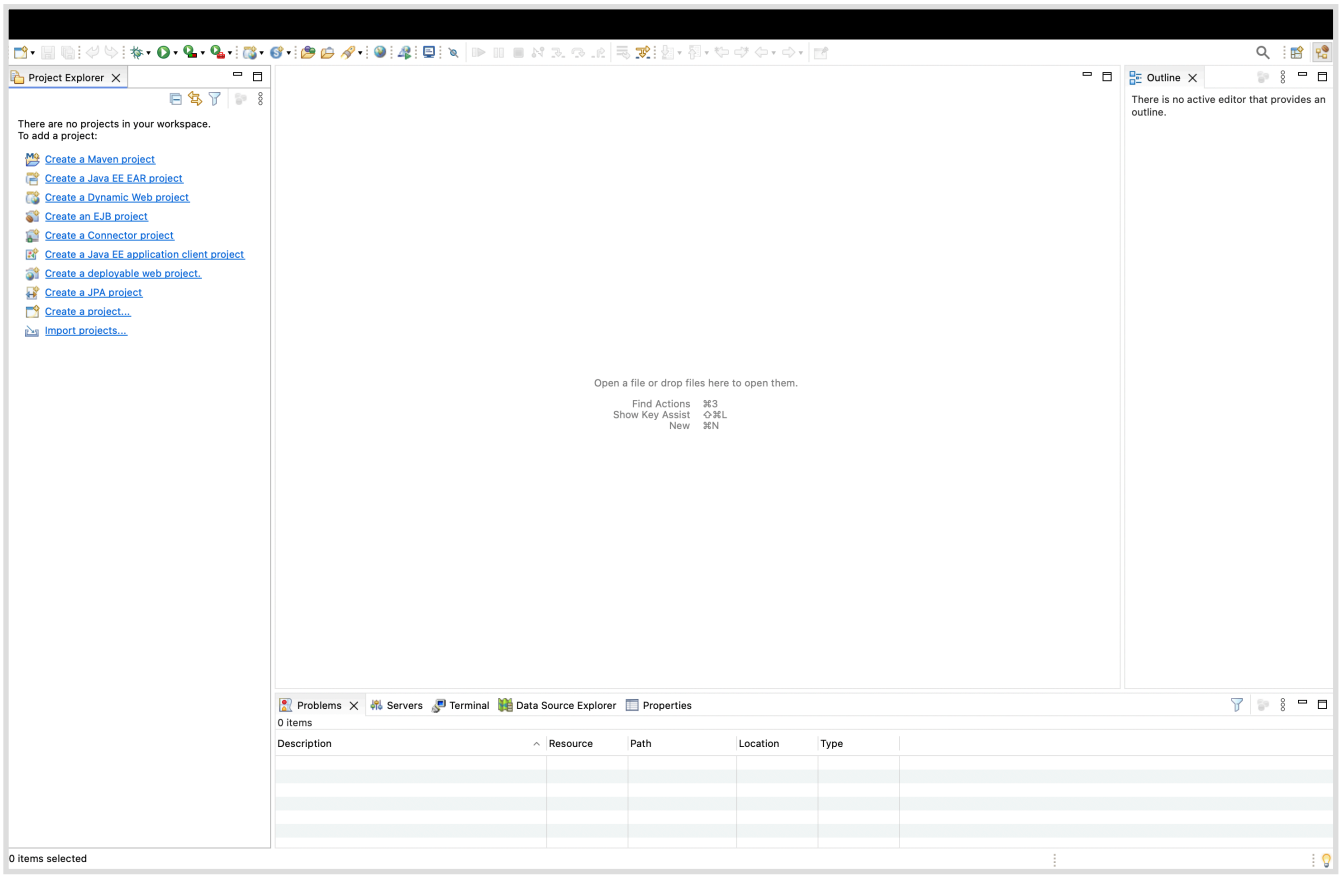
Creation of Maven Java Project

Step 1. Open Eclipse IDE

-> Launch Eclipse workspace



Step 2. Install Maven Plugin (if not installed)

->2.1. Go to "Help" in the top menu

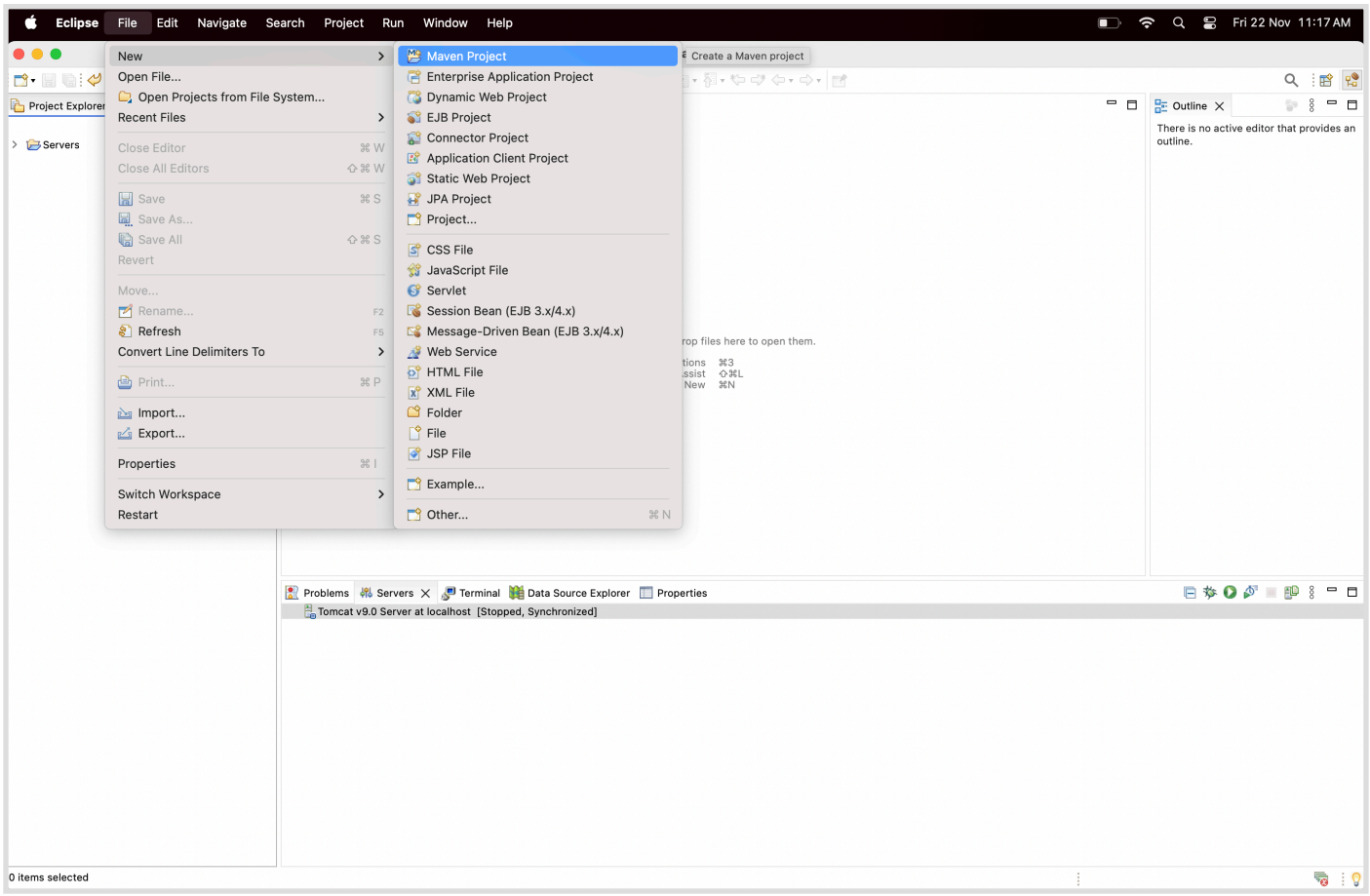
-> 2.1.1. Click "Eclipse Marketplace"

->2.1.2. Search for "Maven Integration for Eclipse"

->2.1.3. Install the plugin if not already installed

Step 3. Create a New Maven Project

->3.1. File -> New -> Project...



->3.1.1. Expand "Maven"

->3.1.2. Select "Maven Project" and click "Next"

Step 4. Set Project Configuration

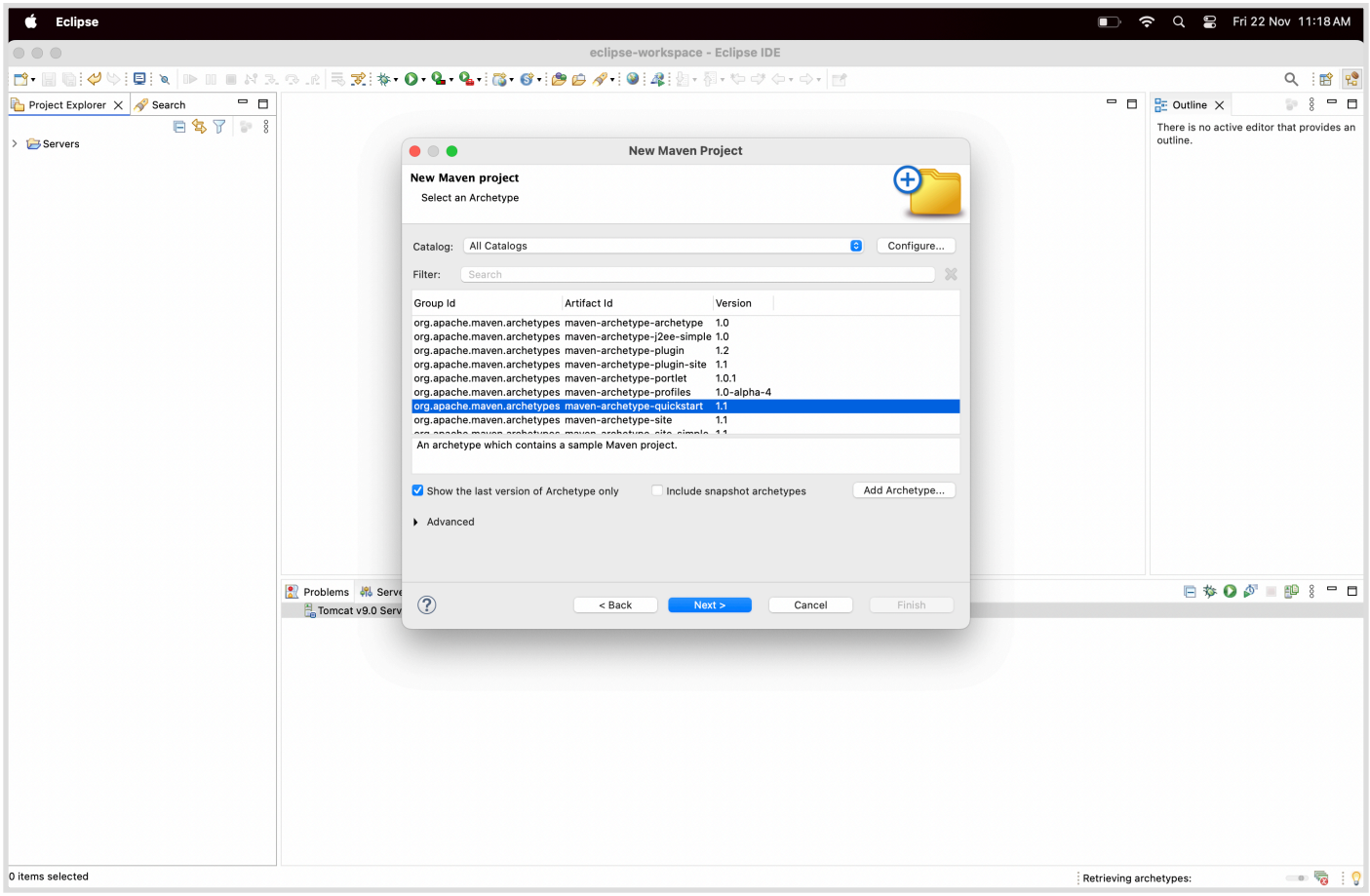
->4.1. Select workspace location (default or custom)

->4.2. Click "Next"

Step 5. Choose Maven Archetype

->5.1. Select an archetype(e.g "org.apache.maven.archetypes -> maven-archetype-quickstart 1.4 ")

->5.2. Click "Next"



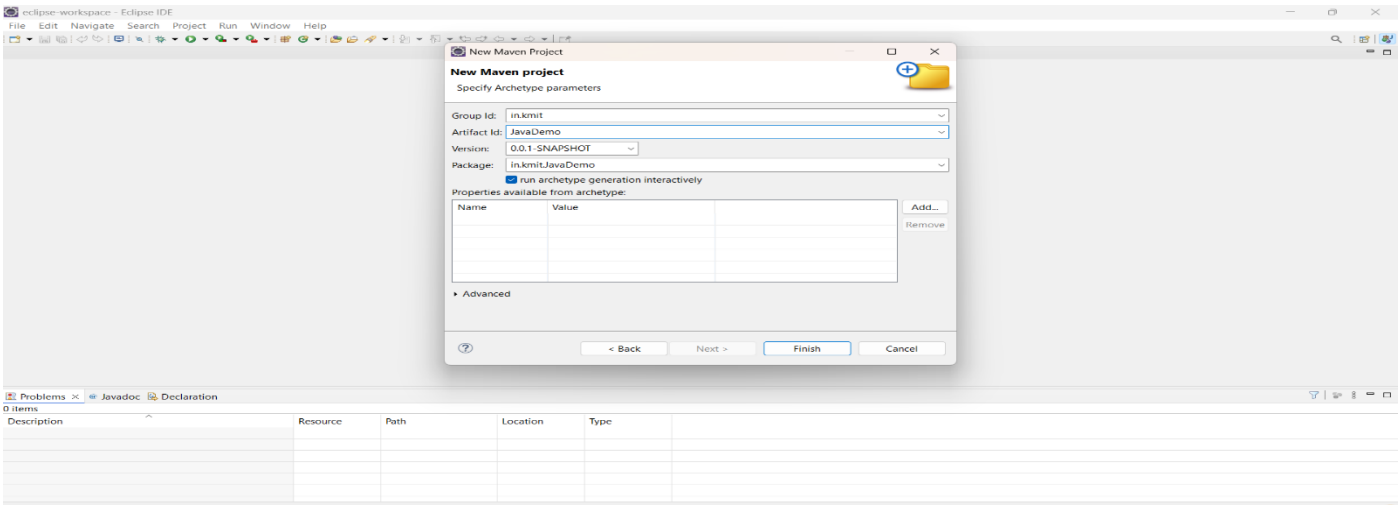
Step 6. Define Project Metadata

-> 6.1. Group ID: (e.g., com.example)

->6.2. Artifact ID: (e.g., my-maven-project)

->6.3. Version: (default is usually fine)

->6.4. Click "Finish"



In Console, artifacts are grouped. When prompted with Y/N, type 'Y'.

Step 7. Maven Project Created

->7.1. Project structure is generated with a standard Maven layout

->7.2. Includes:

-> src/main/java (for Java source code)

-> src/test/java (for test code)

->pom.xml (Maven configuration file)

Step 8. Update Project Settings (if needed)

->8.1. Right-click on the project -> Maven -> Update Project...

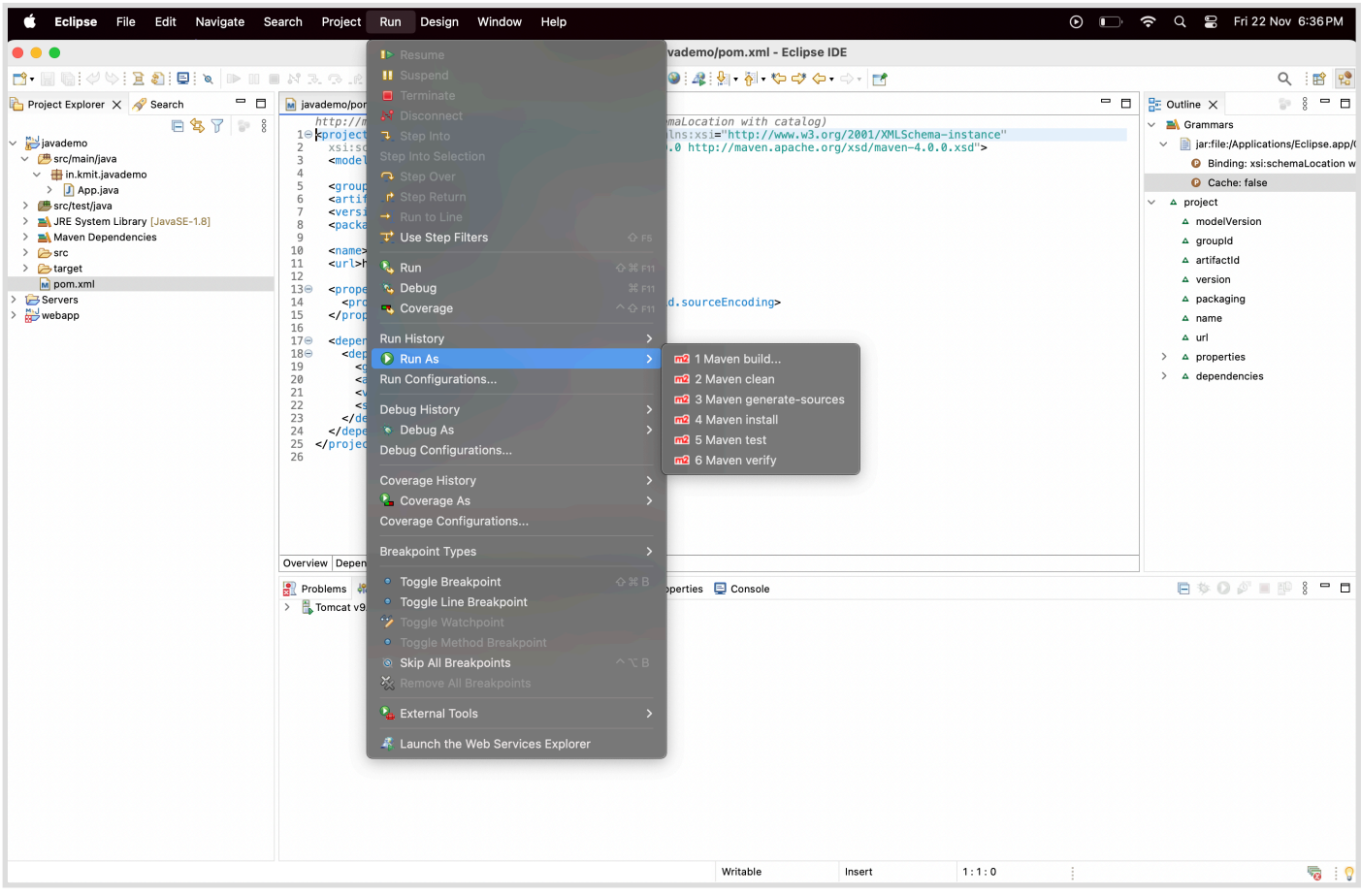
-> 8.2. Ensure dependencies are up to date

Step 9. Build and Run Maven Project

->9.1. Right-click on App.java -> Run As -> Maven Clean

->9.1.1. Right-click on App.java -> Run As -> Maven Install

->9.1.2. Right-click on App.java -> Run As -> Maven Test

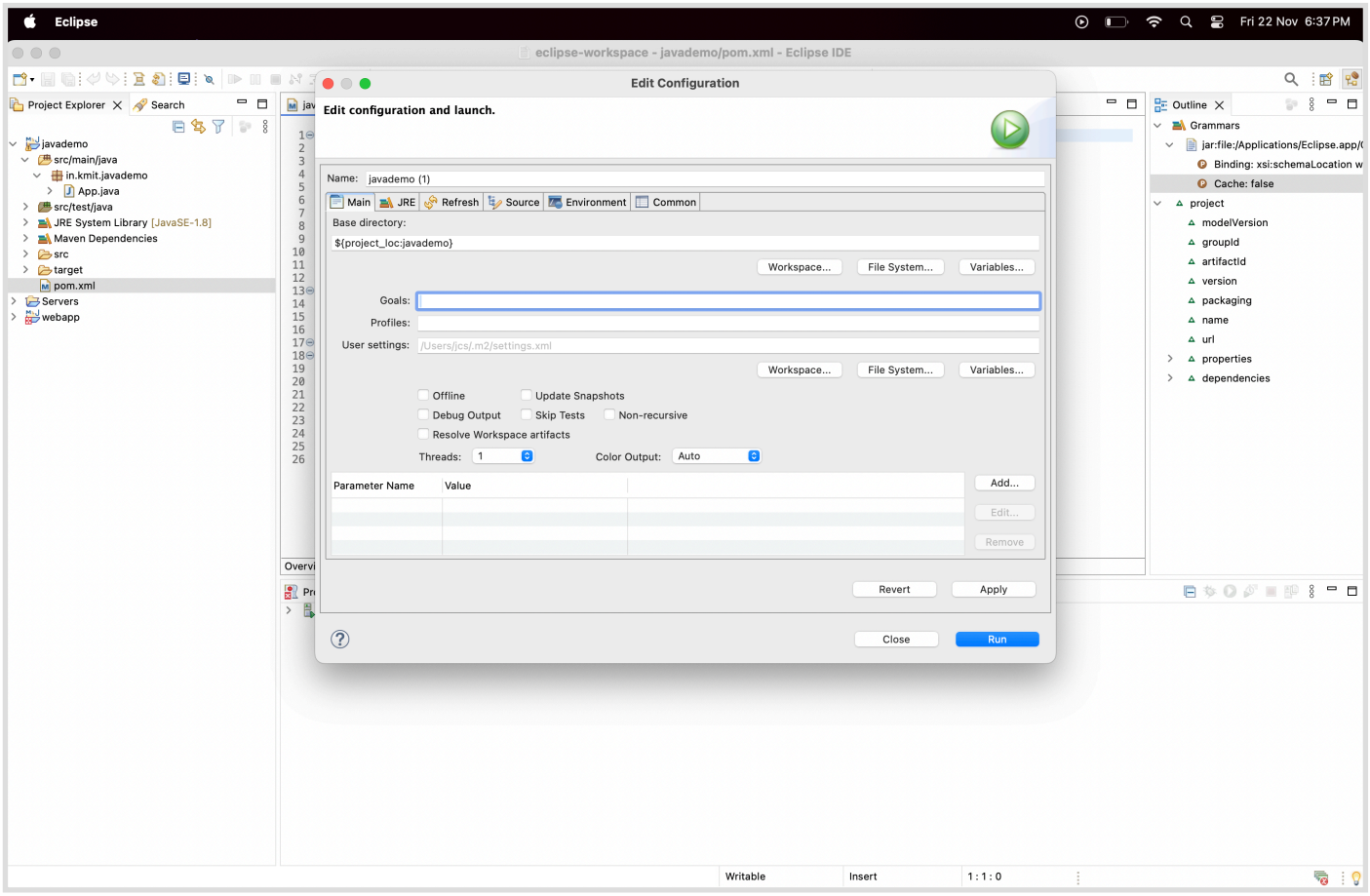


->9.1.3. Right-click on App.java -> Run As -> Maven Build

Step 10. In the Maven Build dialog:

->Enter Goals: clean install test

-> Click on Apply -> Click on Run

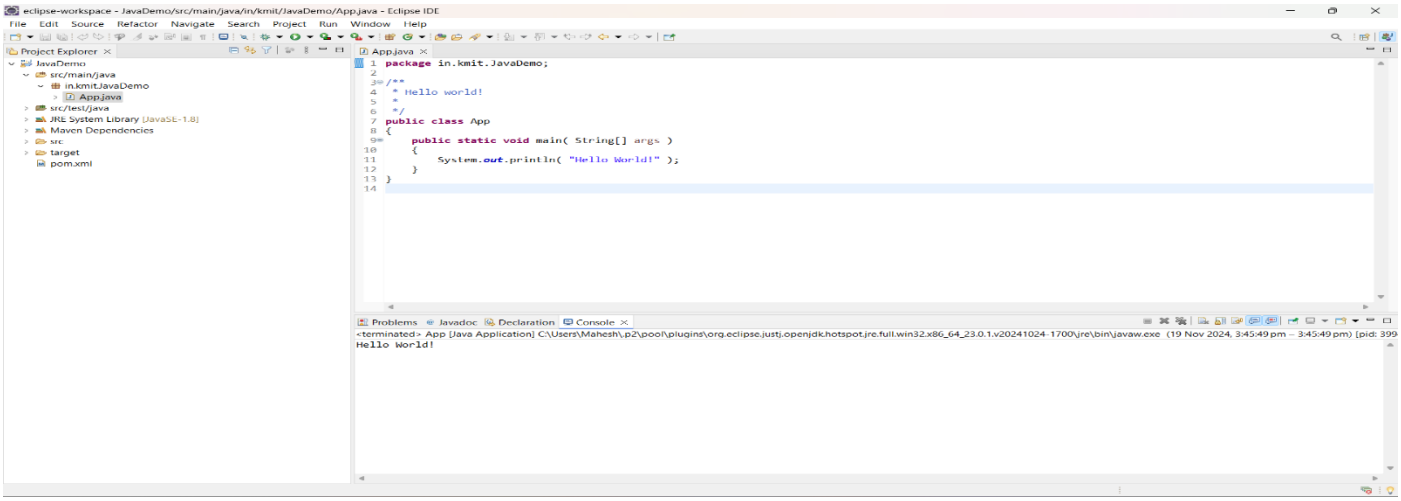


Step 11. Check console for BUILD SUCCESS message.

Step 12. Run the application:

->Right-click on App.java -> Run As -> Java Application

->Output: "Hello World" displayed.



Creation of Maven web Java Project

Step 1: Open Eclipse

->1.1 Launch Eclipse IDE.

->1.2 Select or create a workspace.

Step 2: Create a New Maven Project

->2.1. File -> New -> Project...

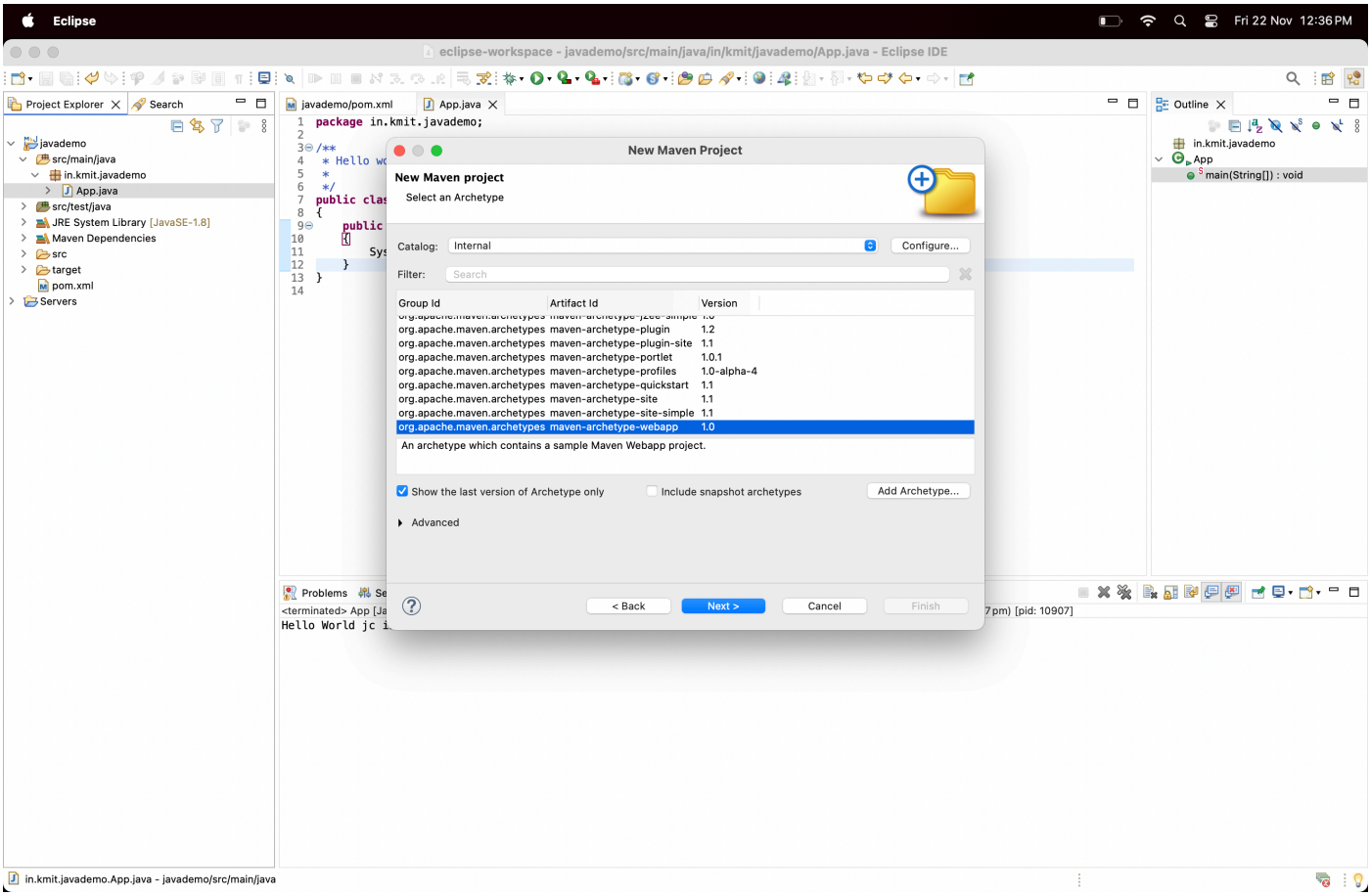
-> 2.1.1. Expand "Maven"

->2.1.2. Select "Maven Project" and click "Next"

Step 3: Choose Maven Archetype

->3.1. Select an archetype(e.g "'org.apache.maven.archetypes' -> 'maven-archetype-webapp' 1.4 ")

->3.2. Click “Next”

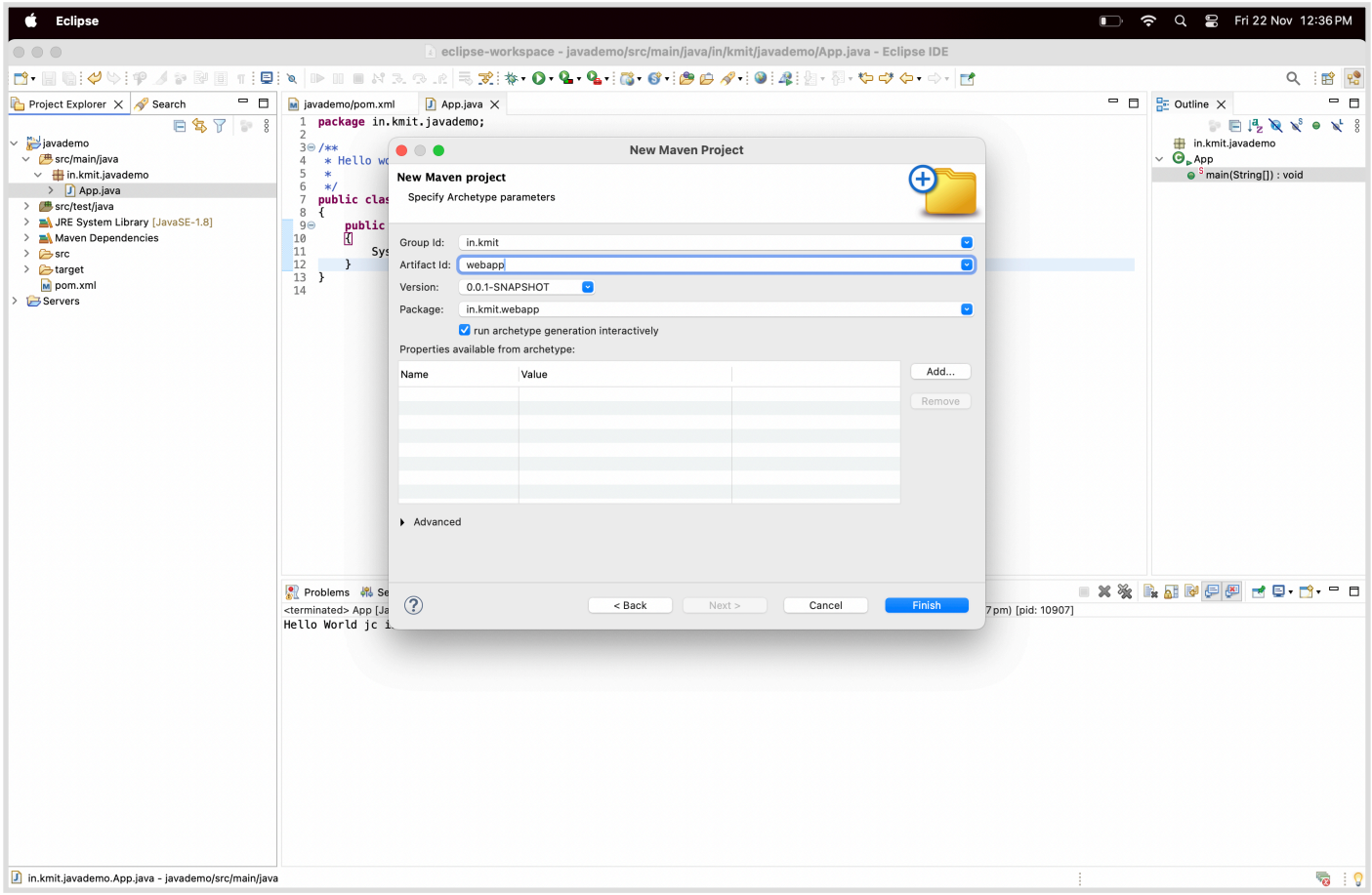


Step 4: Configure the Maven Project

->4.1 Group Id: Enter a group ID (e.g., com.example).

->4.2 Artifact Id: Enter an artifact ID (e.g., my-web-app).

->4.3 Click **Finish** to create the project.



Step 5: Add Maven Dependencies

-> 5.1 Open the **pom.xml** file in the Maven project.

-> 5.2 Add the necessary dependencies for your web project (e.g., Servlet, JSP):

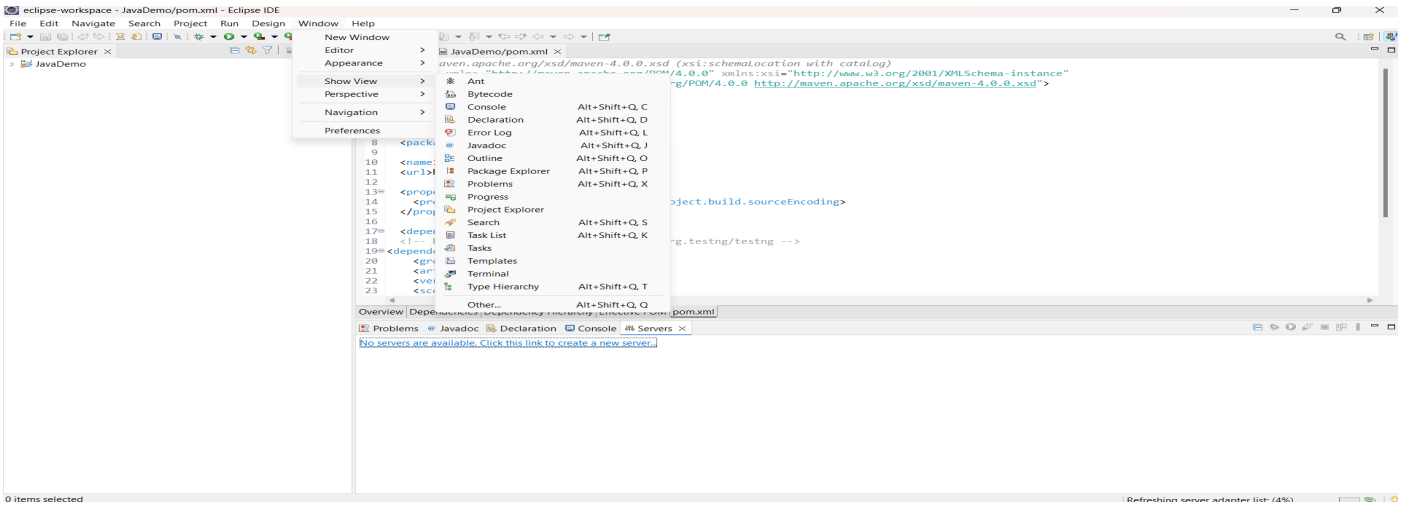
Go to browser -> Open mvnrepository.com

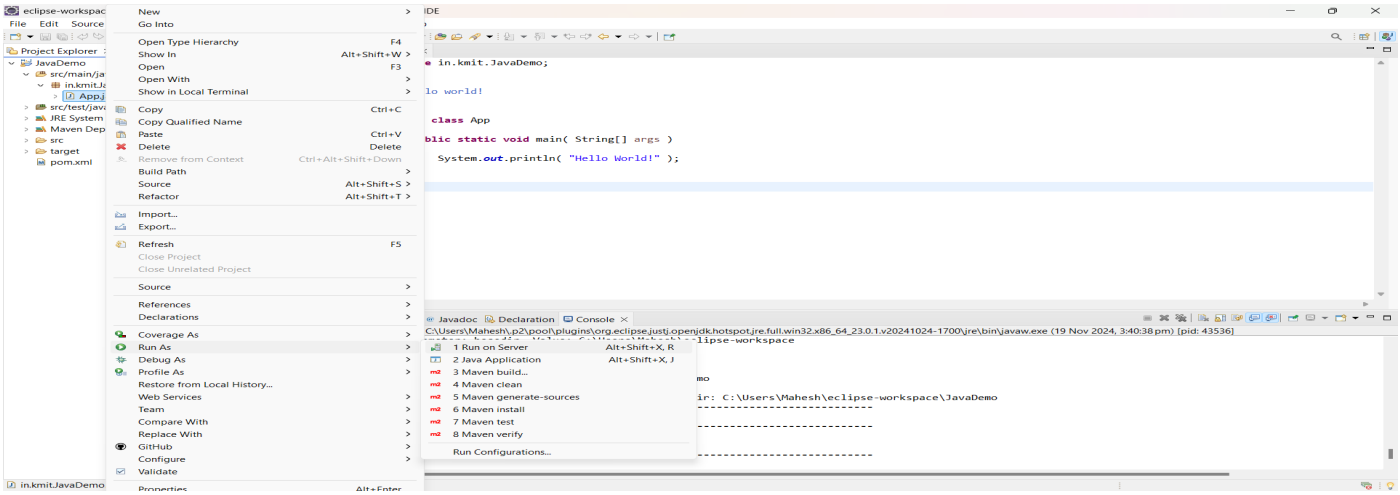
Search for 'Java Servlet API' -> Select the latest version.

Copy the dependency code -> Paste it in MavenWeb’s pom.xml under the target folder

Example:

```xml
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>4.0.1</version>
```

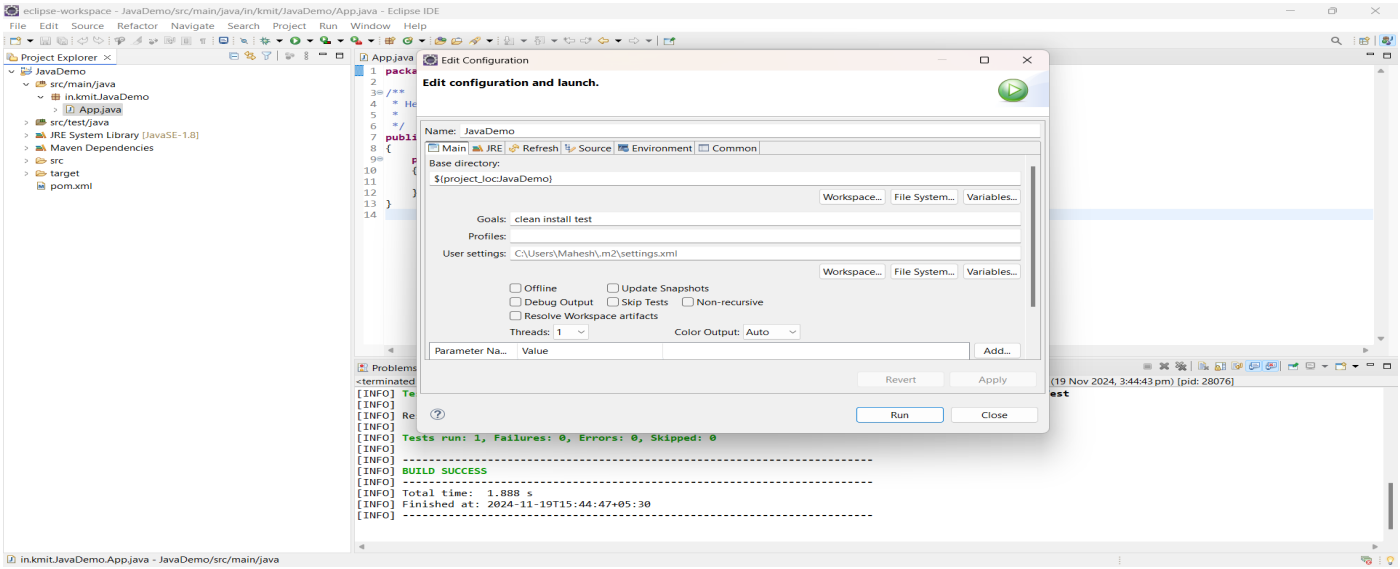
```
        <scope>provided</scope>
    </dependency>
    ```
```

Step 6:-. Configure server:

->Window -> Show View -> Servers

->Add server -> Select Tomcat v9.0 server -> Click Next

->Configure server options (e.g., ports, server location).




Step 7:-. Modify 'tomcat-users.xml':

->Add role and user details under <tomcat-users> tag.

Step 8:. Build the project:

-> Right-click on index.jsp -> Run As -> Maven Clean

->Right-click on index.jsp -> Run As -> Maven Install

->Right-click on index.jsp -> Run As -> Maven Test

->Right-click on index.jsp -> Run As -> Maven Build
```



Step 9. In the Maven Build dialog:

-> Enter Goals: clean install test

->Click on Apply -> Click on Run



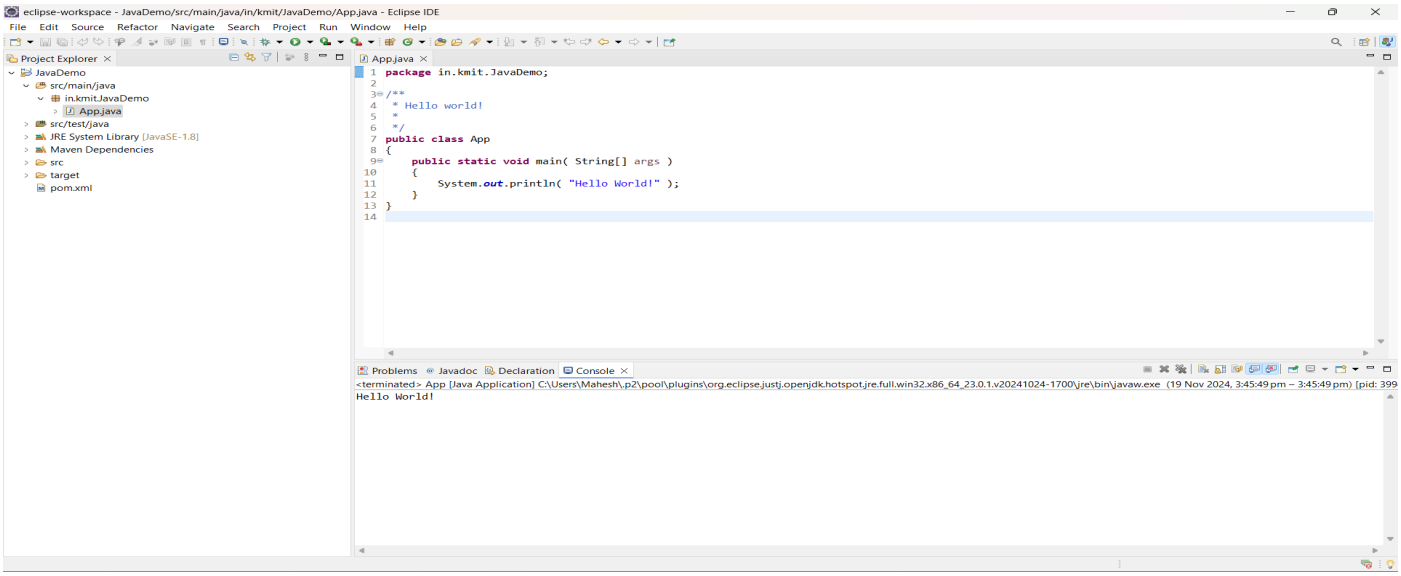
Step 10. Check console for BUILD SUCCESS message.

Step 11. Run the application:

->Right-click on index.jsp -> Run As -> Run on Server

->Select the Tomcat server -> Click on Finish

Step 12. Output: "Hello World" webpage displayed.


eclipse-workspace - JavaDemo/src/main/java/in/kmit/JavaDemo/App.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
JavaDemo
src/main/java
in.kmit.JavaDemo
App.java
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
pom.xml
App.java
1 package in.kmit.JavaDemo;
2
3 /**
4 * Hello world!
5 *
6 */
7 public class App
8 {
9 public static void main(String[] args)
10 {
11 System.out.println("Hello World!");
12 }
13 }
14
Problems Javadoc Declaration Console
<terminated> App [Java Application] C:\Users\Mahesh\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.1.v20241024-1700\jre\bin\javaw.exe (19 Nov 2024, 3:45:49 pm – 3:45:49 pm) [pid: 399
Hello World!