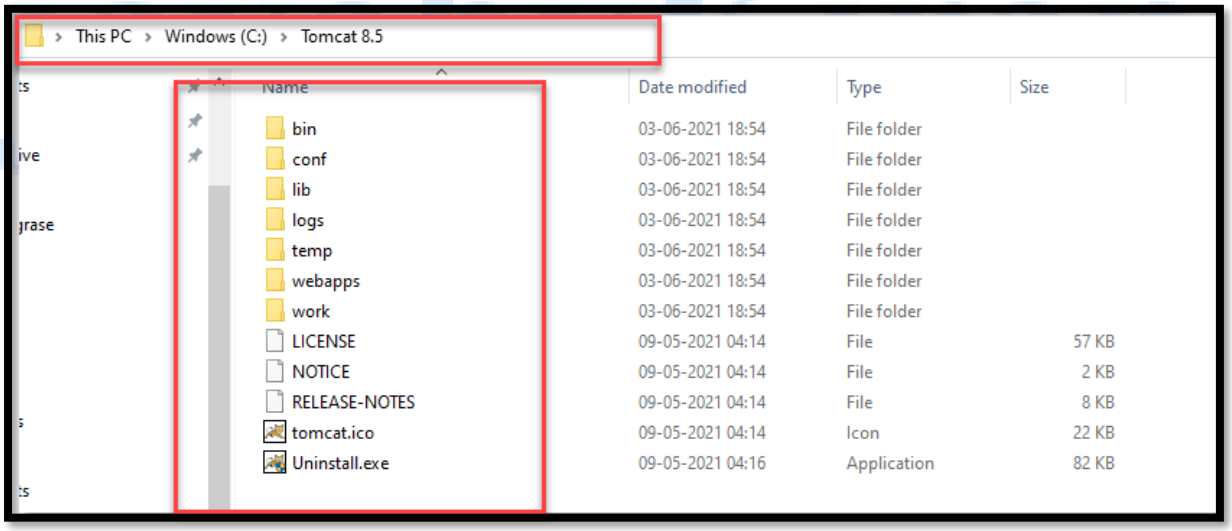
SERVLETS

- ✓ Servlets is Web Technology which is used to develop the Web Application.
- ✓ Servlets is server-side component which receives the Request from Browser and process the Request and sends the Response to Browser.

Steps to Develop the first Servlet example: -

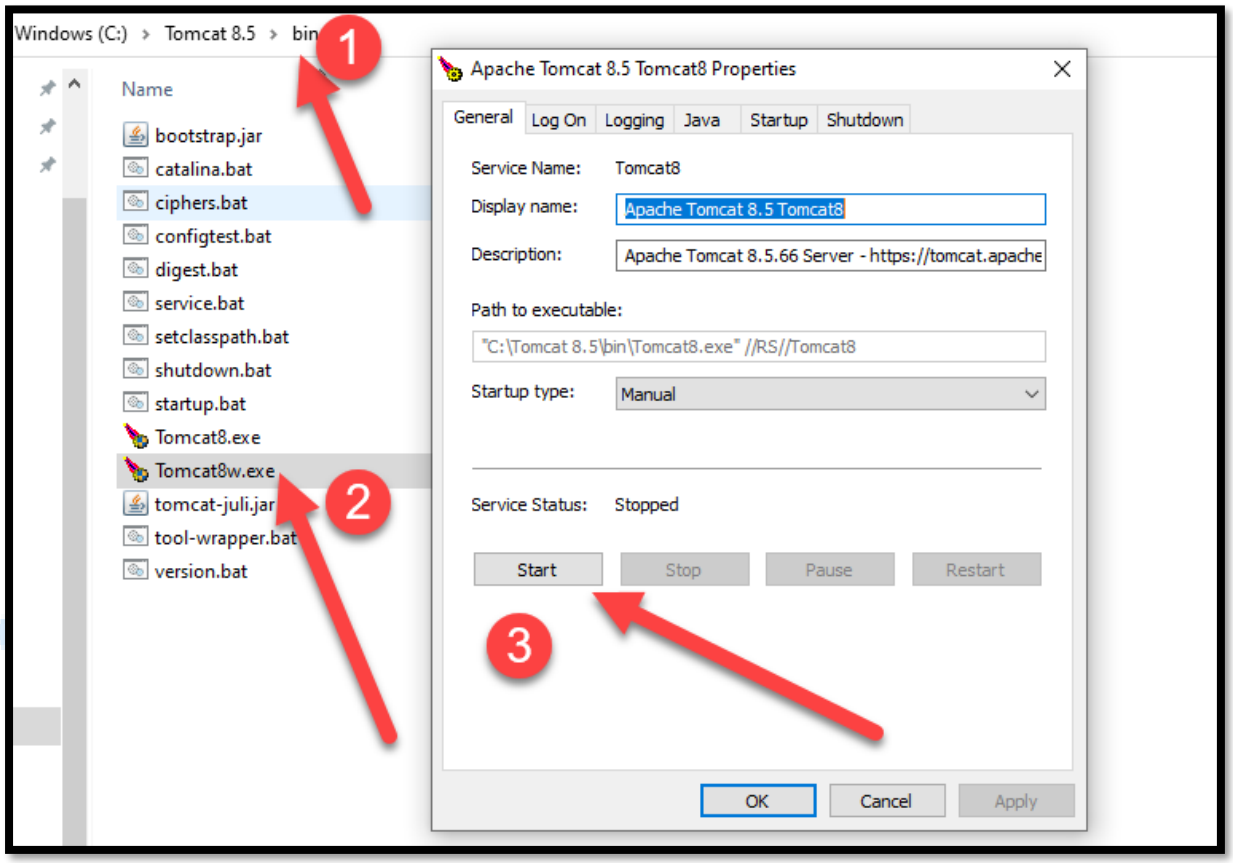
- ✓ Steps to install Apache Tomcat 8.5
- ✓ Click on the installer called tomcat - 8.5.exe. [download from apache site]
- ✓ Click on the next button.
- ✓ Click on I agree Button.
- ✓ Click on the Next Button.
- ✓ Provide **C:\Tomcat 8.5** as destination folder and click on Next Button [try D drive].
- ✓ Provide **port: 9999, username: jbk password:jbk**
  [8080 is default port number if oracle is there on system, it gives problem]
- ✓ Please click on the Next Button and then finish button.

After installing successfully, you can see the following directly structure.



Starting the tomcat [without eclipse]

Click on **C:\Tomcat 8.5\bin** open the highlighted.



Check if tomcat started properly.

Open the browser and type the following: - [make sure you use only port used while installation]

http://localhost:8080/

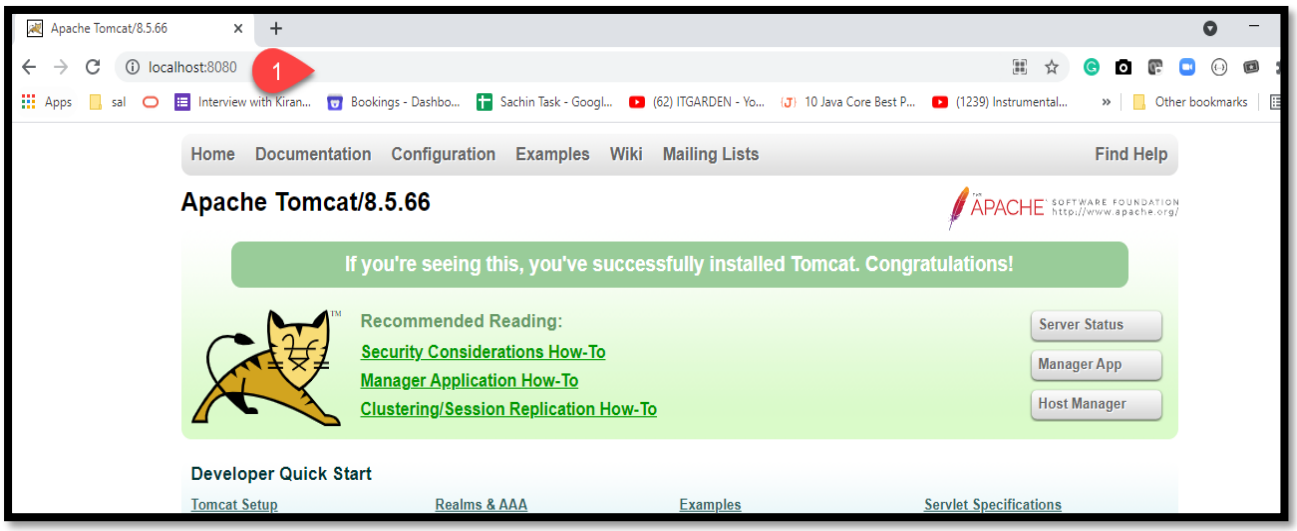
or

http://localhost:8090/

or

http://localhost:9999/

You should see below page



Note: - you can change the port number of the tomcat from the file server.xml which is located in the **C:\Tomcat 8.5\conf\server.xml**

Inside server.xml

<Connector port="8080" protocol="HTTP/1.1"

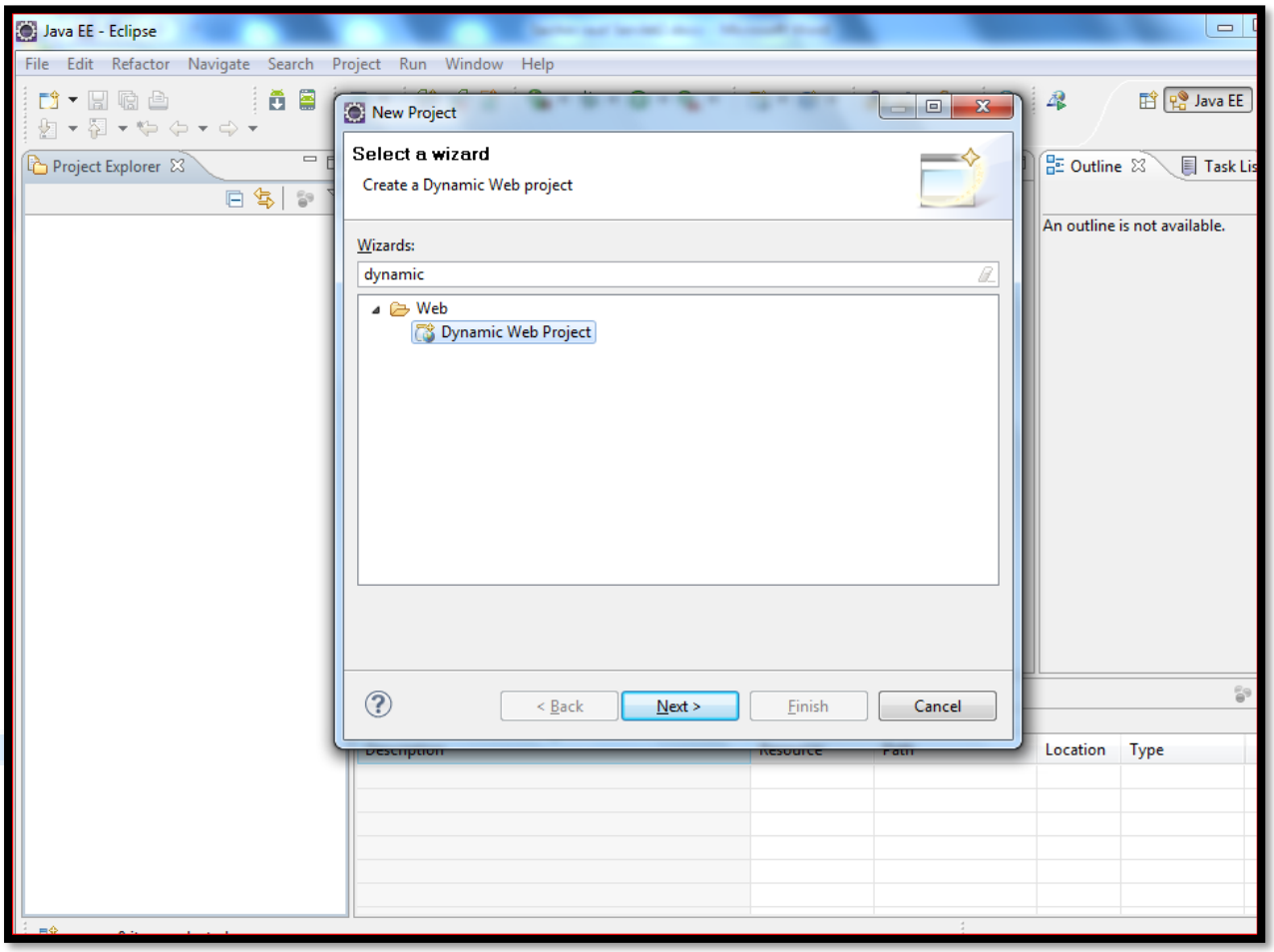
connectionTimeout="20000"

redirectPort="8443" />

tag is used to change the port number.

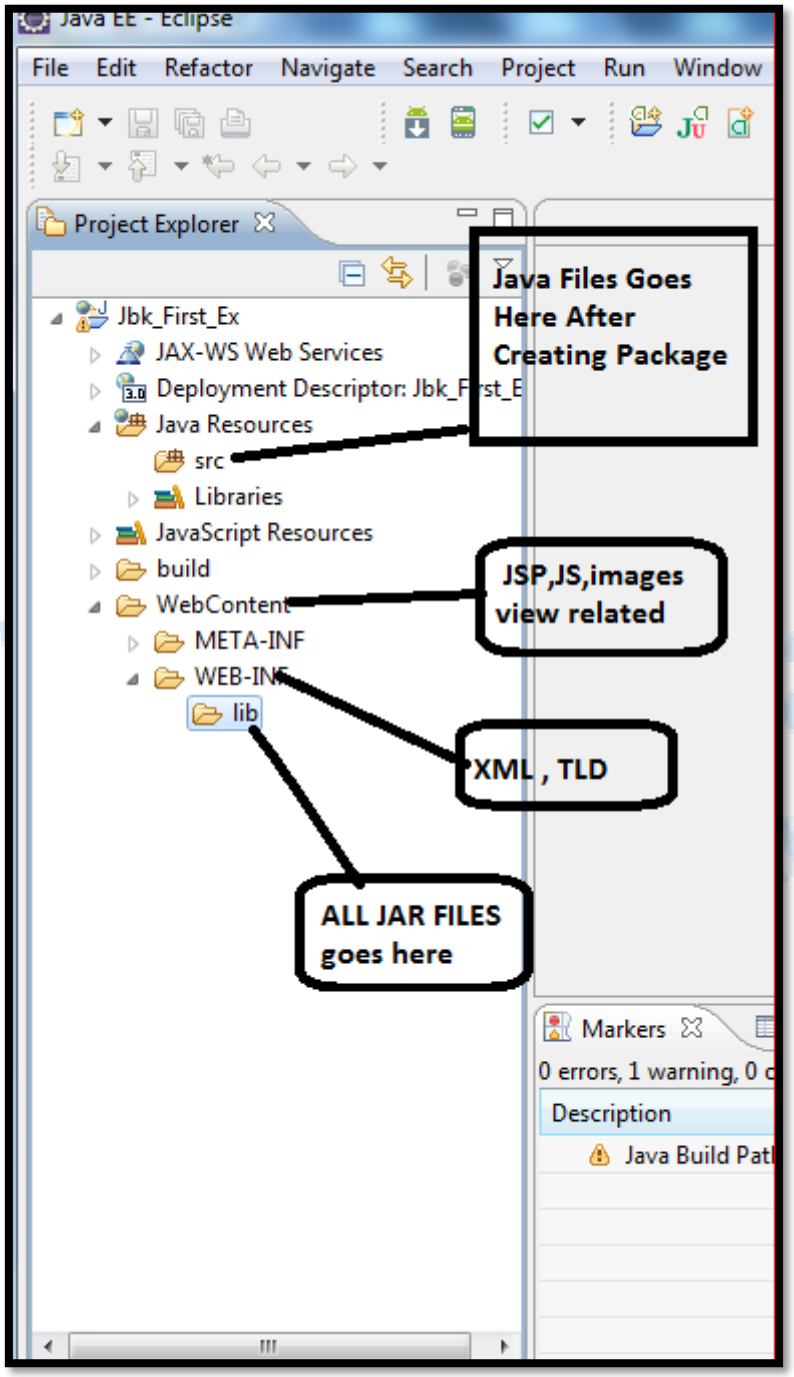
Basic Login Application in ECLIPSE

In eclipse create new project as below. In explorer right click and select new project.

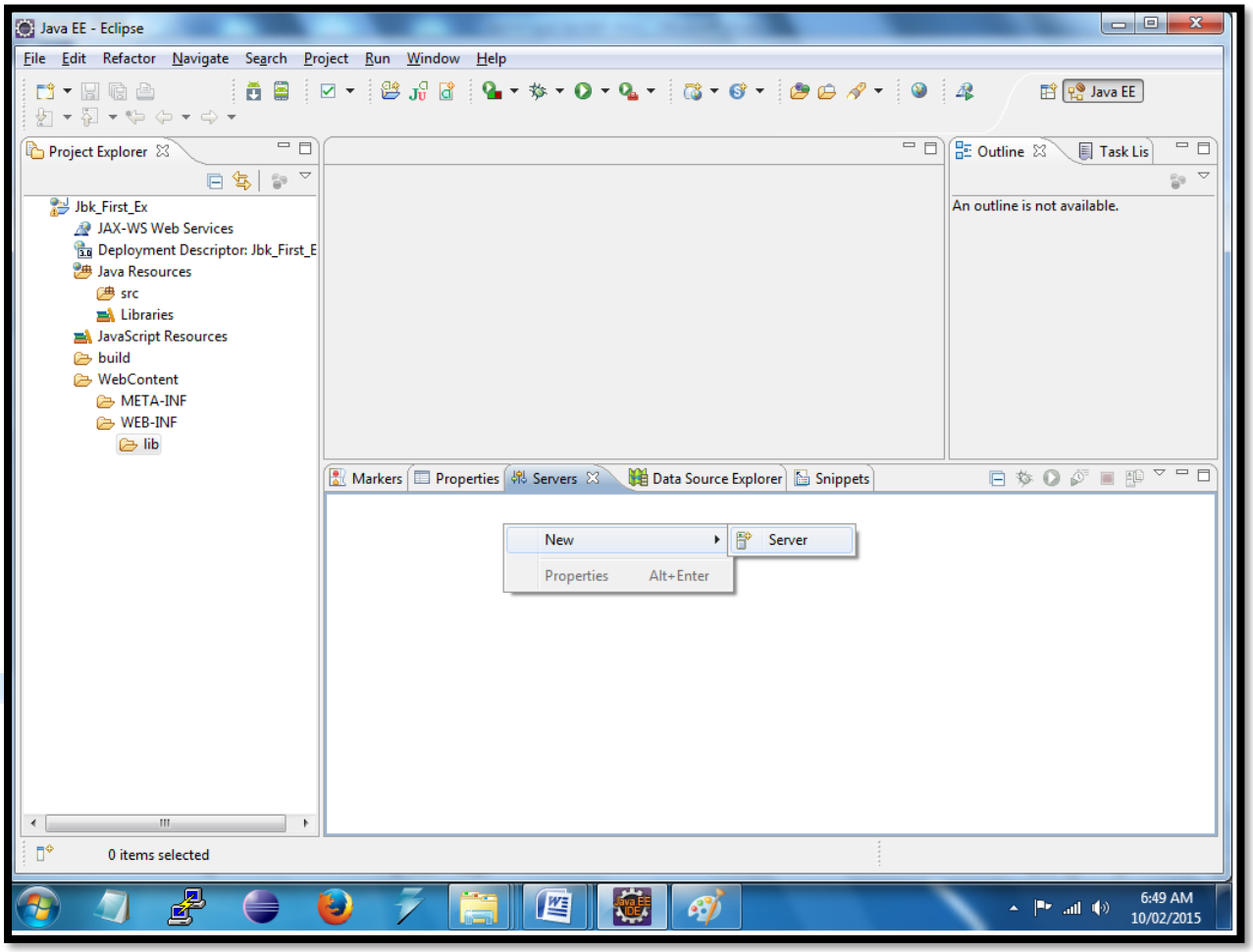


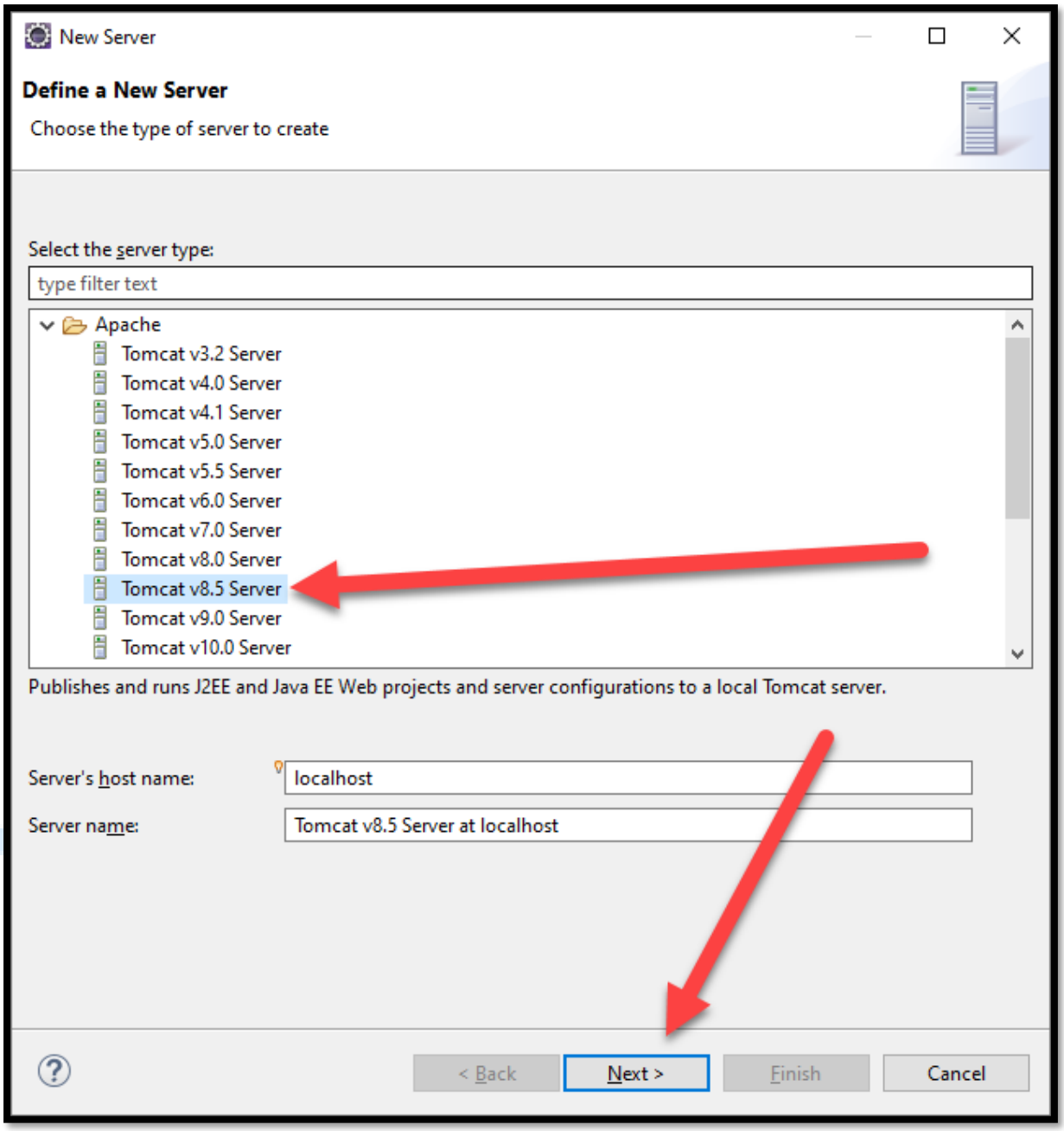
After creation of project in eclipse we can see below in explorer window

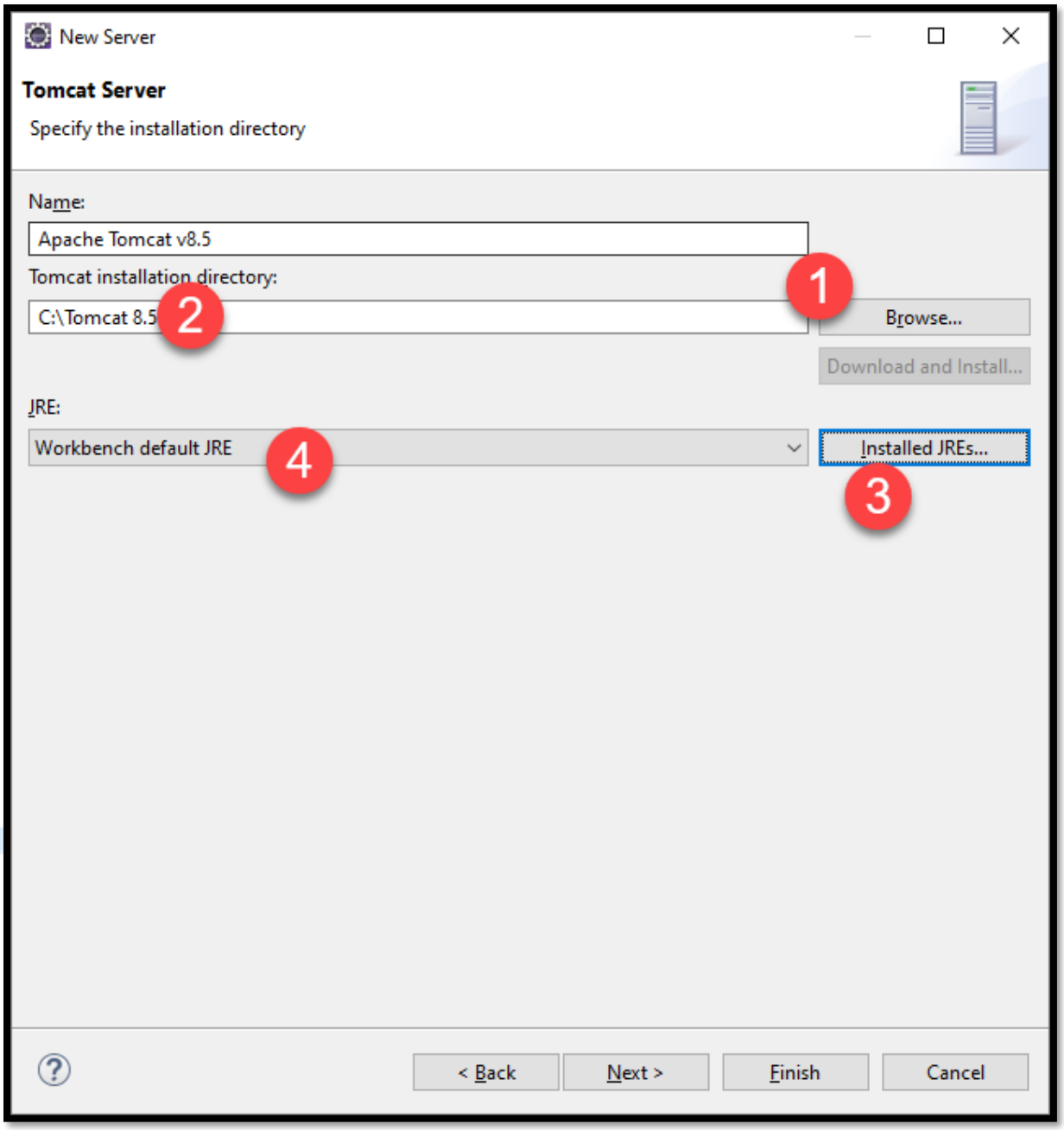
What kind of files needs to created where is been explained in below diagram.



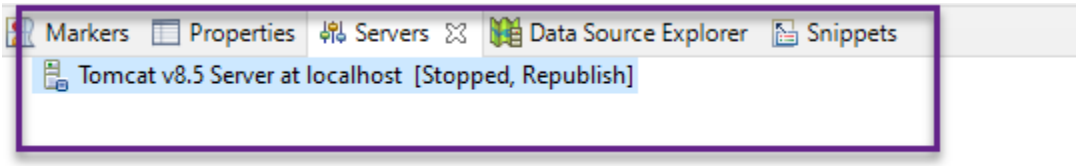
Configure Tomcat in Eclipse



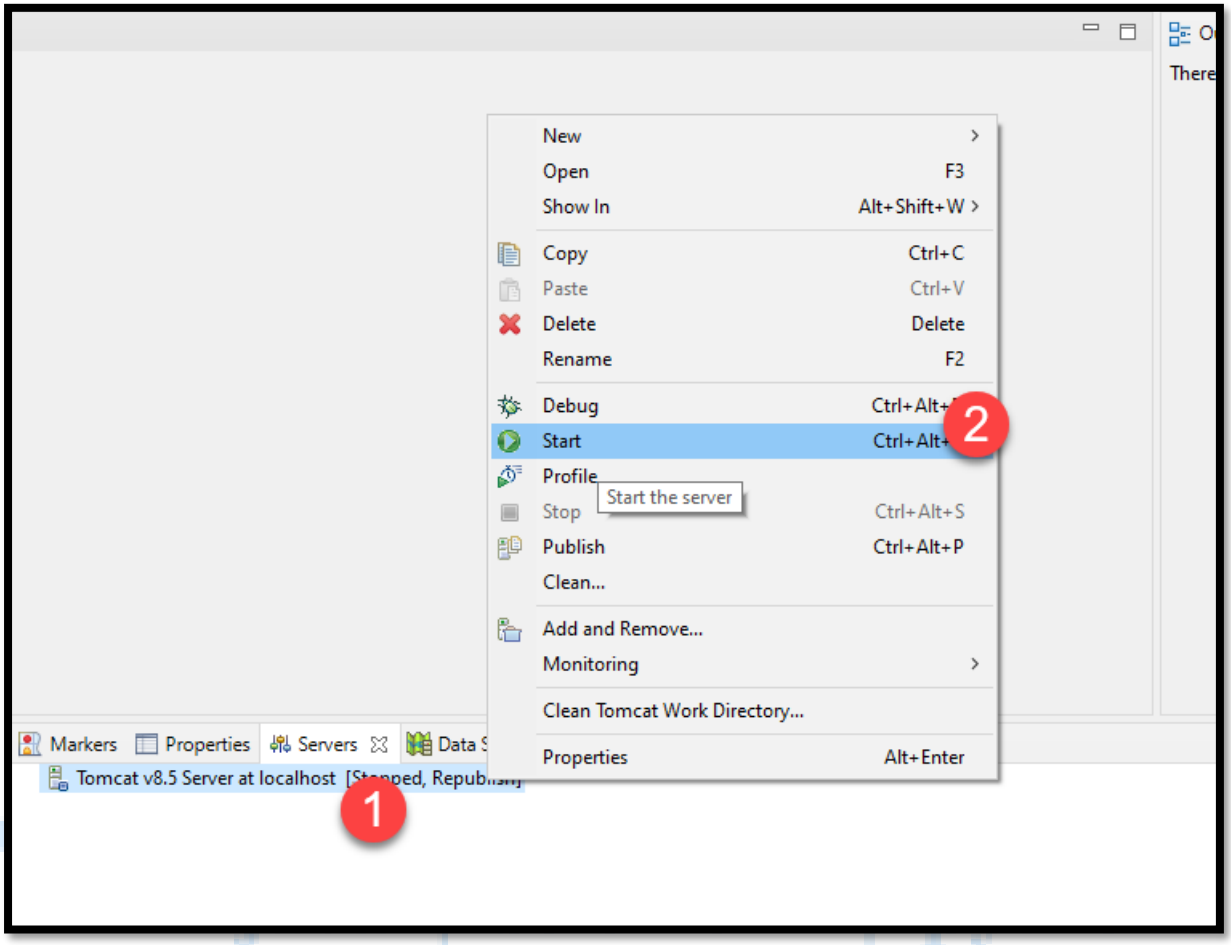

New Server
Define a New Server
Choose the type of server to create
Select the server type:
type filter text
Apache
Tomcat v3.2 Server
Tomcat v4.0 Server
Tomcat v4.1 Server
Tomcat v5.0 Server
Tomcat v5.5 Server
Tomcat v6.0 Server
Tomcat v7.0 Server
Tomcat v8.0 Server
Tomcat v8.5 Server
Tomcat v9.0 Server
Tomcat v10.0 Server
Publishes and runs J2EE and Java EE Web projects and server configurations to a local Tomcat server.
Server's host name:
localhost
Server name:
Tomcat v8.5 Server at localhost
< Back
Next >
Finish
Cancel



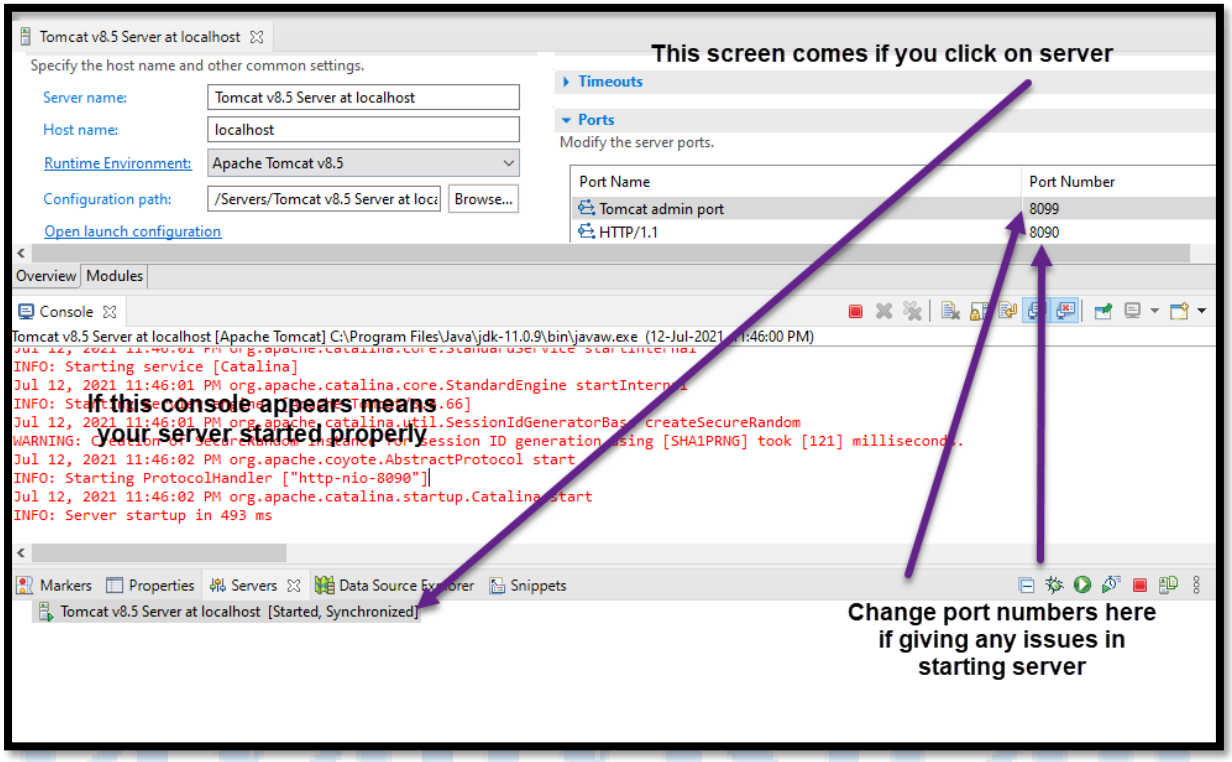
After clicking finish



Right click on server



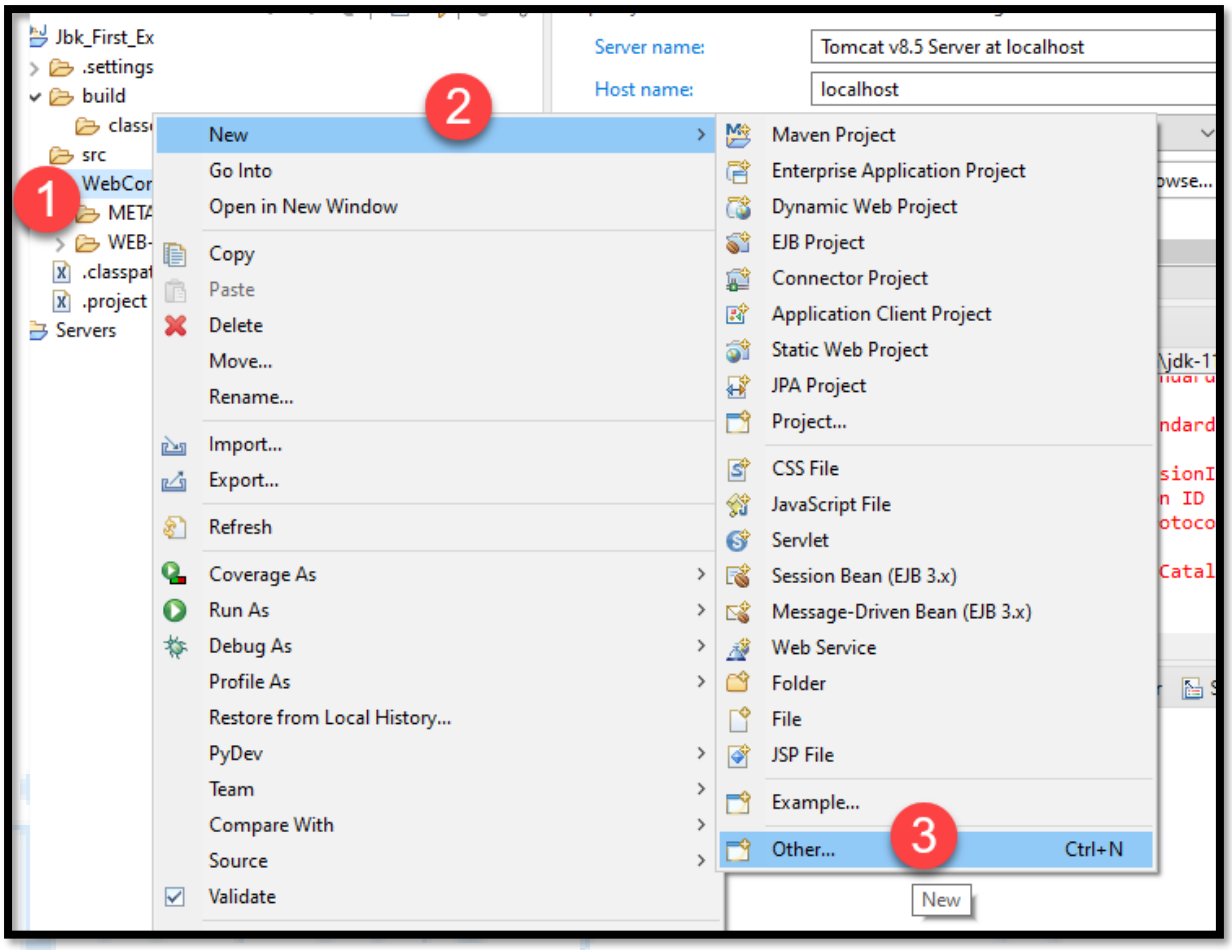
After clicking start observing below – no action needed if no error comes in

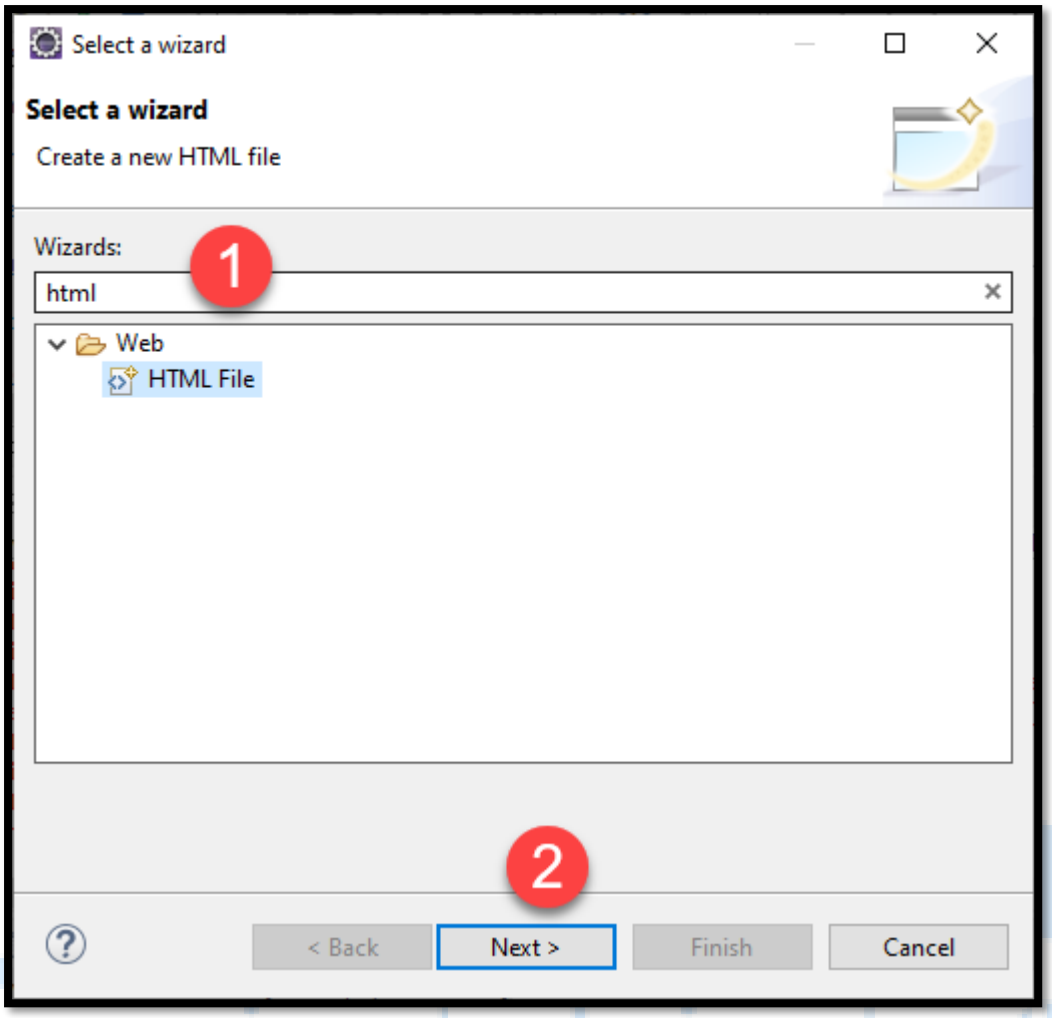


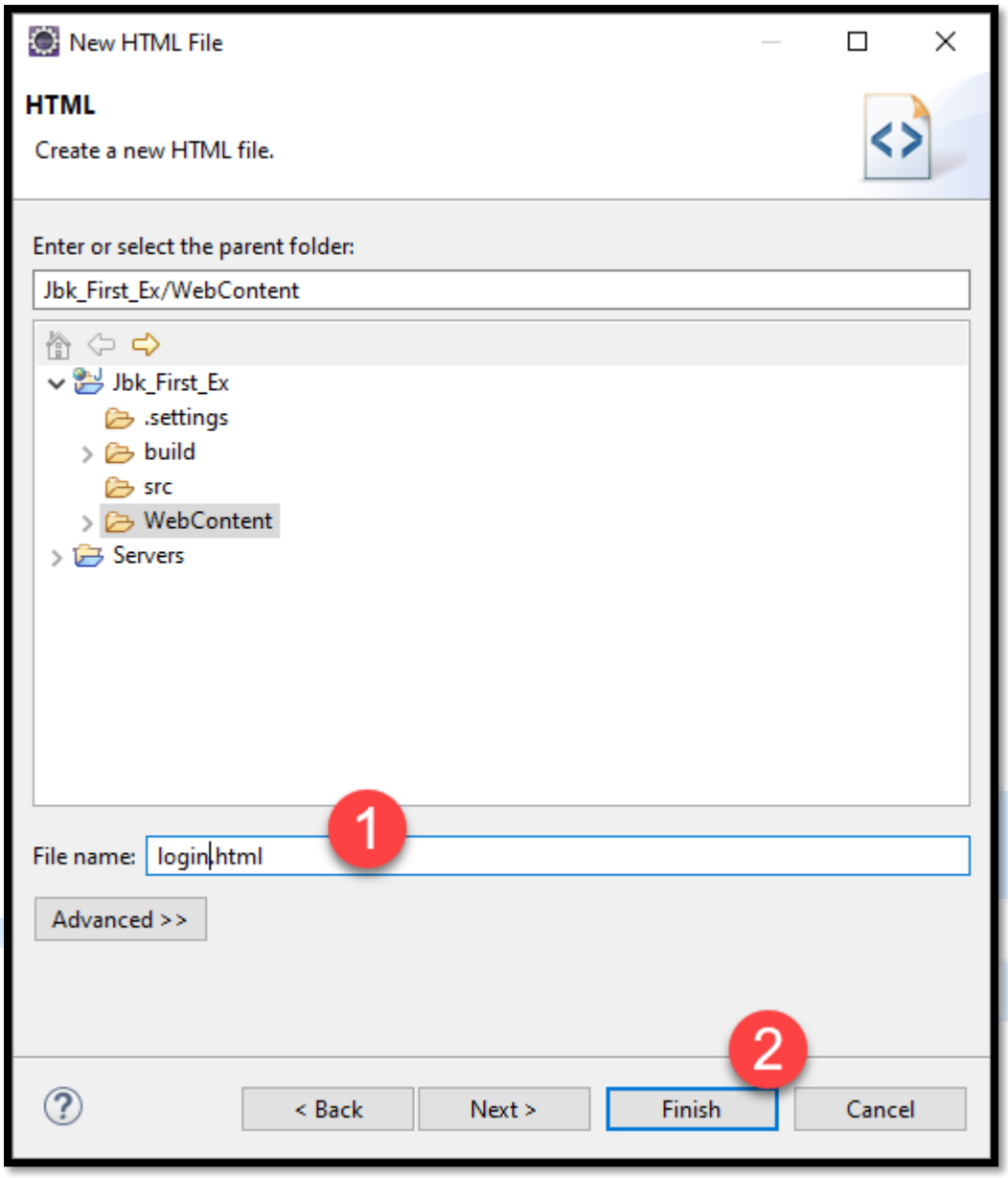
Files required for running the First Application.

login.html

LoginServlet.java



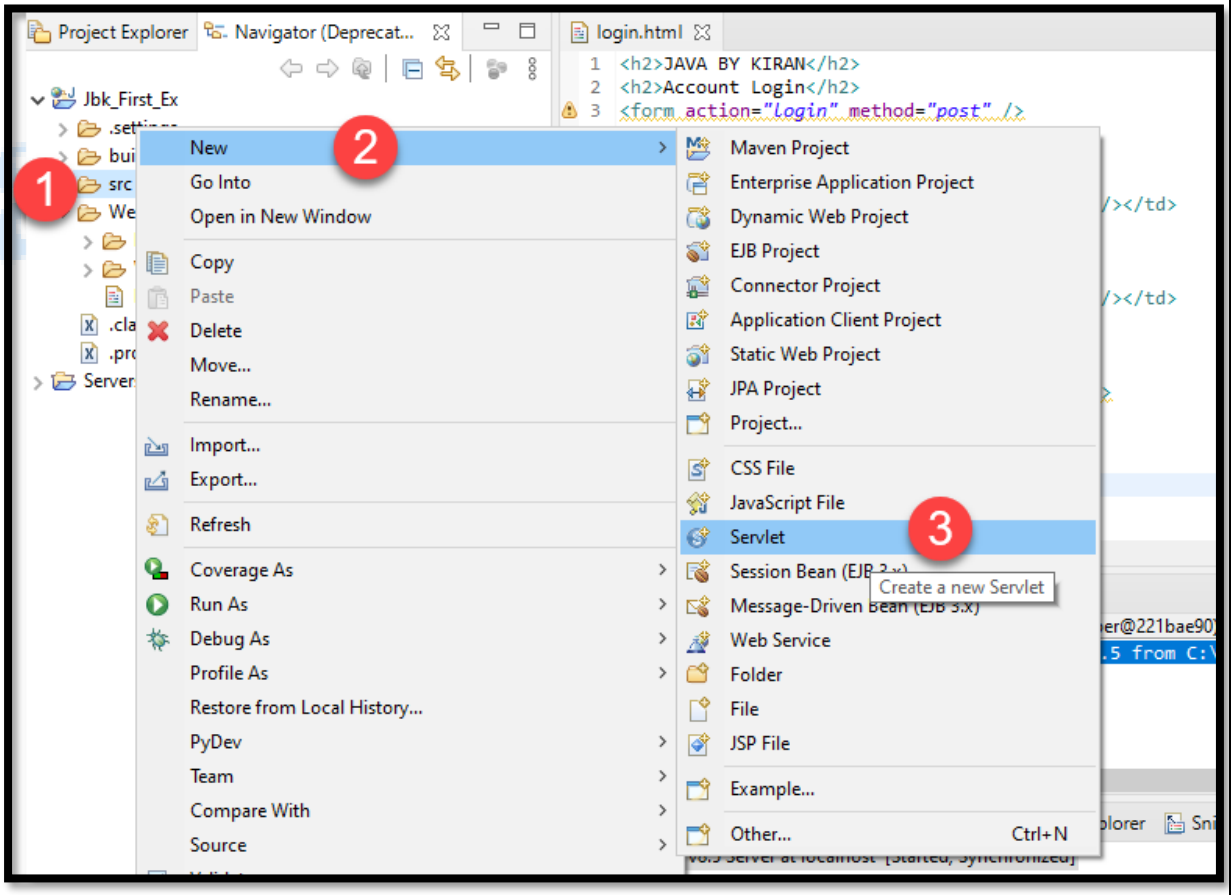


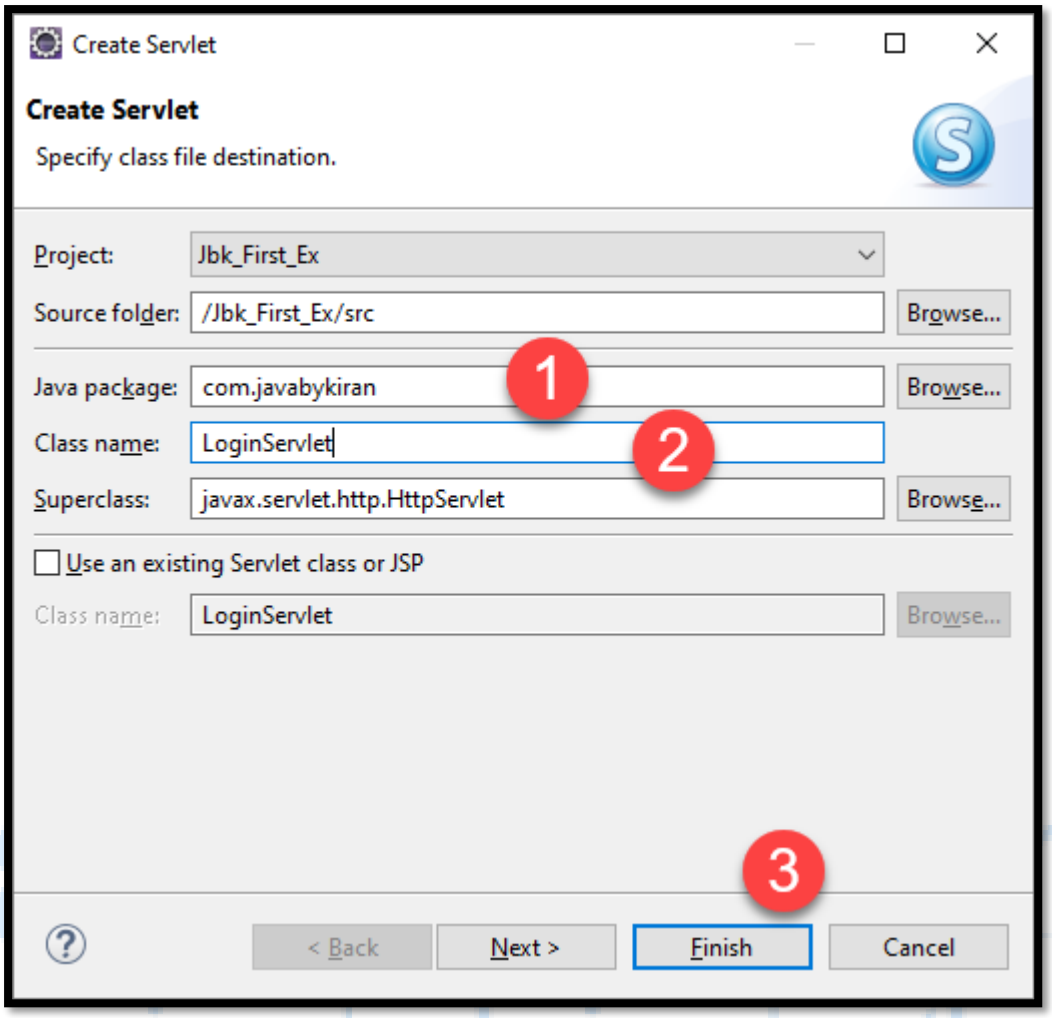


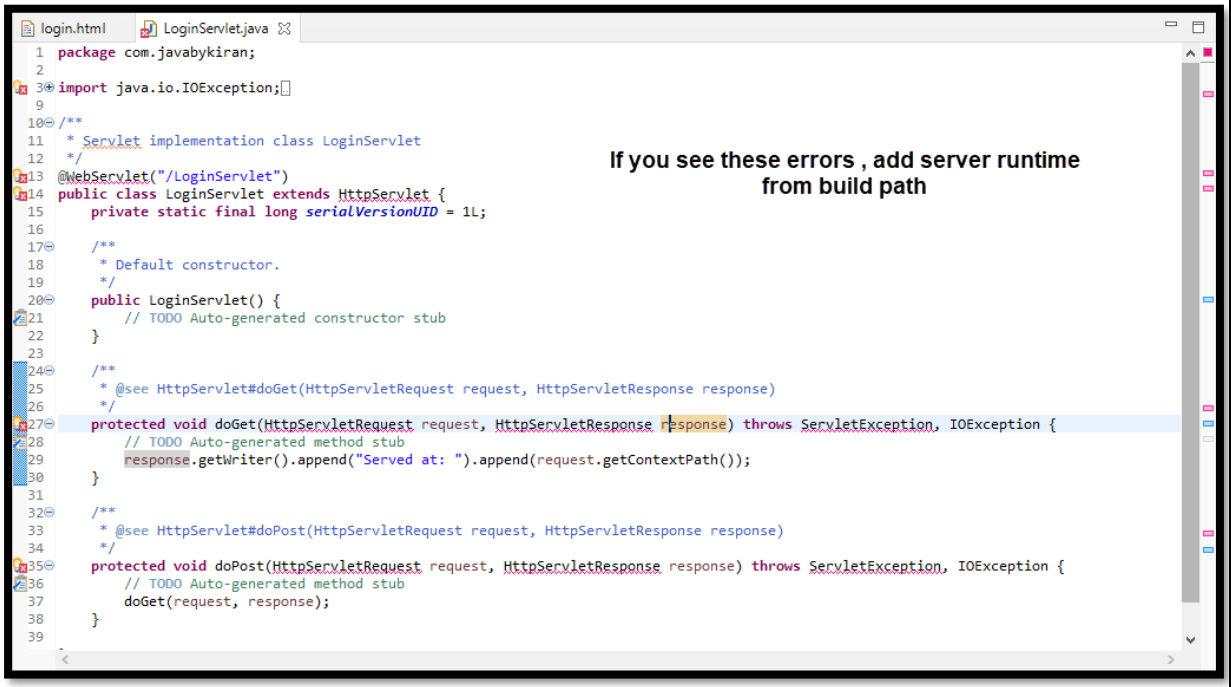
1. **login .html:-**

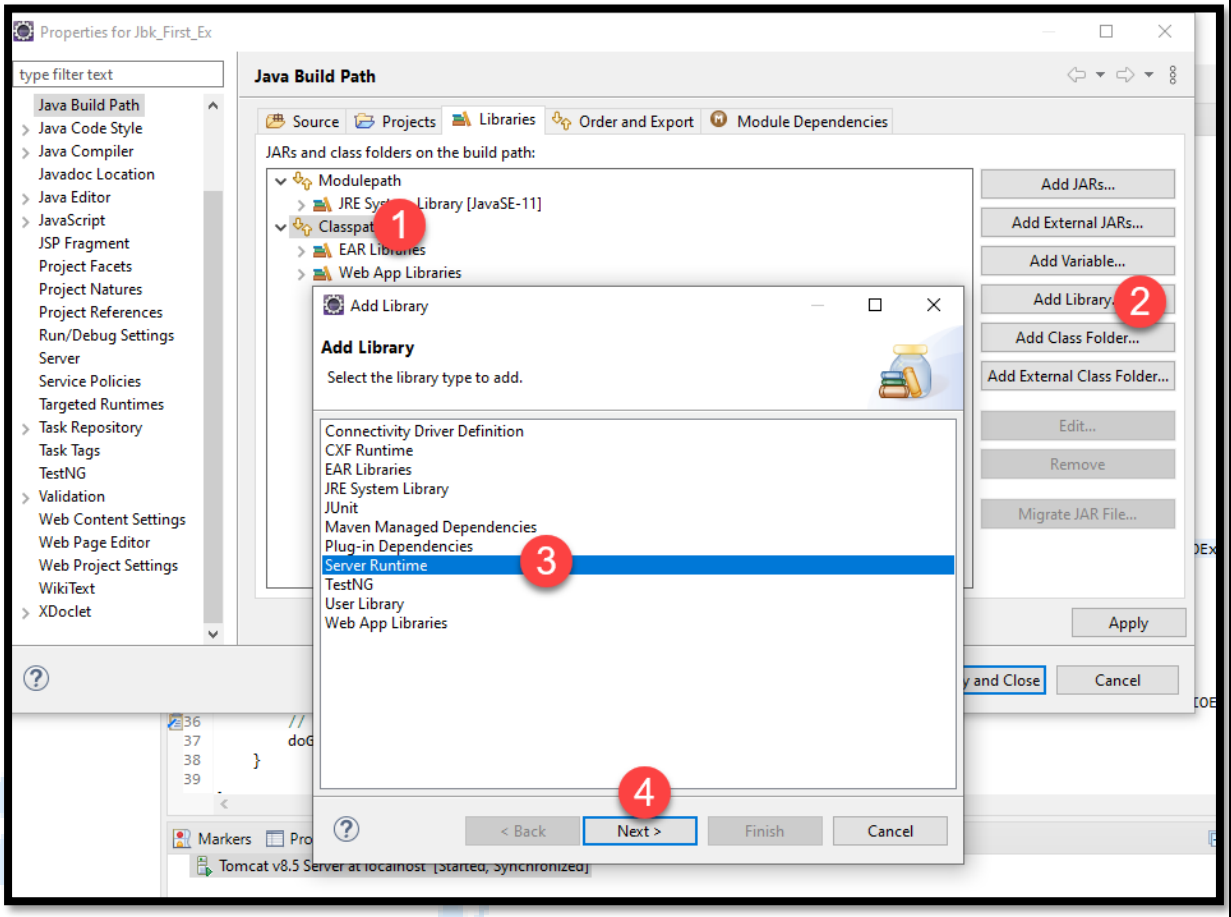
```
<h2>JAVA BY KIRAN</h2>

<h2>Account Login </h2>

<form action ="login" method="post"/>

<table>

<tr>

<td>Username:</td><td>

<input type ="text" name="uname"/></td>
```

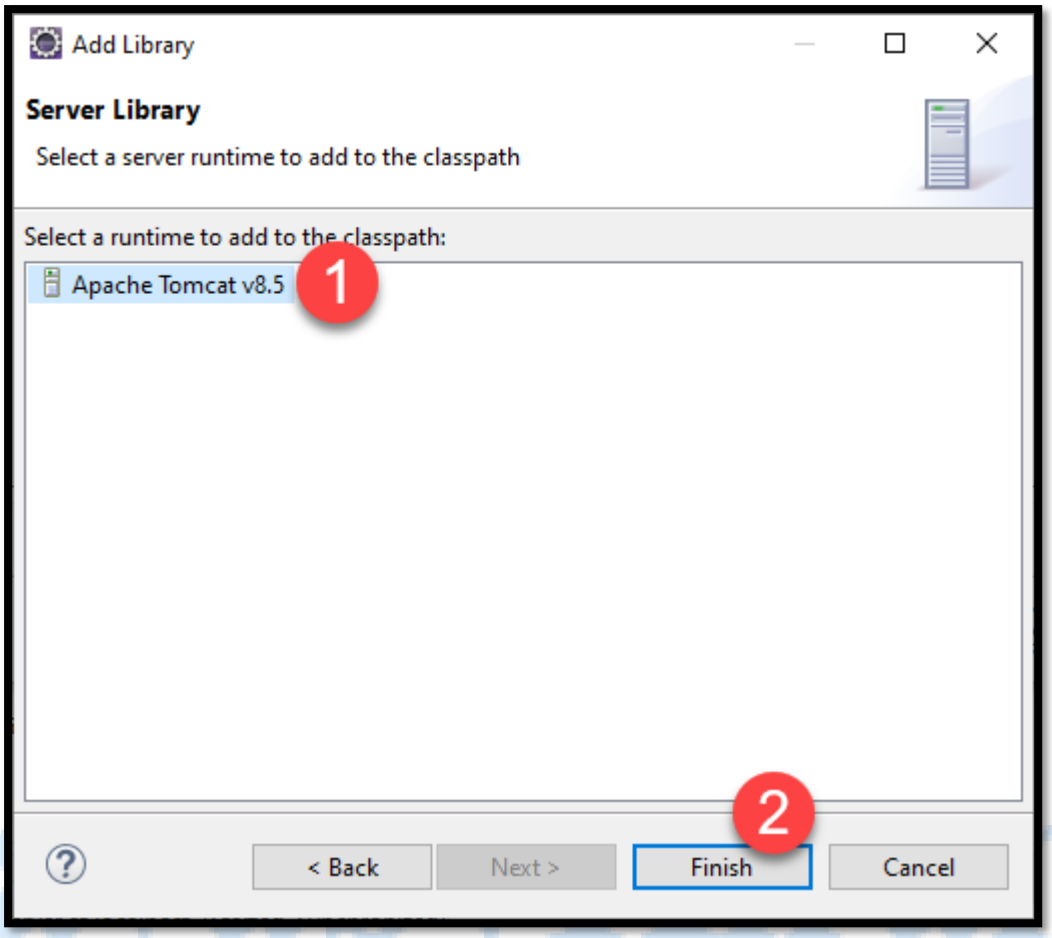
```
</tr>

<tr>

<td>Password:</td>

<td><input type ="text" name="pword"/></td>

</tr>


<tr><input type="submit" value="Login"/></tr>

</table>

</form>
```

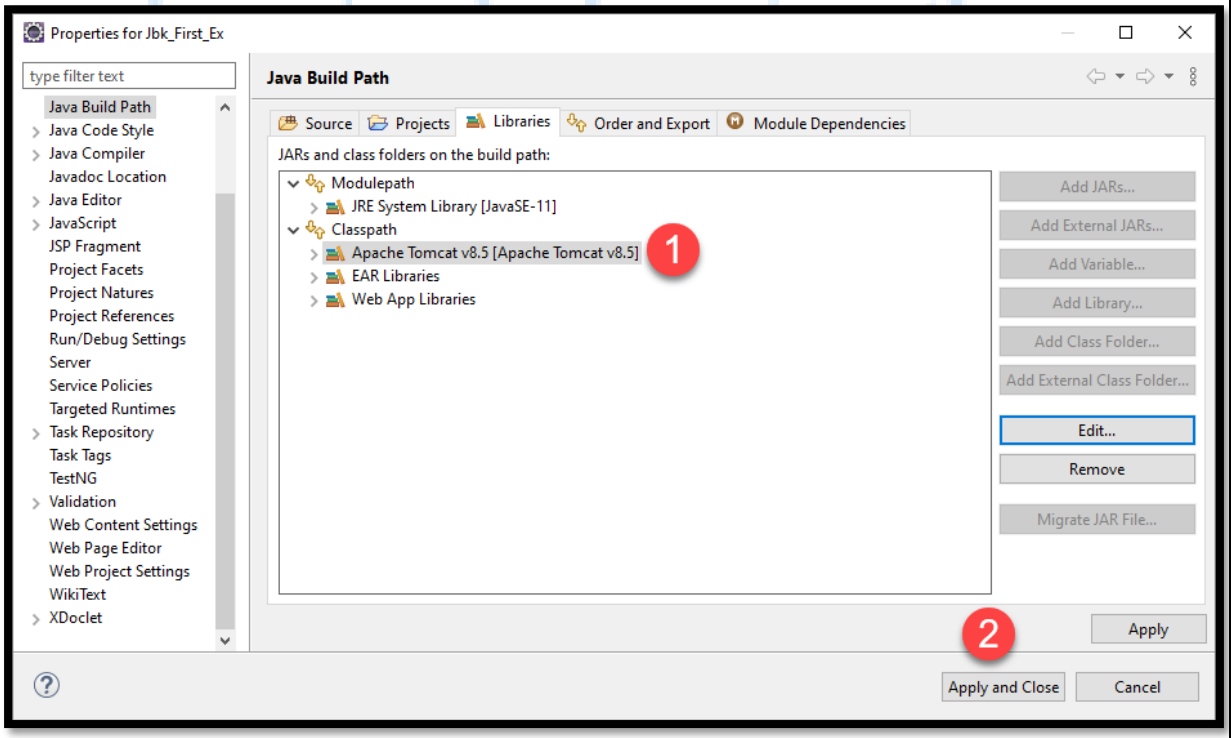


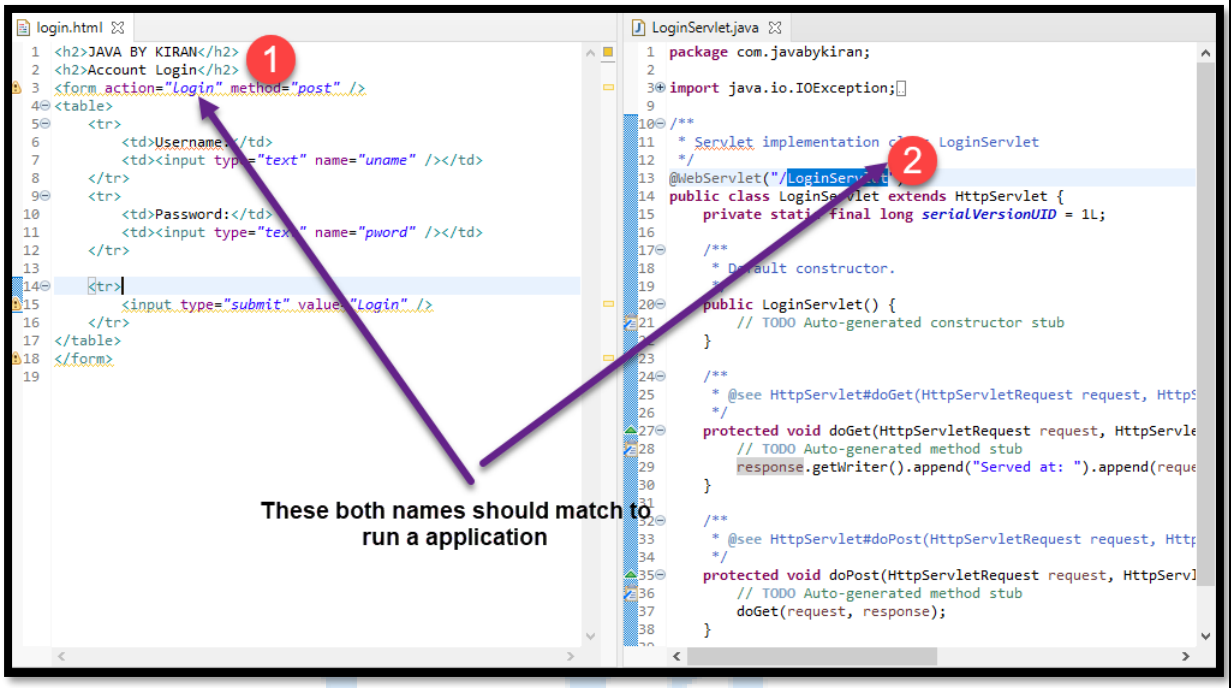

Create Servlet
Create Servlet
Specify class file destination.
Project:
Jbk_First_Ex
Source folder:
/Jbk_First_Ex/src
Browse...
Java package:
com.javabykiran
Class name:
LoginServlet
Superclass:
javax.servlet.http.HttpServlet
Use an existing Servlet class or JSP
Class name:
LoginServlet
1
2
3
< Back
Next >
Finish
Cancel


login.html
LoginServlet.java
If you see these errors , add server runtime from build path
package com.javabykiran;
import java.io.IOException;
/**
* Servlet implementation class LoginServlet
*/
@WebServlet("/LoginServlet")
public class LoginServlet extends HttpServlet {
private static final long serialVersionUID = 1L;
/**
* Default constructor.
*/
public LoginServlet() {
// TODO Auto-generated constructor stub
}
/**
* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
*/
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
// TODO Auto-generated method stub
response.getWriter().append("Served at: ").append(request.getContextPath());
}
/**
* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
*/
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
// TODO Auto-generated method stub
doGet(request, response);
}




Add Library
Server Library
Select a server runtime to add to the classpath
Select a runtime to add to the classpath:
Apache Tomcat v8.5
1
2
< Back
Next >
Finish
Cancel


Properties for Jbk_First_Ex
type filter text
Java Build Path
Java Code Style
Java Compiler
Javadoc Location
Java Editor
JavaScript
JSP Fragment
Project Facets
Project Natures
Project References
Run/Debug Settings
Server
Service Policies
Targeted Runtimes
Task Repository
Task Tags
TestNG
Validation
Web Content Settings
Web Page Editor
Web Project Settings
WikiText
XDoclet
Java Build Path
Source
Projects
Libraries
Order and Export
Module Dependencies
JARs and class folders on the build path:
Modulepath
JRE System Library [JavaSE-11]
Classpath
Apache Tomcat v8.5 [Apache Tomcat v8.5]
EAR Libraries
Web App Libraries
1
Add JARs...
Add External JARs...
Add Variable...
Add Library...
Add Class Folder...
Add External Class Folder...
Edit...
Remove
Migrate JAR File...
2
Apply
Apply and Close
Cancel

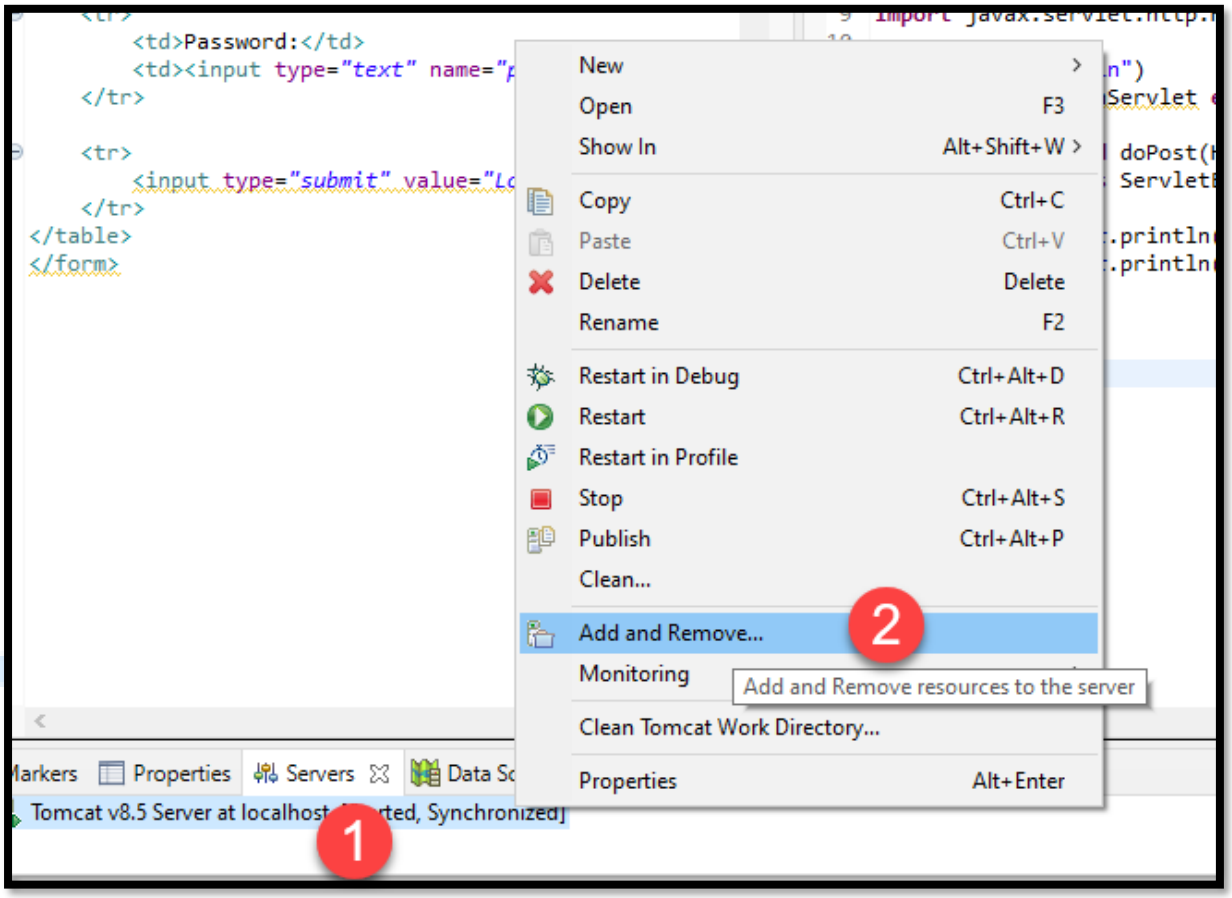


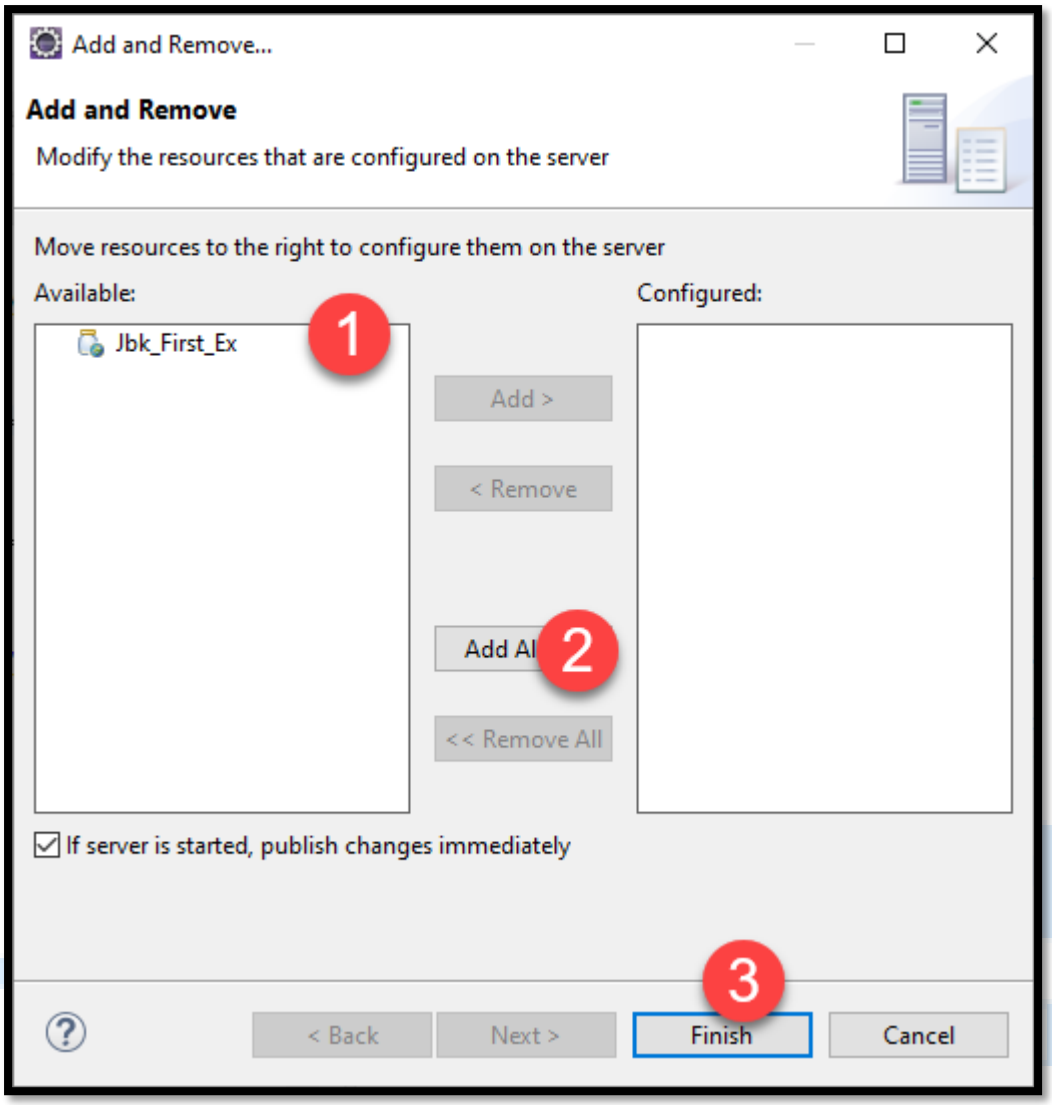
So we will change here servlet @WebServlet annotation value {2} as login as per html page.

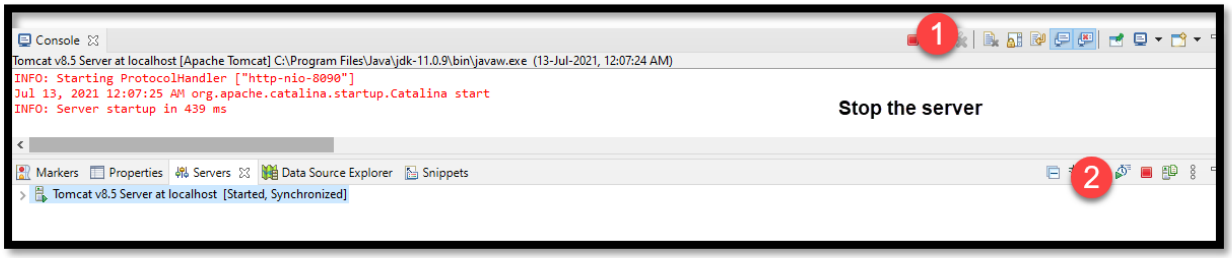
2. LoginSevlet.java

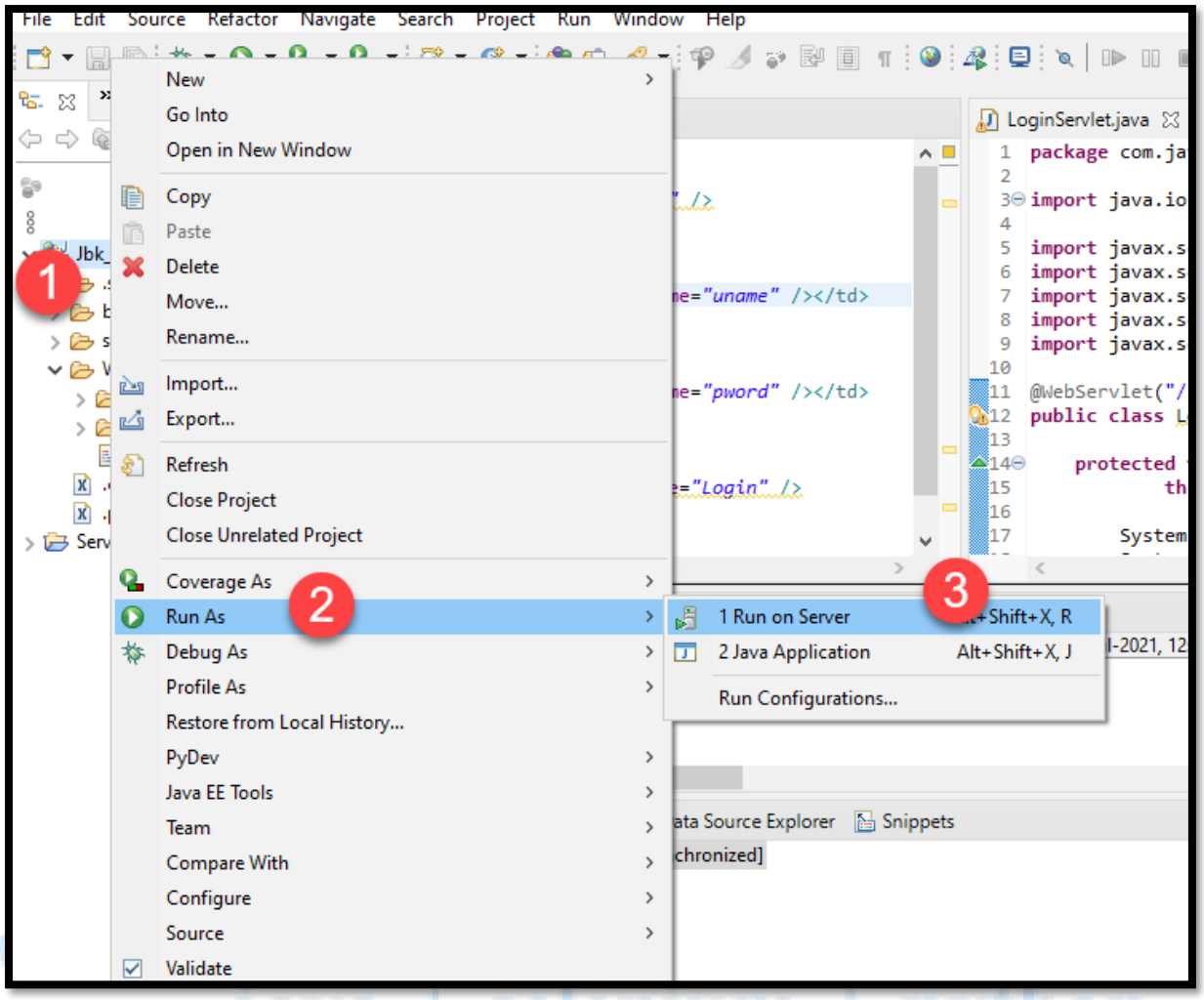
```
package com.javabykiran;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/login")
public class LoginServlet extends HttpServlet {

   protected void doPost(HttpServletRequest request, HttpServletResponse response)
               throws ServletException, IOException {

         System.out.println("I am in do post");
         System.out.println("connectivity is made between html and servlet");
   }

}
```

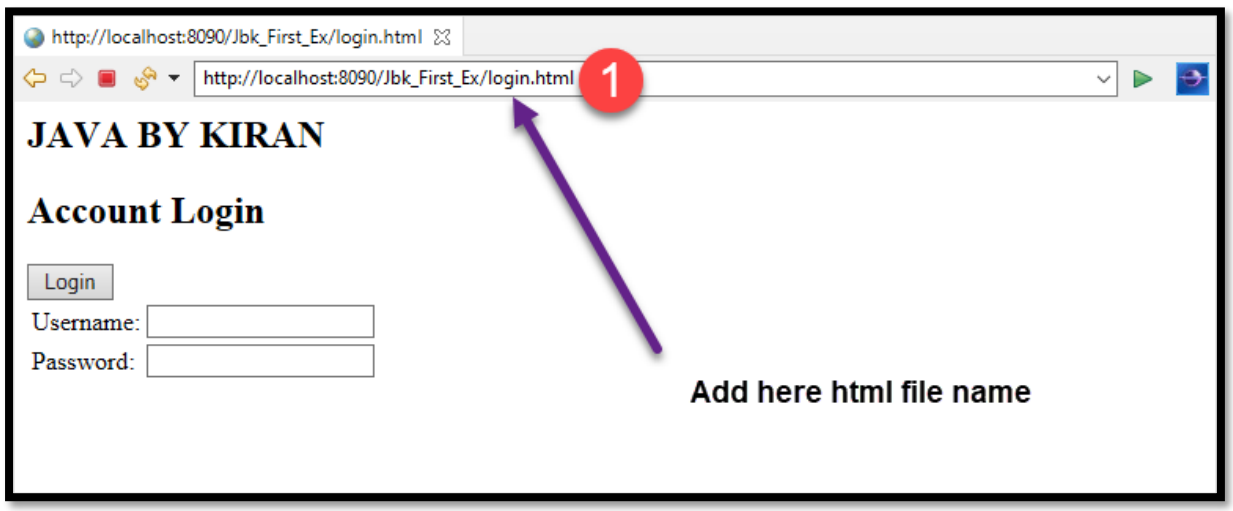
Now run the project



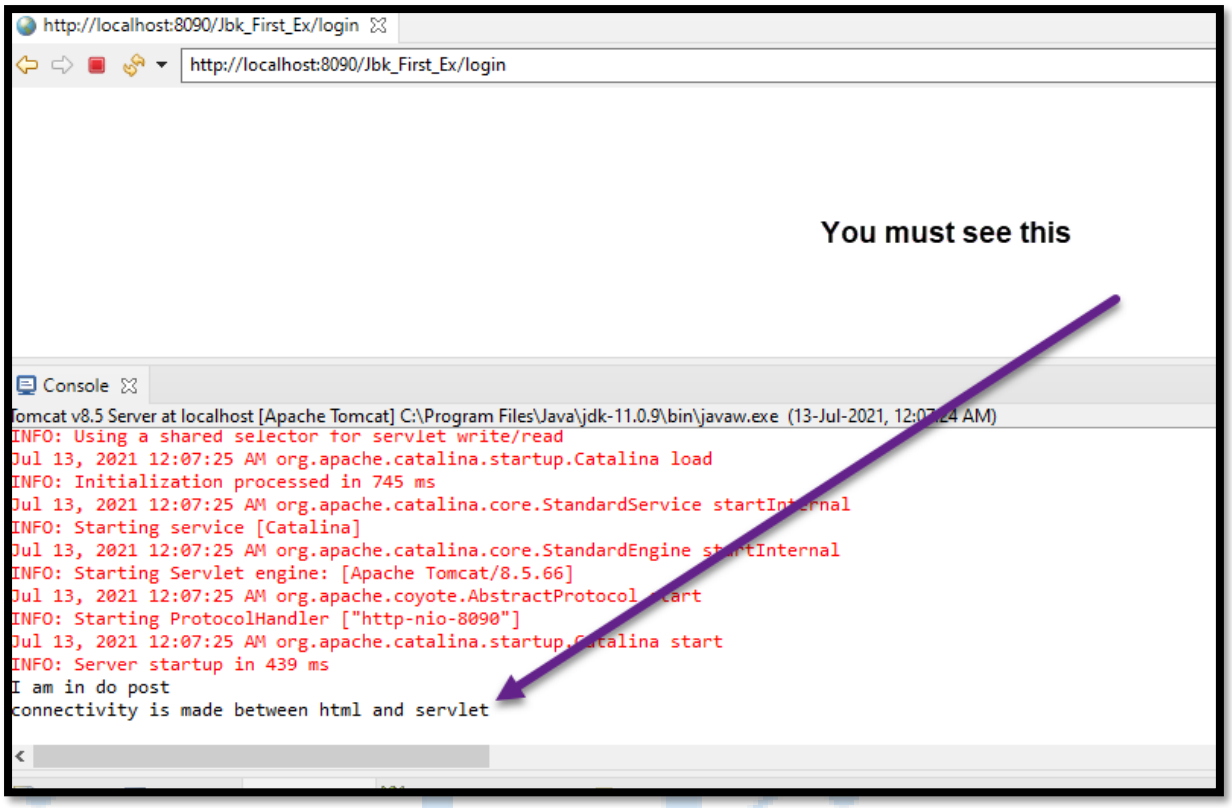

Add and Remove...
Add and Remove
Modify the resources that are configured on the server
Move resources to the right to configure them on the server
Available:
Configured:
Jbk_First_Ex
1
Add >
< Remove
Add Al
2
<< Remove All
If server is started, publish changes immediately
3
< Back
Next >
Finish
Cancel


Console
Tomcat v8.5 Server at localhost [Apache Tomcat] C:\Program Files\Java\jdk-11.0.9\bin\javaw.exe (13-Jul-2021, 12:07:24 AM)
INFO: Starting ProtocolHandler ["http-nio-8090"]
Jul 13, 2021 12:07:25 AM org.apache.catalina.startup.Catalina start
INFO: Server startup in 439 ms
1
Stop the server
Markers
Properties
Servers
Data Source Explorer
Snippets
Tomcat v8.5 Server at localhost [Started, Synchronized]
2





After adding uname and password click on submit



This is complete deployment of simple first project.

Types of Applications: -

- ✓ Standalone application
- ✓ Client server application
- ✓ Web application
- ✓ Distributed application
- ✓ Enterprise application

- **Standalone application:-**application that can be used by so many users at a time is called as standalone application.
  - Ex :- MS WORD, MUSIC PLAYER, etc.
  - Standalone application can be implemented using c,c++,java,etc

  - Problems: -
    Application has to installed across all the machines .this may give maintenance when you change applications features.

- **Client server application :-** in the case of client server application
  - Application is divided into two parts. One part will be installed on server machine and other part will be installed on multiple client machines.
    i.e., multiple clients can be centralized at server machine.
    Ex. yahoo messenger, google-talk etc.
  - Client server application can be implemented using c , c++ ,java.
  - Data sharing is possible.
  - Problems: client S/W has to install across all the machines. This may give maintenance when you change client S/W features.

- **Web application:-** in this case of web application ,application S/W will installed on web server machines only .i.e. multiple clients can access application using any browser.
  Ex. javabykiran.com , gmail.com
  - Web application can be implemented by using servlets, JSP .
  - Data sharing is possible
  - No maintenance problem because modification will happen only at web server

Terms generally used.

- **Web Server**
  - Web server is an application which receives request from web client and processes request with the help of web container and send response to web client

    Ex. Apache server, PWS, IIS.

- **Web Client**
  - Web client is an application which sends the http request to web server and receives the response from web server.

  Ex IE, OPERA, MOZILLA etc.

- **Web Container**
  - web container is an application responsible for managing the complete lifestyle of Servlet or JSP.

- **HTTP:(Hypertext Transfer Protocol)**
  - Http sites on the Top of TCP |IP for transferring web client Info to web server and web server Info to web Client

- **TCP|IP**
  - Original Data transfer will happen through TCP/IP.
  - IP (Internet Protocol) is responsible for carries data from one place to another.
  - TCP (Transmission Control Protocol) placed on the Top TCP|IP and Monitor the Data Transmission.
- **Domain Naming Servers (DNS)**
  - DNS is registry where Domain names will be bound with IP address.
- **Web Technology**
  - Servlets, JSP
- **Web framework**
  - JSP, Spring MVC, etc.