MAVEN& TOMCAT SERVER

Installation Process

ABSTRACT

This guide explains how to create a Maven project and set up the Tomcat server using the Eclipse IDE. It covers how to use Maven to manage project files and dependencies, and how to install and configure Tomcat to run Java web applications. The goal is to help beginners understand how to set up a basic Java web development environment using Eclipse, Maven, and Tomcat.

Installation Guide....

Maveen Project & Tomcat Installation Process

Document: How to Create a Maven Project in Eclipse

1. Introduction:

This document provides a step-by-step guide for creating a Maven project using Eclipse IDE. Maven is a build automation tool used primarily for Java projects.

2. Prerequisites:

2.1 Install Java Development Kit (JDK):

- Download and install **JDK 8 or above** from the Oracle website or OpenJDK.
- Set JAVA_HOME environment variable:

JAVA_HOME "C:\Program Files\Java\jdk-version"

- Verify the installation
  - ➔ java -version
  - ➔ javac -version

2.2 Install Eclipse IDE:

- Download Eclipse IDE for Java Developers from the official Eclipse website.
- Install and launch Eclipse.

3.Creating a Maven Project in Eclipse:

Step 1: Open Eclipse

- Launch Eclipse IDE.

Step 2: Create a New Maven Project

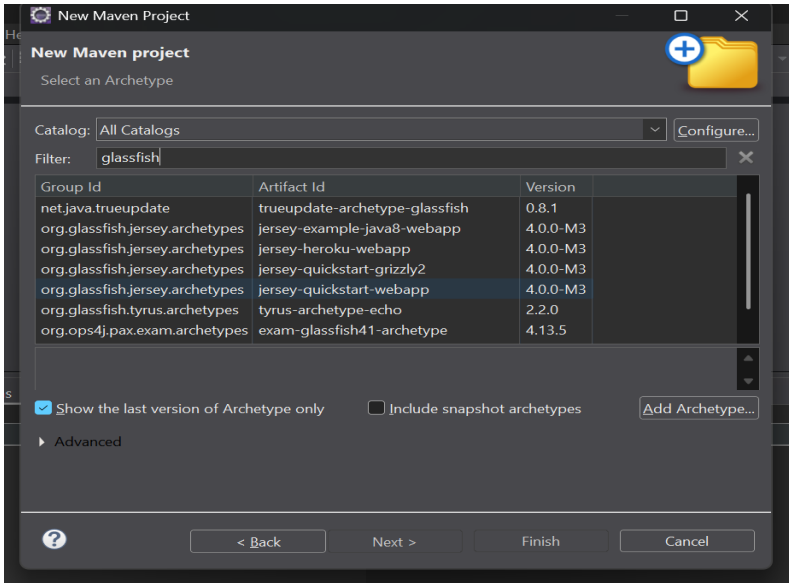
- Go to File > New > Other...
- Select Maven > Maven Project and click Next.

Step 3: Select Project Location

- Leave the default workspace location or specify a different location.
- Click **Next**.

Step 4: Select an Archetype

1. Choose an archetype in the catalog (a project template). For beginners, use:

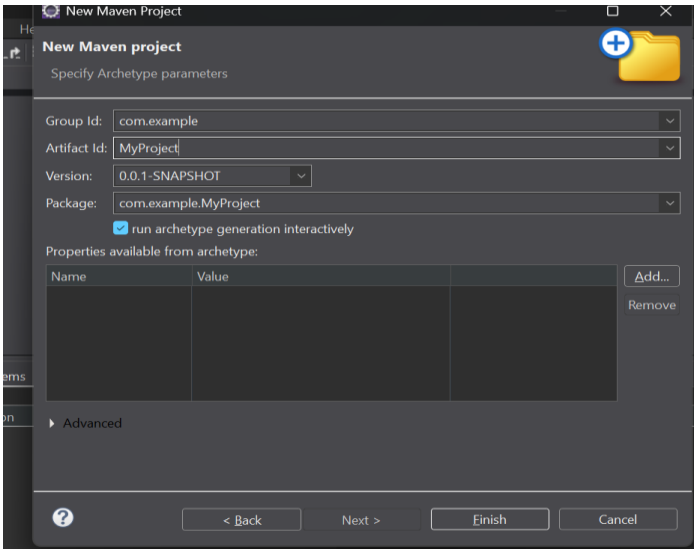


Filters use glassfish (select jersey-quickstart-webapp 4.0.0-M3

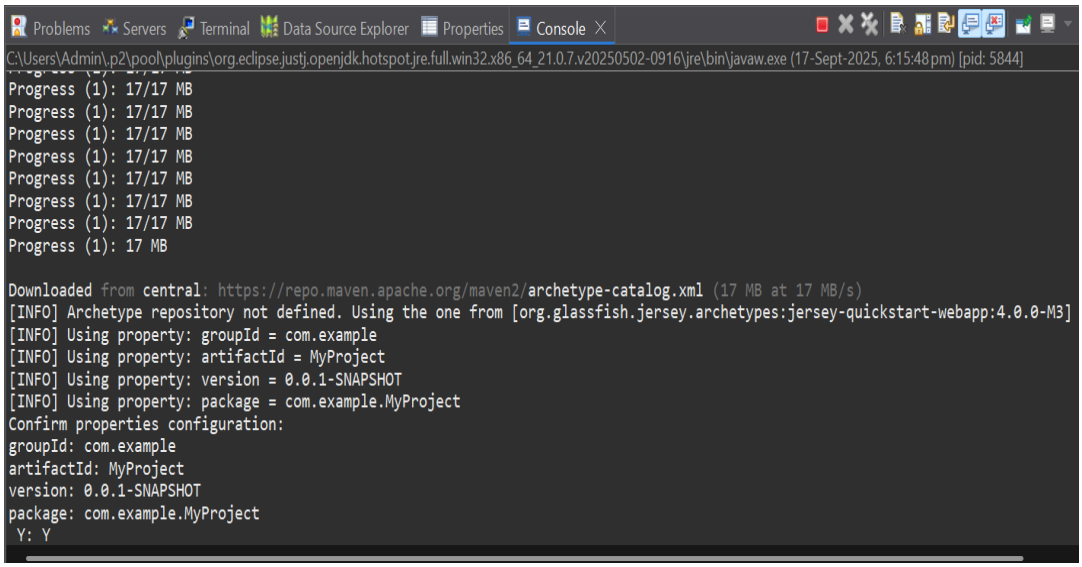
2. Click **Next**.

Step 5: Define Project Coordinates

- Enter Group Id (e.g., com.example)
- Enter Artifact Id (project name, e.g., my-maven-project)
- Enter Version (default is 1.0-SNAPSHOT)
- Click Finish.



- After Clicking Finish in console you will see the maven creation log at last, Enter Y it will create your first maven project

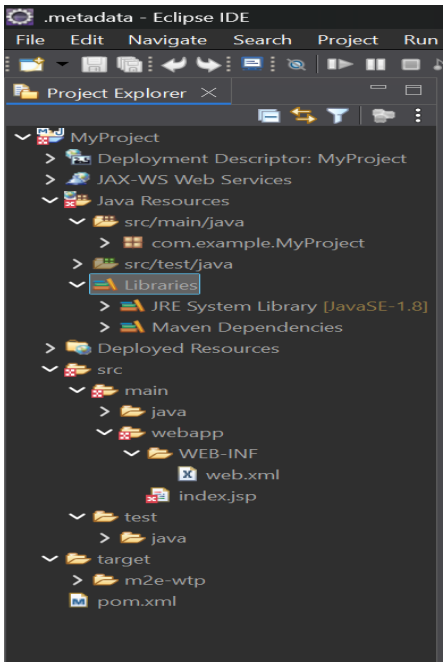


4. Understanding the Project Structure

```
my-maven-project
├── src
│   ├── main
│   │   └── java
│   └── test
│       └── java
├── pom.xml
```

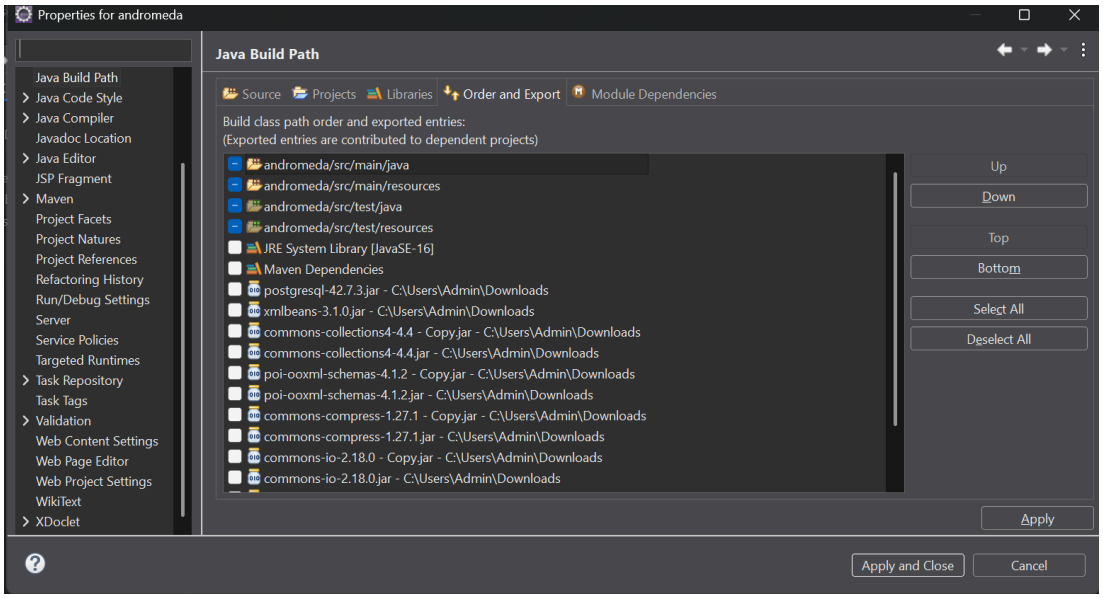
- src/main/java – Place your Java source files here.
- src/test/java – Place your test Java files here.
- src/main/webapps – Place your jsp files here.

- pom.xml – The Project Object Model file where project configurations and dependencies are managed.

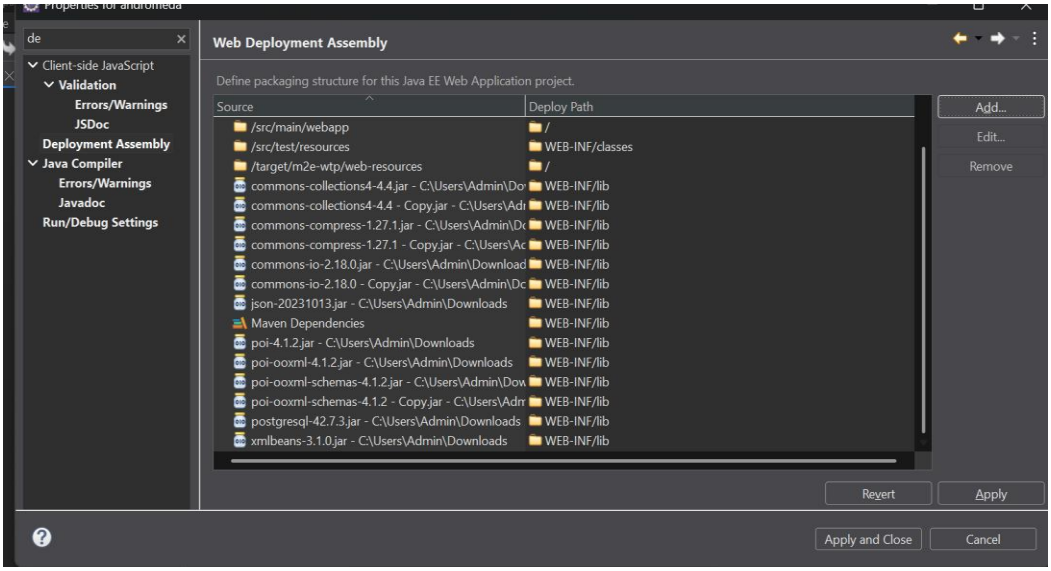


- Once completed your first project creation then add some external jars to your project. Right Click on your project search for Build Path → Configure Path of your project and it opens a dialog box with the details like this click on the Libraries at the right side you may notice some extra options like Add Jars, Add External jars.

- Click on Add External Jars and add your downloaded External jars after adding your jars then it shows like this then Apply and Apply and Close.



- It will add your jars and you may notice some features on your left side menu like Java Build Path search for Deployment Assembly. Once it opens click on Add a small dialog box opens in that click on Java build Path Entries it will show all your added jars select all the jars Click Apply and again Click on Apply and Close.



- It will add your all jars to your project and look like this after adding all your required jars.
- To here your first maven project creation is completed and now the installation of Tomcat server in your eclipse this will guide you to install your tomcat server (Tommy Server)

Installation Guide for Apache Tomcat Server in Eclipse:

1. **Introduction:**

This document provides a step-by-step guide to install and configure Apache Tomcat server in Eclipse IDE, allowing developers to deploy and test Java web applications efficiently.

2. **Installing Tomcat Server in Eclipse:**

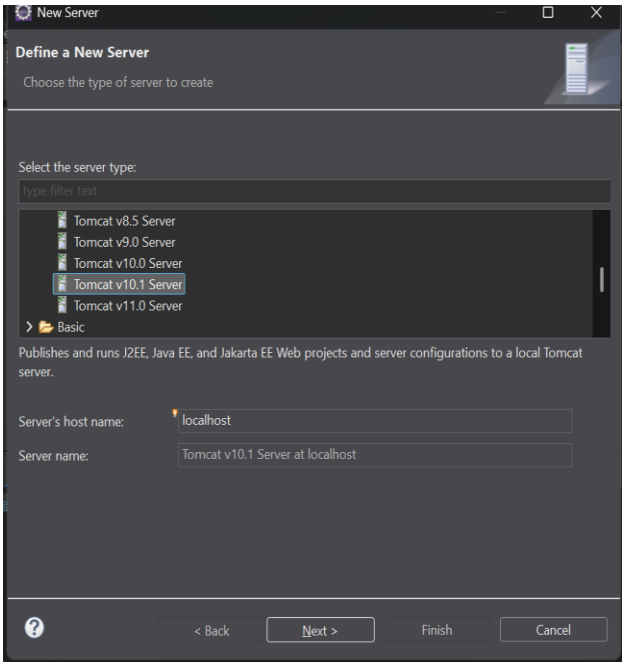
Step 1: Download Apache Tomcat

1. Go to the Apache Tomcat Official Website.
2. Click on **"Download"** for the desired version (e.g., Tomcat 9.0 or 10.0).
3. Choose the **Core zip** or **tar.gz** version and download.
4. Extract the downloaded archive to a known directory (e.g., C:\apache-tomcat-9.0.x)

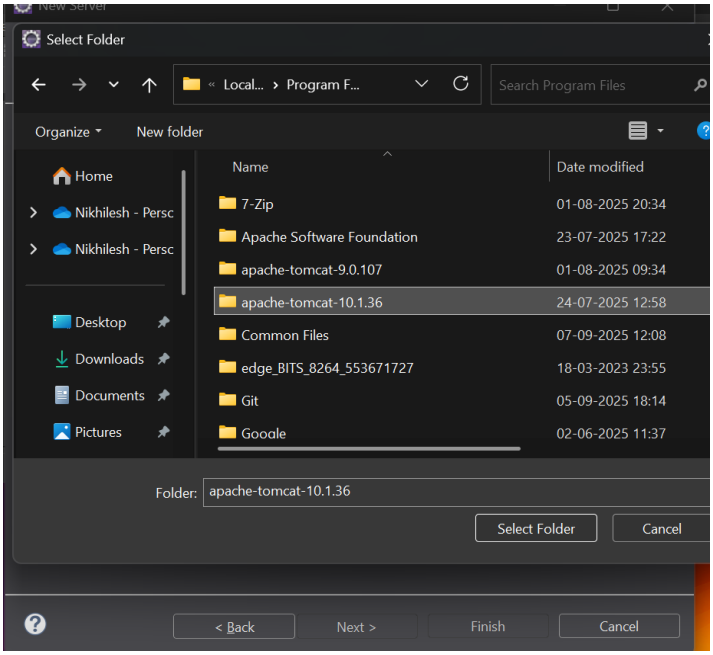
Step 2: Configure Tomcat in Eclipse

1. **Open Eclipse IDE.**
2. Click on **"Servers"** tab at the bottom. *(If not visible: Window > Show View > Servers)*.
3. Click **"No servers are available. Click this link to create a new server..."**
4. In the **New Server Wizard**:
    - Expand **Apache**
    - Select the version of Tomcat you downloaded (e.g., *Apache Tomcat v9.0*)
    - Click **Next**
5. Click **Browse** and select the **Tomcat installation directory** (e.g., C:\apache-tomcat-9.0.x).
6. Click **Finish**. You may see your added server in the **Servers** tab.

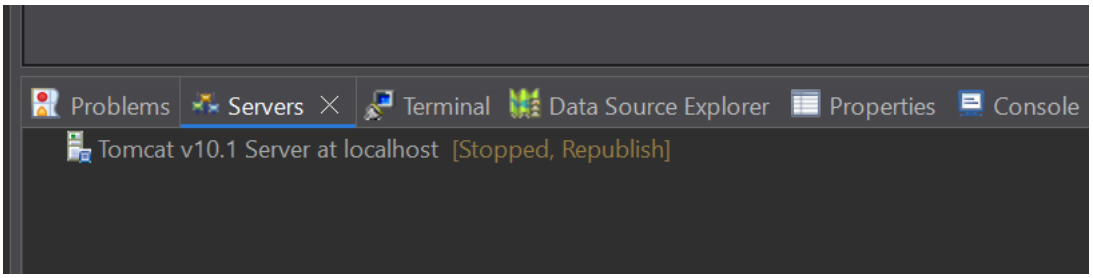
4.1 *



5.1*



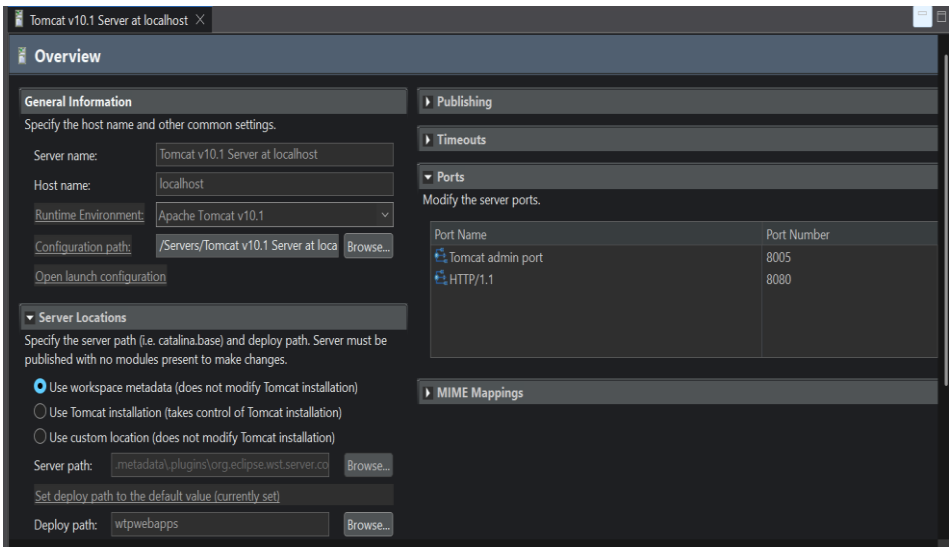
7.1



3. Configure Tomcat Server in Eclipse

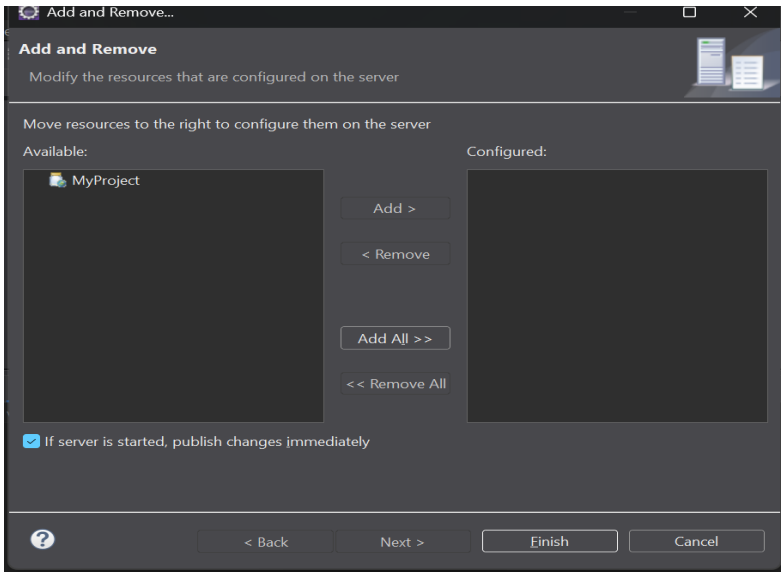
Step 1: Set Server Ports (Optional)

- Double-click the server in the Servers view.
- The server configuration editor opens.
- Change the HTTP/1.1 port (default 8080) if needed to avoid conflicts.
- Save the configuration.



Step 2: Add Projects to the Server

- Right-click the server and choose Add and Remove....
- Select your web project from the left pane.
- Click Add > to deploy it to Tomcat.
- Click Finish**.**



Step 3. Starting and Stopping Tomcat Server

- To start the server, right-click the server in the Servers view and select Start.
- To stop the server, right-click and select Stop.
- Check the Console view for logs to confirm the server is running**.**

How to Configure Tomcat Server in Environment Variable

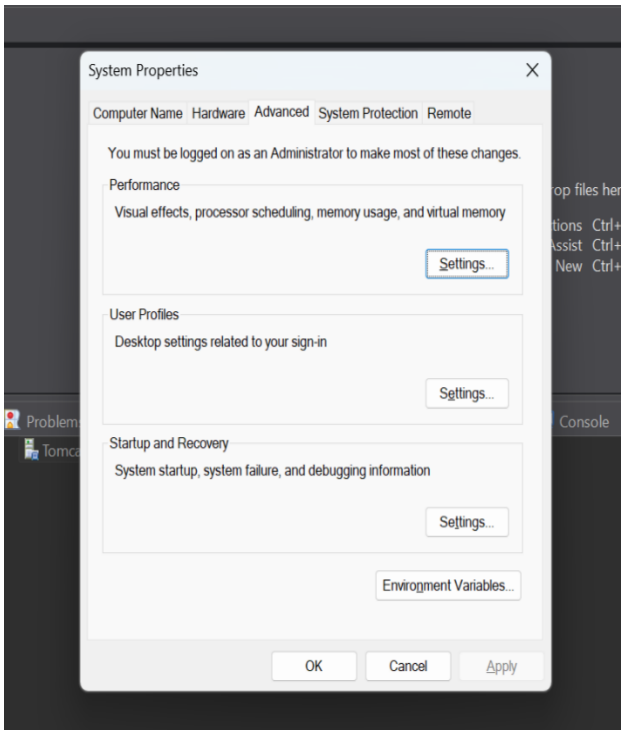
1. Set **CATALINA_HOME** Environment Variable**.** This variable points to your Tomcat installation directory.

On Windows:

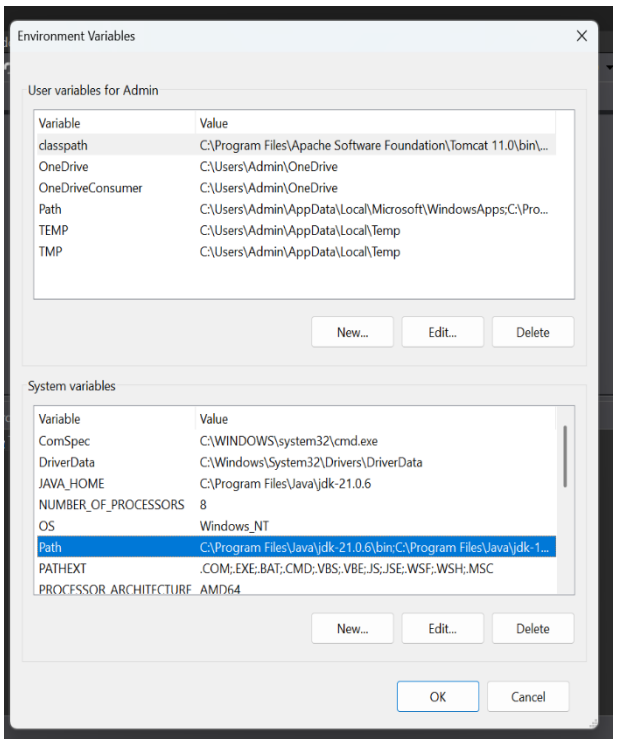
1. **Open Environment Variables:**
    - Right-click on This PC or My Computer > Properties.
    - Click Advanced system settings.
    - Click Environment Variables... button.
2. **Create a New System Variable:**
    - Click New... under the System variables section.
    - Set Variable name as: CATALINA_HOME
    - Set Variable value as the full path of your Tomcat folder, e.g., C:\apache-tomcat-10.0.XX
    - Click OK.

3. **Add Tomcat `bin` Folder to `PATH`:**
    - In **System variables**, find and select the `Path` variable, then click **Edit...**
    - Click **New** and add: `%CATALINA_HOME%\bin`
    - Click **OK** on all dialogs to apply the changes.

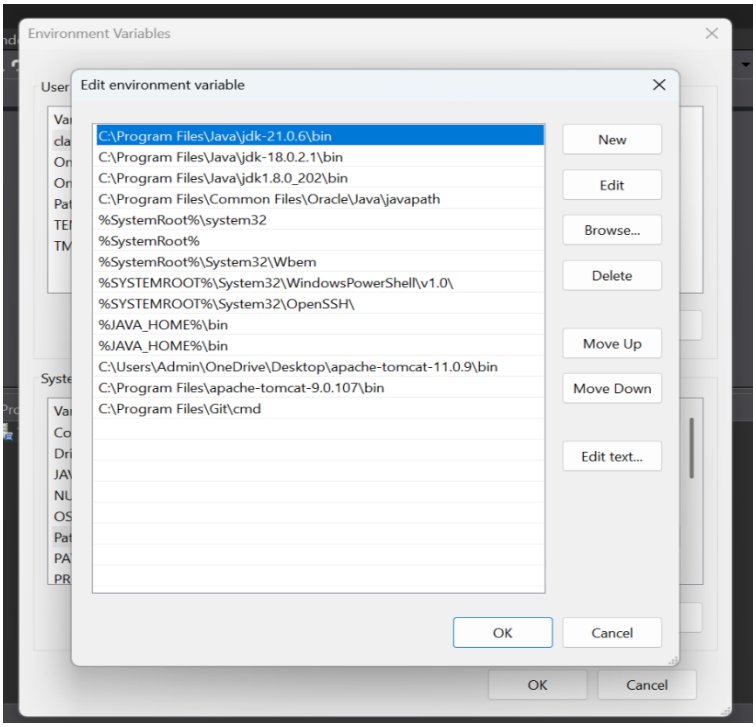
1.1



2.1

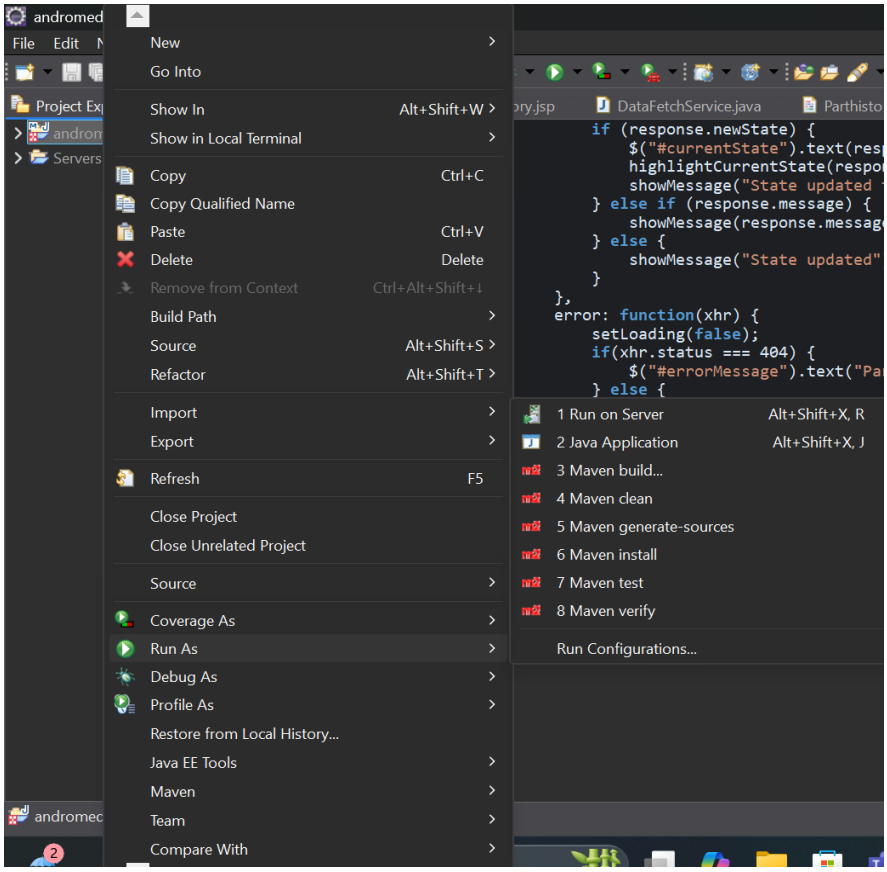


3.1



4. **To run your project on server:**
    - Right-Click on your Maven Project e.g. My Project.
    - It opens a dialog box search for Run AS. For your notice I attach an image please refer 4.1.
    - Again, it opens a small there you can notice Run on Server. Once you click on that your project starts run on the server.
    - You may notice the server running status in the console of your eclipse.

4.1*



Manual Installation and Usage of Apache Tomcat Server:

1. What is Apache Tomcat?

Apache Tomcat is an open-source **Java Servlet Container** and **Web Server** developed by the Apache Software Foundation. It implements several Java EE specifications such as:

- Java Servlet
- Java Server Pages (JSP)
- WebSocket
- Java Expression Language

Tomcat allows you to run Java web applications on your system by providing an HTTP server environment.

2. Manual Installation of Tomcat Server

Step 1: Download Apache Tomcat

- Go to the official Apache Tomcat website: https://tomcat.apache.org
- Navigate to **Download** and select the desired Tomcat version (e.g., 9.x or 10.x).
- Download the **Core** package in zip or tar.gz format.

Step 2: Extract Tomcat: Extract the downloaded archive to a directory of your choice, for example: C:\apache-tomcat-9.0.XX on Windows.

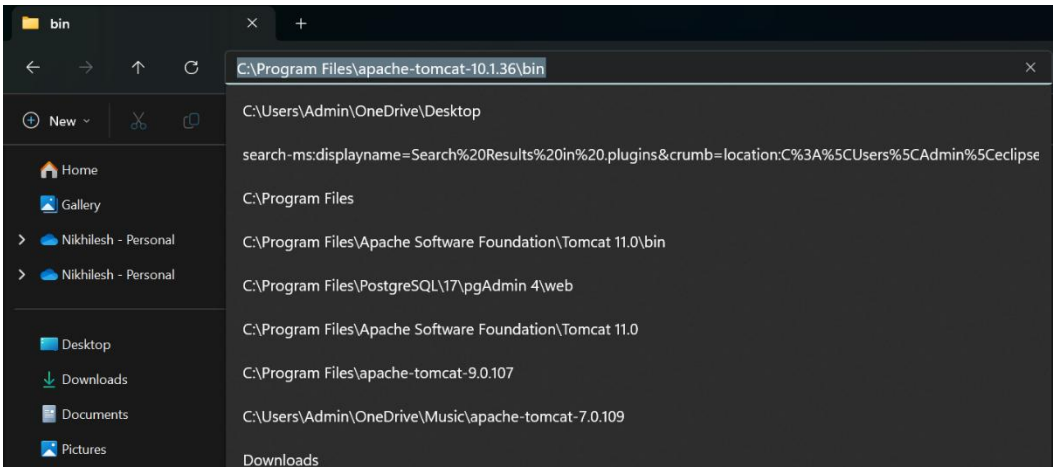
Step 3: Set Environment Variables

Refer to the earlier instructions to set (1.1,2.1,3.1):

- CATALINA_HOME pointing to the Tomcat root directory.
- Add the bin folder inside Tomcat to your system PATH.

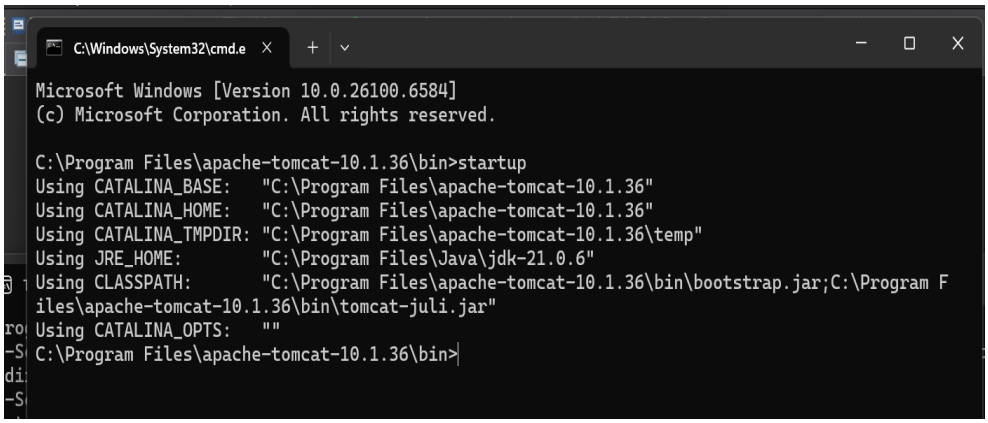
Step 4: Commands to Run and Stop Tomcat Server

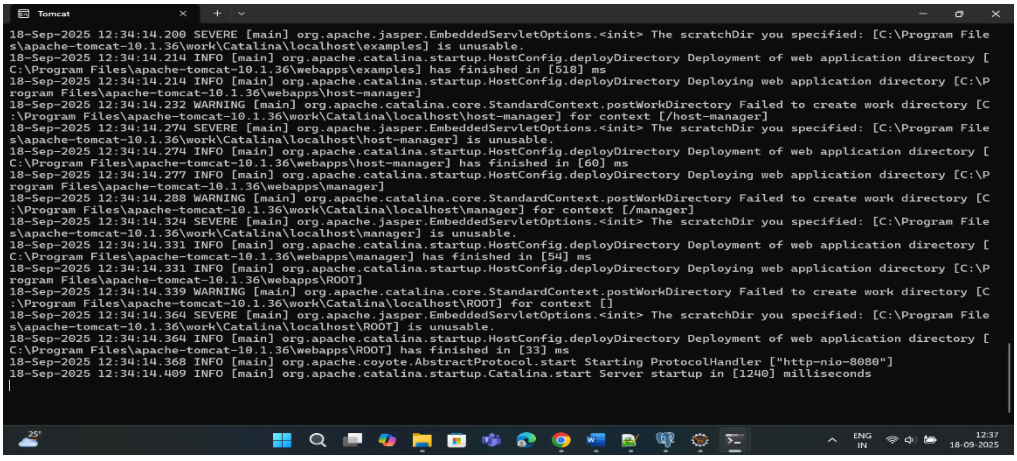
- Open a terminal or command prompt and navigate to the Tomcat bin directory: cd /path/to/apache-tomcat-9.0. XX/bin
- For your better convenient open your Tomcat bin folder in the top toolbar you may see like this enter **cmd.** It will redirect to command prompt.



Starting the Server

- **On windows:** Enter the following command in the command prompt **startup.bat** / **startup**

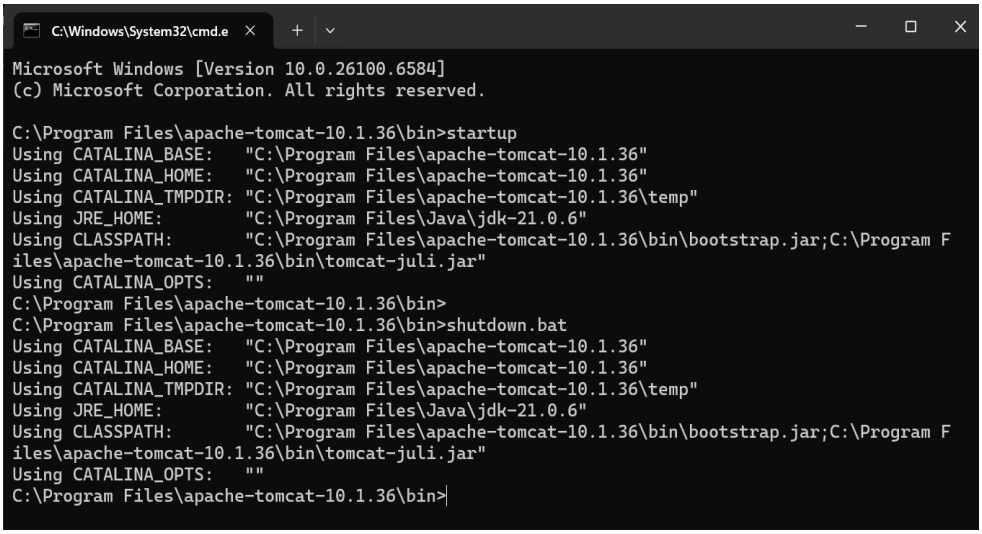




- You may notice the running of your Tomcat server.

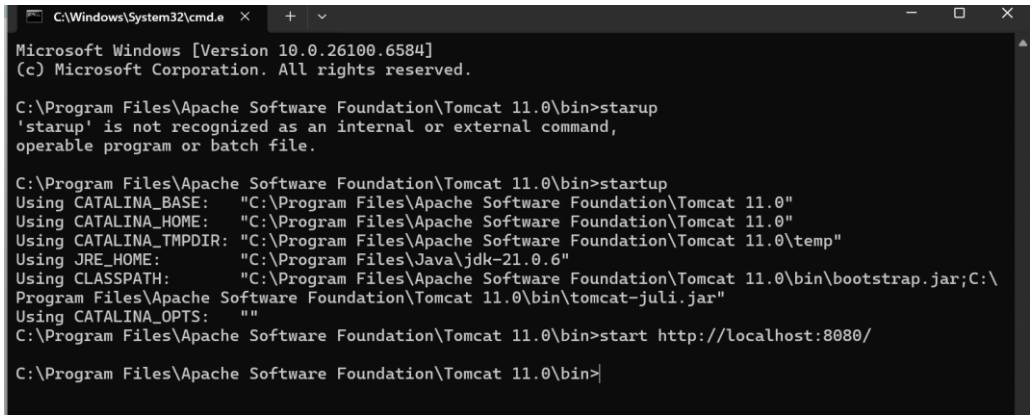
Stopping the Server

- **On Windows:** Enter the following command in the prompt **shutdown.bat.**
- Once you enter the command the server stops.

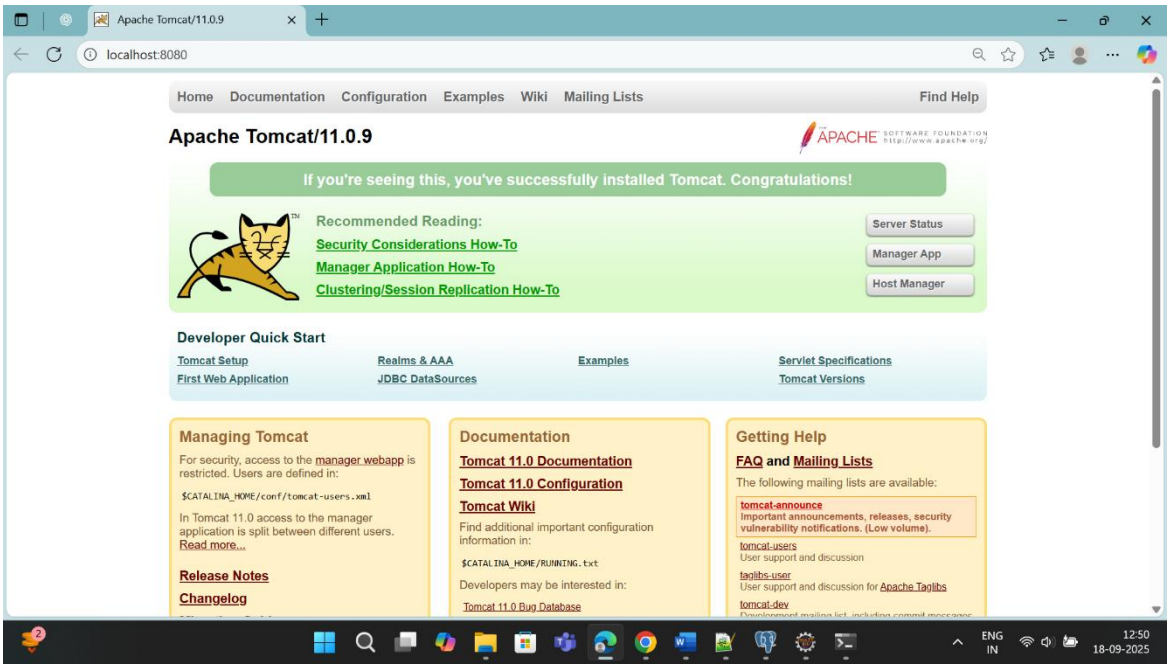


Checking if Tomcat is Running

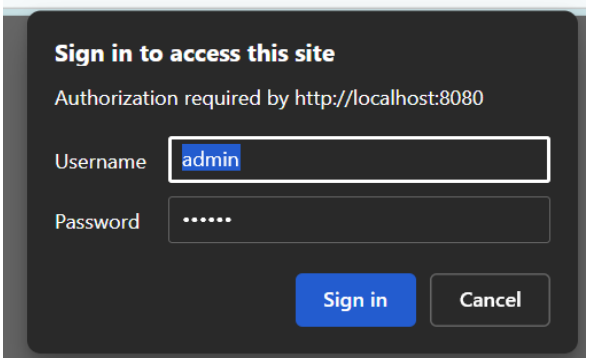
- Open a browser and go to: http://localhost:8080 or else you may enter the following command in the command prompt it directly opens Tomcat welcome page. start http://localhost:8080/



- You should see the Tomcat welcome page like this.



When you click on Manager App it asks Authorization make sure to save your authentication details username and password.



Deploying Web Applications to Tomcat

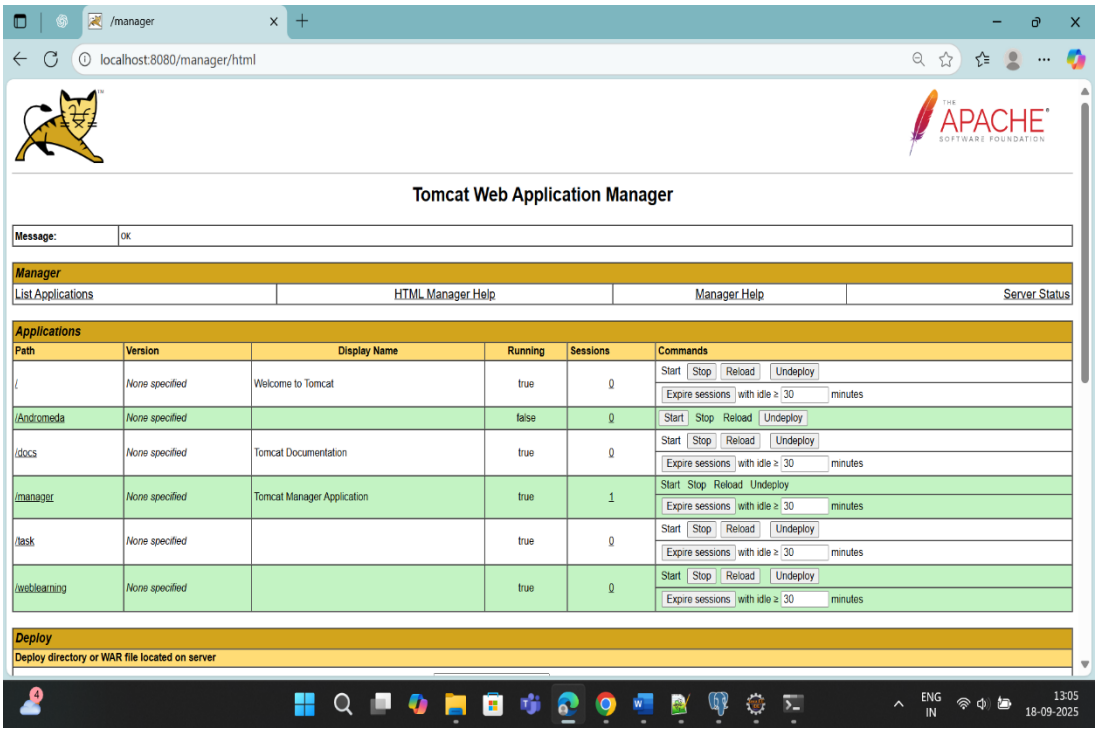
- There are several ways to deploy your Java web application (WAR file or exploded folder) to Tomcat:

Method 1: Manual Deployment

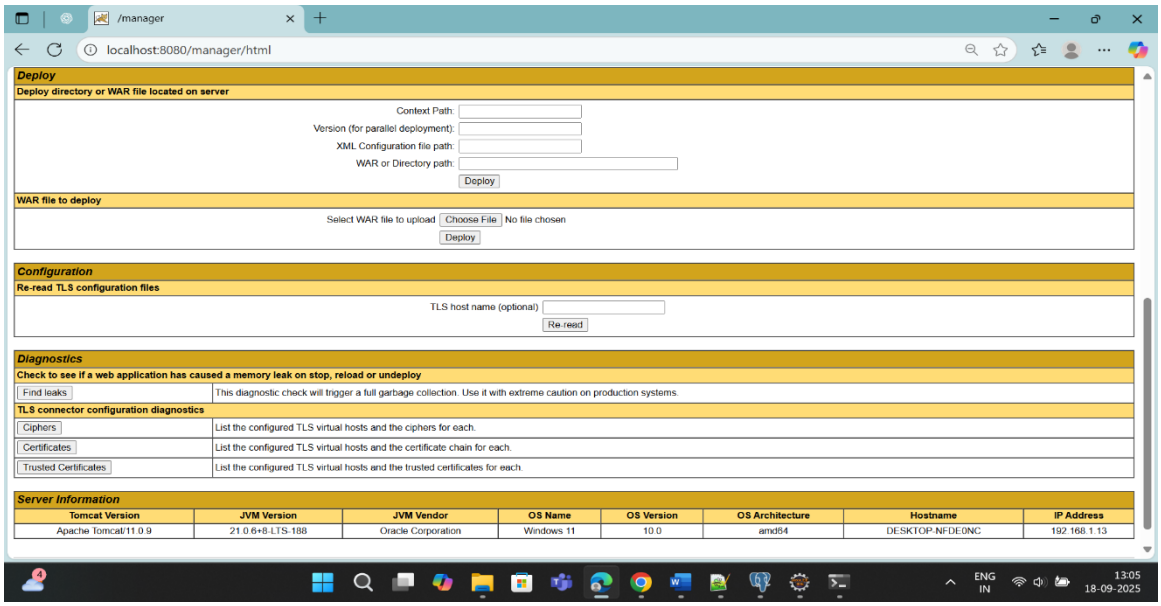
1. Build your web application as a **WAR file** (Web Application Archive).
2. Copy the WAR file to the Tomcat webapps directory `C:\ProgramFiles\apache-tomcat-10.1.36\webapps`
3. Tomcat automatically extracts and deploys the WAR when started or restarted.
4. Access the application in your browser via: `http://localhost:8080/YourAppName`

Method 2: Deployment via Tomcat Manager (Web Interface)

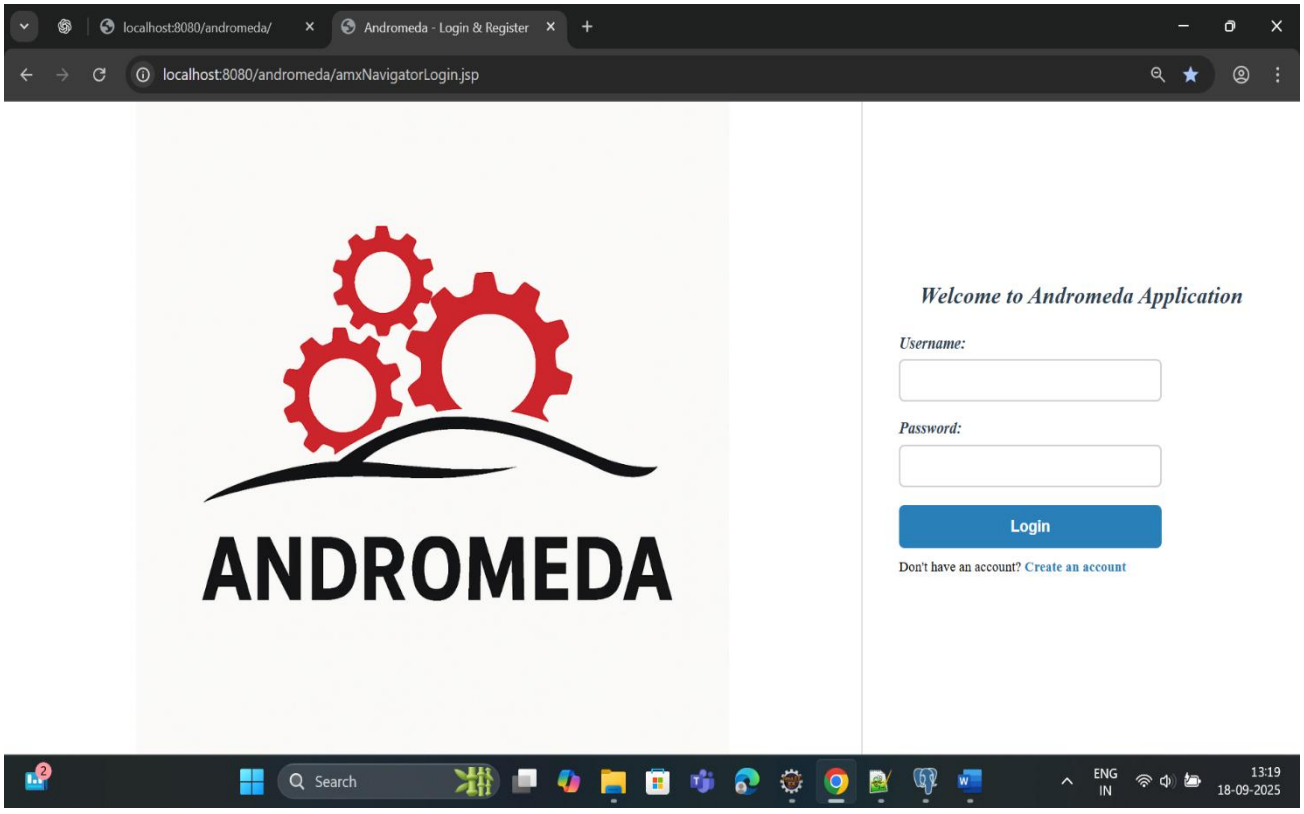
1. Enable the **Tomcat Manager** application (comes pre-installed).
2. Configure user roles in the conf/tomcat-users.xml file: Add the following inside <tomcat-users>tag <role rolename="manager-gui"/> <user username="admin" password="password" roles="manager-gui"/>
3. Restart Tomcat.
4. Access Tomcat Manager at: `http://localhost:8080/manager/html`
5. Use the interface to upload WAR files or deploy applications.
6. Open your **Tomcat Web Application Manager** you may see like this



7. Scroll down you may notice **Deploy** below you may see **WAR file to deploy** there you can deploy your war file.



- Once you deploy your file you can see your deployed file in the **Applications tab.** I deployed a war file called **task** you can see the war file in that /task and you can notice the commands box there that my war file is deployed and the running status.
- Now you can open your project application that will be available `http://localhost:8080/your_projectname` E.g. this is my project application `http://localhost:8080/andromeda/amxNavigatorLogin.jsp`
- Like this it will open your project application.



Tomcat Directory Structure

- **bin/** — Startup and shutdown scripts.
- **conf/** — Configuration files, including server.xml and tomcat-users.xml.
- **lib/** — Libraries required by Tomcat.
- **logs/** — Log files.
- **webapps/** — Default directory where you deploy your web applications.
- **work/** — Temporary working directory.
- **temp/** — Temporary files.

Important Configuration Files

- server.xml: Main configuration file for the Tomcat server settings (ports, connectors, etc.).
- web.xml: Default servlet configuration.
- tomcat-users.xml: User and role management for Tomcat Manager and Host Manager apps.