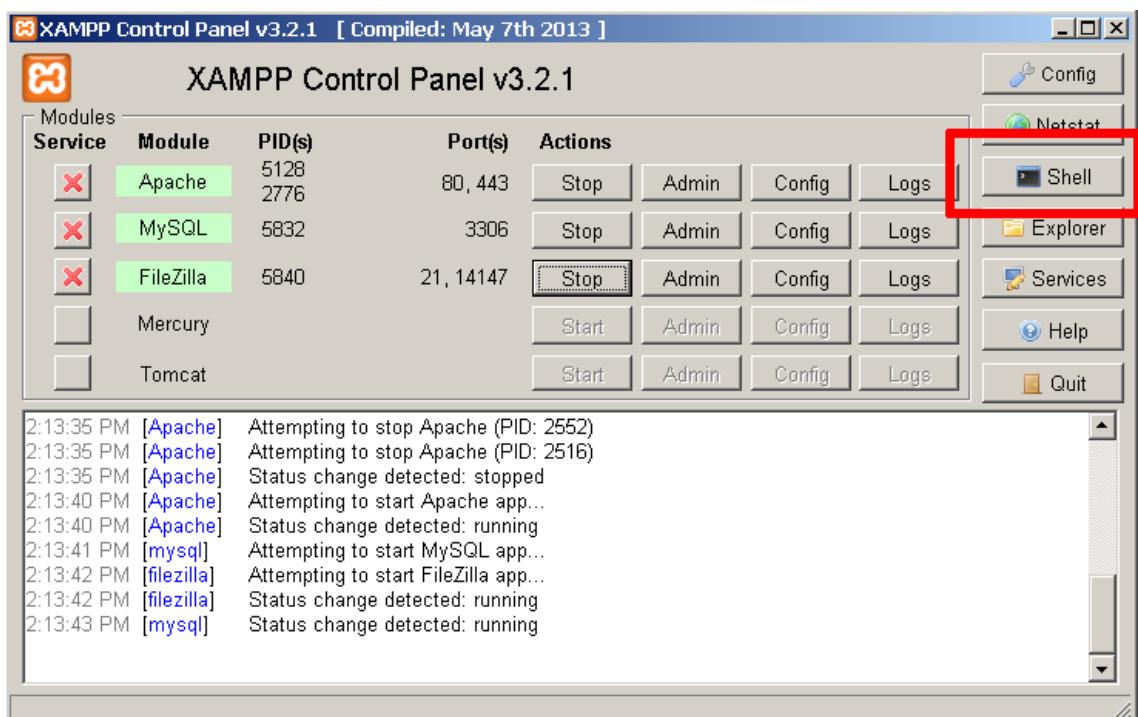
Description:

5 minutes

By default, the MySQL installation that ships with XAMPP have an empty root password. It is a severe security risk, especially if you plan to use XAMPP in production scenarios.

To change the MySQL root password, follow these steps:

1. Ensure that the MySQL server is running.
2. Open your Windows command prompt by clicking the "Shell" button in the XAMPP control panel.



3. Use the mysqladmin command-line utility to alter the MySQL password, using the following syntax (Note: We will use “admin” as our password during the lesson):

```
mysqladmin --user=root password "admin"
```

4. To test that your password change has been accepted, by attempting to connect to the MySQL server using the MySQL command-line client in the same directory. For example, you could use the command below to connect to the server and return the results of a calculation:

```
mysql --user=root --password=admin -e "SELECT 1+1"
```



The screenshot shows a terminal window titled "XAMPP for Windows". The command entered was "mysql --user=root --password=admin -e \"SELECT 1+1\"". The output shows the result of the query: "1+1" followed by a vertical bar, then "2", followed by another vertical bar, and finally a "#".

```
c:\xampp\mysql\bin>mysql --user=root --password=admin -e "SELECT 1+1"
+-----+
| 1+1 |
+-----+
| 2 |
+-----+
c:\xampp\mysql\bin>
```

5. To update the password in the future, you can use the following syntax:

```
mysqladmin --user=root --password=oldpassword
password "newpassword"
```

6. To test that your password change has been accepted, by attempting to connect to the MySQL server using the MySQL command-line client in the same directory. For example, you could use the command below to connect to the server and return the results of a calculation:

```
mysql --user=root --password=gue55me -e "SELECT 1+1"
```

IMPORTANT:

To avoid forgetting the password during our lesson, please set the MySQL password as **admin** ONLY..!

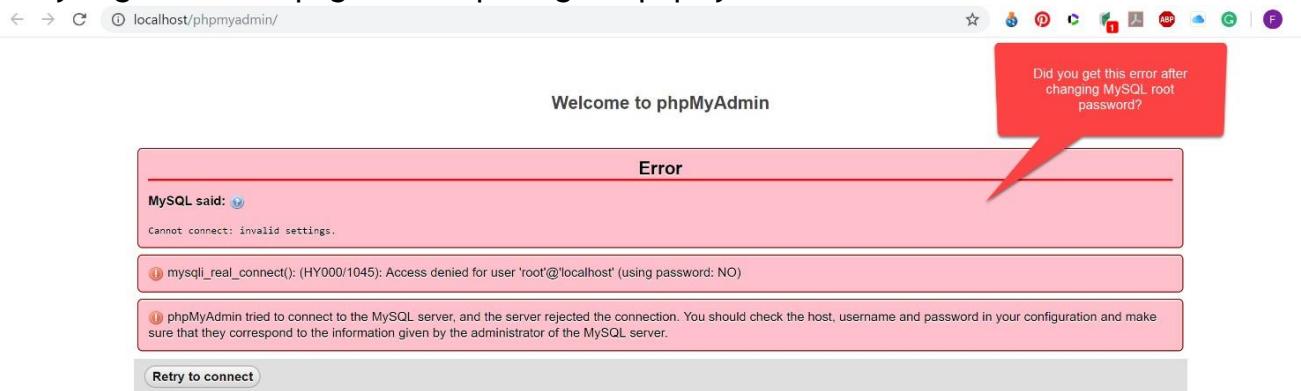
Remark: The above instructions are taken from Apache Friends Documentation. You may retrieve the documentation at <http://localhost/dashboard/docs/reset-mysql-password.html> after done with the XAMPP installation.

Reflection

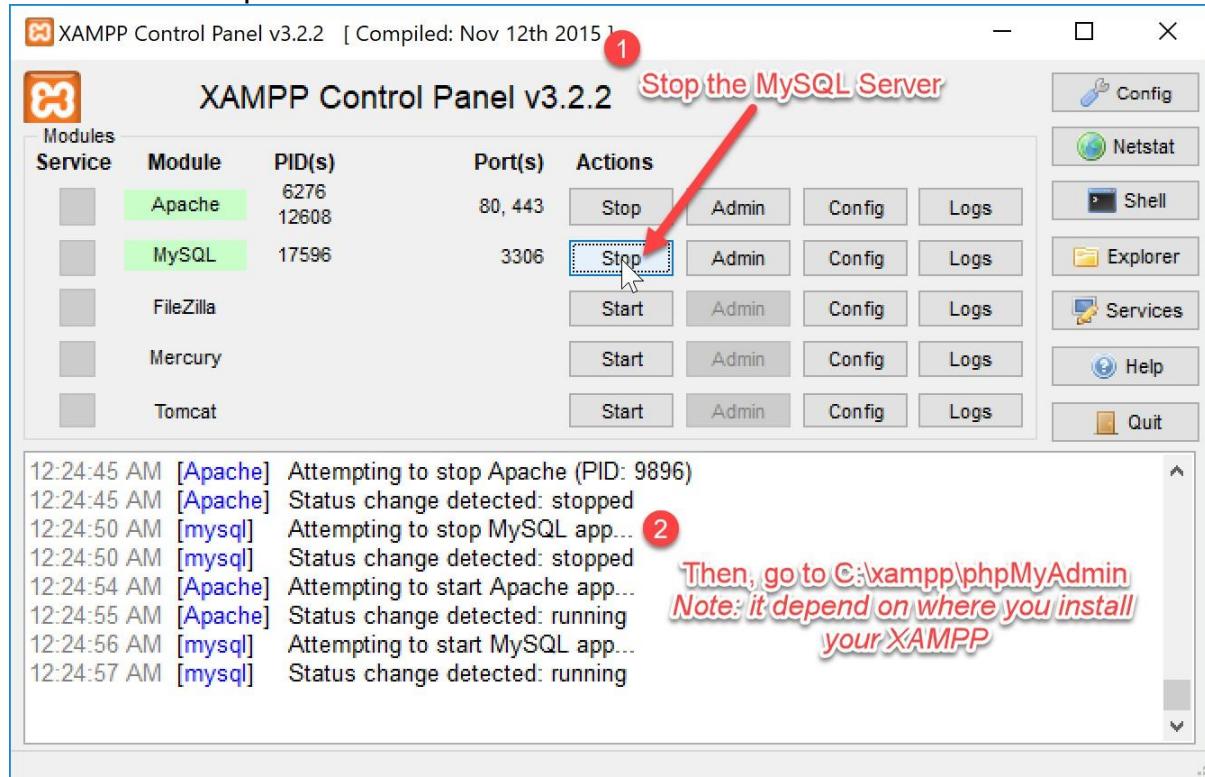
What have you learned from this exercise?

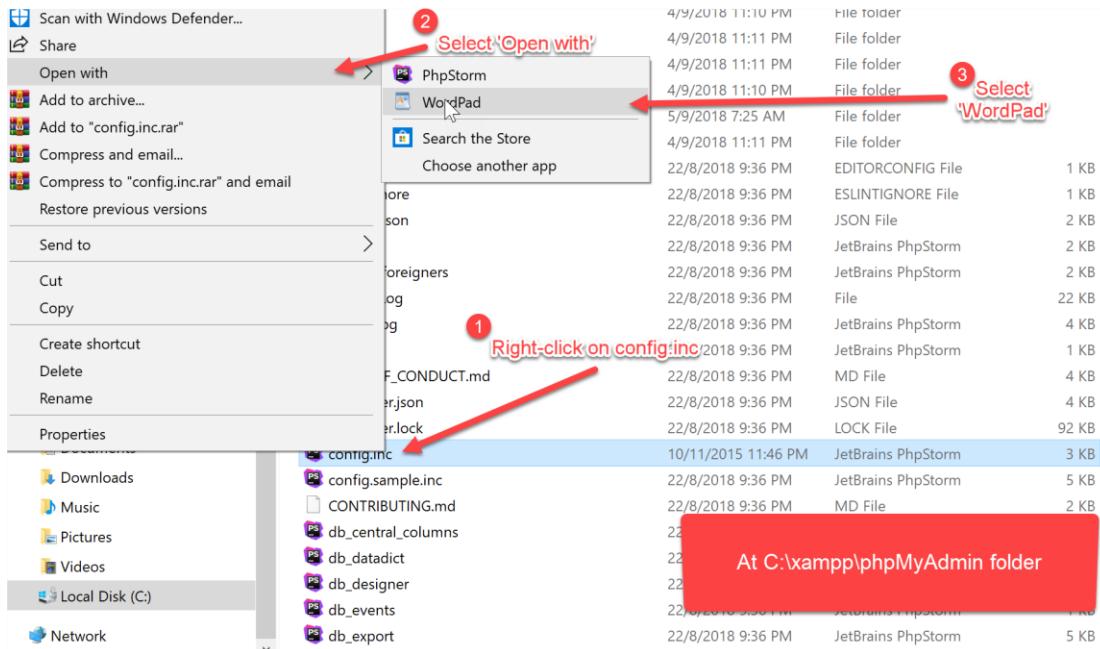
Troubleshooting Notes: Fixing phpMyAdmin Error

Did you get an error page when opening the phpMyAdmin?



Follow these steps to fix it.





```

password in
 * cookie
 */
$cfg['blowfish_secret'] = 'xampp'; /* YOU SHOULD CHANGE THIS
FOR A MORE SECURE COOKIE AUTH! */

/*
 * Servers configuration
 */
$i = 0;

/*
 * First server
 */
$i++;

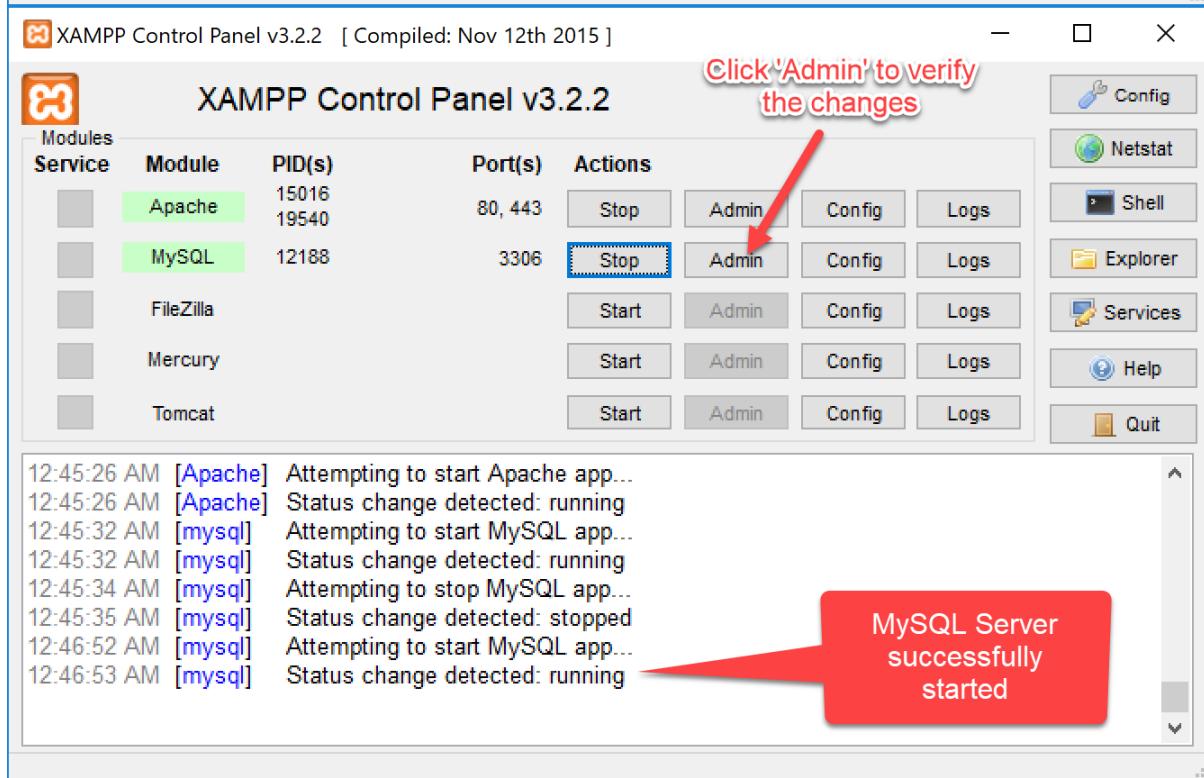
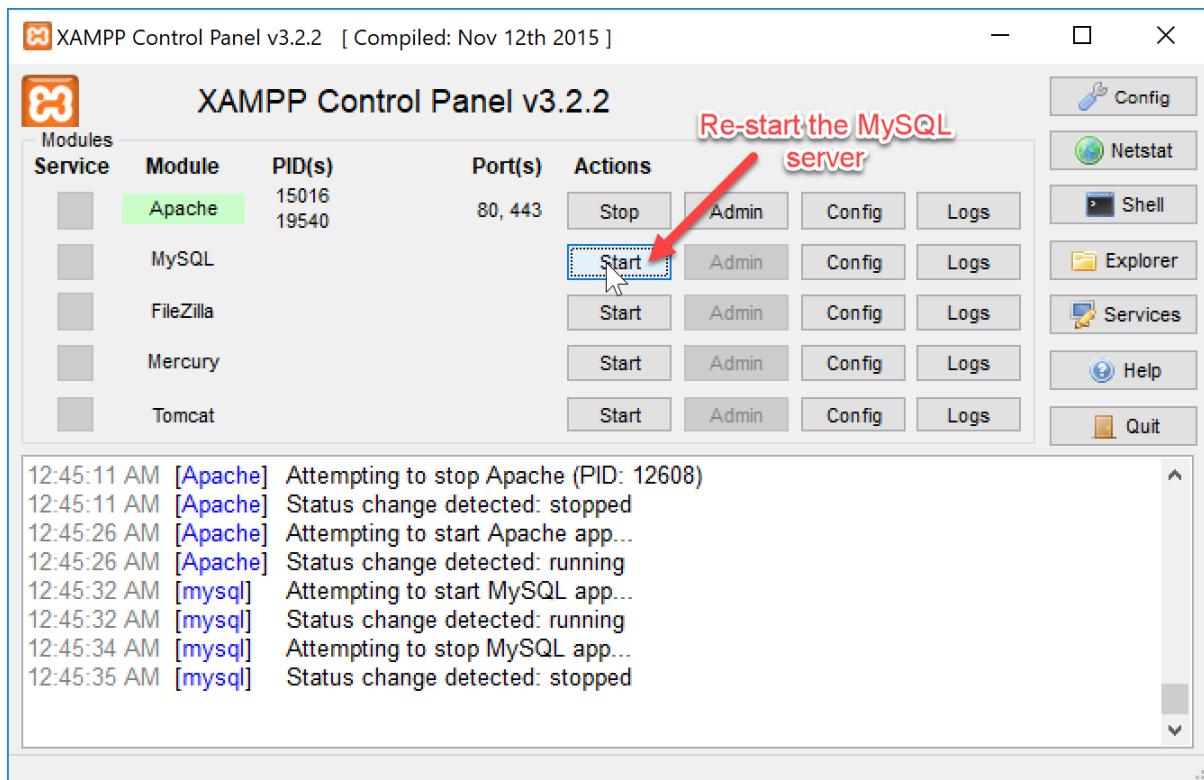
/* Authentication type and info */
$cfg['Servers'][$i]['auth_type'] = 'config';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = 'admin'; ① Put your root password here. In this example, 'admin' is the root password.
$cfg['Servers'][$i]['extension'] = 'mysqli';
$cfg['Servers'][$i]['AllowNoPassword'] = false; ② Change the value from 'true' to 'false'
$cfg['Lang'] = '';

/* Bind to the localhost ipv4 address and tcp */
$cfg['Servers'][$i]['host'] = '127.0.0.1';
$cfg['Servers'][$i]['connect_type'] = 'tcp';

/* User for advanced features */
$cfg['Servers'][$i]['controluser'] = 'pma';
$cfg['Servers'][$i]['controlpass'] = '';

/* Advanced phpMyAdmin features */
$cfg['Servers'][$i]['pmadb'] = 'phpmyadmin';
$cfg['Servers'][$i]['bookmarktable'] = 'pma_bookmark';
$cfg['Servers'][$i]['relation'] = 'pma_relation';
$cfg['Servers'][$i]['table_info'] = 'pma_table_info';
$cfg['Servers'][$i]['table_coords'] = 'pma_table_coords';
$cfg['Servers'][$i]['pdf_pages'] = 'pma_pdf_pages';
$cfg['Servers'][$i]['column_info'] = 'pma_column_info';

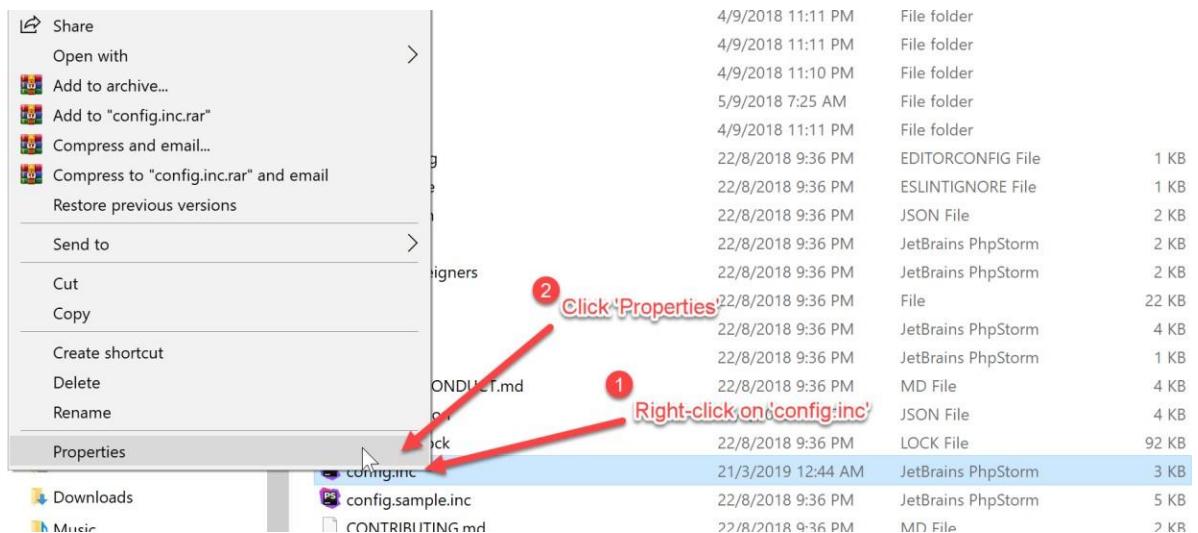
```

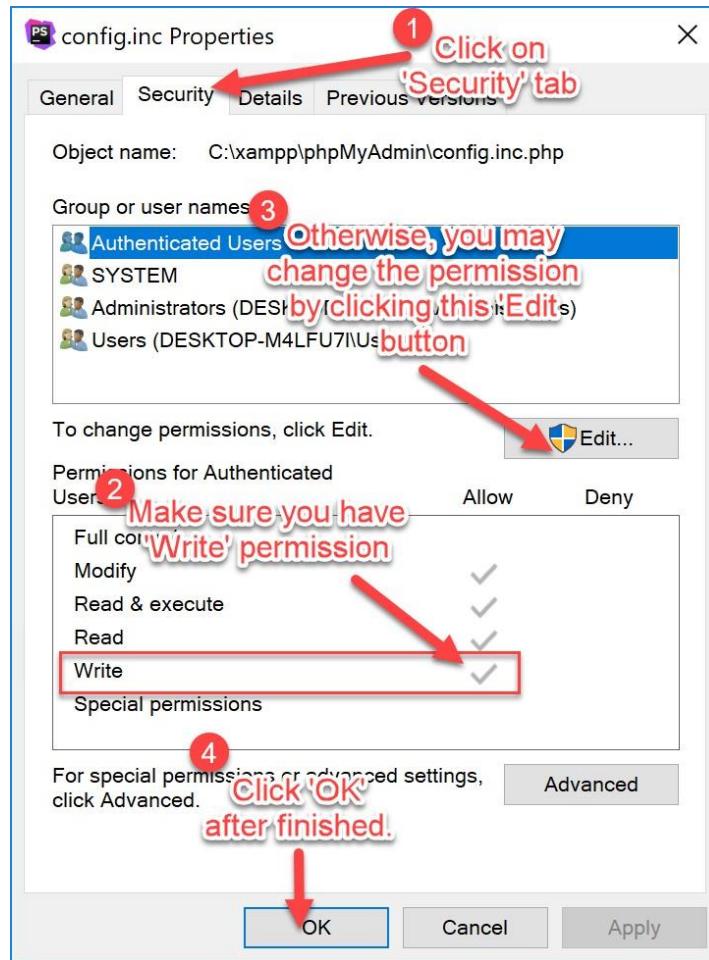


The screenshot shows the phpMyAdmin interface on a local server at 127.0.0.1. The left sidebar lists databases: information_schema, mysql, performance_schema, and phpmyadmin. The main area has two tabs: 'General settings' and 'Appearance settings'. In 'General settings', the 'Server connection collation' is set to 'utf8mb4_unicode_ci'. In 'Appearance settings', the 'Language' is set to 'English', the 'Theme' is 'pmahomme', and the 'Font size' is '82%'. A red speech bubble points to the bottom right of the page with the text: 'If you done it correctly, then you should see this page. Now, you can start to create a database and the tables. Good luck!'

If you done it correctly, then
you should see this page.
Now, you can start to create
a database and the tables.
Good luck!

If you can't edit or save the `config.inc`, follow the steps below:





Task 3: Managing Apache Tomcat

Objective:

Testing the access and add a new user to Apache Tomcat in XAMPP

Problem**Description:**

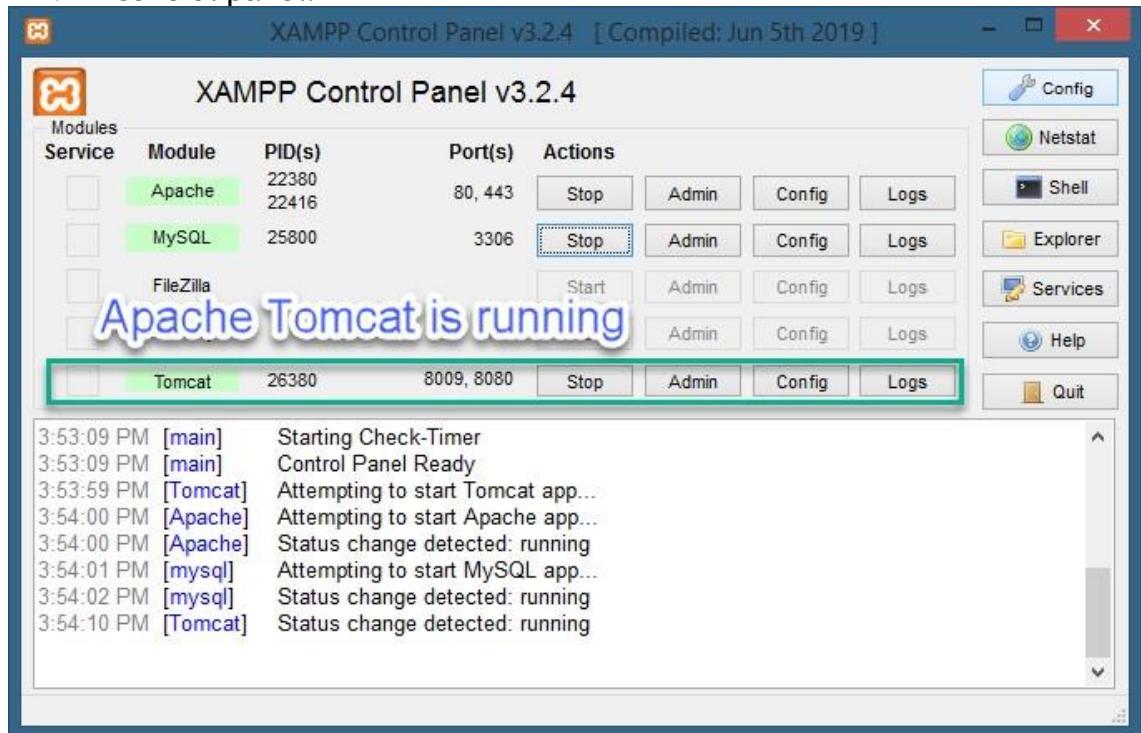
To configure user access for Apache Tomcat in XAMPP and test the access at localhost.

Estimated time:

30 minutes

In the previous task, we have successfully installed Apache Tomcat which is part of XAMPP software package.

1. Make sure Apache Tomcat is running. You can see that status from XAMPP control panel.



2. Open the browser, and go to <http://localhost:8080/>. If Apache Tomcat is running correctly, you will see a homepage as follows:

- Click on the “Manager App” of Apache Tomcat/7.0.99 homepage and there will appear a page ask for username and password. Do not panic, just click “Cancel”.

- As our Apache Tomcat currently does not has any defined user, a page below will be shown on the browser.

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.

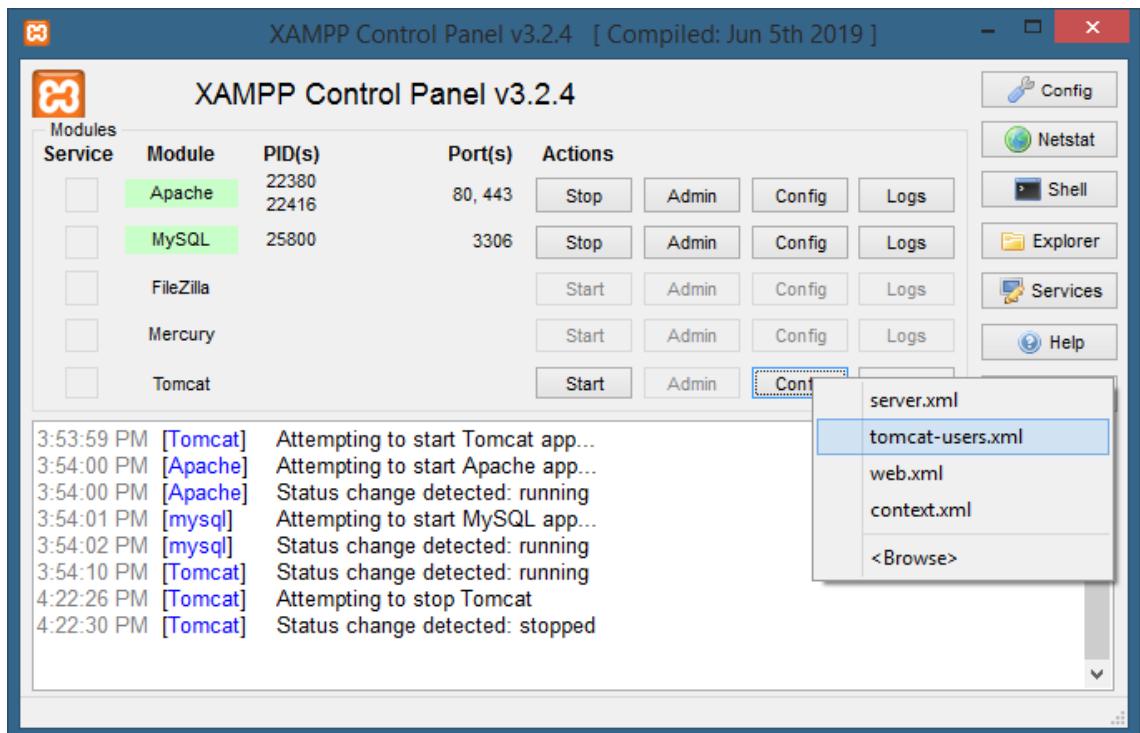
- `manager-gui` - allows access to the HTML GUI and the status pages
- `manager-script` - allows access to the text interface and the status pages
- `manager-jmx` - allows access to the JMX proxy and the status pages
- `manager-status` - allows access to the status pages only

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](#).

- To define a new user, click Stop Apache Tomcat server at the XAMPP Control Panel and click “Config” > `tomcat-users.xml`



6. The tomcat-users.xml will be opened using the default editor in your system. Scroll down the file until you see the </tomcat-users> tag. Add the tags below at the top of </tomcat-users> tag.

```
<role rolename="manager-script"/>
<role rolename="manager-gui"/>
<user username="admin" password="admin"
roles="manager-script,manager-gui"/>
```

7. Save the file. Finally, tomcat-users.xml will look like this

The screenshot shows a Notepad window titled 'tomcat-users - Notepad'. The XML content is as follows:

```

<!--
  NOTE: By default, no user is included in the "manager-gui" role required
  to operate the "/manager/html" web application. If you wish to use this app,
  you must define such a user - the username and password are arbitrary. It is
  strongly recommended that you do NOT use one of the users in the commented out
  section below since they are intended for use with the examples web
  application.
-->
<!--
  NOTE: 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>

```

8. We have added a new user “admin” with a password “admin”. Be sure to change the password when you deploy your application on a production server.

9. Now, start the Apache Tomcat and repeat Step 2 and Step 3. This time, put the username as “admin” and password as “admin”.
10. If you did the above steps correctly, you will see the screen as below:

The screenshot shows the Tomcat Web Application Manager interface. At the top, there's a header bar with the Apache logo and the title "Tomcat Web Application Manager". Below the header, there's a message box that says "Message: OK". The main area is divided into sections: "Manager", "List Applications", "HTML Manager Help", "Manager Help", and "Server Status". The "Applications" section is the most prominent, displaying a table of deployed applications. The table has columns for Path, Version, Display Name, Running, Sessions, and Commands. The applications listed are: "/ (Welcome to Tomcat), "/MyFirstServlet (with a warning about session expiration), "/docs (Tomcat Documentation), "/examples (Servlet and JSP Examples), "/host-manager (Tomcat Host Manager Application), and "/manager (Tomcat Manager Application). The "/manager row shows 1 session. The "Commands" column for each application includes buttons for Start, Stop, Reload, and Undeploy, along with a link to "Expire sessions with idle ≥ [30] minutes". Below the "Applications" section is a "Deploy" section with a placeholder "Deploy directory or WAR file located on server".

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

Task 4: Netbeans 12.3 IDE Installation

Objective:

Netbeans 12.3 IDE Installation

Problem Description:

To setup a proper environment for Java Web Application development.

Estimated time:

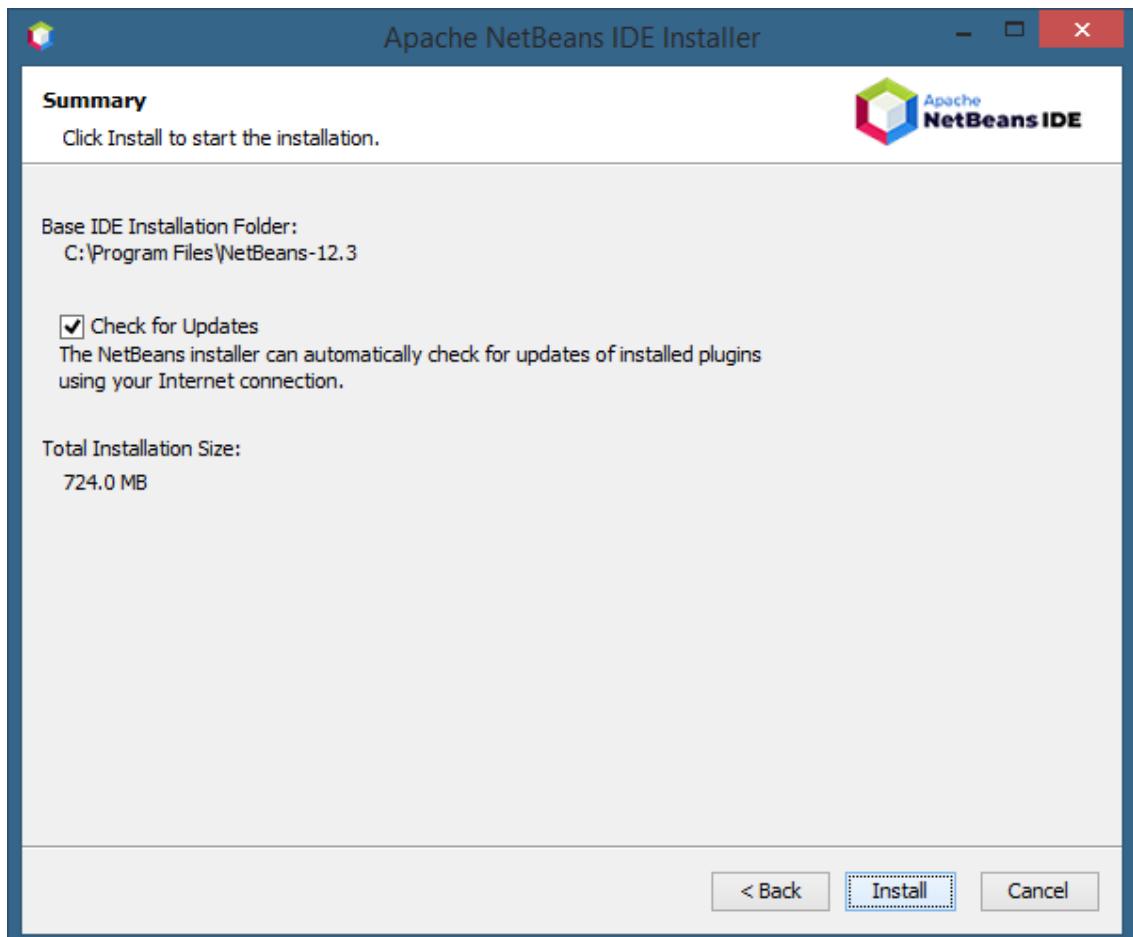
30 minutes

To start the Netbeans 12.3 installation process, follow the steps below:

11. Go to <https://netbeans.apache.org/download/index.html>
12. Click “Download”. Choose the installer that appropriate with your operating system and computer architecture (x32 or x64). A correct installation program could be something like this *Apache-NetBeans-12.3-bin-windows-x64.exe*

The screenshot shows the Apache Software Foundation's download page for Netbeans 12.3. The URL in the address bar is <https://apache.org/dyn/closer.cgi/netbeans/netbeans/12.3/Apache-NetBeans-12.3-bin-windows-x64.exe>. The page features the Apache logo and navigation links for News, About, Make a Donation, The Apache Way, Join Us, Downloads, and a search bar. Below the navigation, there's a section titled "COMMUNITY-LED DEVELOPMENT 'THE APACHE WAY'" with links for Projects, People, Community, License, and Sponsors. A "SUPPORT APACHE" button is also present. The main content area displays download links for various mirror sites, including an HTTP link to <https://downloads.apache.org/netbeans/netbeans/12.3/Apache-NetBeans-12.3-bin-windows-x64.exe>.

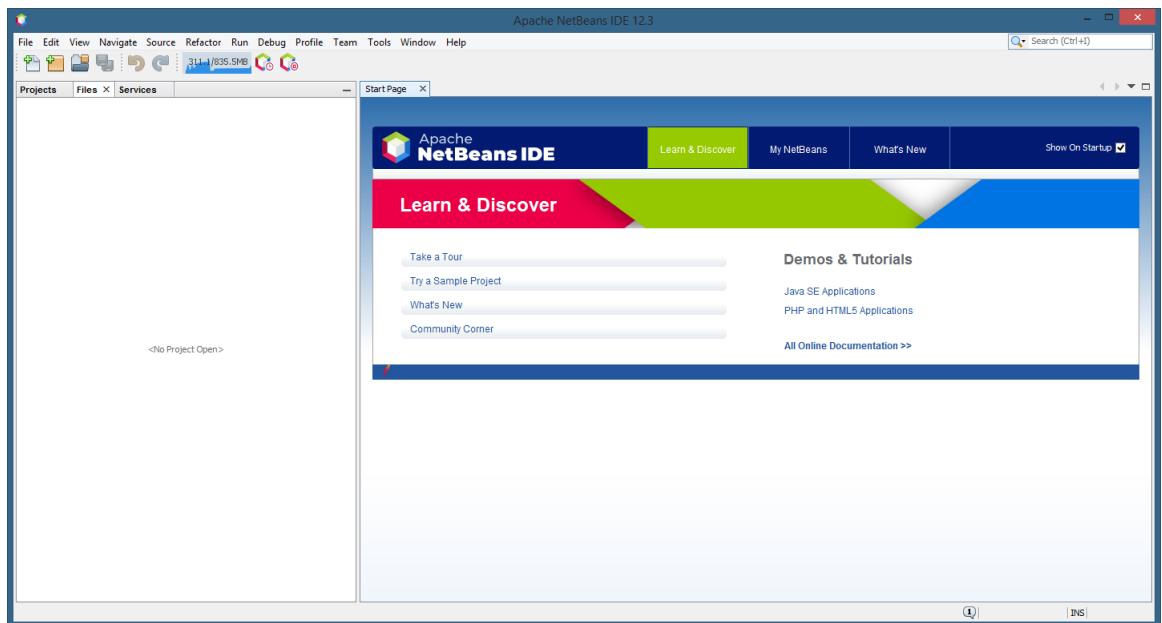
13. After the download finish, double-click on the installer and click “Next” until you reach the following screen:



14. Make sure you tick the “Check for Updates”, then click Install. Wait until the installation finish.
15. Run the Netbeans. If you have previous installation, the pop-up windows as below may appear and simply click “Yes”. This will import your previous configuration to a new one.



16. Finally, you will get the following screen.



17. Now, you are ready to start developing your first Java web application.
Well done!

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

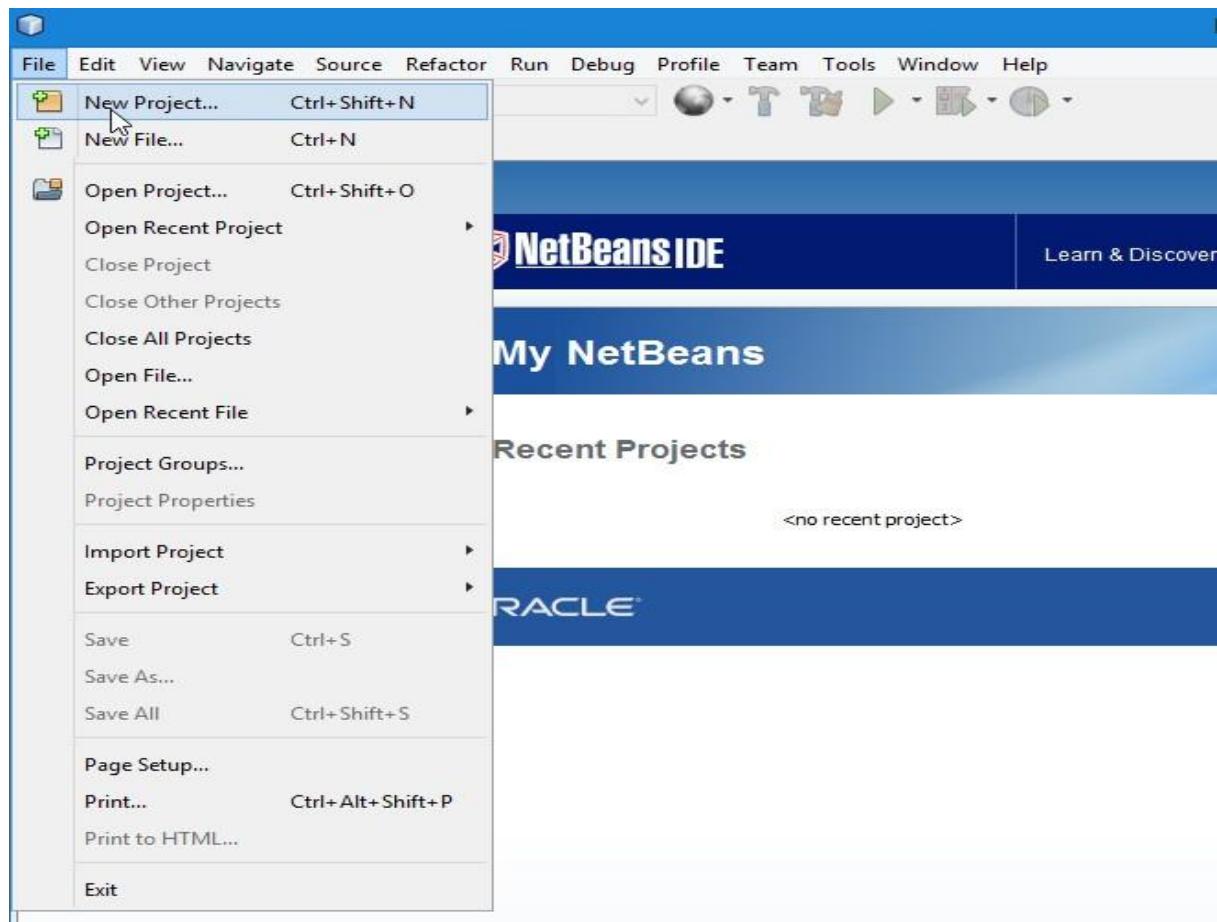
Problem

To write a simple Java Servlet

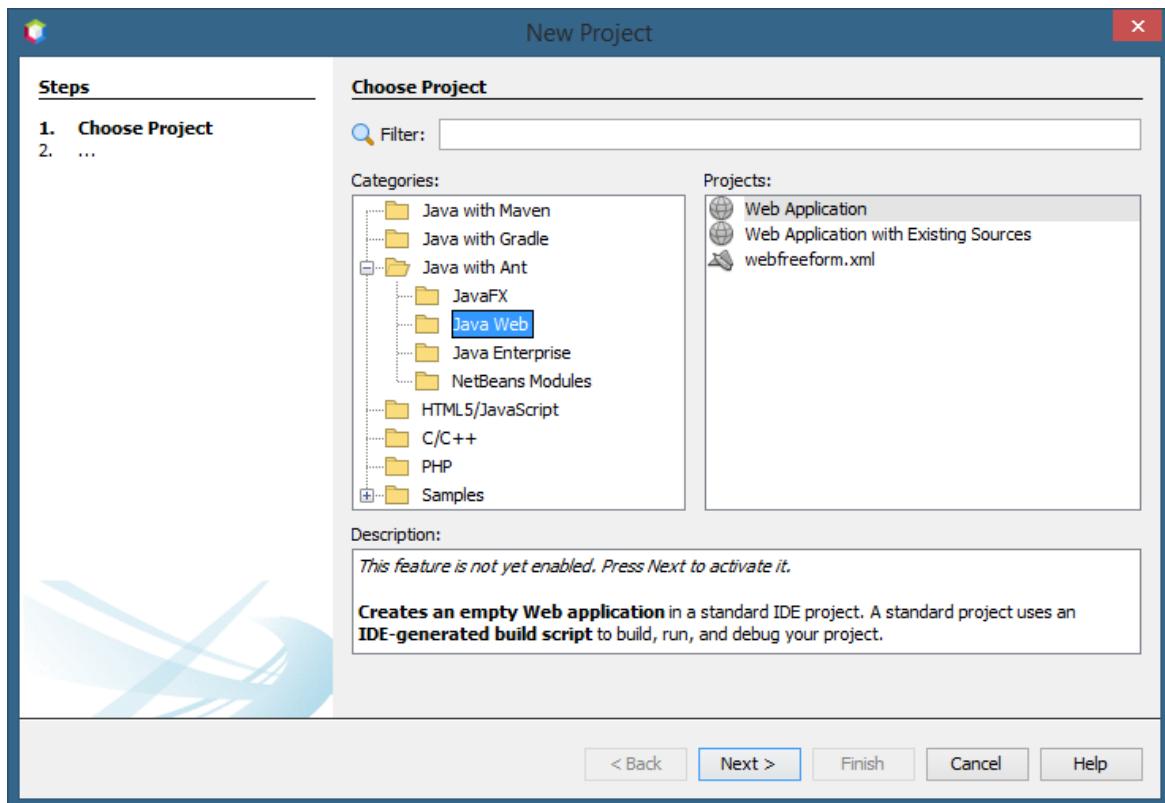
Description:**Estimated time:**

15 minutes

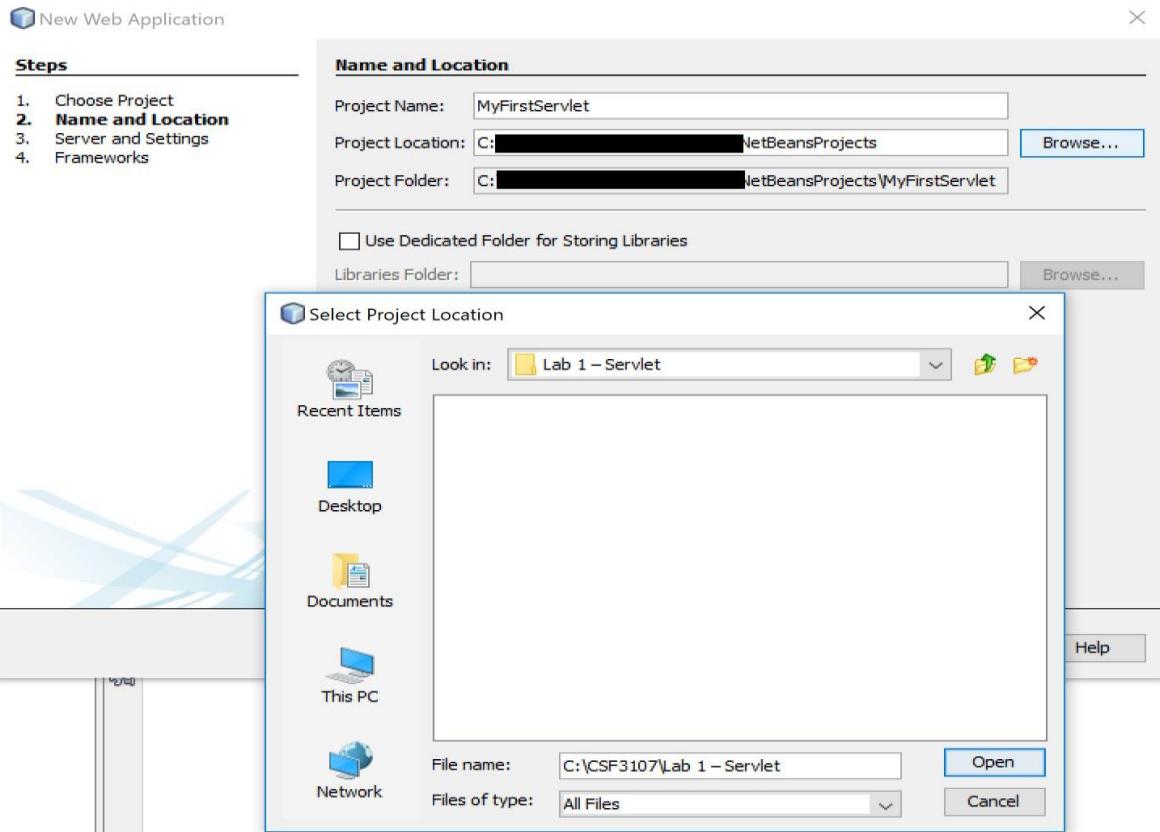
1. Create a directory *C:\[MATRICNUMBER]*.
2. Go to *C:\[MATRICNUMBER]* Lab's directory and create sub-directory as *Lab 1 - Servlet*.
3. Open your NetBeans.
4. Go to File -> New Project



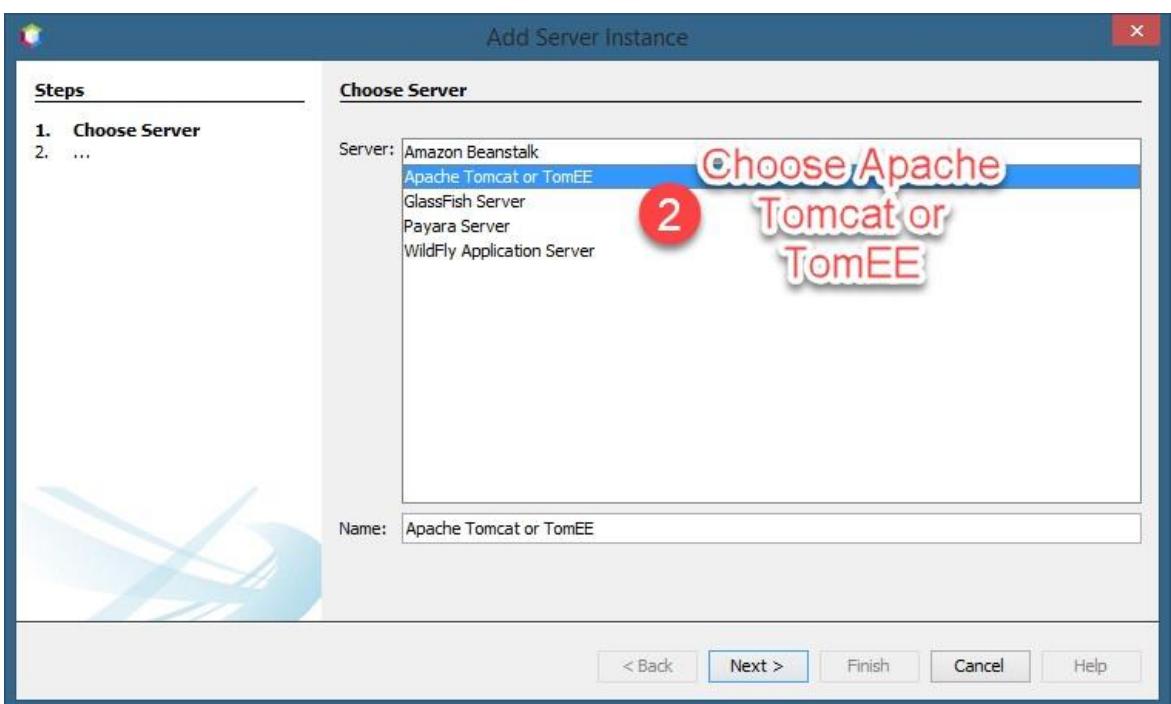
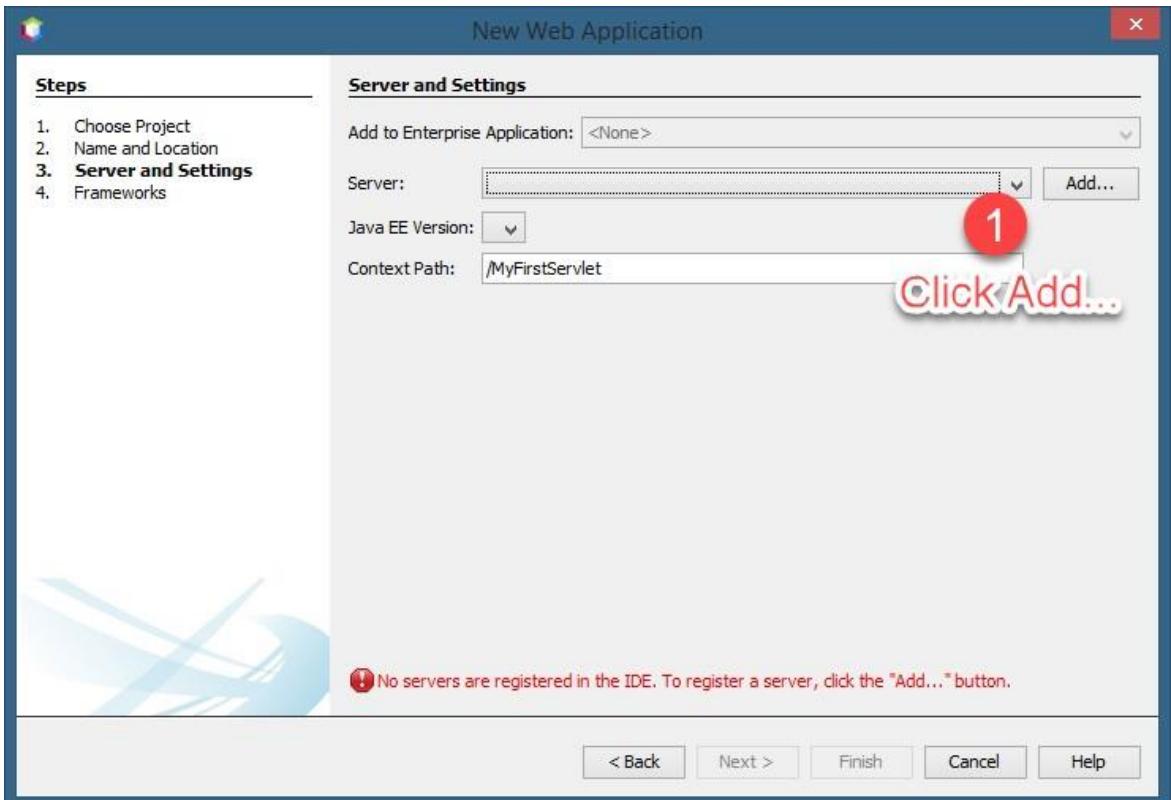
5. Select Java with Ant -> Java Web -> Web Application and click Next.

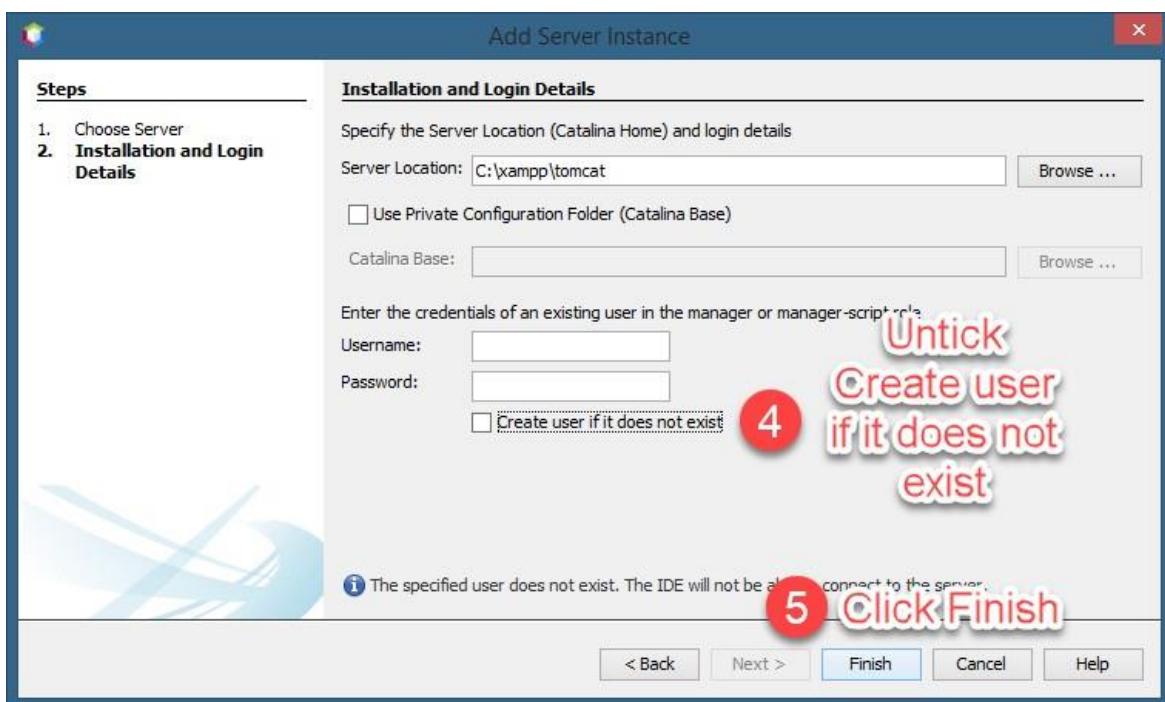
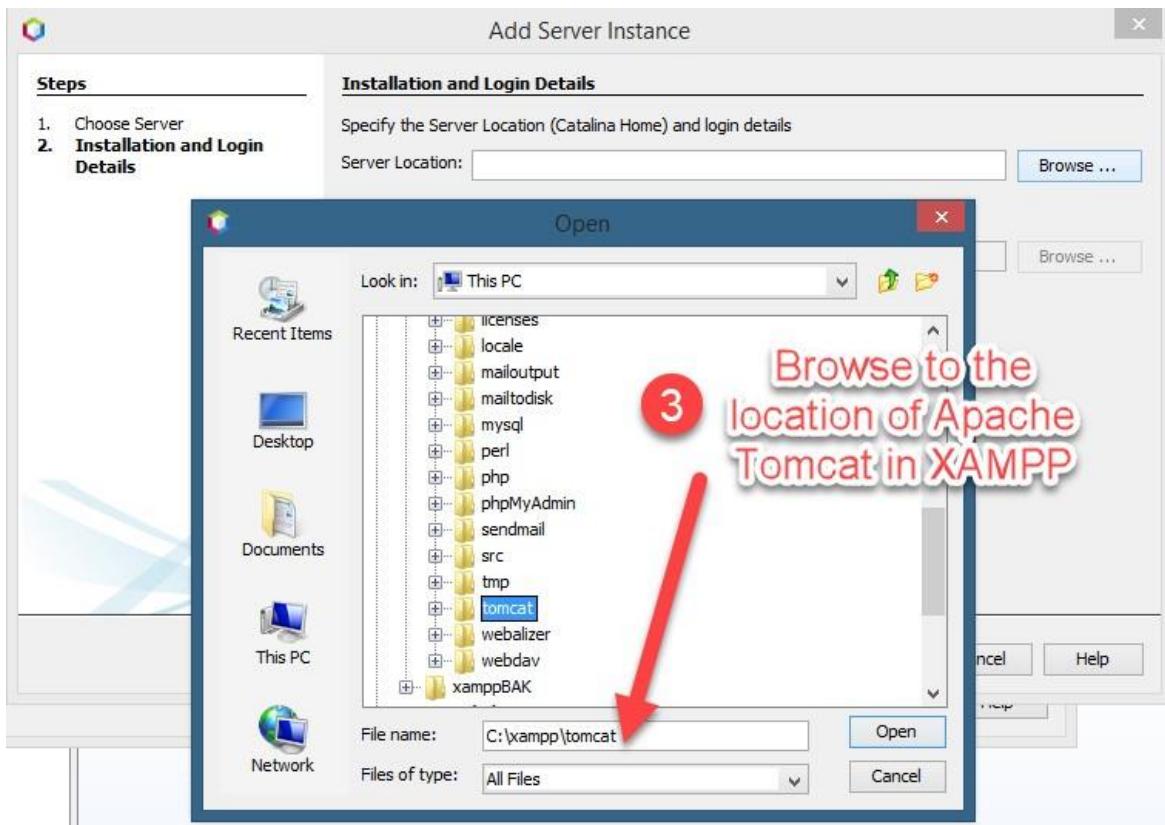


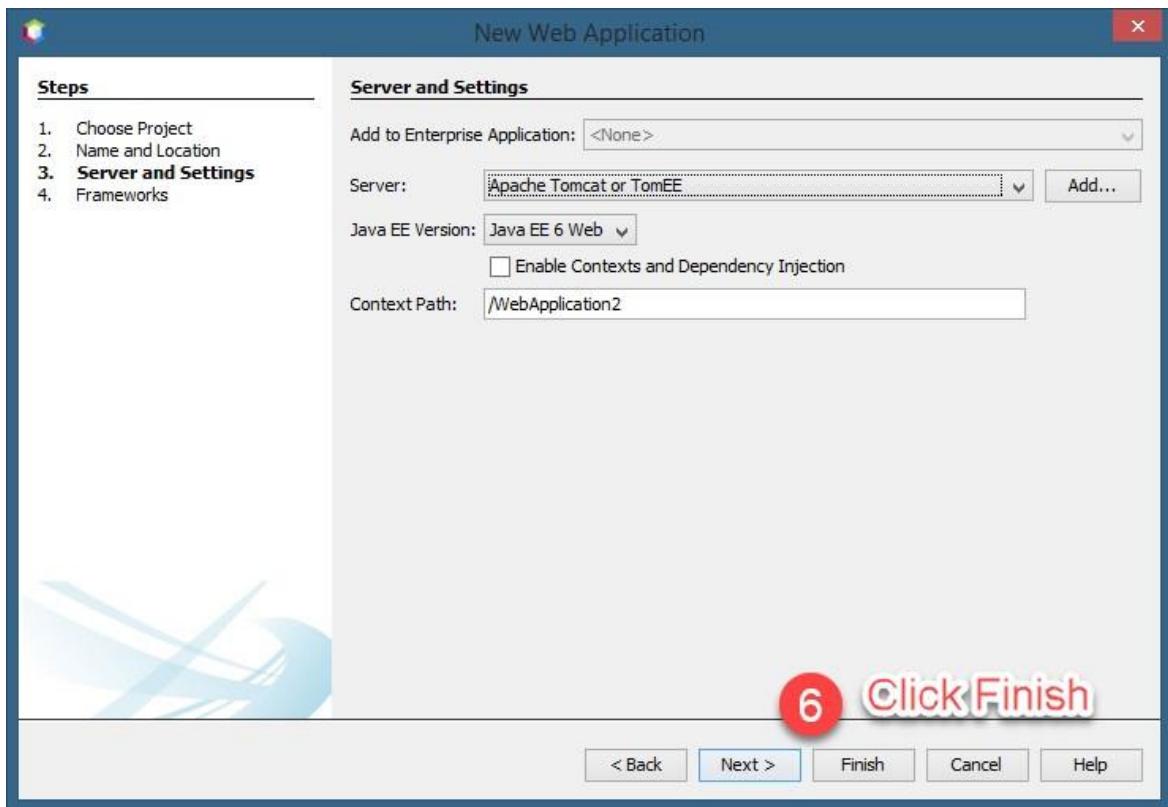
1. Type Project Name: *MyFirstServlet*.
2. Click Browse and choose Project Location: *C:\[MATRICNUMBER]\Lab 1 - Servlet*. Then click the Next button.



3. Before we can run our Java Web Application, we need to link it to Apache Tomcat Server. Follow the diagrams below to link Netbeans with Apache Tomcat in XAMPP.

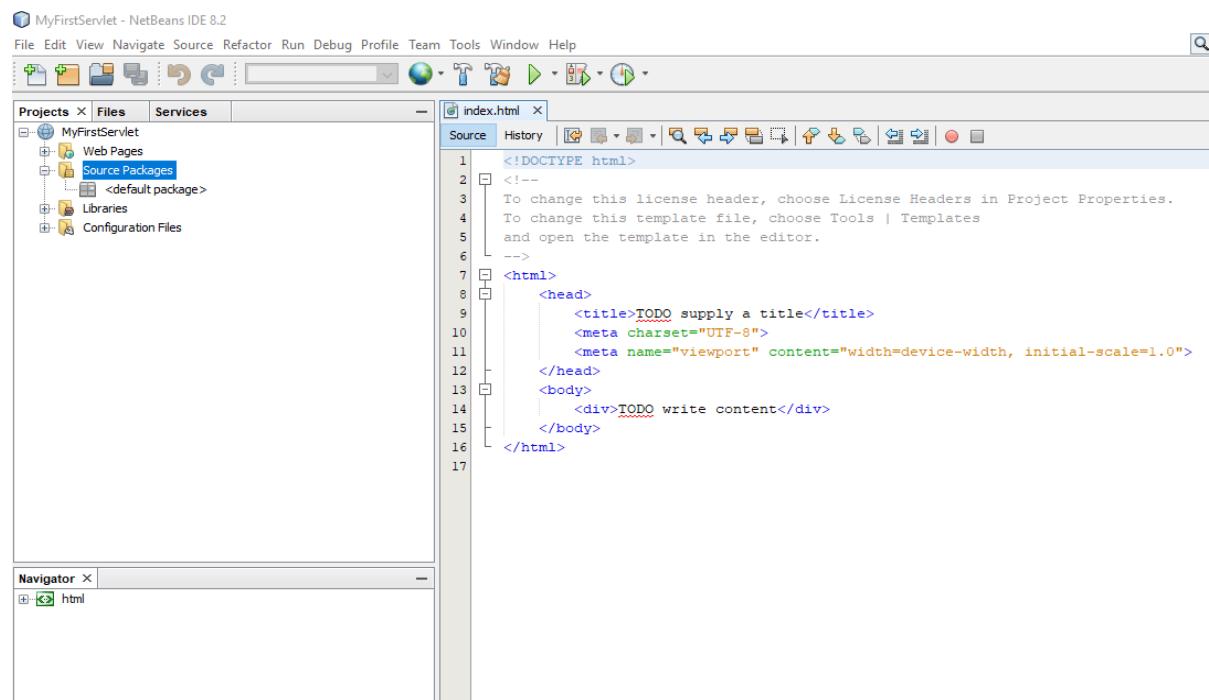




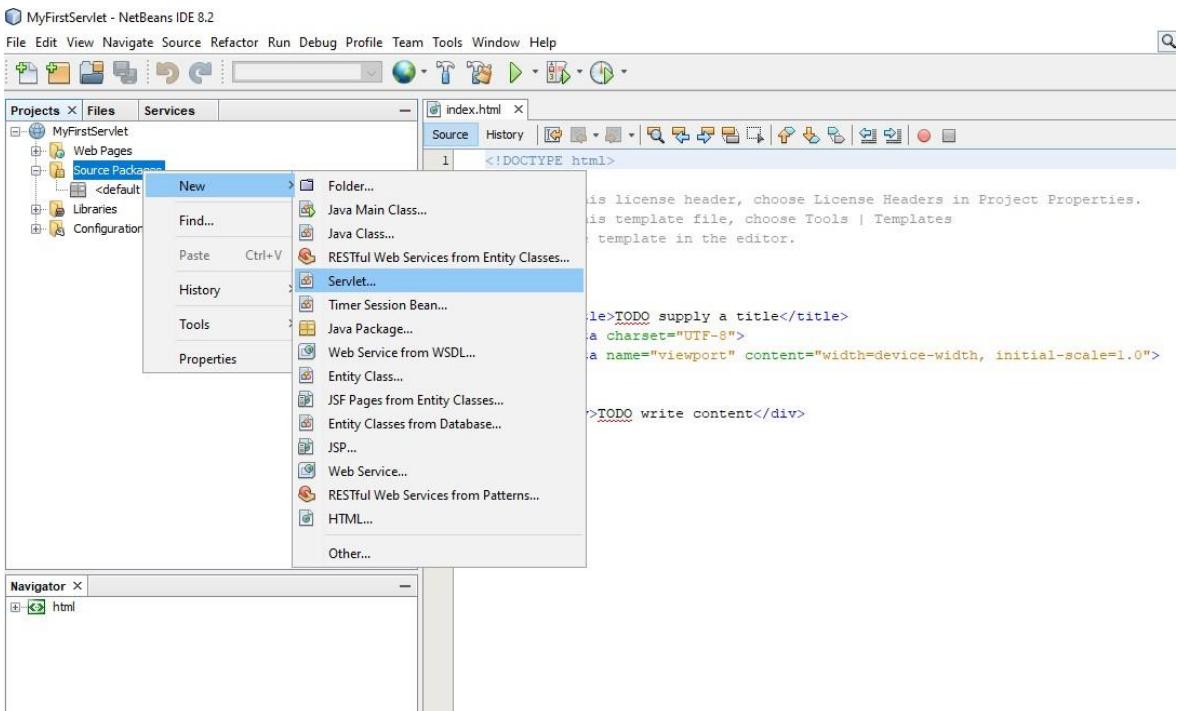


Note: the above steps only need to be ran for the first time you link the Netbeans to Apache Tomcat in XAMPP. After this, you just need to select it from the screen.

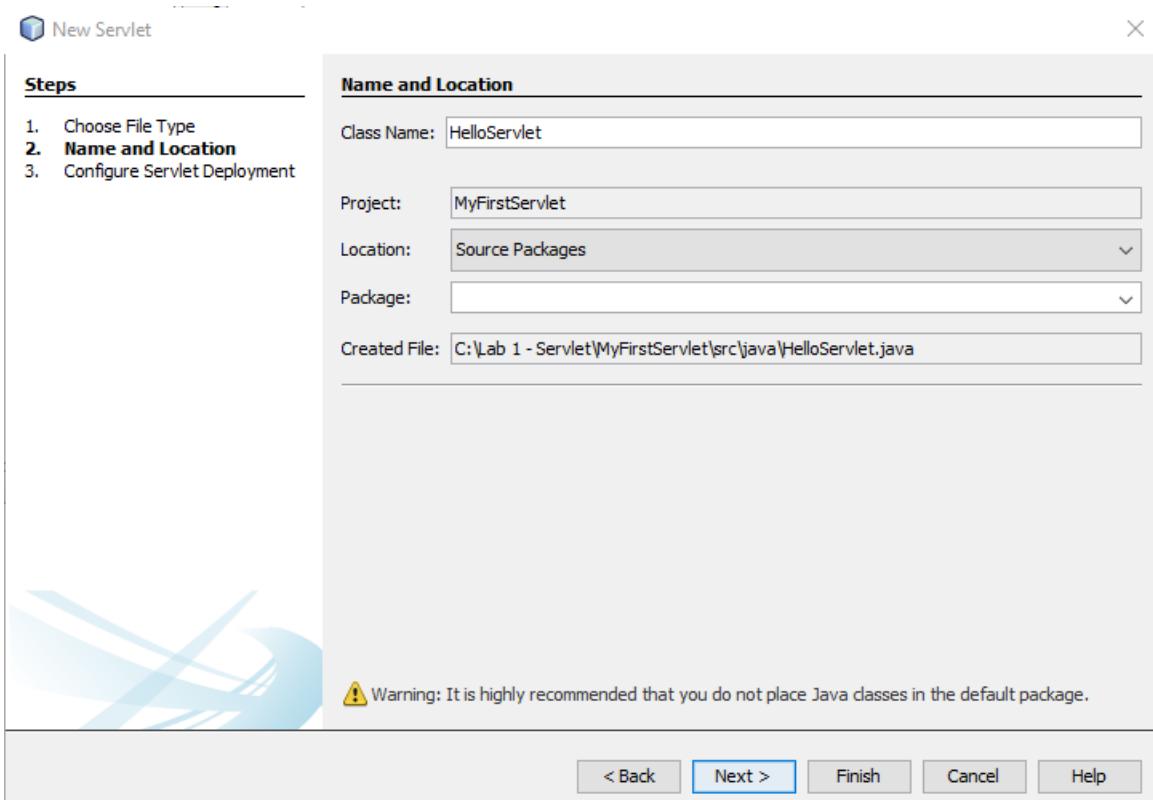
4. If everything okay, you will see the following screen:



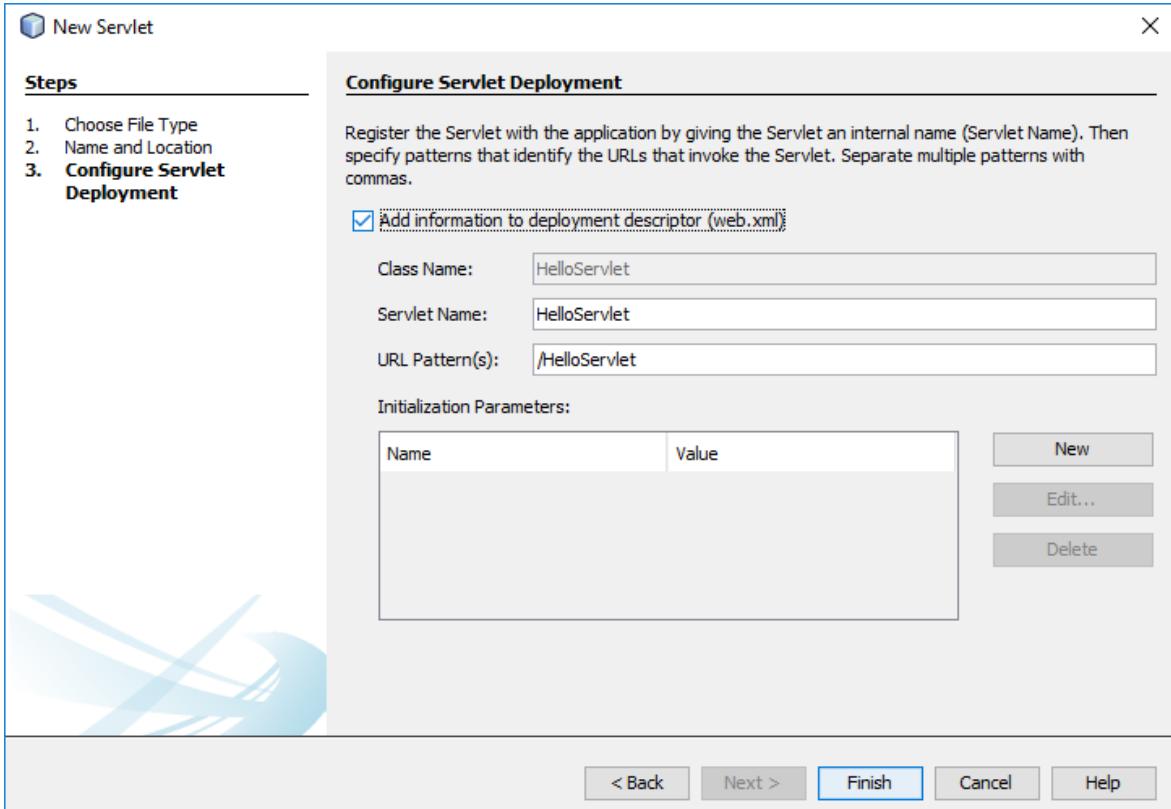
5. Create a new Servlet file.



6. Name your servlet as *HelloServlet*.



7. Tick the “Add information to deployment descriptor (web.xml)”



8. NetBeans has produced a new file named *HelloServlet.java*. You may see the location of it on the left side of NetBeans editor. On the right, are the servlet codes generated by NetBeans.

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author Fakhru Adli
 */
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()) {
            /* ... */
        }
    }

    // ...
}

```

9. Browse through the servlet file until you find the processRequest() method.

The screenshot shows a Java IDE interface with two tabs open: "index.html" and "HelloServlet.java". The "HelloServlet.java" tab is active, displaying the following code:

```
16 * @author Fakhru Adli
17 */
18 public class HelloServlet extends HttpServlet {
19
20     /**
21      * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
22      * methods.
23      *
24      * @param request servlet request
25      * @param response servlet response
26      * @throws ServletException if a servlet-specific error occurs
27      * @throws IOException if an I/O error occurs
28      */
29
30     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
31         throws ServletException, IOException {
32         response.setContentType("text/html;charset=UTF-8");
33         try (PrintWriter out = response.getWriter()) {
34             /* TODO output your page here. You may use following sample code. */
35             out.println("<!DOCTYPE html>");
36             out.println("<html>");
37             out.println("<head>");
38             out.println("<title>Servlet HelloServlet</title>");
39             out.println("</head>");
40             out.println("<body>");
41             out.println("<h1>Servlet HelloServlet at " + request.getContextPath() + "</h1>");
42             out.println("</body>");
43             out.println("</html>");
44         }
45     }
46 }
```

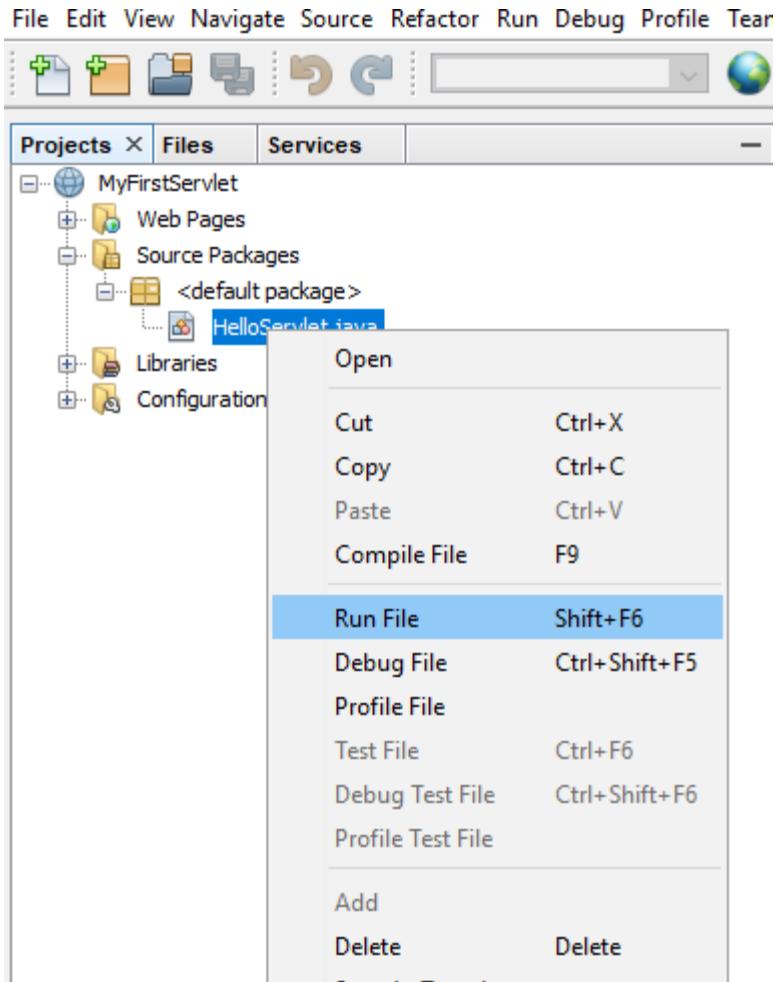
10. We are going to make a simple modification to the existing codes. Modify the them as follows. Refer to line 37, 40 and 41.

The screenshot shows a Java IDE interface with the following details:

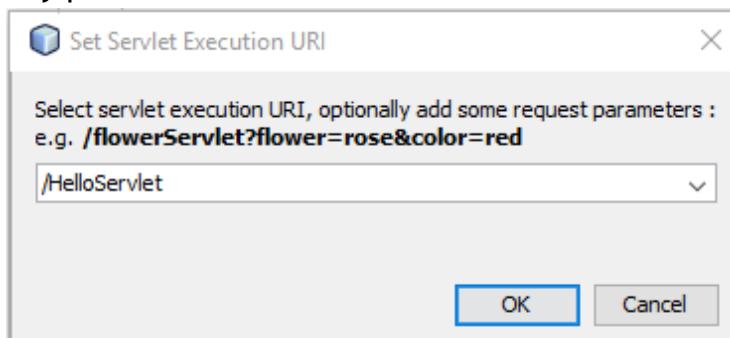
- Project Structure:** The project is named "HelloServlet" and contains two files: "index.html" and "HelloServlet.java".
- Code Editor:** The "HelloServlet.java" file is open, displaying Java code for a servlet. The code includes annotations for parameters and throws clauses, and uses PrintWriter to print HTML content.
- Toolbars:** Standard Java development toolbars for file operations, search, and navigation are visible at the top.
- Sidebar:** A sidebar on the right side of the interface typically used for navigating between files or viewing code metrics.

```
16 * @author Fahrul Adli
17 */
18 public class HelloServlet extends HttpServlet {
19
20     /**
21      * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
22      * methods.
23      *
24      * @param request servlet request
25      * @param response servlet response
26      * @throws ServletException if a servlet-specific error occurs
27      * @throws IOException if an I/O error occurs
28      */
29
30     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
31         throws ServletException, IOException {
32         response.setContentType("text/html;charset=UTF-8");
33         try (PrintWriter out = response.getWriter()) {
34             /* TODO output your page here. You may use following sample code. */
35             out.println("<!DOCTYPE html>");
36             out.println("<html>");
37             out.println("<head>");
38             out.println("<title>Servlet Saya Yang Pertama</title>");
39             out.println("</head>");
40             out.println("<body>");
41             out.println("<h1>Hello, Servlet!</h1>");
42             out.println("<h2>Servlet HelloServlet at " + request.getContextPath() + "</h2>");
43             out.println("</body>");
44             out.println("</html>");
45         }
46     }
47 }
```

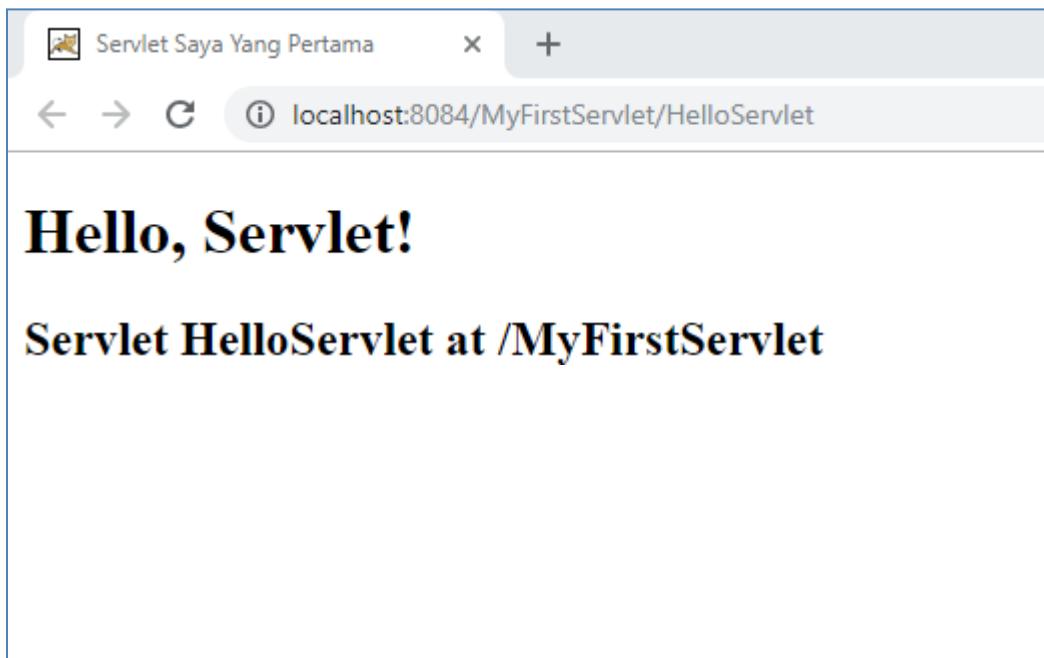
11. After finished your modification on the previous step, run *HelloServlet.java* by right-clicking on it and select *Run File*.



12. Before you can see the output, a dialogue box will appear. This dialogue box allows us to add some request parameter. At this moment, we will not supply any parameter. Just click OK button.



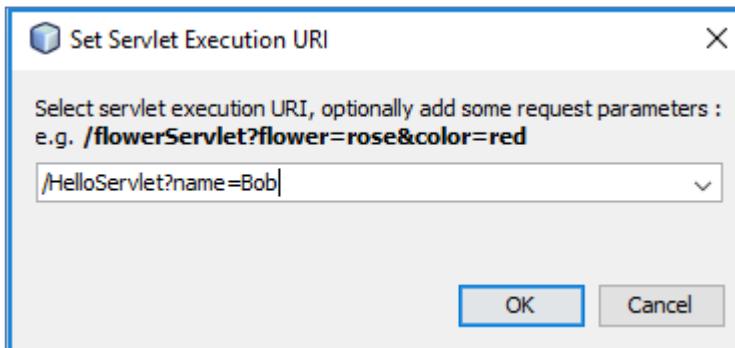
13. You will see the following output on your browser.



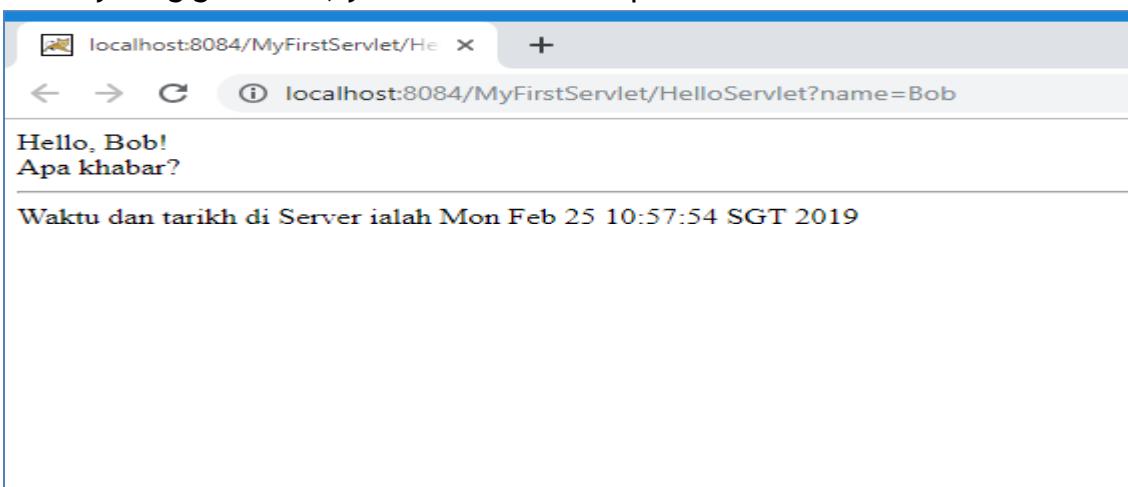
14. Now, we will make further modification to our servlet file. This time, we are going to modify the `doGet()` method. Browse through the codes in `HelloServlet.java` until you find the desired method. Type the codes as provided in the screenshot below. Read the comments carefully and try to understand what each of the codes does. **Remark: Please remove `processRequest()` method from the body of `doGet()` method.**

A screenshot of an IDE showing the code for `HelloServlet.java`. The code is as follows:Start Page X index.html X HelloServlet.java X
Source History 55 */
56 @Override
57 protected void doGet(HttpServletRequest request, HttpServletResponse response)
58 throws ServletException, IOException {
59
60 /* Step 1: Set the content type (tell the browser what is the type of
61 | the response data; e.g. text/html, text/plain. In our case, we will
62 | responds with html data. */
63 response.setContentType("text/html");
64
65 /* Step 2: Create the PrintWriter object. We name it as 'out'
66 */
67 PrintWriter out = response.getWriter();
68
69 /* Step 3: Read GET parameter sent by the user through the web browser
70 */
71 String name = request.getParameter("name");
72
73 /*Step 4: Generate content for our HTML response. Print the name
74 */
75 out.println("<html><body>");
76
77 out.println("Hello, "+name+"!"+
");
78 out.println("Apa khabar?"+<hr>");
79 out.println("Waktu dan tarikh di Server ialah "+new java.util.Date());
80 out.println("</html></body>");
81
82 }
83The code is annotated with line numbers and comments explaining the steps taken to generate an HTML response.

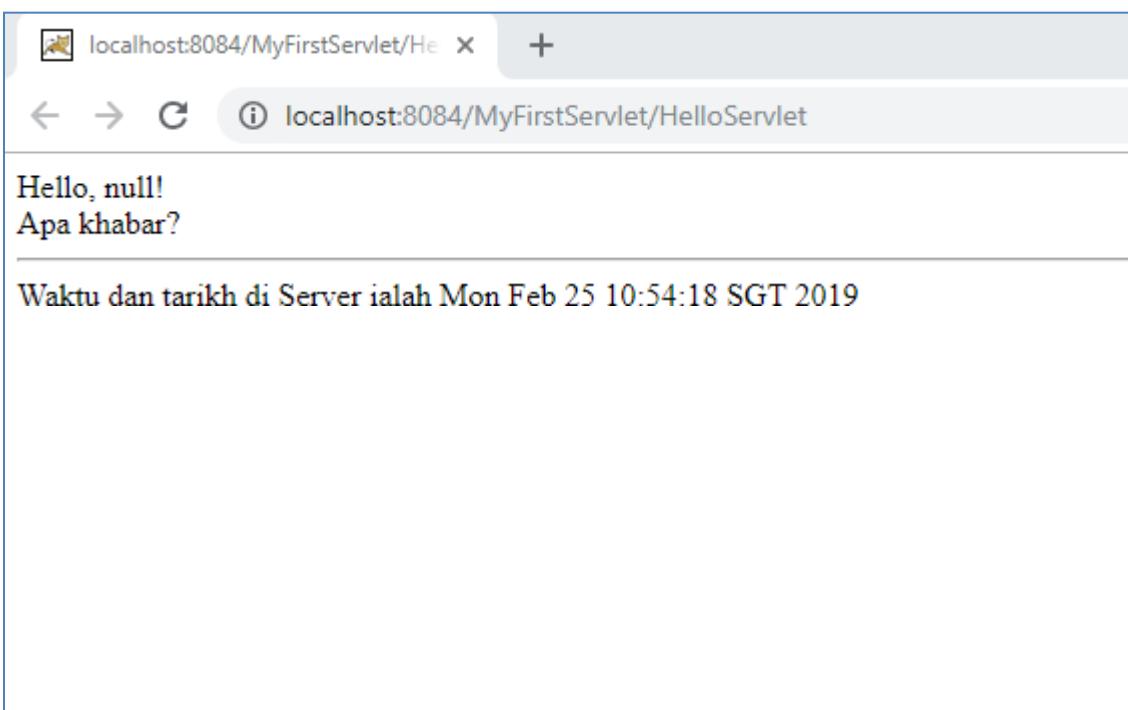
15. After finish, right click on *HelloServlet.java* and click Run. As you have seen previously, a dialogue box shows. This time, we will supply a value *Bob* to the parameter *name*. Then, click the OK button.



16. If everything goes well, you will see the output as below:



17. Rerun step 20 again, this time do not supply any request parameter. What do you see from the output? Do you see something like the following screenshot? How to avoid the *null* from being displayed?



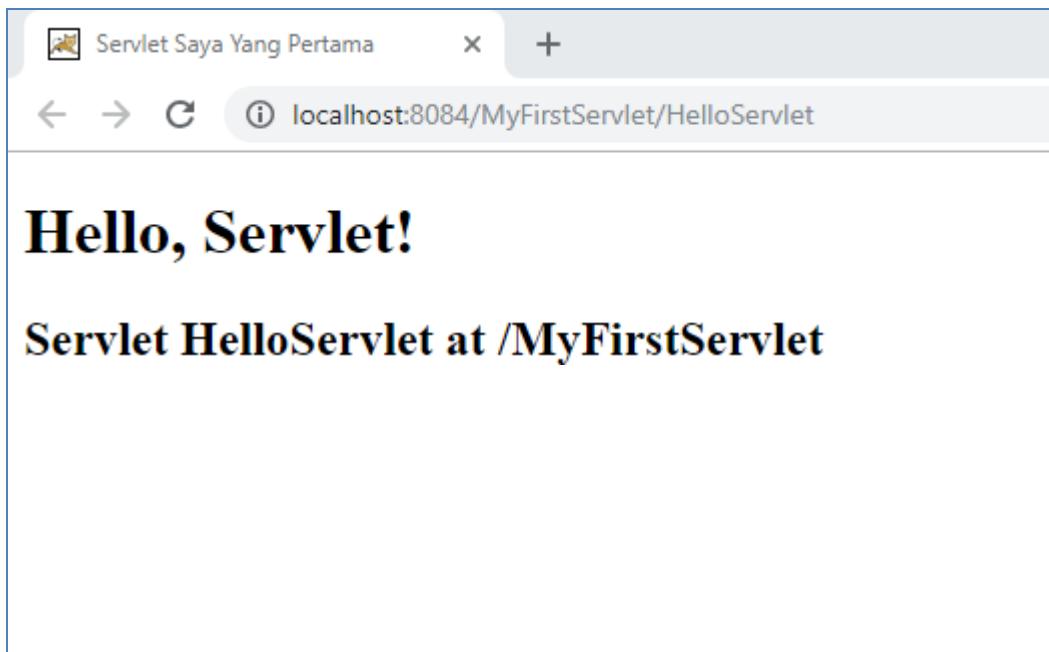
18. You can upgrade your code in HelloServlet.java by putting the following codes into it. By doing this, if no value supplied to the parameter *name*, the request will be passed to *processRequest()* method, and this will avoid from the *null* value appears on the browser.

The screenshot shows a Java IDE interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, Print, Find, Copy, Paste, Cut, Undo, Redo), a search bar, and other development tools.
- Tab Bar:** Displays three tabs: "Start Page", "index.html", and "HelloServlet.java".
- Source Editor:** The main window contains the Java code for "HelloServlet.java".
- Code Content:** The code is as follows:

```
Start Page x index.html x HelloServlet.java x
Source History | [New] [Open] [Save] [Print] [Find] [Copy] [Paste] [Cut] [Undo] [Redo] | [Run] [Stop] [Minimize] [Maximize] [Close]
55  /*
56  * @Override
57  * protected void doGet(HttpServletRequest request, HttpServletResponse response)
58  *     throws ServletException, IOException {
59
60      /* Step 1: Set the content type (tell the browser what is the type of
61         the response data; e.g. text/html, text/plain. In our case, we will
62         respond with html data. */
63      response.setContentType("text/html");
64
65      /* Step 2: Create the PrintWriter object. We name it as 'out'
66      */
67      PrintWriter out = response.getWriter();
68
69      /* Step 3: Read GET parameter sent by the user through the web browser
70      */
71      String name = request.getParameter("name");
72
73      /*
74      Additional: if no value for parameter "name", call processRequest method.
75      */
76      if(name==null)
77          processRequest(request, response);
78
79      /*Step 4: Generate content for our HTML response. Print the name
80      */
81      out.println("<html><body>");
82
83      out.println("Hello, "+name+"!"+<br>");
```

19. So, if you rerun the file and without supplying any parameter, you will see the output as follows:



It is the same output as can be seen in Step 18: why?

- processRequest() will be response and display their output. Name is null so NullPointerException will be throw.

Coding

```

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;
import jakarta.annotation.*;

public class HelloServlet extends HttpServlet
{

    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() + "</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}

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

    response.setContentType("text/html");

    PrintWriter out = response.getWriter();

    String name = request.getParameter("name");

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

    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("</html></body>");
    }

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

@Override
public String getServletInfo() {
    return "Short description";
}// </editor-fold>

}

```

```

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;
import jakarta.annotation.*;

public class HelloServlet extends HttpServlet
{

    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() + "</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }
}

```

```

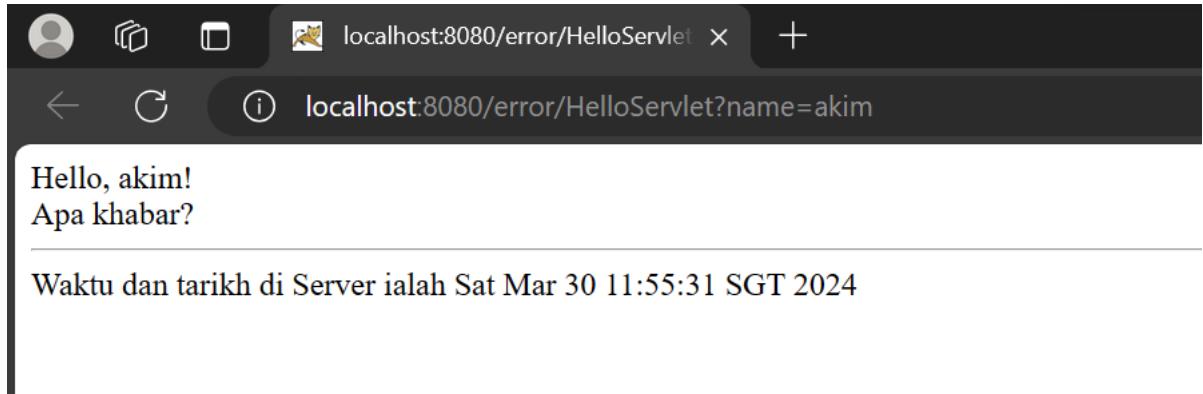
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String name = request.getParameter("name");
    if(name==null)
        processRequest(request, response);
    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("</html></body>");
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}

```

Output



Task 6: Writing a Simple JSP Program

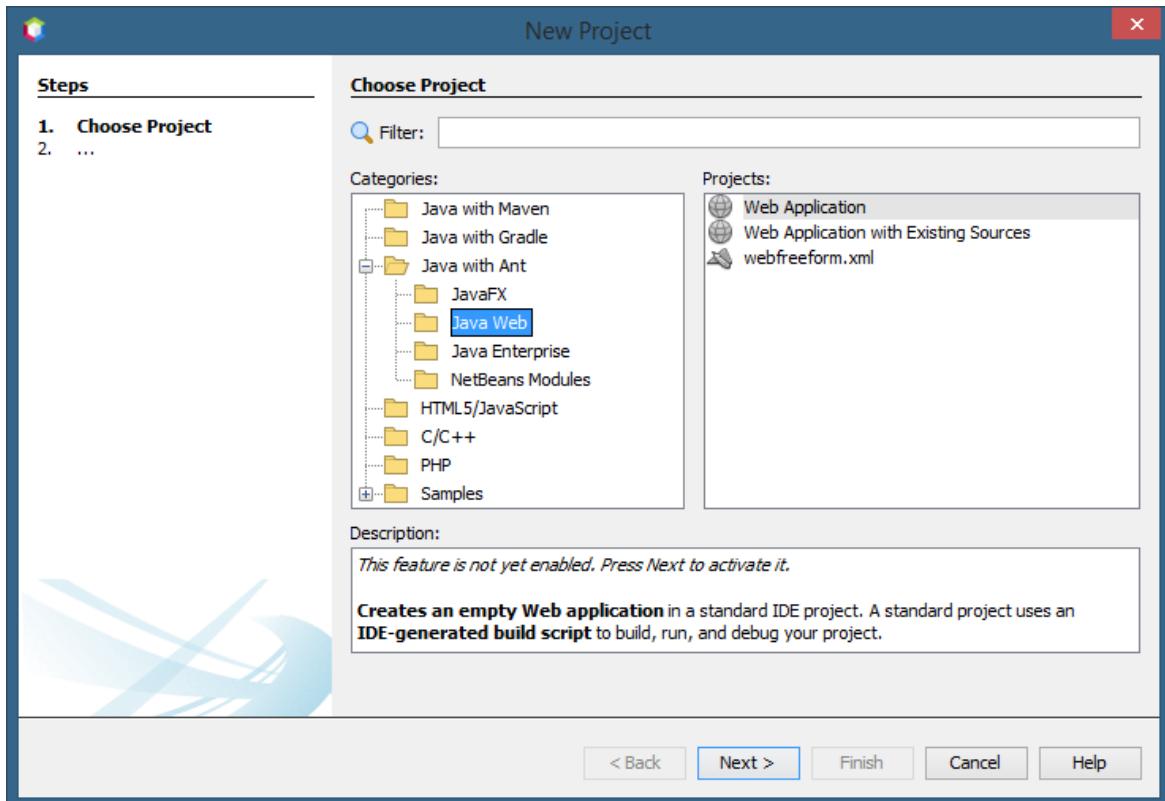
Objective : Writing a simple plain JSP program

Problem

Description : Write a simple plain JSP program to display
“Welcome to [MATRICNUMBER]..!”

Estimated time : 15 minutes

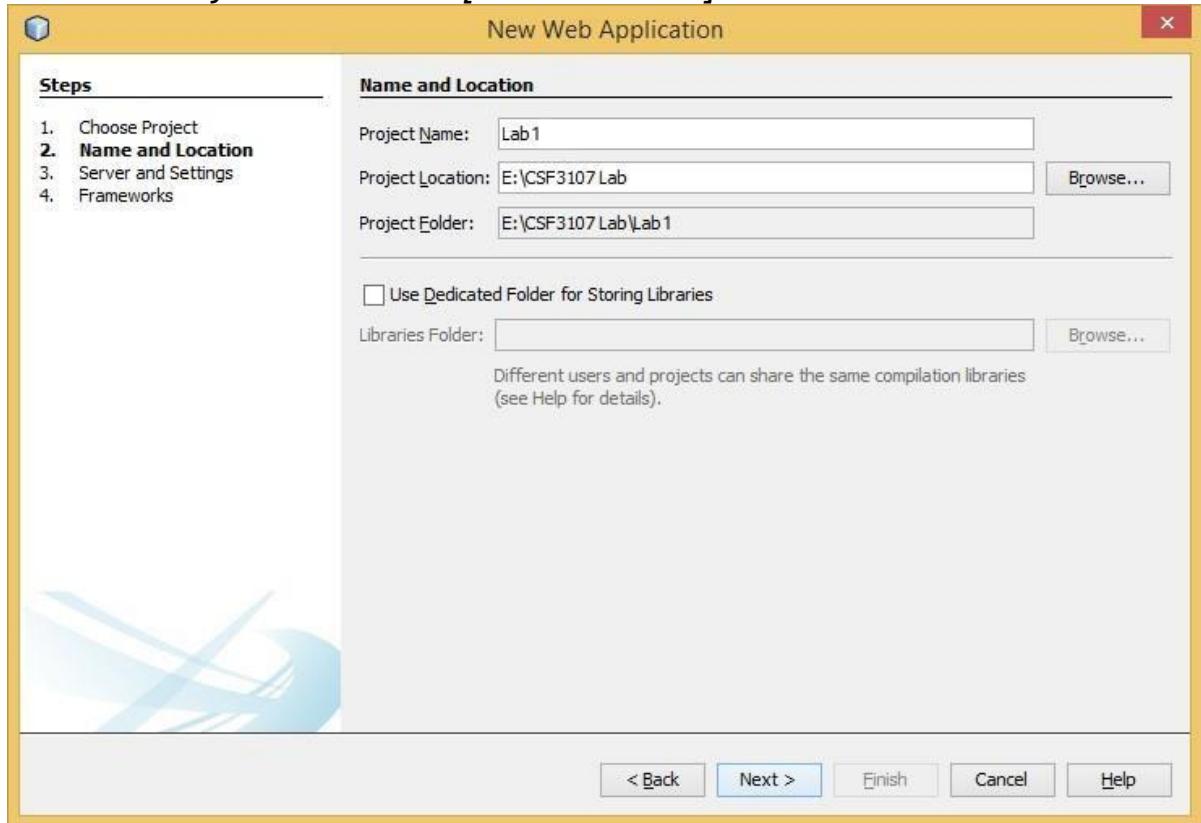
1. Create a directory C:\[MatricNumber] Lab
2. Go to C:\ [MatricNumber] Lab's directory and create sub-directory as Lab 1 - JSP.
3. Open your NetBeans.
4. Go to File -> New Project
5. Select Select Java with Ant -> Java Web -> Web Application and click Next.



6. Click the Next button.

7. Type Project Name: *Lab1*.

8. Choose Project Location: *C:\[MATRICNUMBER]\Lab 1.*

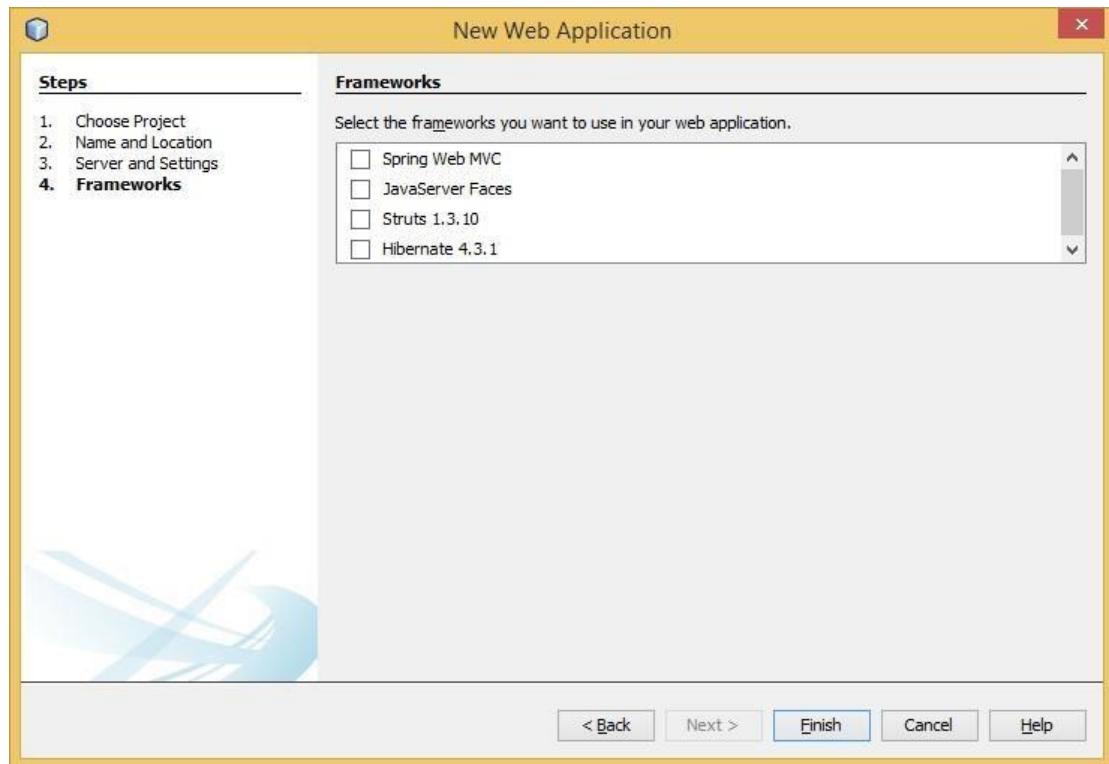


9. Click the *Next* button.

10. Select Server: *Apache Tomcat or TomEE*

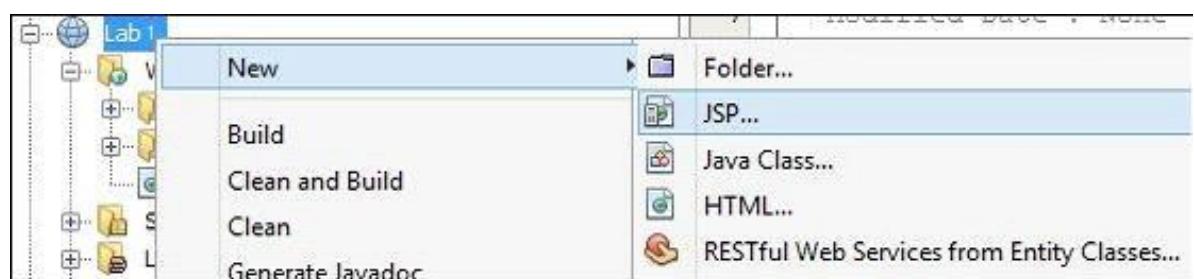
11. Select Java EE Version: *Java EE 6 Web.*

12. Click the Next button.

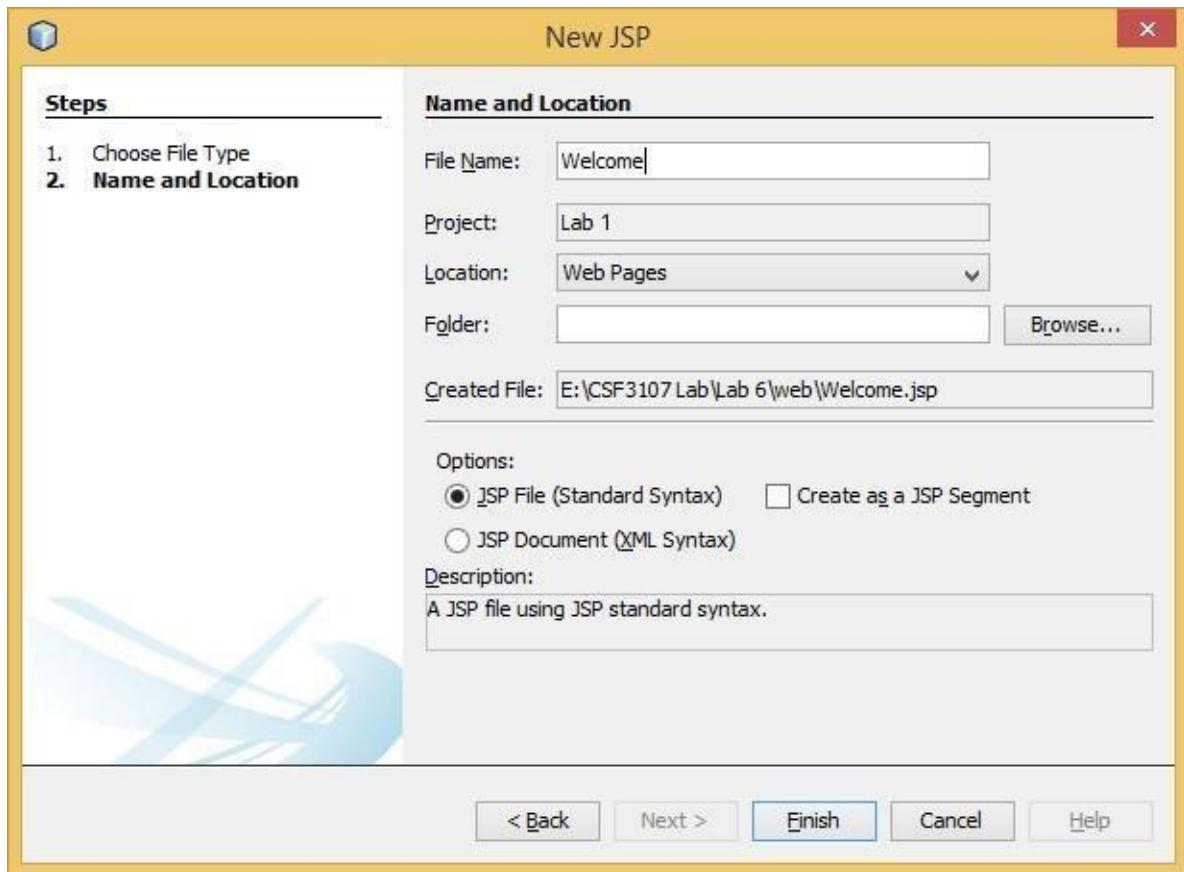


13. Click the Finish button.

14. Create a new JSP's file.



15. Type file name as *Welcome*.



16. Click the *Finish* button.

17. Type title as *[MATRICNUMBER] - Web Programming 2*

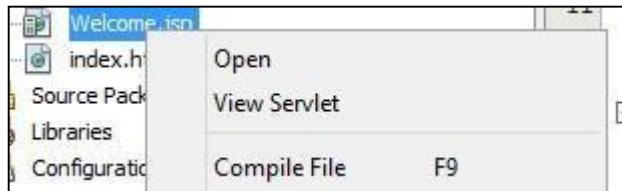
18. Type header1 as *Welcome to [MATRICNUMBER]...!*

```
<%--  
2 Document : Welcome.jsp  
3 Created on : 29-Mar-2016, 09:46:05  
4 Author : Mohamad Nor Hassan  
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>CSF3107 - Web Programming 2</title>  
13 </head>  
14 <body>  
15 <h1>Welcome to CSF3107...!</h1>  
16 </body>  
17 </html>
```

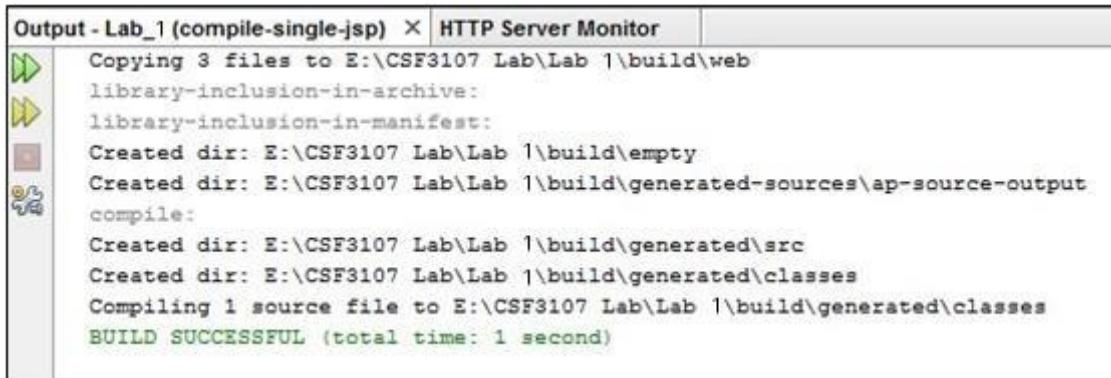
19. Click *SaveAll* icon



20. Right-click file *Welcome.jsp* and click *Compile File* (F9).



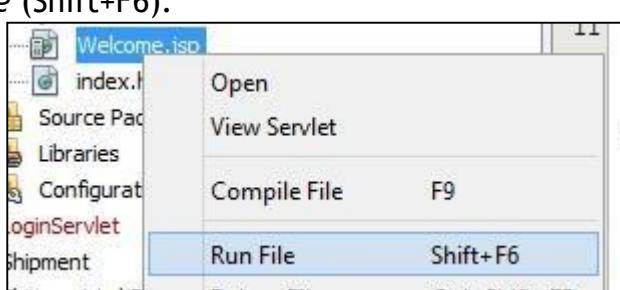
21. You will get notification message the bottom of Netbeans IDE with the green colour.



Note: Before running any JSP's files for the first time upon opening your Netbeans IDE, you need to start your web server (i.e., Apache Tomcat).

Note: Avoid these steps if Apache Tomcat already starts.

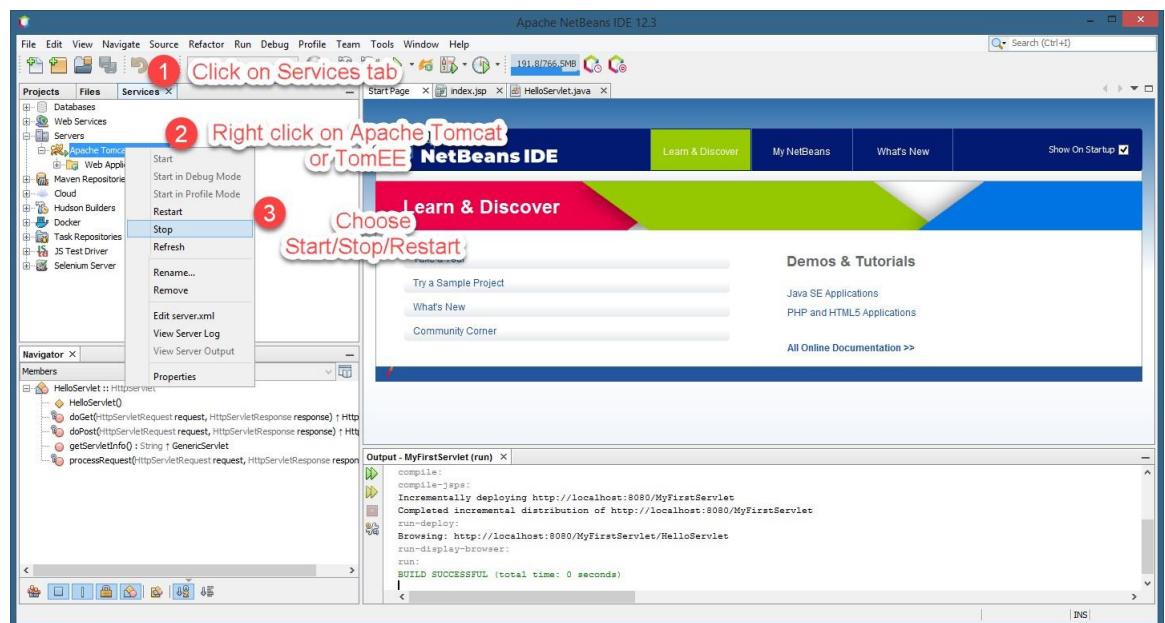
26. Go to the Project's tab. Then right click *Welcome.jsp* file and click *Run File* (Shift+F6).



27. The output will appear in a web browser.



Note: Beside using XAMPP to start or stop the Apache Tomcat, you may also do it directly from Netbeans as shown in the figure below.



Reflection

1. What have you learned from this exercise?

- Code for jsp is same with html but It's a server-side technology used in web application development. Also can use both HTML and JSP tags.

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

- JSP is Integrated in HTML content to create a basic JSP page. The HTML will displayed as a web page. When the file is processed on the server, data and server-side code will be called via the embedded JSP tags.

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?

JSP	SERVLET
<ul style="list-style-type: none"> - JSP is a HTML-based compilation code - JSP is easy to code as it is java in HTML - JSP only accepts HTTP requests. 	<ul style="list-style-type: none"> - Servlet is a java code - Writing code for servlet is harder than JSP as it is HTML in java - Servlet can accept all protocol requests.

Coding

```
<%--
 Document : Welcome
 Created on : 27 Mar 2024, 5:47:13 pm
 Author   : Luqman Hakim
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>CSF3107 - Web Programming 2</title>
</head>
<body>
<h1>Welcome to CSF3107...!</h1>
</body>
</html>
```

```
<%--
 Document : Welcome
 Created on : 27 Mar 2024, 5:47:13 pm
 Author   : Luqman Hakim
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>CSF3107 - Web Programming 2</title>
</head>
<body>
<h1>Welcome to CSF3107...!</h1>
</body>
</html>
```

Output

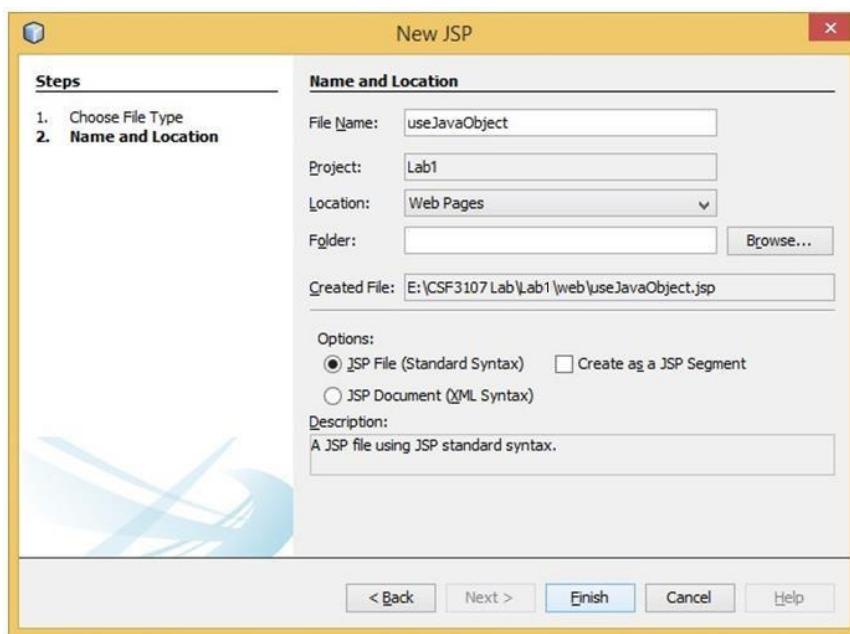


Task 7: Use Java Reference Datatype/Class Wrapper in JSP

Objective	: Using Java's object in JSP page.
Problem	: Display the current date, perform auto refresh header in
Description	JSP's page.

Estimated time : 20 minutes

1. Go to project *Lab1*.
2. Create a new JSP's file as *useJavaObject*.



3. Click the *Finish* button.
4. Change the title *Using Java's object in JSP page*.
5. Change the <h1> as *Display Current Date* and perform auto refresh header.

6. Add *Java util* package for Date reference.

```
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <%@page import="java.util.Date.*"%>
9
```

7. Write a Java Scriptlet to create Date's object and display the current date and time.

```
<body>
    <h1>Display Current Date and perform simple Mathematics operations </h1>

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

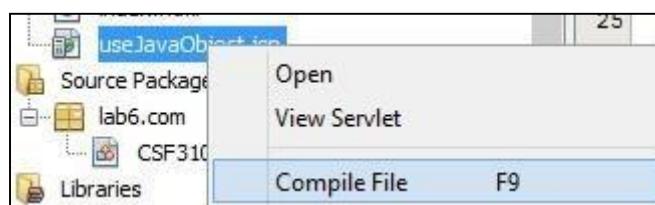
8. Continue writing a code to perform auto refresh header.

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

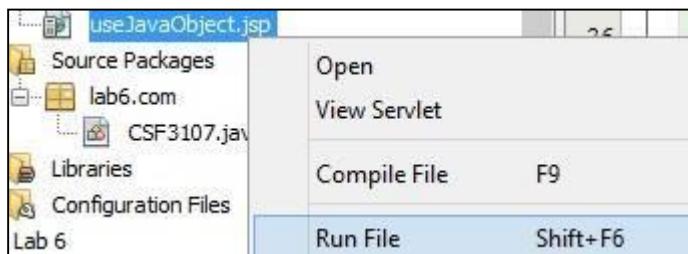
%>
```

9. Save your file.

10. Right-click *useJavaObject.jsp* and compile the program.



11. Finally, right-click *useJavaObject.jsp* and choose Run File to run the program.



12. Review the output display in the browser.



Reflection

1. What have you learnt from this exercise?

- Auto Refresh Header response header to enable auto-refresh of the page every 5 seconds. This is done using `response.setIntHeader("Refresh", 5)`

2. What is Java Scriptlet?

- is a Java program that extends the capabilities of a server. servlets can respond to any type of request, they are commonly used to extend the applications hosted by web servers.

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

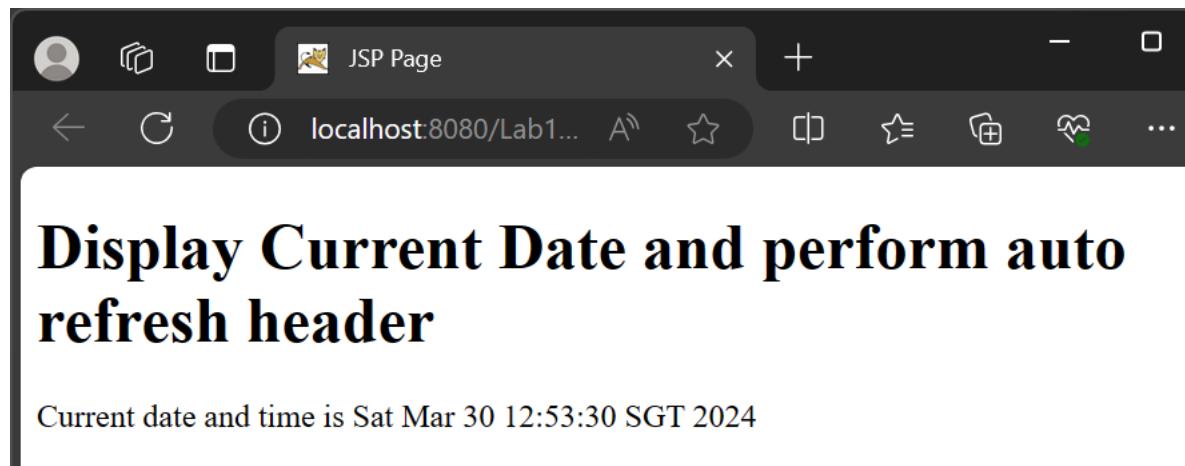
- To use Java code within JSP page, you can embed Java code snippets directly into the HTML markup using scriptlet tags (`<% %>`).

CODING

```
<%--  
Document : useJavaObject  
Created on : 27 Mar 2024, 5:53:45 pm  
Author   : Luqman Hakim  
--%>  
  
<%@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>JSP Page</title>  
    </head>  
    <body>  
        <h1>Display Current Date and perform auto refresh header</h1>  
  
        <%  
            Date todayDate = new Date();  
            out.println("<p>Current date and time is " + todayDate.toString() + "<p>");  
        %>  
  
        <%  
            response.setIntHeader("Refresh", 5);  
        %>  
    </body>  
</html>
```

```
<%--  
    Document      : useJavaObject  
    Created on   : 27 Mar 2024, 5:53:45 pm  
    Author       : Luqman Hakim  
--%>  
  
<%@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>JSP Page</title>  
    </head>  
    <body>  
        <h1>Display Current Date and perform auto refresh header</h1>  
  
        <%  
            Date todayDate = new Date();  
            out.println("<p>Current date and time is " + todayDate.toString() + "</p>");  
        %>  
  
        <%  
            response.setIntHeader("Refresh", 5);  
        %>  
    </body>  
</html>
```

OUTPUT



Task 8: Using JSP Implicit object in JSP page

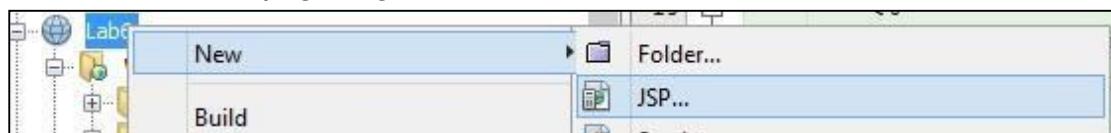
Objective : Using JSP Implicit object (Session) in JSP page.

Problem : Using Session object, perform simple Mathematics

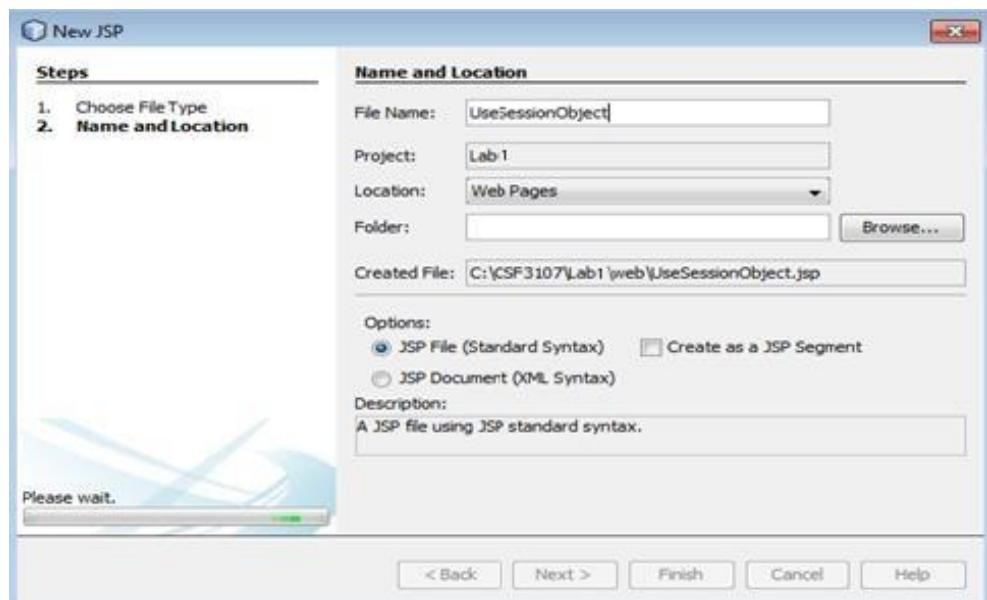
Description operations in JSP's page.

Estimated time : 30 minutes

1. Go to Project *Lab1*.
2. To create a JSP's page, right click *Lab1*-> *New* -> *JSP*.



3. Create a new JSP's file as *AttributesSet*.



4. Click the *Finish* button.
5. Source code for *AttributelsSet.jsp* will appear.
6. Write the code below:

```
<%-->
Document : jsp
Created on : 20-Feb-2018
Author : Dr. Faizah Aplop
--%>

<%@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", "Fouad Abdulameer");%>
    <a href="GetAttribute.jsp">Click here to get user name </a>

  </body>
</html>
```

9. Save and compile *AttributelsSet.jsp* file.
10. Run the *AttributelsSet.jsp* file, and you should get the interface as below:



11. Repeat step 1 and step 2.
12. Key-in File Name: *GetAttribute*.

13. Click the *Finish* button.

14. Source code for *GetAttribute.jsp* will appear.

15. Write the *GetAttribute* code.

```
<%@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>
```

17. Compile *GetAttribute.jsp* file.

18. Run the *AttributelsSet.jsp* file and click on the link.

19. Add math package to perform simple Mathematics operation in your page.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.math.*"%>
```

20. Create a new JSP's file as *MathematicsOperations* to perform addition, multiplication and find the square roots of the number.

```
<%
int num1 = 25;
int num2 = 10;
int addition_output;
int 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;

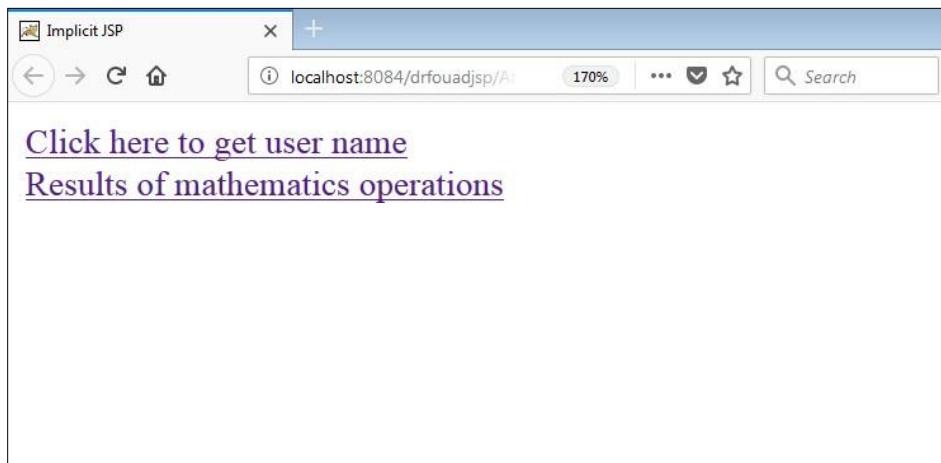
//Find square root for variable num1..
squareroot = (double)(Math.sqrt(num1));

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

%>
```

21. You should get the interface as below:



22. Review the outputs display in the browser.

Reflection

1. How do you want to submit specific information from one form to next form?
 - using GET method, include the specific information in the URL as parameters. Information will be visible in the URL and can be accessed by the next form using request parameters.
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?
 - If the field name specified in `request.getParameter("field_name")` in the second page is different from the field name defined in the first page, the value retrieved in the second page will be null.

CODING

```
<%--  
    Document : AttributeIsSet  
    Created on : 30 Mar 2024, 2:11:28 pm  
    Author : Luqman Hakim  
--%>  
  
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<!DOCTYPE html>  
<html>  
    <head>  
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
        <title>Implicit JSB</title>  
    </head>  
    <body>  
  
        <% session.setAttribute("user", "Foud Abdulameer");%>  
        <a href="GetAttribute.jsp">Click here to get user name</a>  
        <br>  
        <a href="MathematicsOperations.jsp">Results of mathematics operations</a>  
  
    </body>  
</html>
```

```
<%--  
    Document : AttributeIsSet  
    Created on : 30 Mar 2024, 2:11:28 pm  
    Author : Luqman Hakim  
--%>  
  
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<!DOCTYPE html>  
<html>  
    <head>  
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
        <title>Implicit JSB</title>  
    </head>  
    <body>  
  
        <% session.setAttribute("user", "Foud Abdulameer");%>  
        <a href="GetAttribute.jsp">Click here to get user name</a>  
        <br>  
        <a href="MathematicsOperations.jsp">Results of mathematics operations</a>  
  
    </body>  
</html>
```

```

<%--
 Document : GetAttribute
 Created on : 30 Mar 2024, 2:15:37 pm
 Author    : Luqman Hakim
--%>

<%@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     : GetAttribute
 Created on   : 30 Mar 2024, 2:15:37 pm
 Author       : Luqman Hakim
--%>

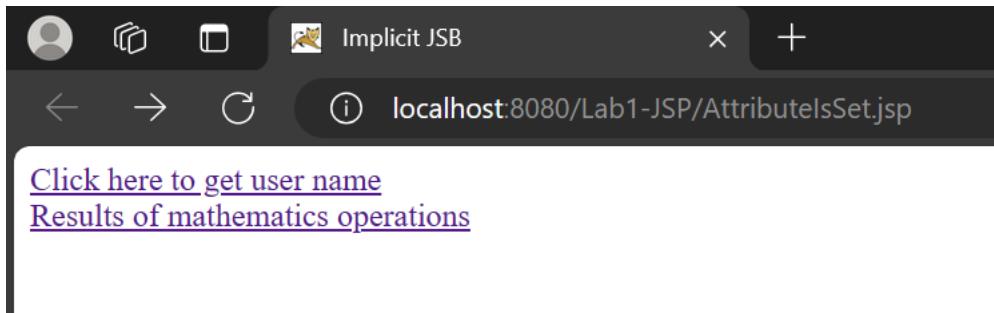
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Implicit JSP</title>
  </head>
  <body>

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

  </body>
</html>

```

OUTPUT



CODING

```
<%--  
    Document : MathematicsOperations  
    Created on : 30 Mar 2024, 2:37:15 pm  
    Author   : Luqman Hakim  
--%>  
  
<%@page import="java.math.*"%>  
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<!DOCTYPE html>  
<html>  
    <head>  
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
        <title>JSP Page</title>  
    </head>  
    <body>  
        <%  
            int num1 = 25;  
            int num2 = 10;  
            int addition_output;  
            int multiply_output;  
            double squareroot = 0.00;
```

```

addition_output = num1 + num2;
multiply_output = num1 * num2;
squareroot = Math.sqrt(num1);
%>

<p>Addition num1 and num2 is <%= addition_output %></p>
<p>Multiplication num1 and num2 is <%= multiply_output %></p>
<p>Square root of <%= num1 %> is <%= String.format("%.2f", squareroot) %></p>
</body>
</html>

```

```

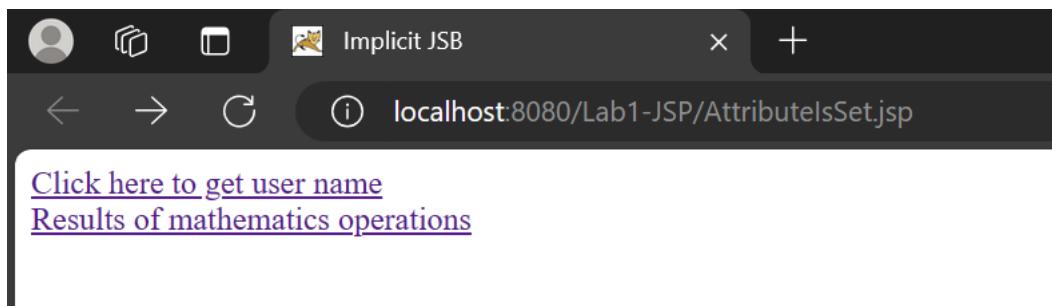
<%@page import="java.math.*"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <%
            int num1 = 25;
            int num2 = 10;
            int addition_output;
            int multiply_output;
            double squareroot = 0.00;

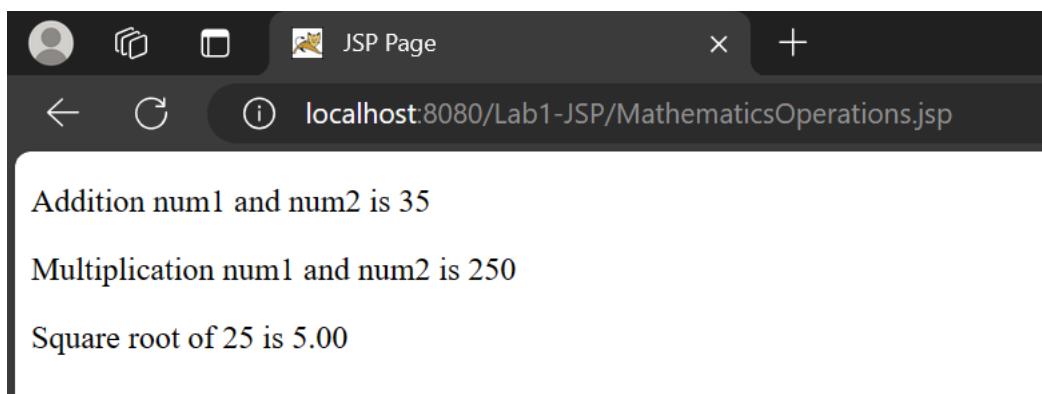
            addition_output = num1 + num2;
            multiply_output = num1 * num2;
            squareroot = Math.sqrt(num1);
        %>

        <p>Addition num1 and num2 is <%= addition_output %></p>
        <p>Multiplication num1 and num2 is <%= multiply_output %></p>
        <p>Square root of <%= num1 %> is <%= String.format("%.2f", squareroot) %></p>
    </body>
</html>

```

OUTPUT





Task 9: Populate Array values into HTML's Table

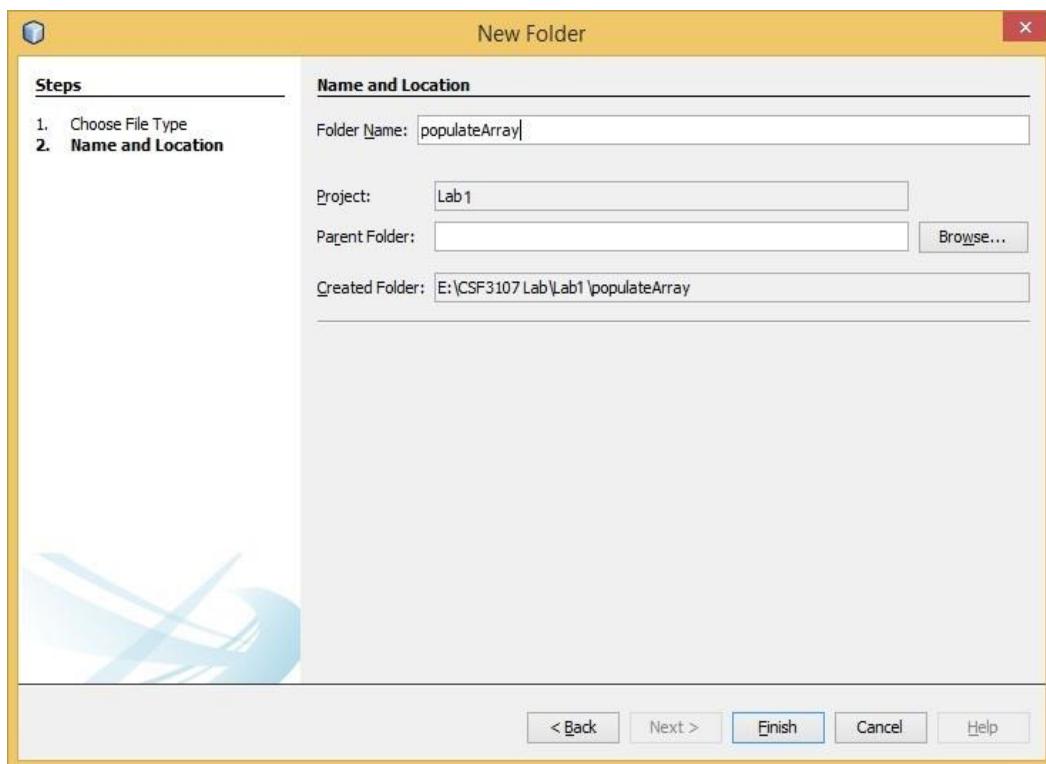
Objective : Read Java array and populate it into HTML's table.

Problem : i. Create a 2D array that store sales data.

Description : ii. Then, read an array and populate into HTML's table.

Estimated time : 50 minutes

1. Go to Project *Lab1*.
2. To create a JSP's page, right click *Lab1* -> *New* -> *JSP*.
3. Key-in File Name: *populateArray*.



4. Click the *Finish* button.

5. Prepare standard HTML's markup for page *populateArray.jsp*.
6. Write a Java Scriptlet and store the following information into an array;

	Jan	Feb	Mac
Salesman 1	2500	2100	2200
Salesman 2	2000	1900	2400
Salesman 3	1800	2200	2450

7. Read the array and populate its value into HTML's table.
8. Save and compile *populateArray.jsp* file.
9. Run the *populateArray.jsp* file and sample of output shows as below:

The screenshot shows a web browser window with the URL 'localhost:8084/Lab1/populateArray.jsp'. The page title is 'Read Java array and populate it into HTML's table'. Below the title is a table with four columns: Salesman, Jan, Feb, and Mac. The data rows correspond to the values in the first table. At the bottom left of the page, there is a copyright notice: '©2016-Mohamad Nor'.

Salesman	Jan	Feb	Mac
Salesman 1	2500	2100	2200
Salesman 2	2000	1900	2400
Salesman 3	1800	2200	2450

Reflection

1. Write a sample syntax to declare 2D Java array.
 - `dataType[][] arrayName = new dataType[m][n];`
2. Define a sequence of steps on how you accomplish Task 7.
 - `<%`
`List<String[]> salesmans = new ArrayList<>();`
`salesmans.add(new String[]{"Salesman 1", "2500", "2100", "2200"});`
`salesmans.add(new String[]{"Salesman 2", "2000", "1900", "2400"});`
`salesmans.add(new String[]{"Salesman 3", "1800", "2200", "2450"});`
`%>`
`<% for (String[] salesman : salesmans) { %>`

```

<tr>
    <td><%= salesman[0] %></td>
    <td><%= salesman[1] %></td>
    <td><%= salesman[2] %></td>
    <td><%= salesman[3] %></td>
</tr>
<% } %>

```

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

JSP	HTML
<ul style="list-style-type: none"> - JSP generated dynamic web pages only. - JSP runs straight on the Web Server and local JVM. - JSP is termed as server-side scripting language. 	<ul style="list-style-type: none"> - Html generated static web pages only. - HTML runs in the Web Browser. - HTML is termed as client-side scripting language.

CODING

```

<%--
 Document : populateArray
 Created on : 30 Mar 2024, 3:13:48 pm
 Author   : Luqman Hakim
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>populate Array</title>
    <style>
        table {
            border-collapse: collapse;
            width: 50%;
            margin: 20px auto;
            float: left;
        }
        th, td {
            padding: 10px;
            border: 1px solid #ddd;
            text-align: left;
        }
    </style>
</head>
<body>
    <table border="1">
        <tr>
            <th>Name</th>
            <th>Address</th>
            <th>City</th>
            <th>State</th>
            <th>Zip</th>
        </tr>
        <% for (int i = 0; i < salesman.length; i++) { %>
            <tr>
                <td><%= salesman[i].name %></td>
                <td><%= salesman[i].address %></td>
                <td><%= salesman[i].city %></td>
                <td><%= salesman[i].state %></td>
                <td><%= salesman[i].zip %></td>
            </tr>
        <% } %>
    </table>
</body>
</html>

```

```

        th {
            background-color: #f2f2f2;
        }
    </style>
</head>
<body>
    <h1>Read Java array and populate it into HTML's table</h1>

    <%@ page import="java.util.ArrayList" %>
    <%@ page import="java.util.List" %>
    <%@ page import="java.util.Arrays" %>

    <%
        List<String[]> salesmans = new ArrayList<>();
        salesmans.add(new String[]{"Salesman 1", "2500", "2100", "2200"});
        salesmans.add(new String[]{"Salesman 2", "2000", "1900", "2400"});
        salesmans.add(new String[]{"Salesman 3", "1800", "2200", "2450"});
    %>

    <table border="1">
        <tr>
            <th>Salesman</th>
            <th>Jan</th>
            <th>Feb</th>
            <th>Mac</th>
        </tr>
        <% for (String[] salesman : salesmans) { %>
            <tr>
                <td><%= salesman[0] %></td>
                <td><%= salesman[1] %></td>
                <td><%= salesman[2] %></td>
                <td><%= salesman[3] %></td>
            </tr>
        <% } %>
    </table>

    </body>
</html>

```

```
<%--  
    Document : populateArray  
    Created on : 30 Mar 2024, 3:13:48 pm  
    Author : Luqman Hakim  
--%>  
  
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<!DOCTYPE html>  
<html>  
<head>  
    <meta charset="UTF-8">  
    <title>populate Array</title>  
    <style>  
        table {  
            border-collapse: collapse;  
            width: 50%;  
            margin: 20px auto;  
            float: left;  
        }  
        th, td {  
            padding: 10px;  
            border: 1px solid #ddd;  
            text-align: left;  
        }  
        th {  
            background-color: #f2f2f2;  
        }  
    </style>  
</head>
```

```

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

    <%@ page import="java.util.ArrayList" %>
    <%@ page import="java.util.List" %>
    <%@ page import="java.util.Arrays" %>

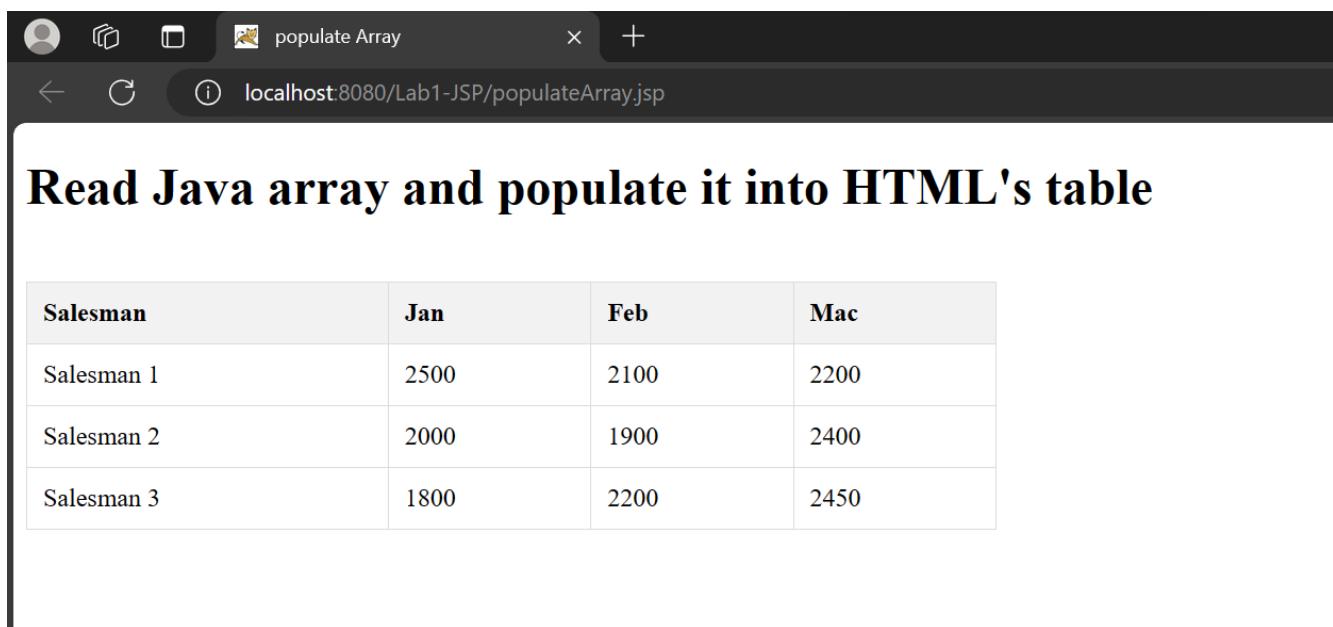
    <%
        List<String[]> salesmans = new ArrayList<>();
        salesmans.add(new String[]{"Salesman 1", "2500", "2100", "2200"});
        salesmans.add(new String[]{"Salesman 2", "2000", "1900", "2400"});
        salesmans.add(new String[]{"Salesman 3", "1800", "2200", "2450"});
    %>

    <table border="1">
        <tr>
            <th>Salesman</th>
            <th>Jan</th>
            <th>Feb</th>
            <th>Mac</th>
        </tr>
        <% for (String[] salesman : salesmans) { %>
            <tr>
                <td><%= salesman[0] %></td>
                <td><%= salesman[1] %></td>
                <td><%= salesman[2] %></td>
                <td><%= salesman[3] %></td>
            </tr>
        <% } %>
    </table>

</body>
</html>

```

OUTPUT

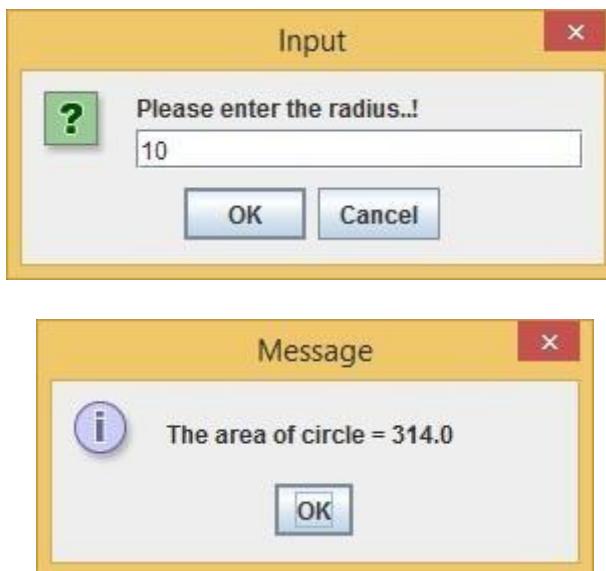


The screenshot shows a web browser window with the title "populate Array". The URL in the address bar is "localhost:8080/Lab1-JSP/populateArray.jsp". The main content of the page is a heading "Read Java array and populate it into HTML's table" followed by a table with three rows of data.

Salesman	Jan	Feb	Mac
Salesman 1	2500	2100	2200
Salesman 2	2000	1900	2400
Salesman 3	1800	2200	2450

Exercise

1. The following program is developed using Java Standard Edition.



Convert this program into Web-Based application where the user can dynamically key-in radius and submit the request to get the area of a circle. You must ensure to accept only number as a radius.

Coding

<%--

Document : calculateArea

Created on : 31 Mar 2024, 12:58:51 am

Author : Luqman Hakim

--%>

```
<%@ 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>

```

```

<%--
Document      : calculateArea
Created on   : 31 Mar 2024, 12:58:51 am
Author        : Luqman Hakim
--%>

<%@ 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>

```

<%--

Document : getRadius

Created on : 31 Mar 2024, 12:56:03 am

Author : User

--%>

<%@page contentType="text/html" 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>

```

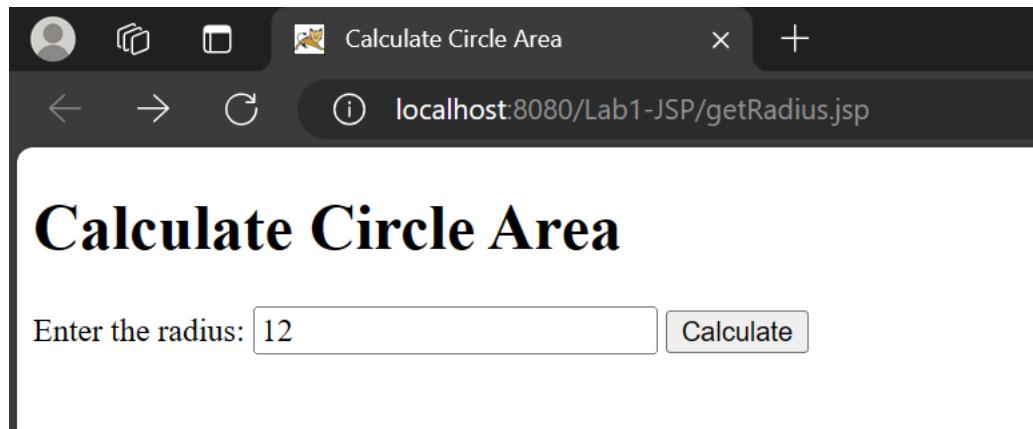
```

<%-->
    Document      : getRadius
    Created on   : 31 Mar 2024, 12:56:03 am
    Author        : User
--%>

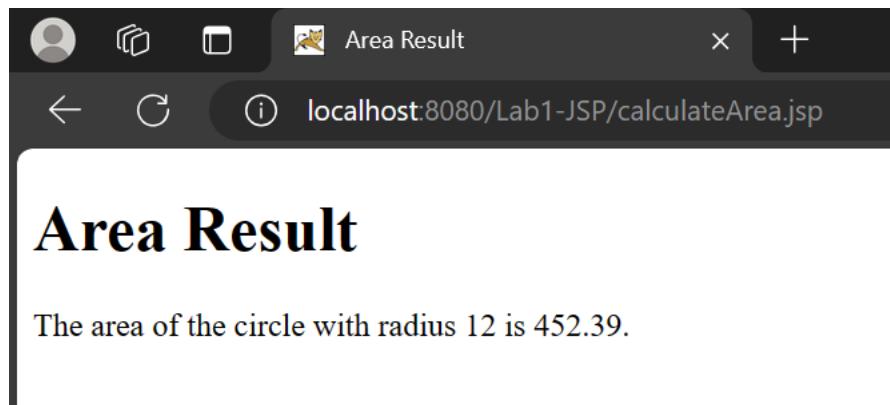
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Calculate Circle Area</title>
    </head>
    <body>
        <h1>Calculate Circle Area</h1>
        <form method="post" action="calculateArea.jsp">
            <label 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>

```

Output

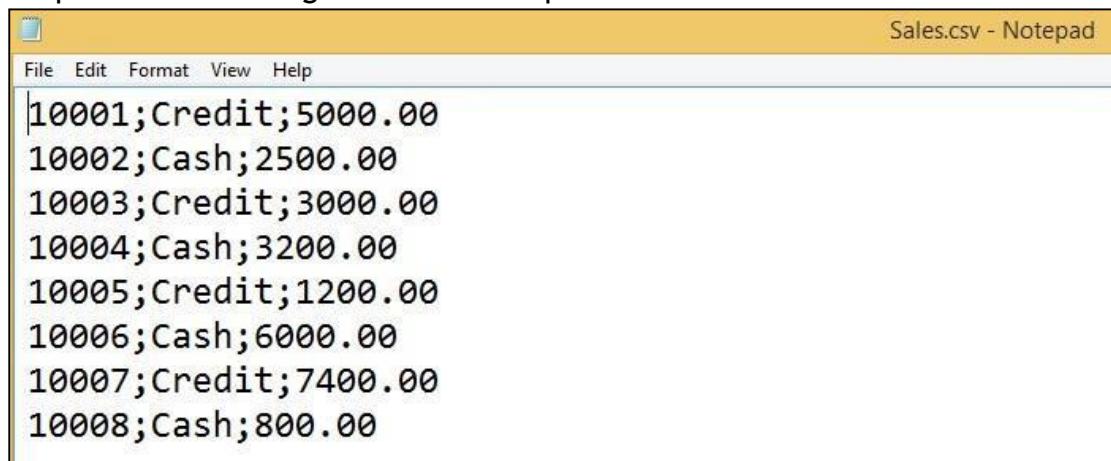


A screenshot of a web browser window titled "Calculate Circle Area". The address bar shows "localhost:8080/Lab1-JSP/getRadius.jsp". The main content area displays the heading "Calculate Circle Area" and a form with the placeholder "Enter the radius:" followed by an input field containing "12" and a "Calculate" button.



A screenshot of a web browser window titled "Area Result". The address bar shows "localhost:8080/Lab1-JSP/calculateArea.jsp". The main content area displays the heading "Area Result" and the text "The area of the circle with radius 12 is 452.39."

2. Prepare the following file in the Notepad and save it as *Sales.csv*.



A screenshot of a Windows Notepad window titled "Sales.csv - Notepad". The window contains the following CSV data:

10001	Credit	5000.00
10002	Cash	2500.00
10003	Credit	3000.00
10004	Cash	3200.00
10005	Credit	1200.00
10006	Cash	6000.00
10007	Credit	7400.00
10008	Cash	800.00

Create a simple JSP's page to read Sales.csv file. Upon reading Sales.csv's file, calculate the discount and populate HTML's table in the JSP's page.

Customer	Cust. Type	Purchase	Discount
10001	Credit	5000	0.00
10002	Cash	2500	250.00
10003	Credit	3000	0.00
10004	Cash	3200	320.00
10005	Credit	1200	0.00
10006	Cash	6000	600.00
10007	Credit	7400	0.00
10008	Cash	800	80.00

Cash customers are eligible to get a 10% discount.

Coding

```
<%--  
Document : readFile  
Created on : 31 Mar 2024, 1:08:56 am  
Author   : Luqman Hakim  
--%>  
  
<%@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>

<%
String filePath = "C:/CSM3023/S66292_Lab1/Sales.csv";

List<String[]> csvData = new ArrayList<>();

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>CustType</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;
    }
%>
<td><%= customer %></td>
<td><%= custType %></td>
<td><%= purchase %></td>
<td><%= String.format("%.2f", discount) %></td>
</tr>
<% } %>
</tbody>
</table>
</body>
</html>
```

```

<%--
    Document : readFile
    Created on : 31 Mar 2024, 1:08:56 am
    Author : Luqman Hakim
--%>

<%@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>

```

<%
    String filePath = "C:/CSM3023/S66292_Lab1/Sales.csv";

    List<String[]> csvData = new ArrayList<>();

    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>CustType</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;
            }
        %>
        <td><%= customer %></td>
        <td><%= custType %></td>
        <td><%= purchase %></td>
        <td><%= String.format("%.2f", discount) %></td>
    </tr>
<% } %>
</tbody>
</table>
</body>
</html>

```

Output

Read CSV files and populate it into HTML's table

Customer	CustType	Purchase	Discount
10001	Credit	5000	0.00
10002	Cash	2500	250.00
10003	Credit	3000	0.00
10004	Cash	3200	320.00
10005	Credit	1200	0.00
10006	Cash	6000	600.00
10007	Credit	7400	0.00
10008	Cash	800	80.00