Advance java practical no.7:

**Steps to Implement and Deploy Servlets in Apache Tomcat:**

1] **Download Apache Tomcat** from the official website:  
<https://tomcat.apache.org/download-90.cgi>

2] **Extract Tomcat** to a location like:

C:\Users\User\apache-tomcat-9.0.102

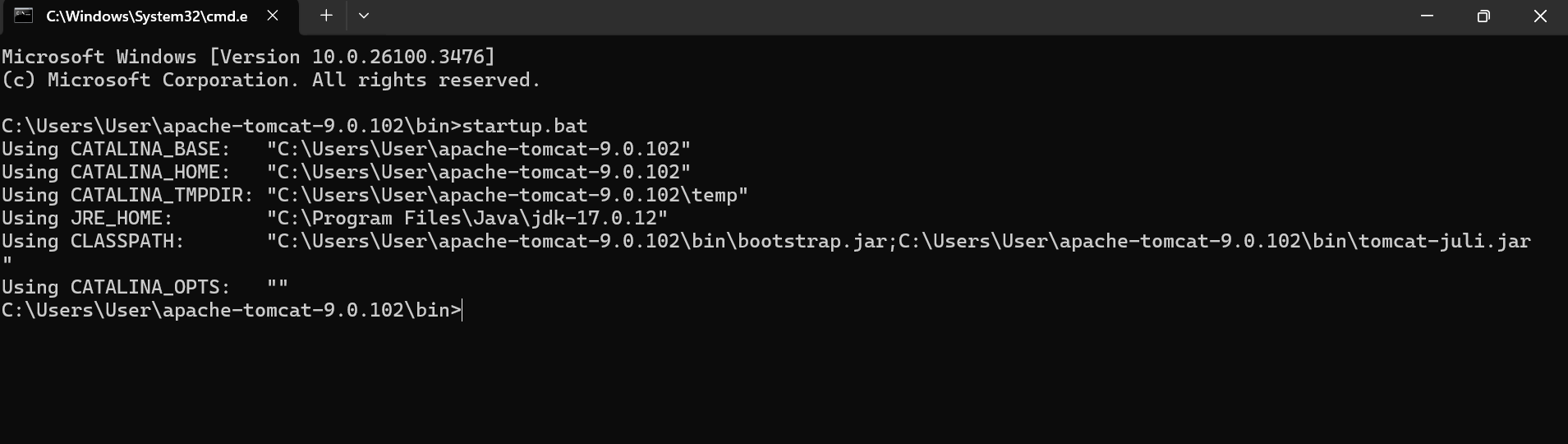
3] **Start Tomcat**

* Open cmd and navigate to the Tomcat bin folder:

cd C:\Users\User\apache-tomcat-9.0.102\bin

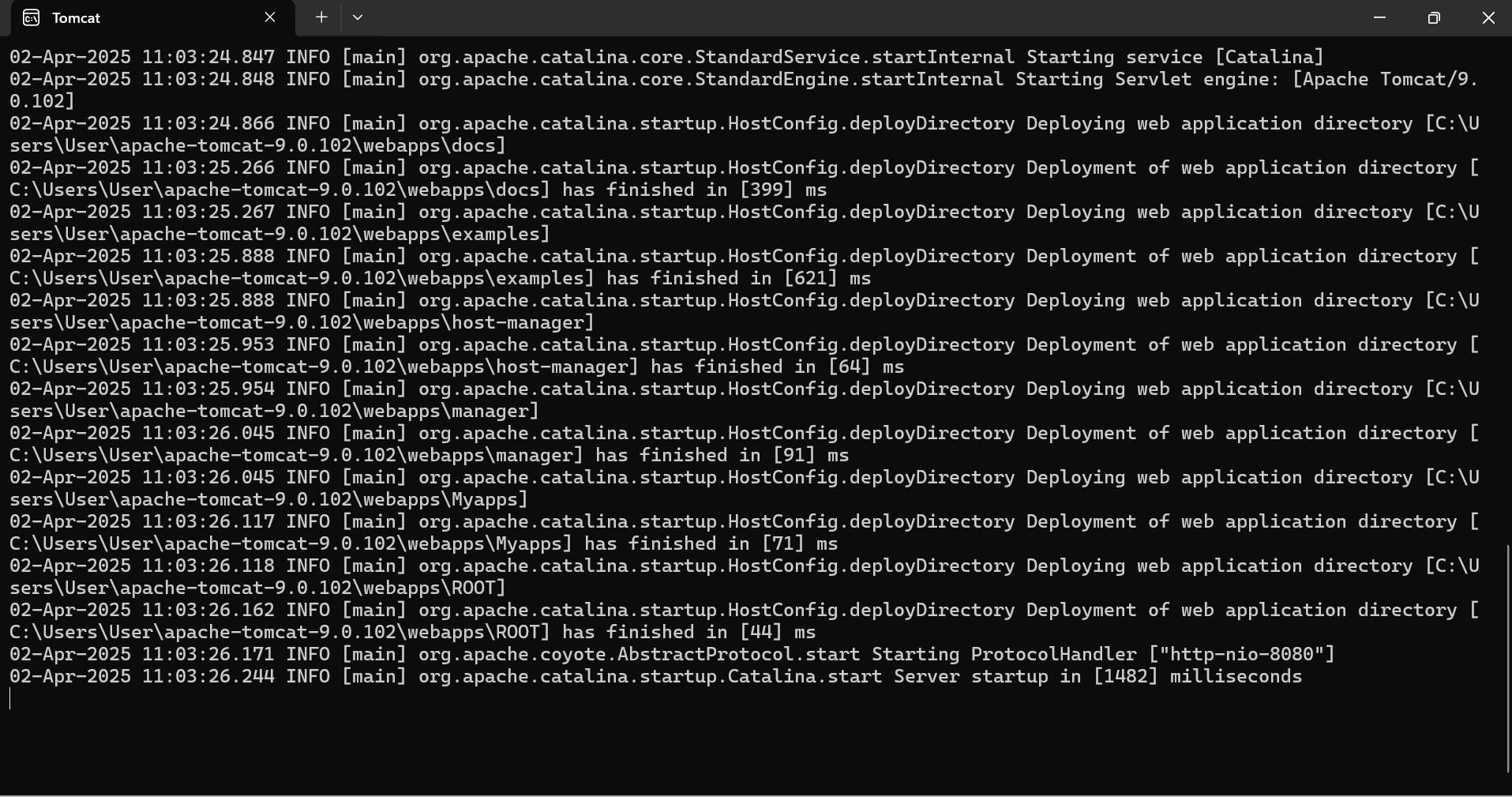
* Run:

startup.bat



Another cmd will get opened named tomcat automatically.

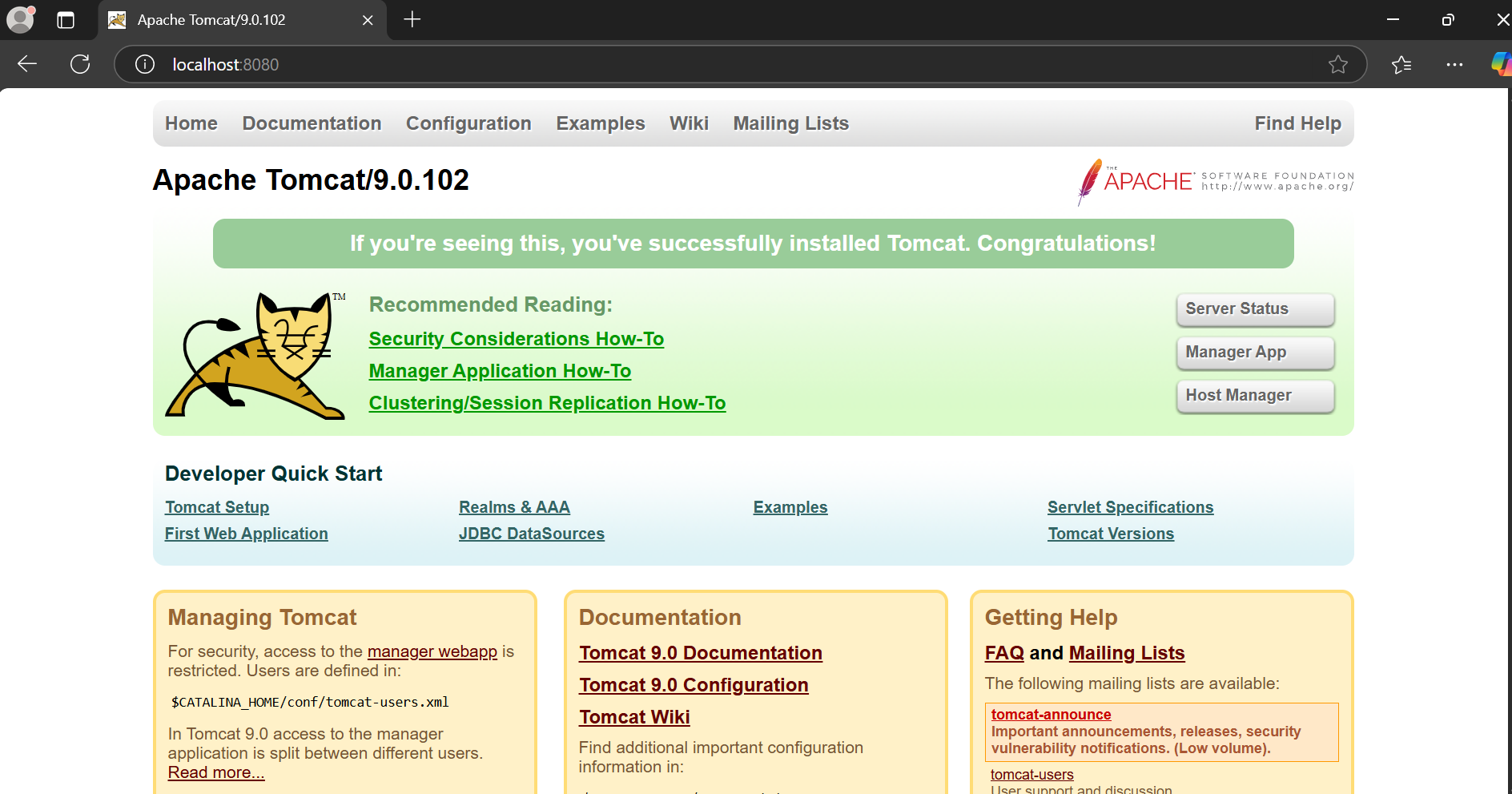
View:



Keep it running in background .to do other work.

If successful, visit:  
<http://localhost:8080> in your browser.

If this page appers means the tomcat is running properly else we have to run the tomcat again.



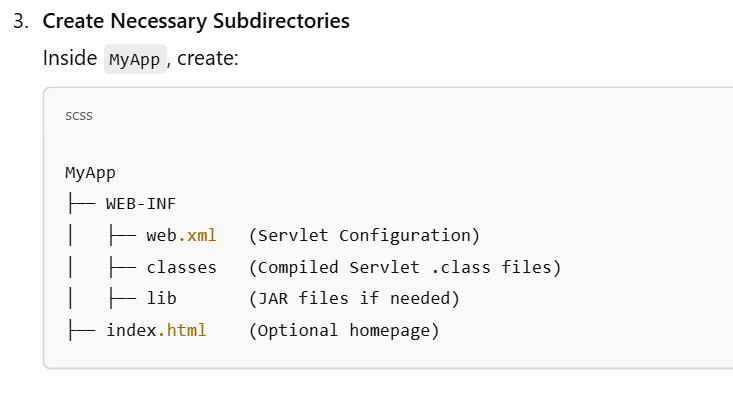
**Create a Web Application Directory in Tomcat:**

1]**Navigate to the Tomcat webapps folder**:

C:\Users\User\apache-tomcat-9.0.102\webapps

2] **Create a New Web Application Folder**

C:\Users\User\apache-tomcat-9.0.102\webapps\MyApp



We have to create this folder structure before beginning the java program part.

**Write the Servlet Code**

Client-Side Servlet (Accepts Username & Password)

**Create a new Java file:** ClientServlet.java

import java.io.IOException;

import java.io.PrintWriter;

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("/ClientServlet")

public class ClientServlet extends HttpServlet {

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

response.setContentType("text/html");

PrintWriter out = response.getWriter();

out.println("<html><body>");

out.println("<h2>Login Page</h2>");

out.println("<form action='ServerServlet' method='post'>");

out.println("Username: <input type='text' name='username'><br>");

out.println("Password: <input type='password' name='password'><br>");

out.println("<input type='submit' value='Login'>");

out.println("</form>");

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

out.close();

}

}

**Server-Side Servlet (Handles Login Data)**

Create another Java file:  
 **ServerServlet.java**

import java.io.IOException;

import java.io.PrintWriter;

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("/ServerServlet")

public class ServerServlet extends HttpServlet {

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

response.setContentType("text/html");

PrintWriter out = response.getWriter();

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

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

out.println("<html><body>");

out.println("<h2>Login Details</h2>");

out.println("<p>Username: " + username + "</p>");

out.println("<p>Password: " + password + "</p>");

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

out.close();

}

}

**Configure web.xml (Servlet Deployment Descriptor)**

Create the web.xml file inside WEB-INF:

<web-app>

<servlet>

<servlet-name>ClientServlet</servlet-name>

<servlet-class>ClientServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ClientServlet</servlet-name>

<url-pattern>/ClientServlet</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>ServerServlet</servlet-name>

<servlet-class>ServerServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ServerServlet</servlet-name>

<url-pattern>/ServerServlet</url-pattern>

</servlet-mapping>

</web-app>

We have to complie the both the java program in the folder classes which we have already created while creating the folder structure .by the the we will get the .class files to them in the classes folder.

**Compile Java Files**

* Navigate to MyApp\WEB-INF\classes\:

cd C:\Users\User\apache-tomcat-9.0.102\webapps\MyApp\WEB-INF\classes

* Compile servlets:

javac -cp "C:\Users\User\apache-tomcat-9.0.102\lib\servlet-api.jar" ClientServlet.java ServerServlet.java

* This will generate .class files inside classes

**Restart Tomcat**

* Stop Tomcat:

shutdown.bat

* Start Tomcat again:

startup.bat

**Run the Web Application**

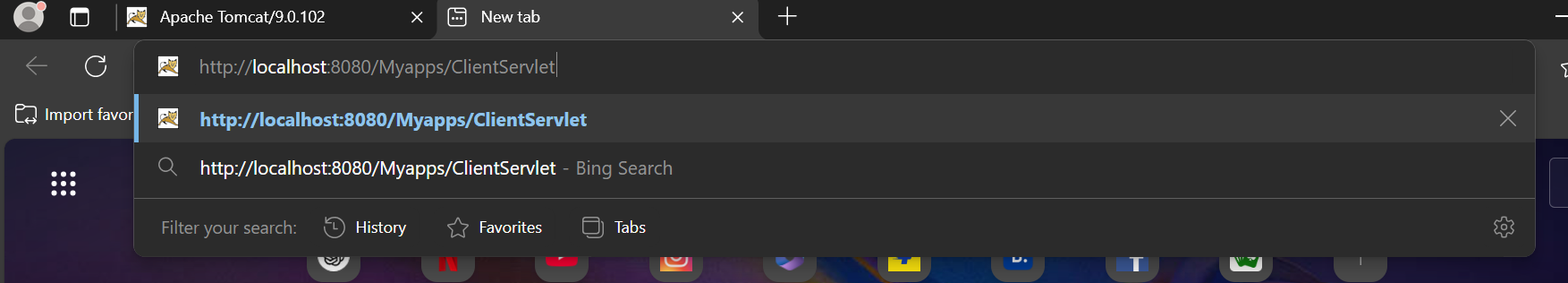
1. **Open a Browser** and go to:

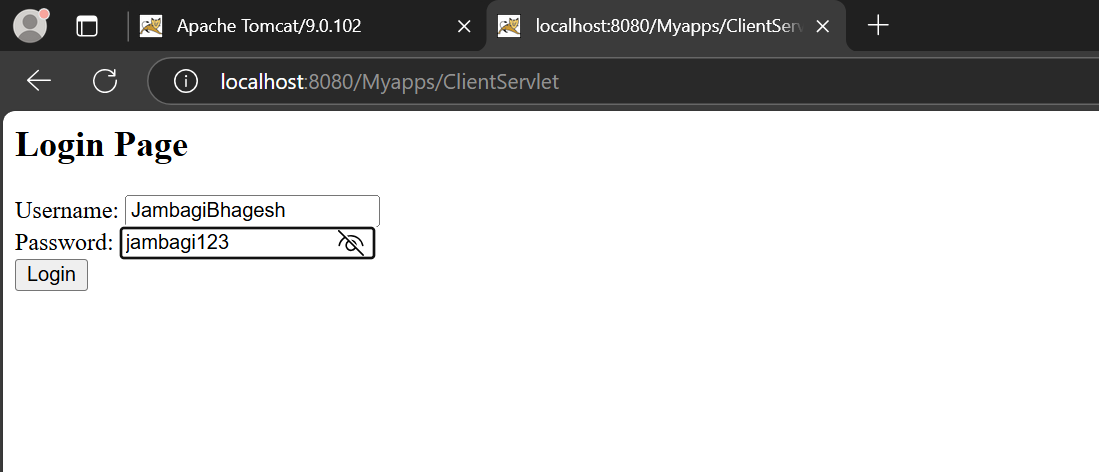
http://localhost:8080/MyApp/ClientServlet

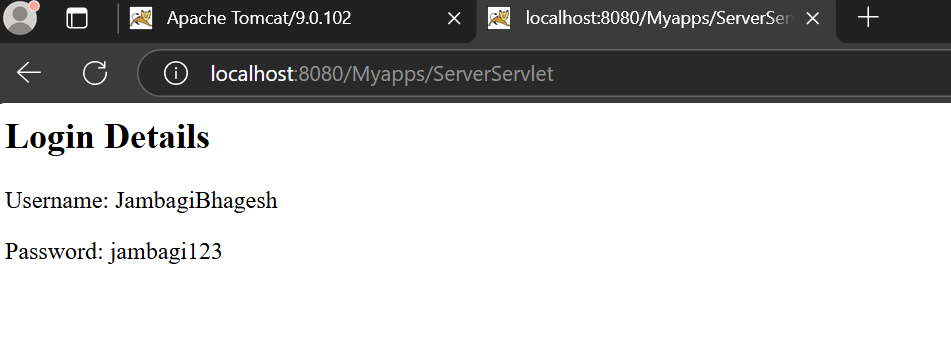
1. **Enter a Username and Password** → Click **Login**.
2. The **ServerServlet** will process the input and display it.

Example :

“http://localhost:8080/Myapps/ClientServlet “on browser.

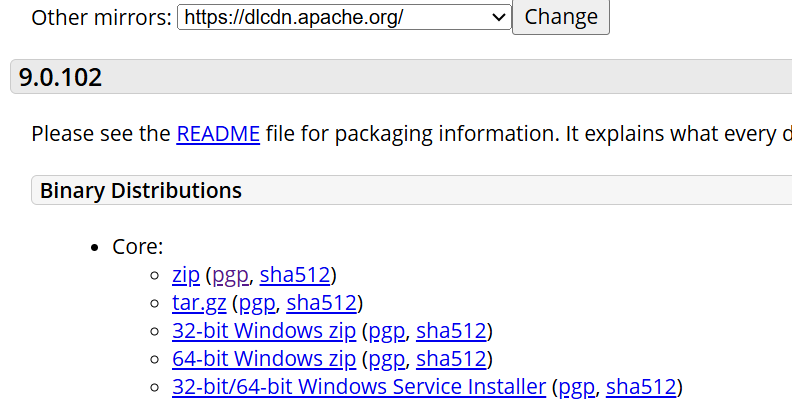






Common expected error while running this experiment:

1] tomcat zip file not extracted properly.



Click on zip only to download properly.

2] improper folder structure .

While this experiment to work ,folder structure is important .

Action on tomcat like ,starting it ,stopping it or restrating it must be done from the cmd which is open on bin location of tomcat

Ex. cd C:\Users\User\apache-tomcat-9.0.102\bin then open cmd

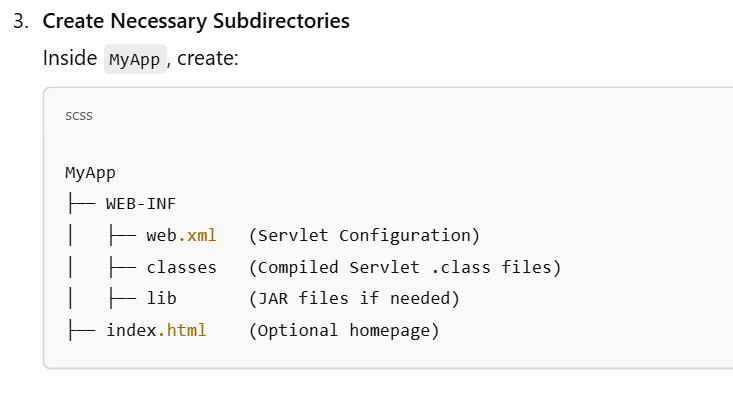
* Stop Tomcat:

shutdown.bat

* Start Tomcat :

startup.bat

the folder structure:



When opening the tomcat folder , select the webapp folder inside which we can create our own folder.

I have created a folder named myapps inside where above folder structure resides.