

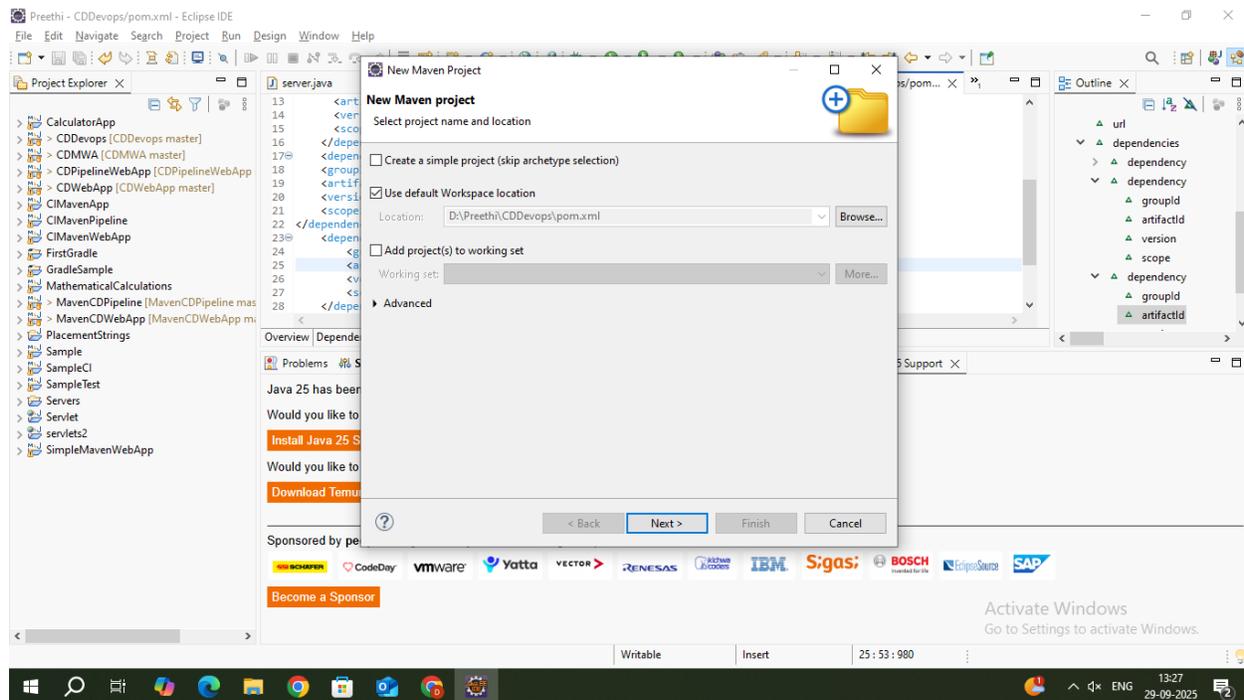
**UNIT III CONTINUOUS INTEGRATION USING JENKINS****6**

Install & Configure Jenkins, Jenkins Architecture Overview, Creating a Jenkins Job, Configuring a Jenkins job, Introduction to Plugins, Adding Plugins to Jenkins, Commonly used plugins (Git Plugin, Parameter Plugin, HTML Publisher, Copy Artifact and Extended choice parameters). Configuring Jenkins to work with java, Git and Maven, Creating a Jenkins Build and Jenkins workspace.

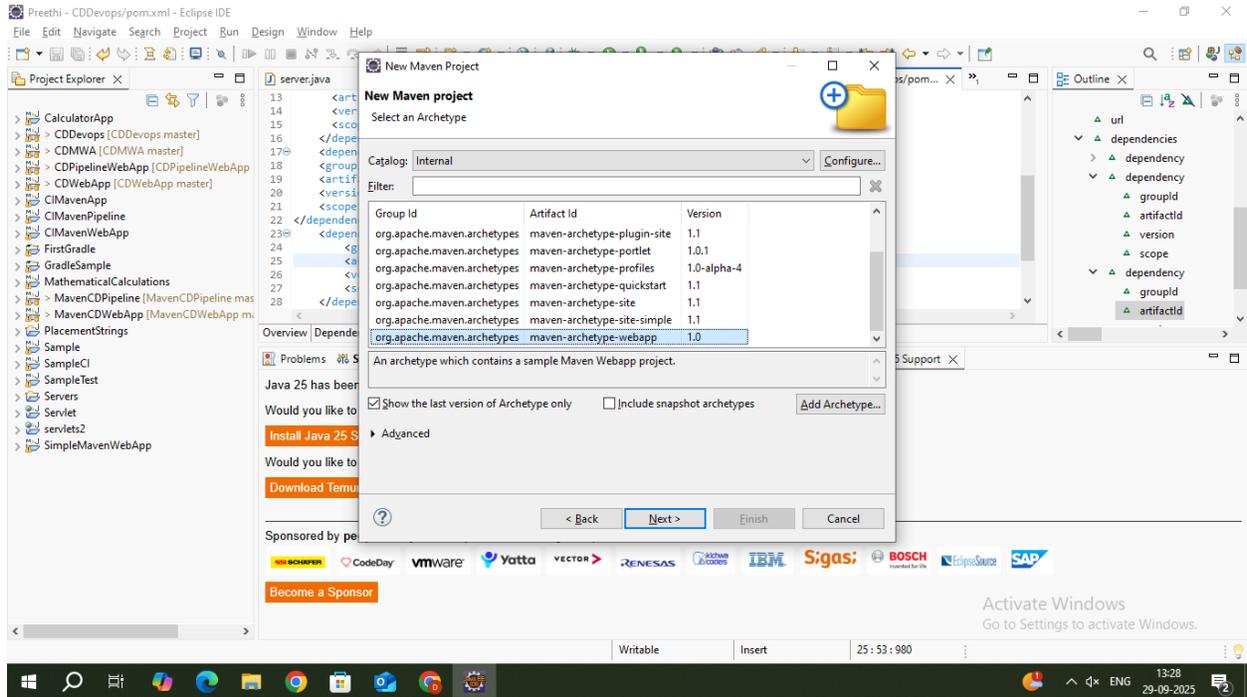
**CONFIGURING JENKINS TO WORK WITH MAVEN, JAVA AND GITHUB****Stage 1: Creating Maven Web Application**

**Step 1:** Create a Maven Web Application by selecting File→ New → Maven Project

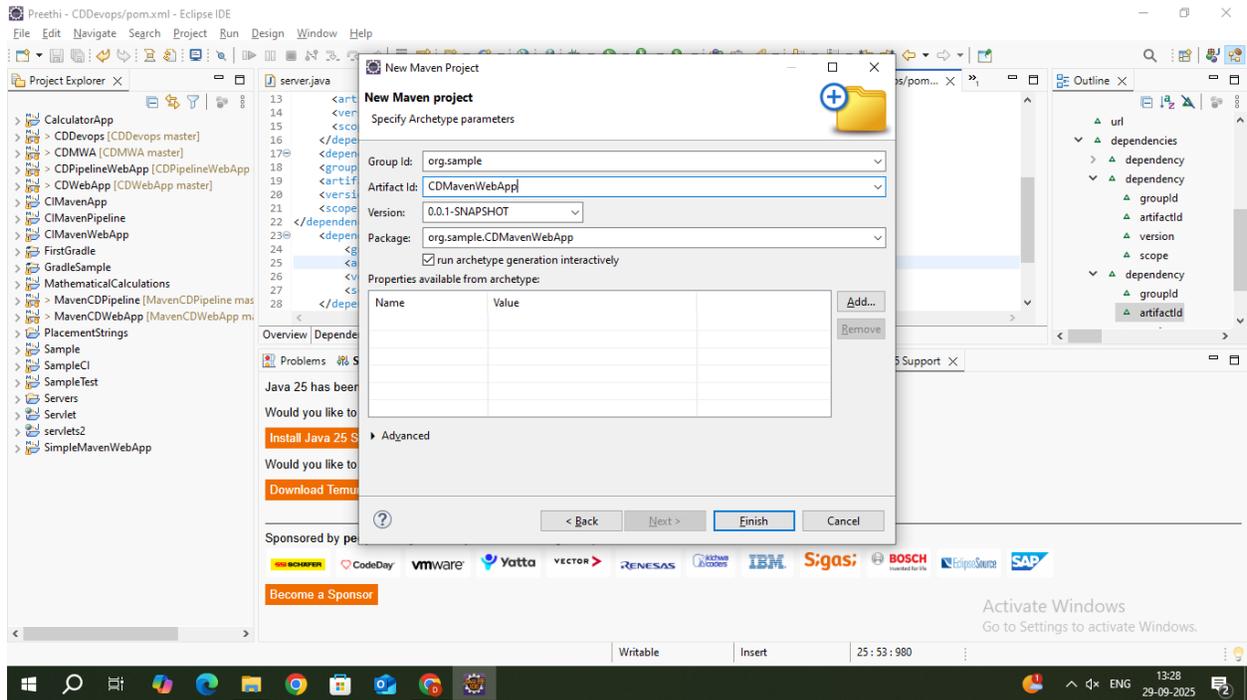
**Step 2:** Click Next (Do not select Create a Simple Application)



**Step 3:** Choose Internal in Catalog and select the Artifactid maven-archetype-webapps in the org.apache.maven.archetypes and click next.

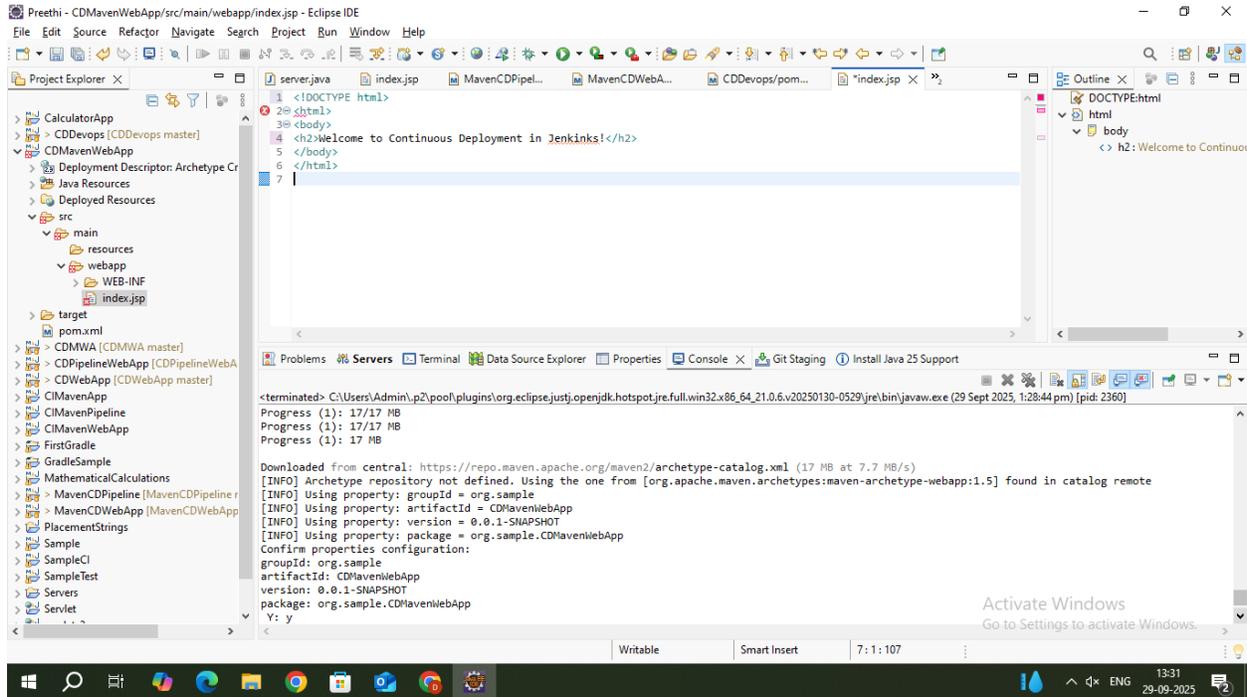


**Step 4:** Give the groupId and artifactId for the project and click Finish



**Step 5:** Open the index.jsp file

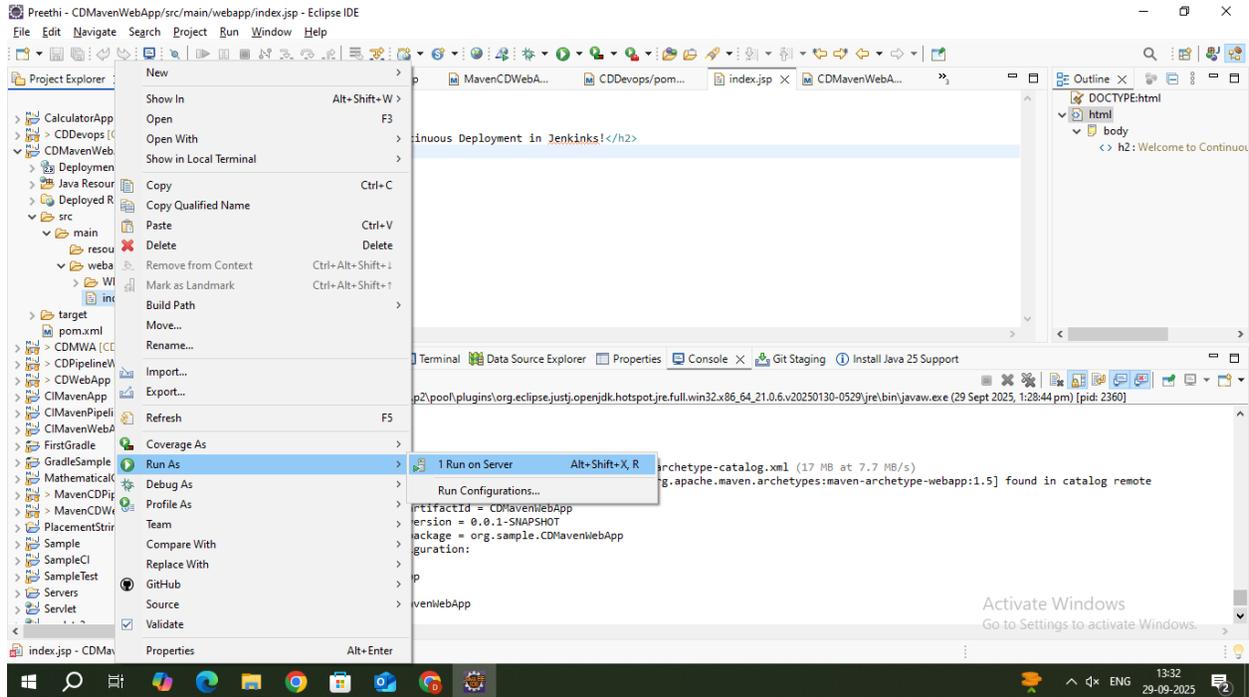
**Step 6:** Update the file for the application



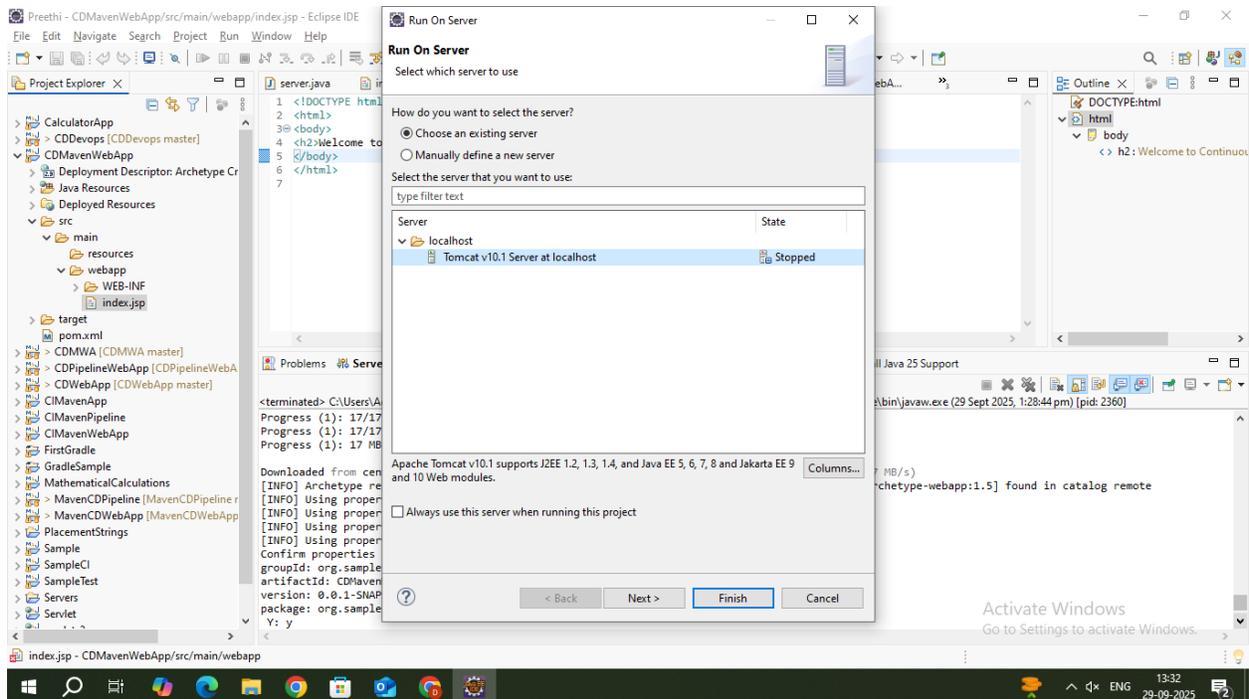
**Step 7:** Add the following dependencies in pom.xml file

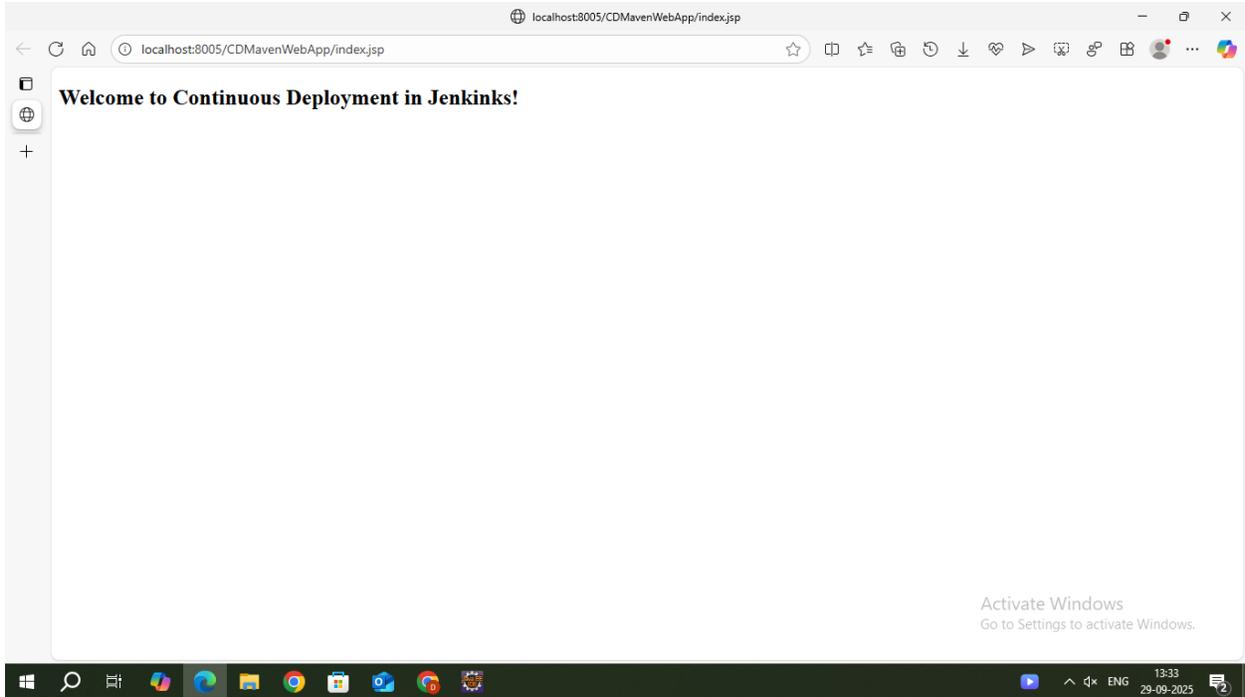
```
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>javax.servlet-api</artifactId>
  <version>4.0.1</version>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>jakarta.servlet</groupId>
  <artifactId>jakarta.servlet-api</artifactId>
  <version>6.0.0</version>
  <scope>provided</scope>
</dependency>
```

**Step 8:** Right click inex.jsp file and click as run on server



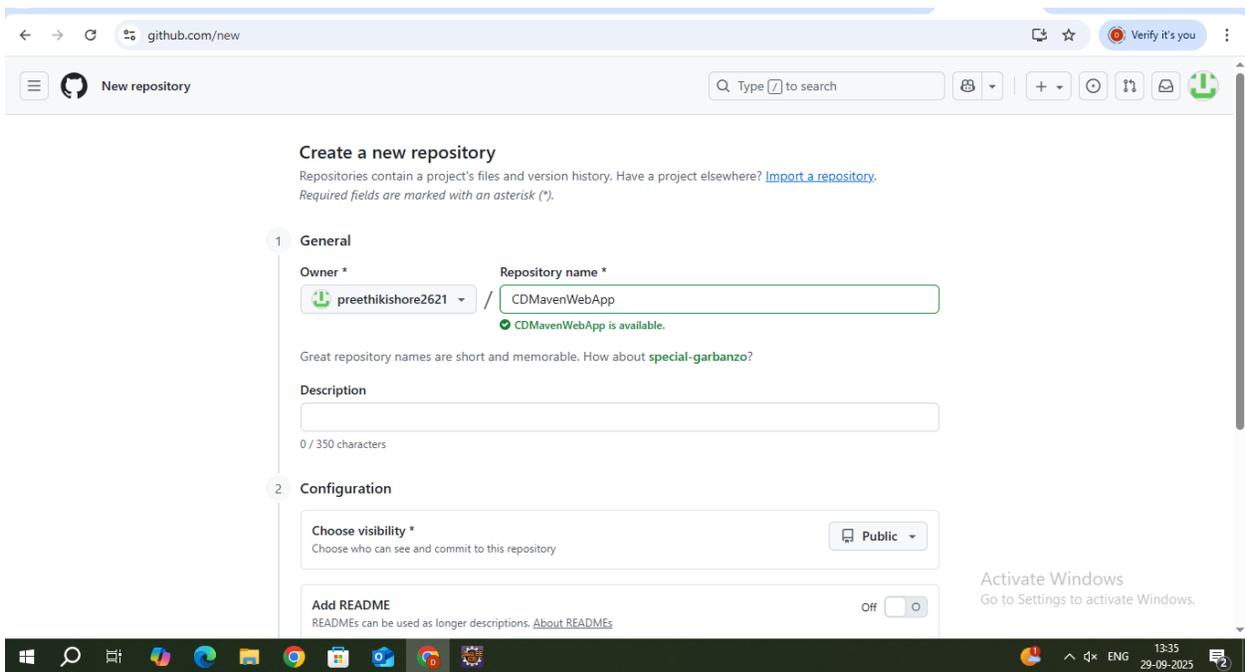
**Step 9:** Choose the server Apache Tomcat 10.1 and click Finish. The application is deployed on tomcat server



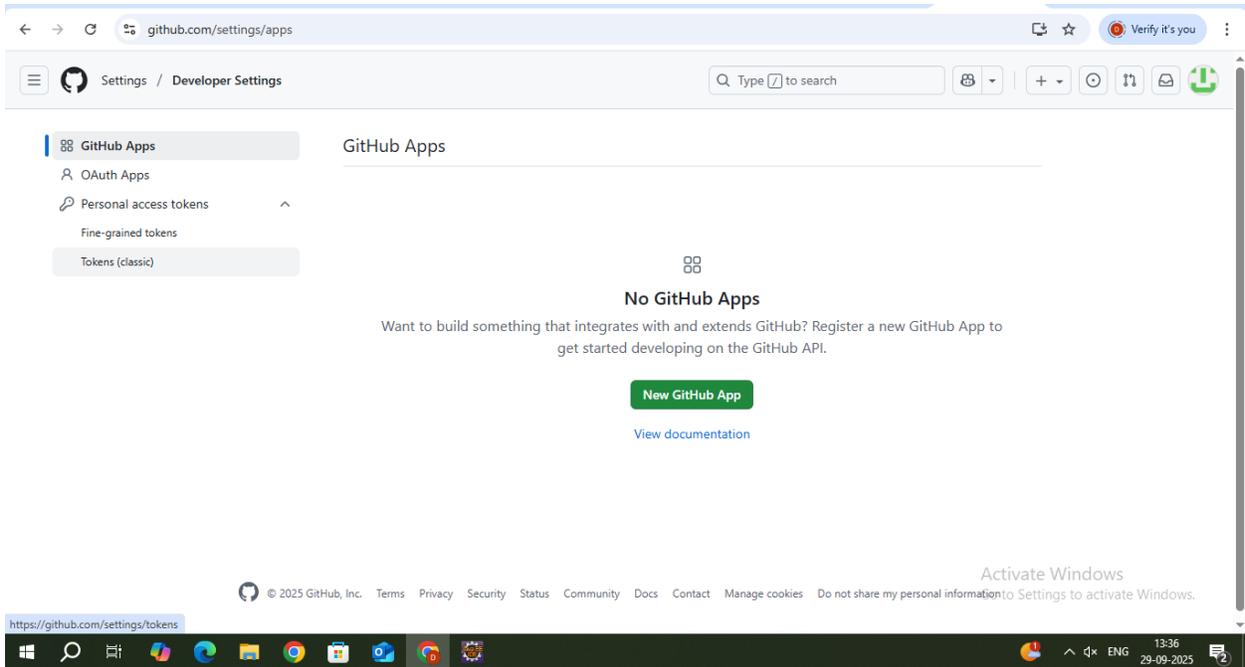


## Stage 2: Pushing the Web application to github

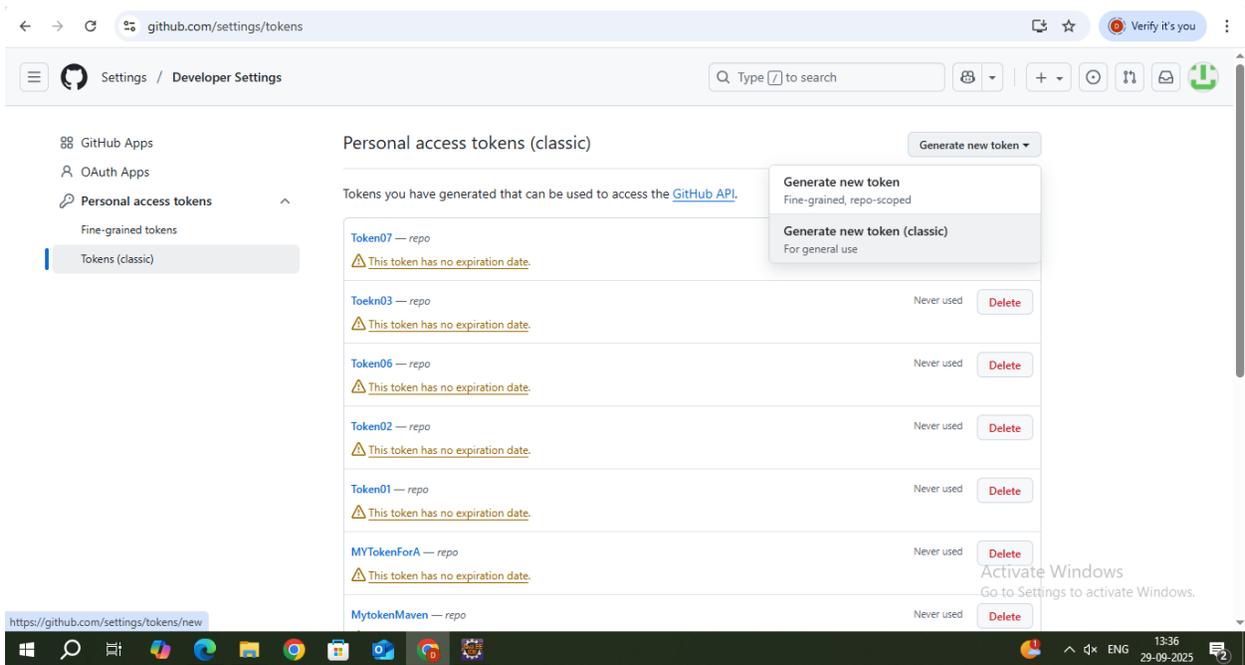
**Step 1:** Open Github and create a repository in the name same as the artifactId of maven project



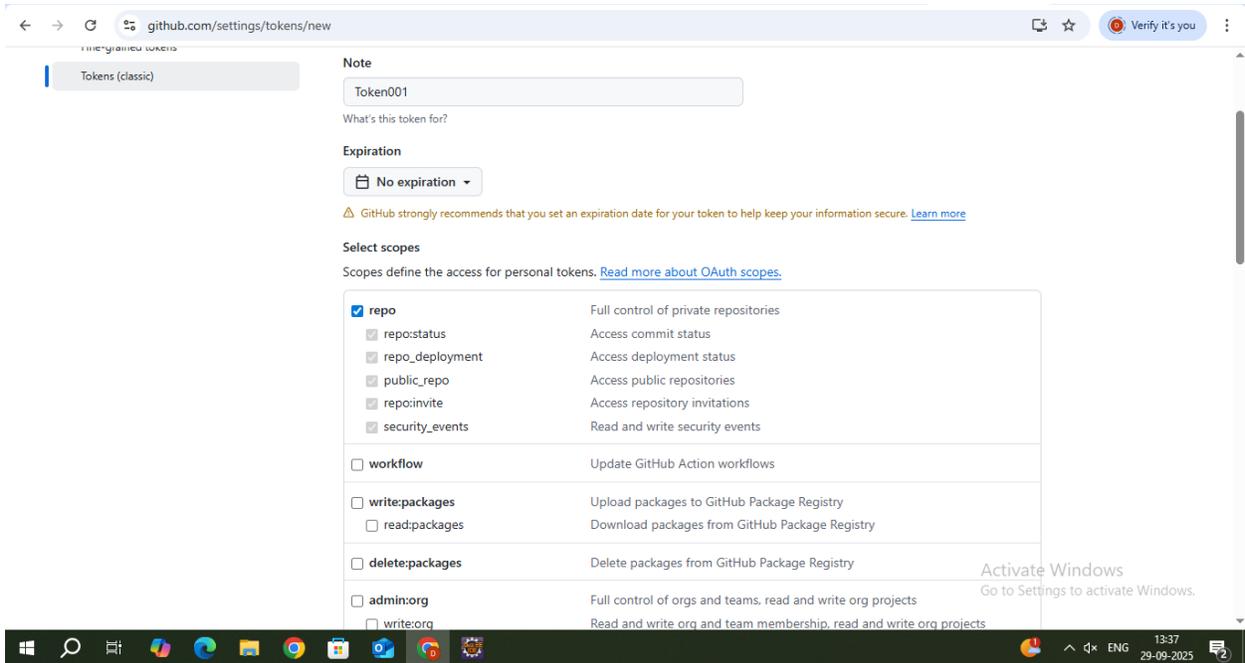
**Step 2:** Open the account settings and move on to developer settings and choose Personalized tokens(Classic)



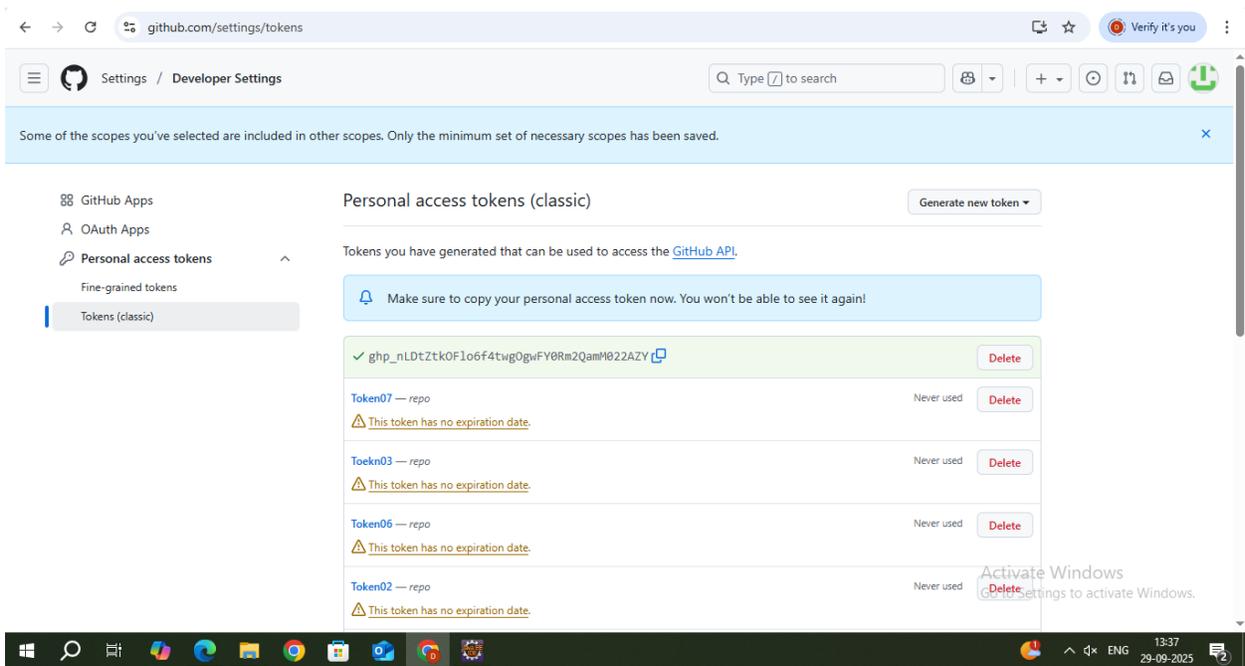
**Step 3: Click on generate token and choose token classic**



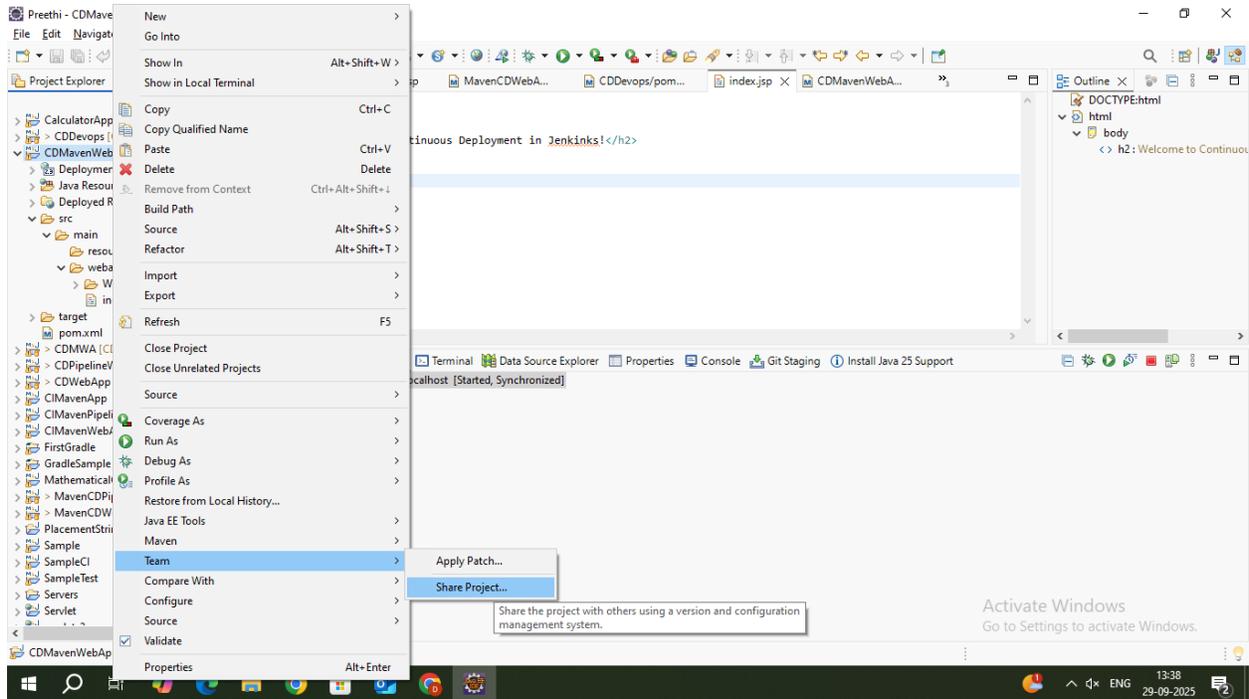
**Step 4: Provide a suitable name for the token and choose the expiration as no expiration and provide accessible privileges for the repository from the third party by selecting repo checkbox and click generate token**



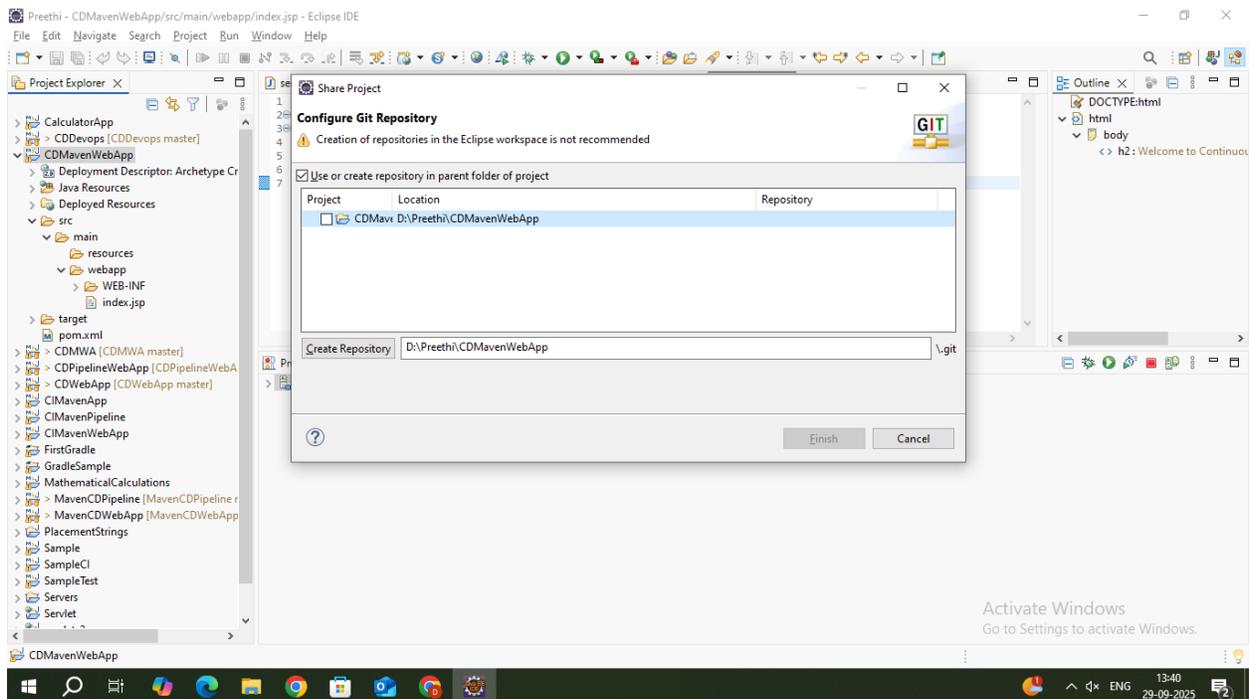
**Step 5:** Copy the token password for further use.

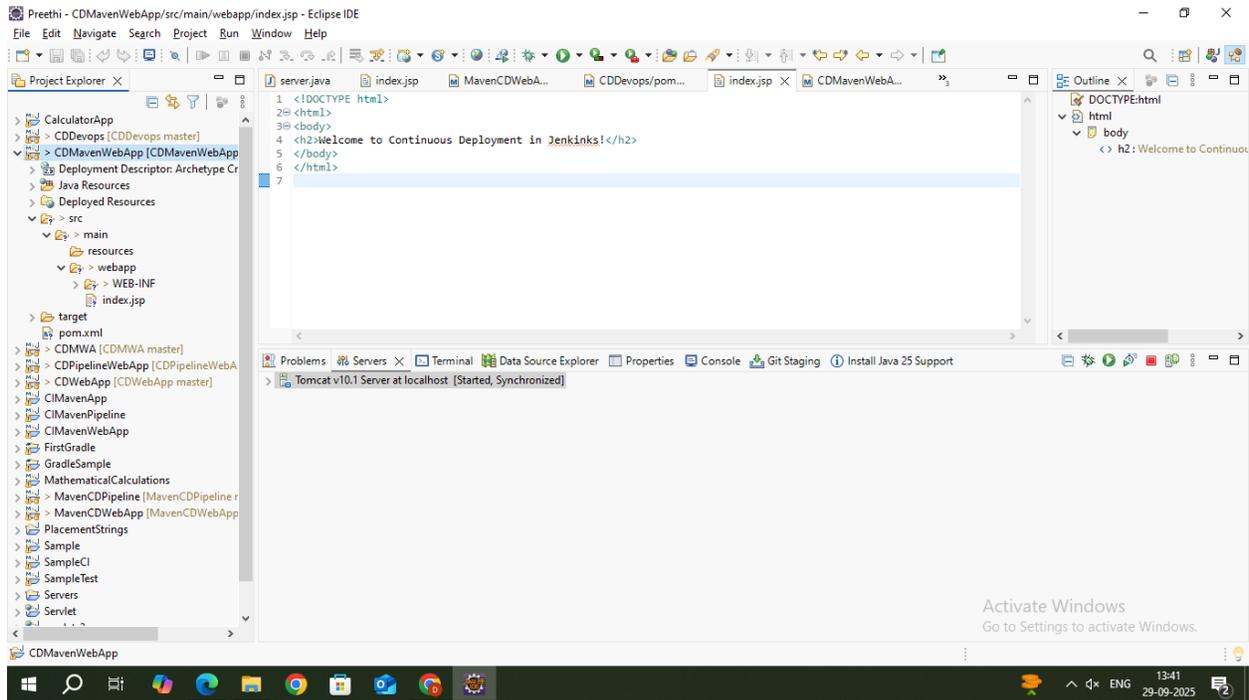
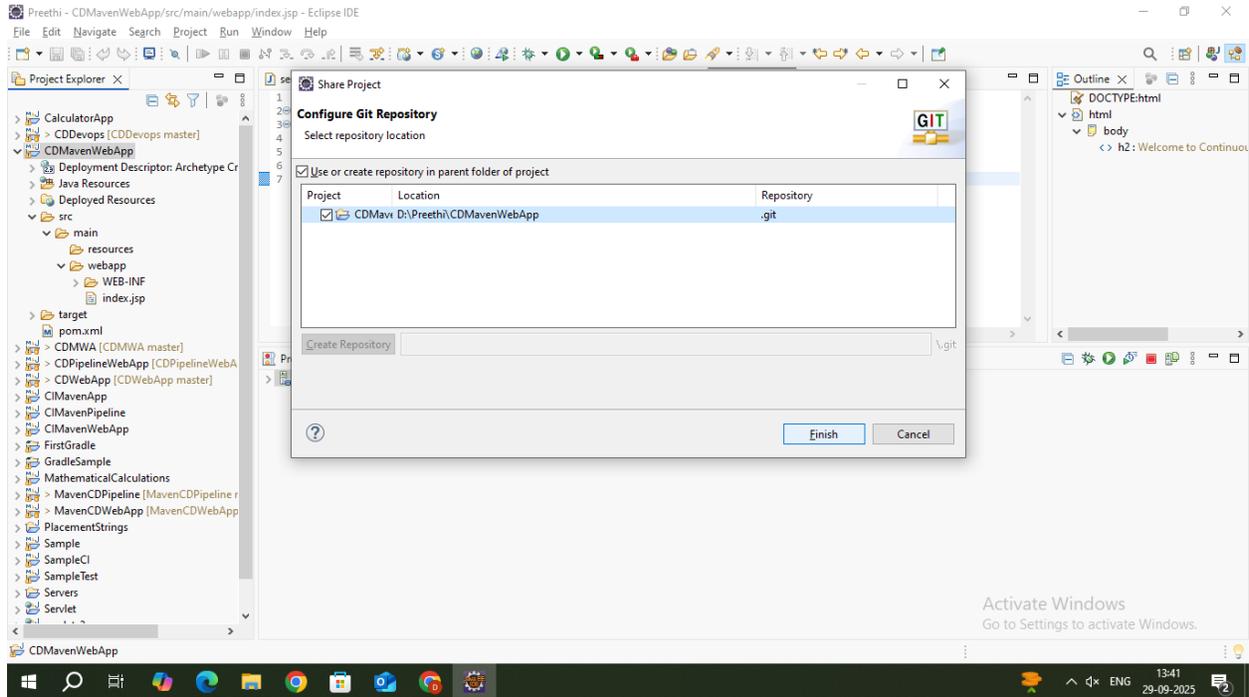


**Step 6:** Open Eclipse IDE and right click the project and select Team → Share project

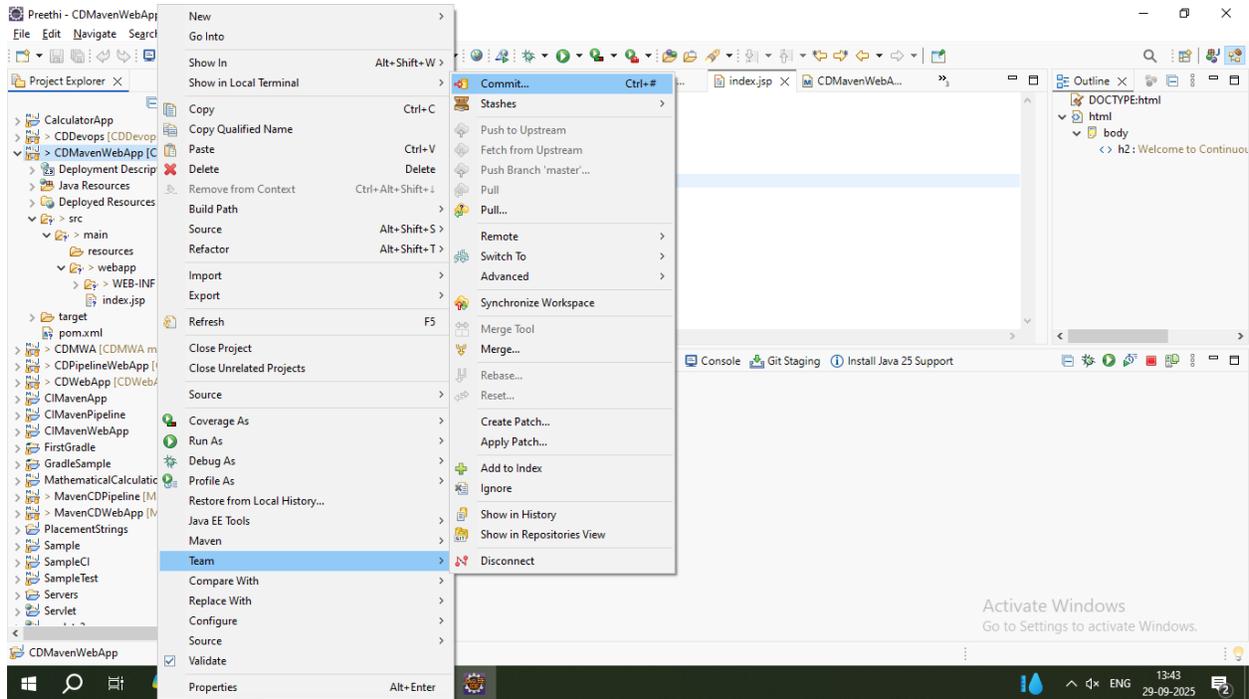


**Step 7:** Configure the Git repository by clicking on the check box for Use or create repository in parent folder of project and choose our project and click create repository to make as git and click Finish to create the master branch

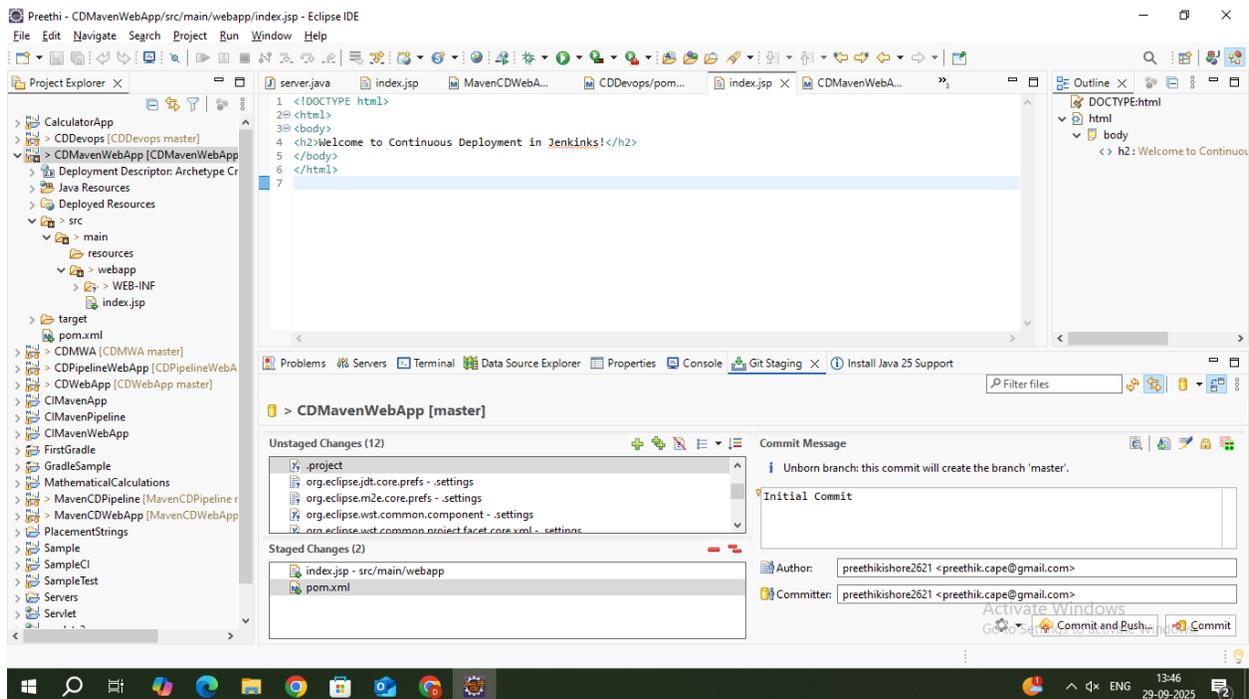




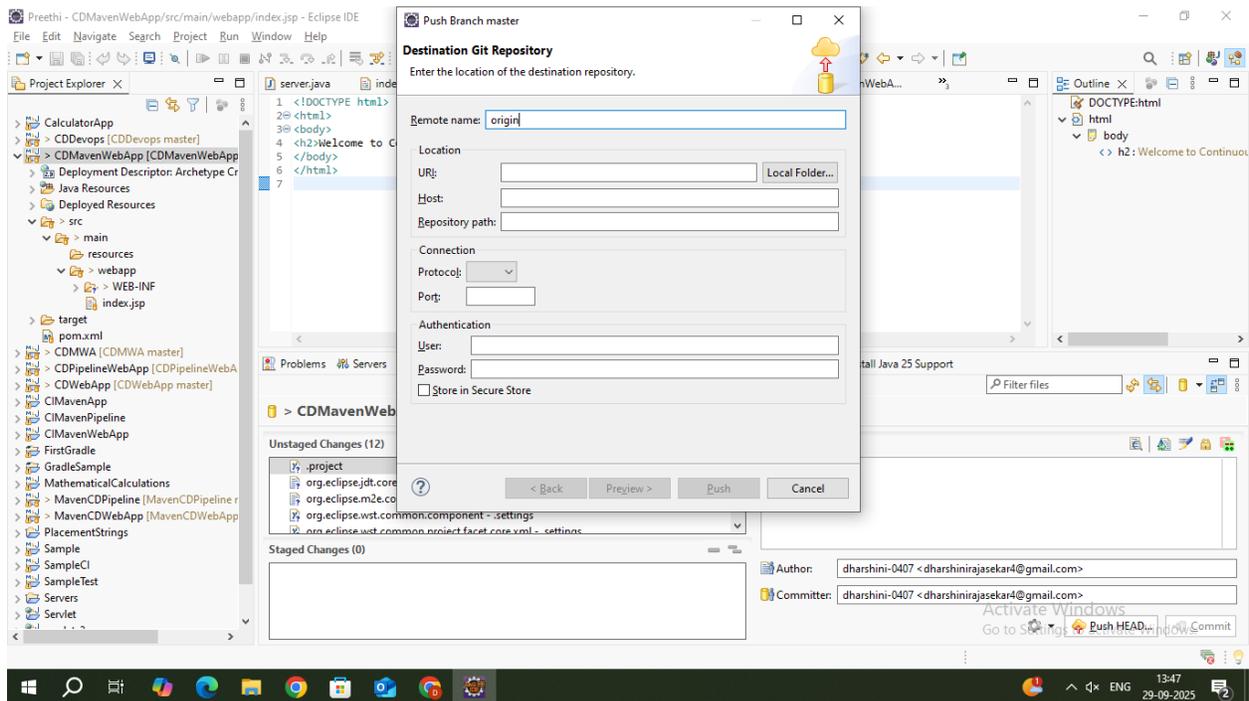
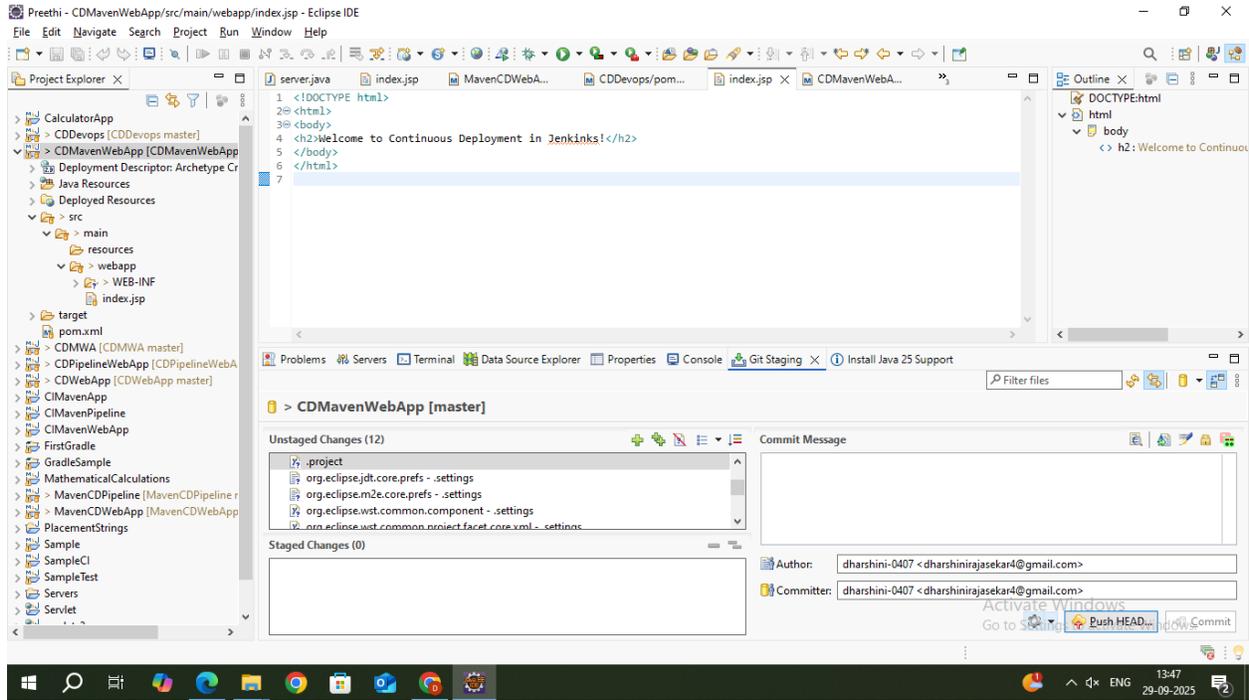
**Step 8:** Right click the project again and choose Team → Commit



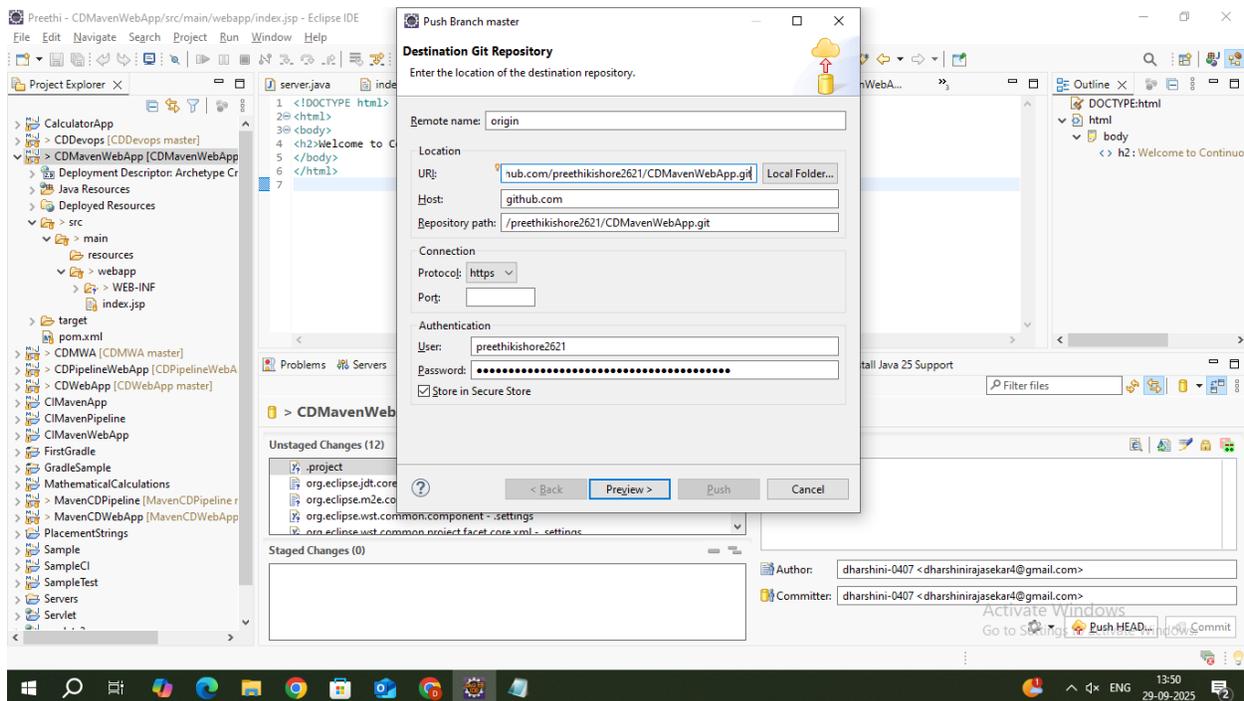
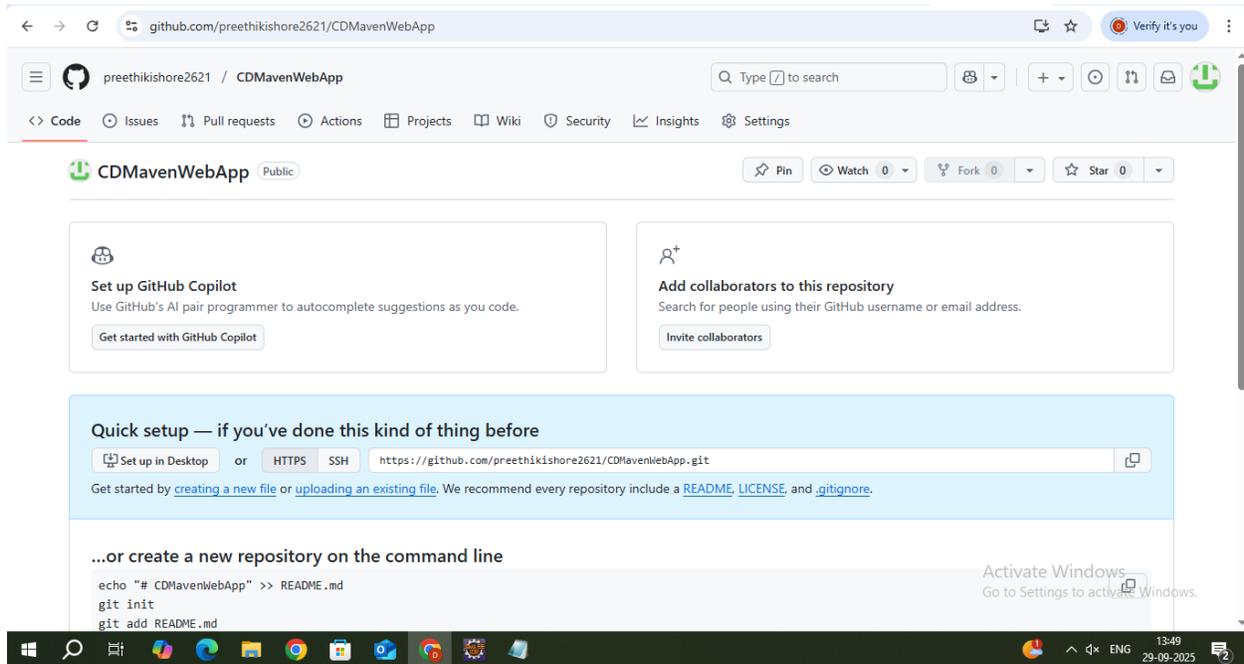
**Step 9:** Drag the files index.jsp and pom.xml from the Unstage changes to Staged Changes and Enter a commit message and provide the github account details of Author and the Committer and click Commit and Push

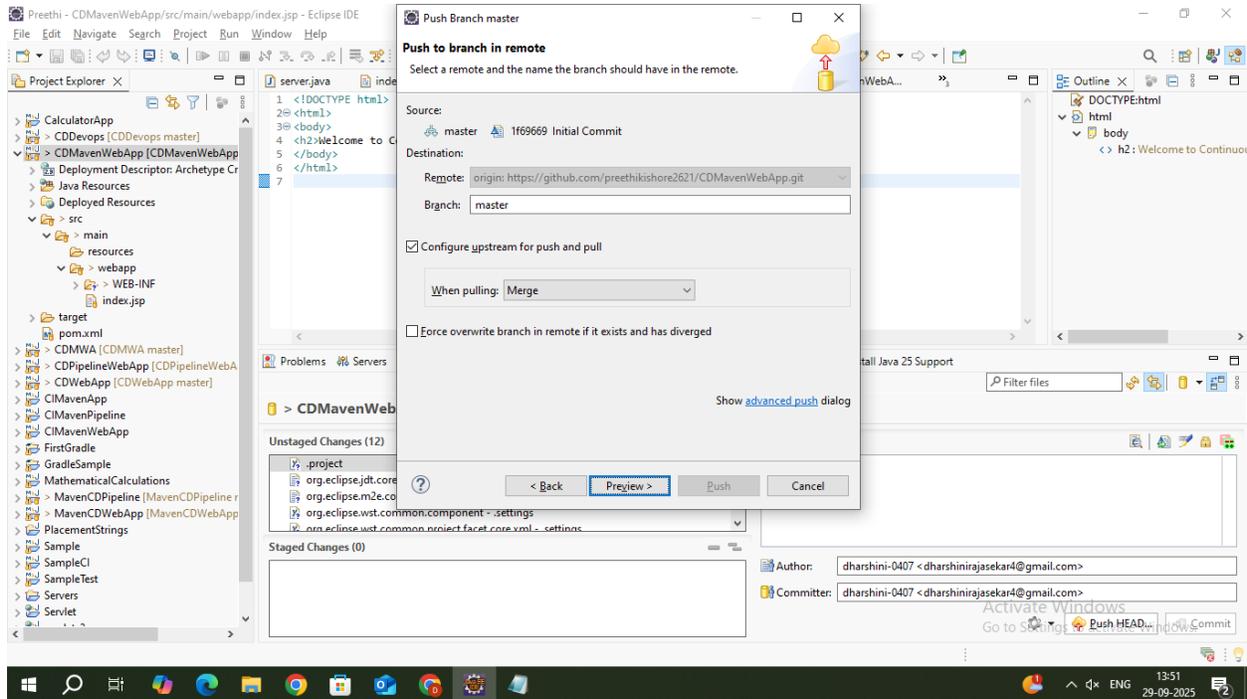


**Step 10:** Now click Push Head to configure the destination git repository

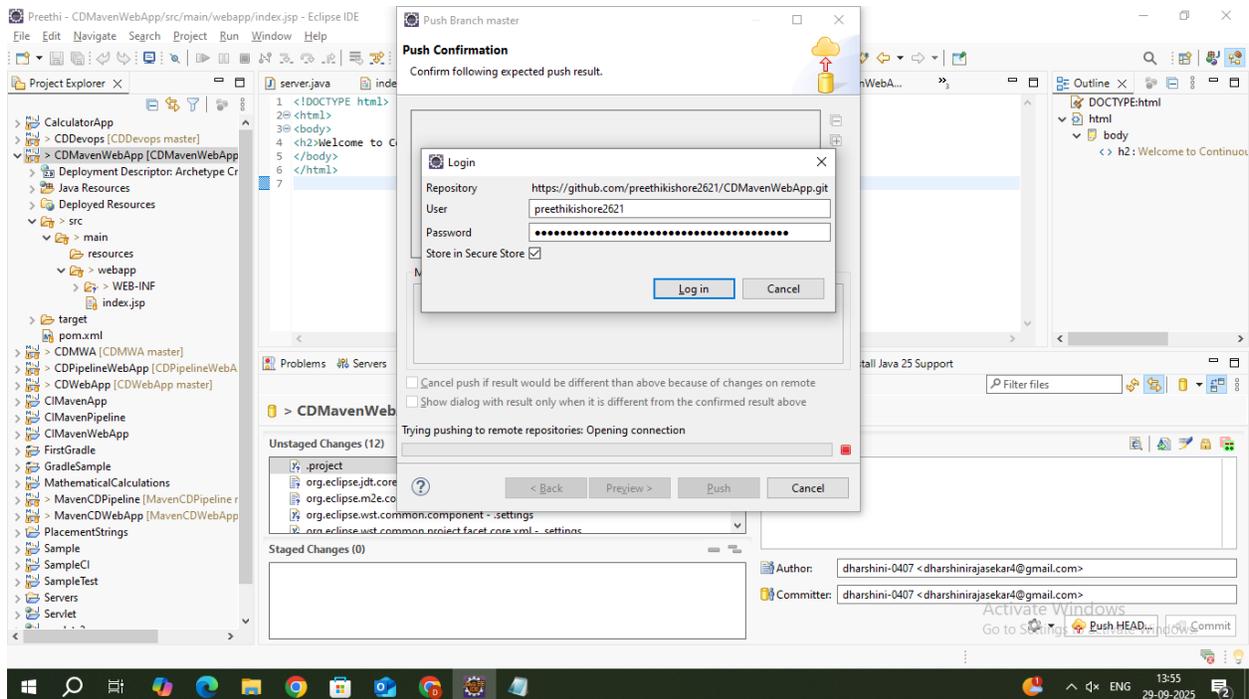
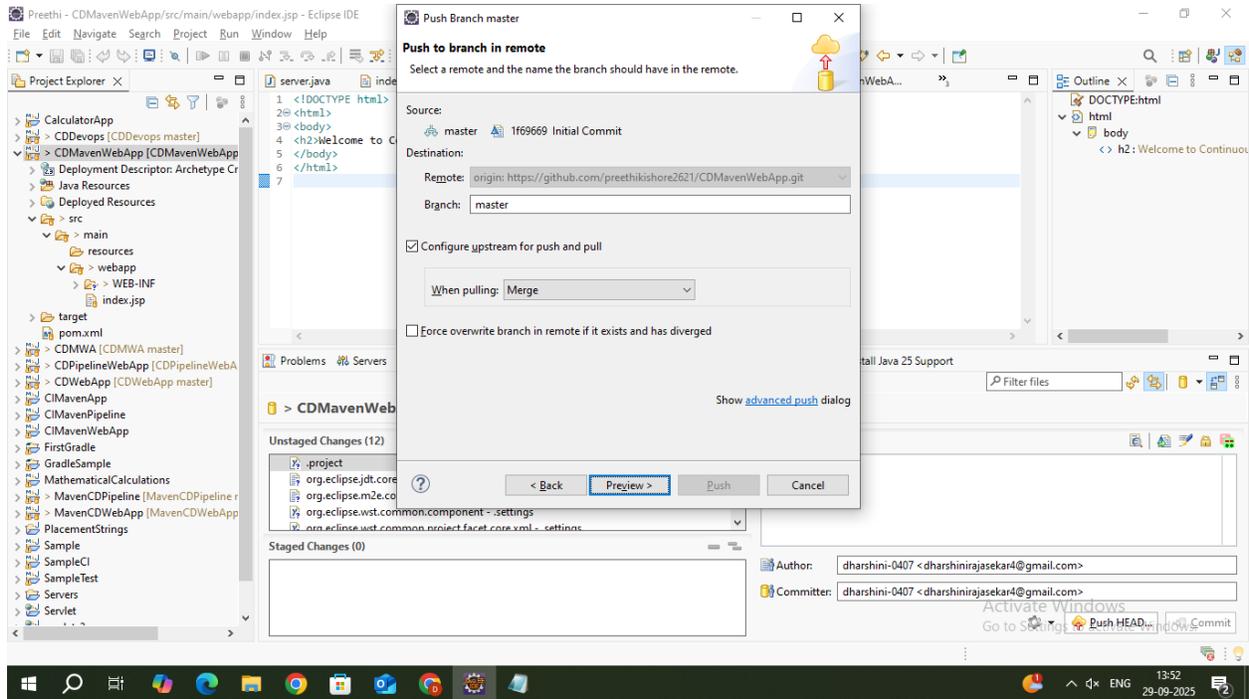


**Step 11:** Copy the url of the github repository and paste it in the url and provide the authentication credentials and click preview

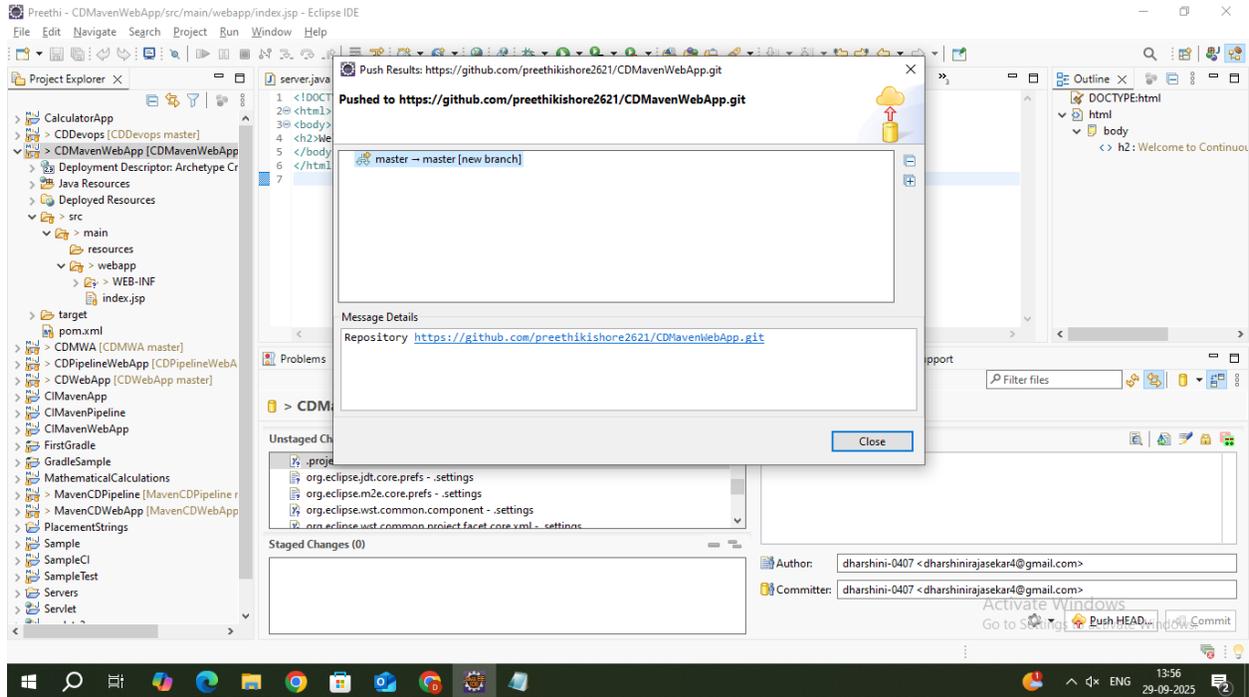


**Step 12: Push the branch to remote by selecting master and click preview**

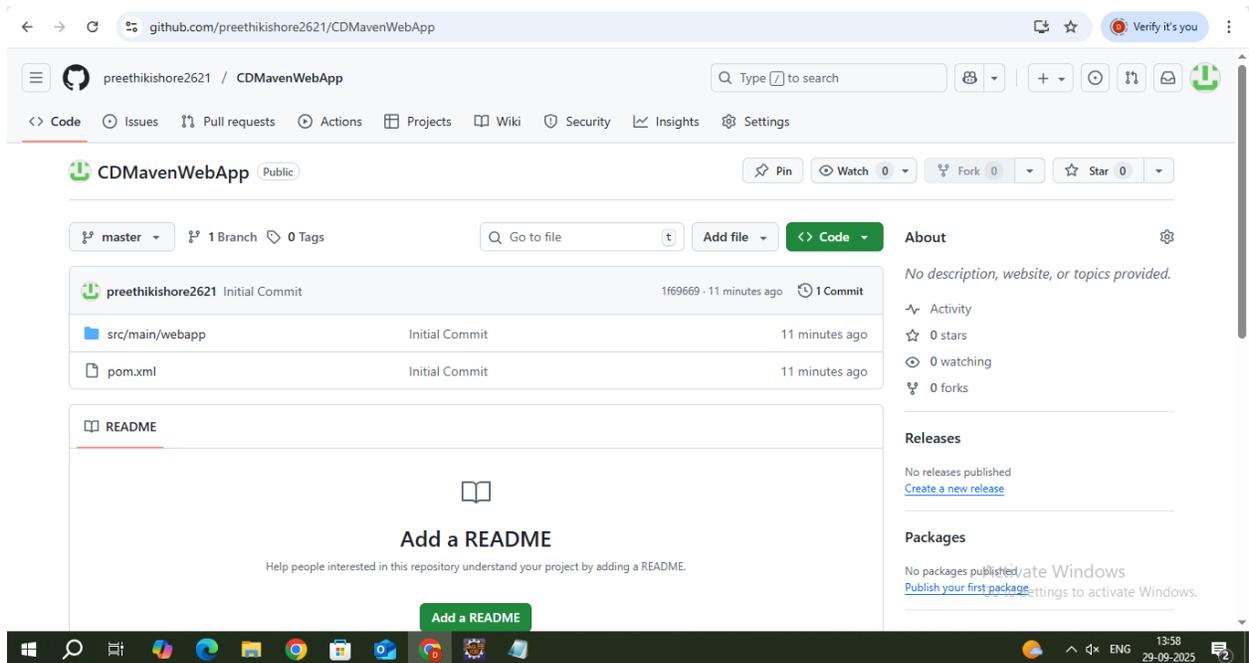
**Step 13: Push the branch to remote by selecting master and click preview the user of github account and provide the token password and check the Store in Secure Store and finish by login**



**Step 14:** Click push and again provide the login credentials same as previous step and close the dialog box



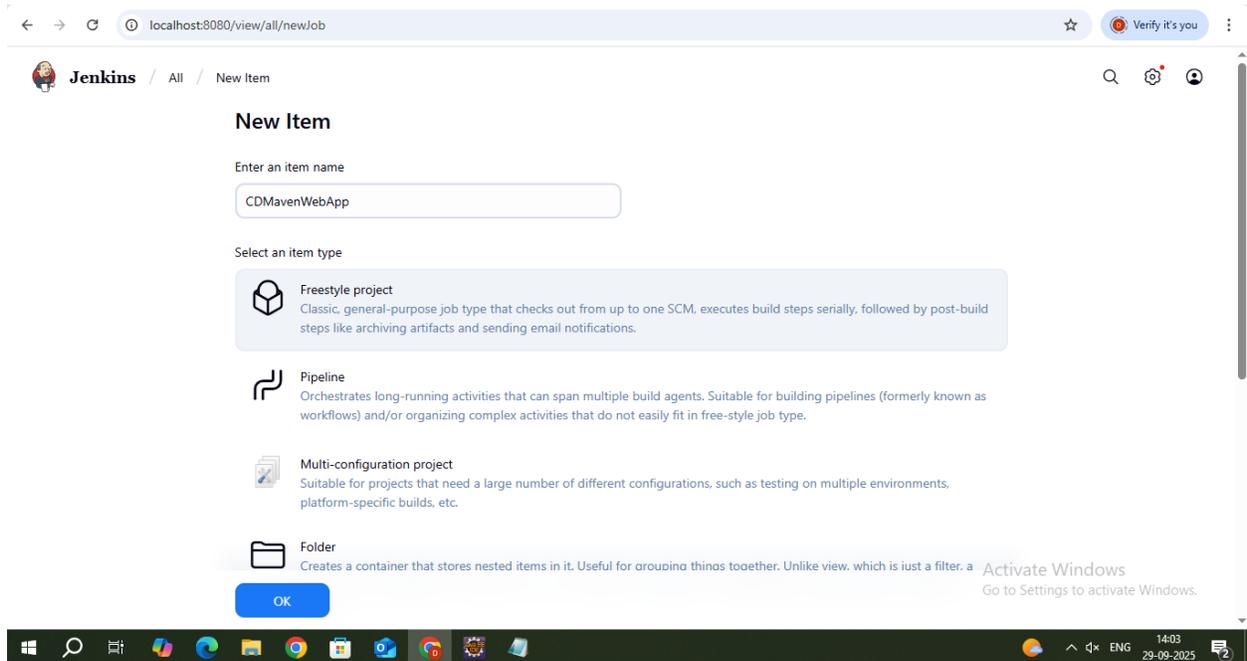
**Step 15: Open github repository and refresh to see the changes**



## Stage 3: Configure the Application in github and build in Jenkins

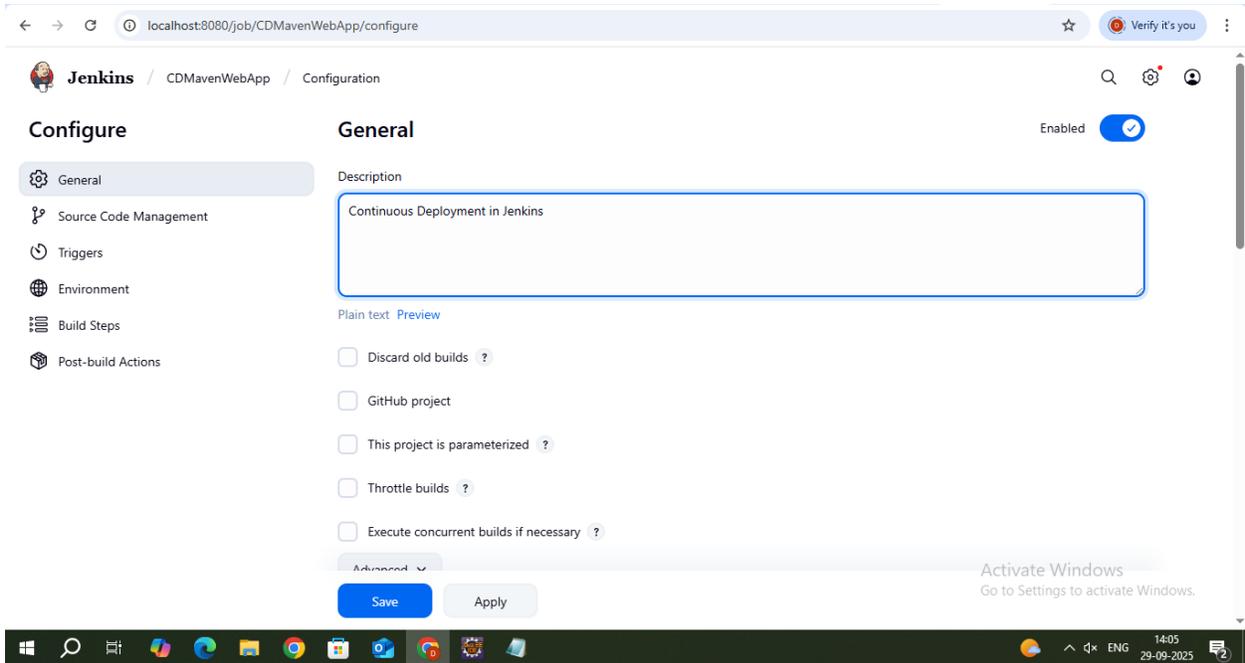
**Step 1:** Open jenkins service

**Step 2:** Create a new free style project by selecting New Item and give the project name as same as artifactId

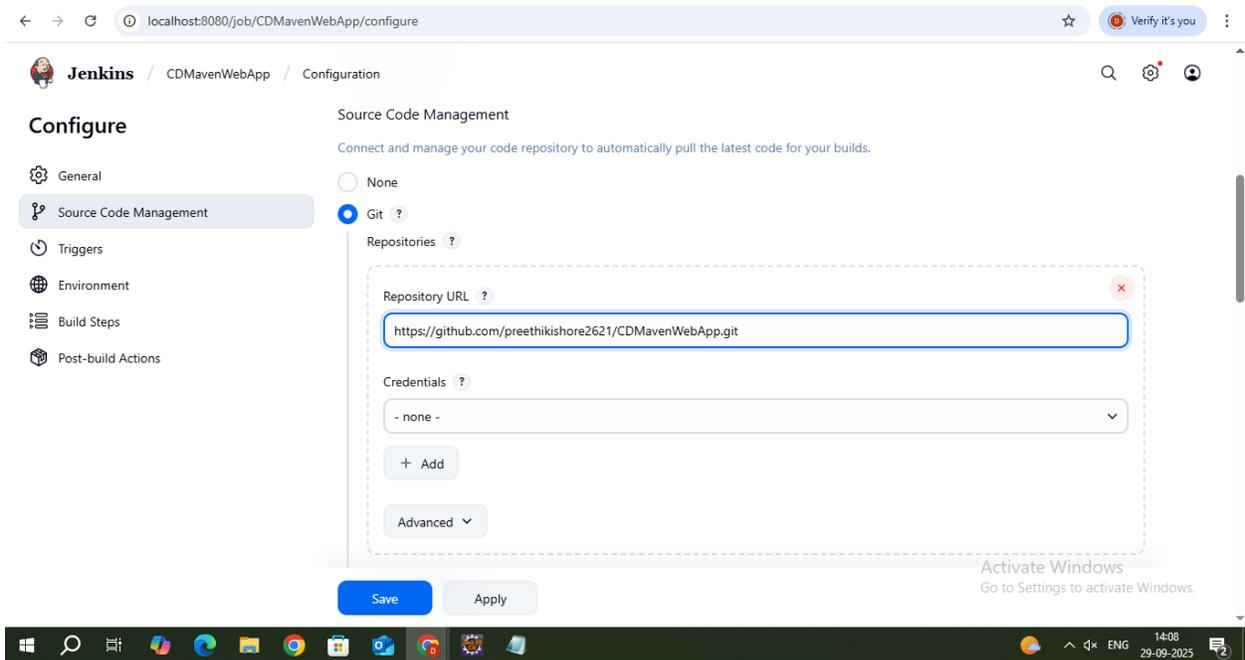


**Step 3:** Configure the project as follows

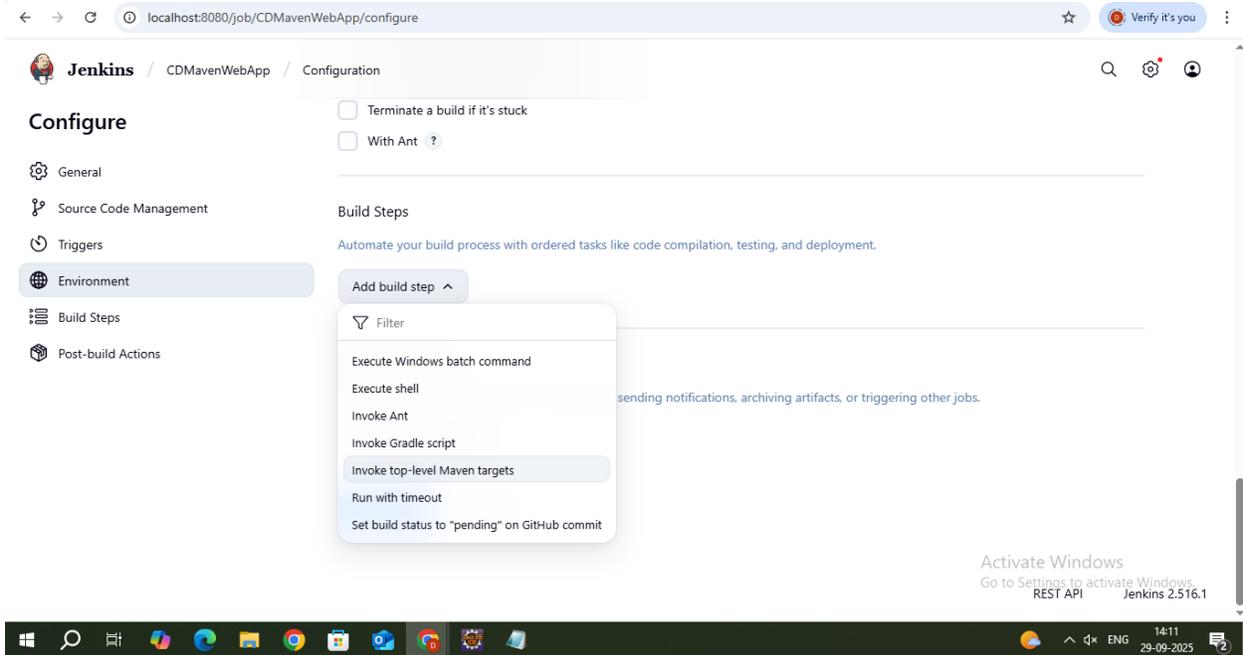
1. Type the suitable Description



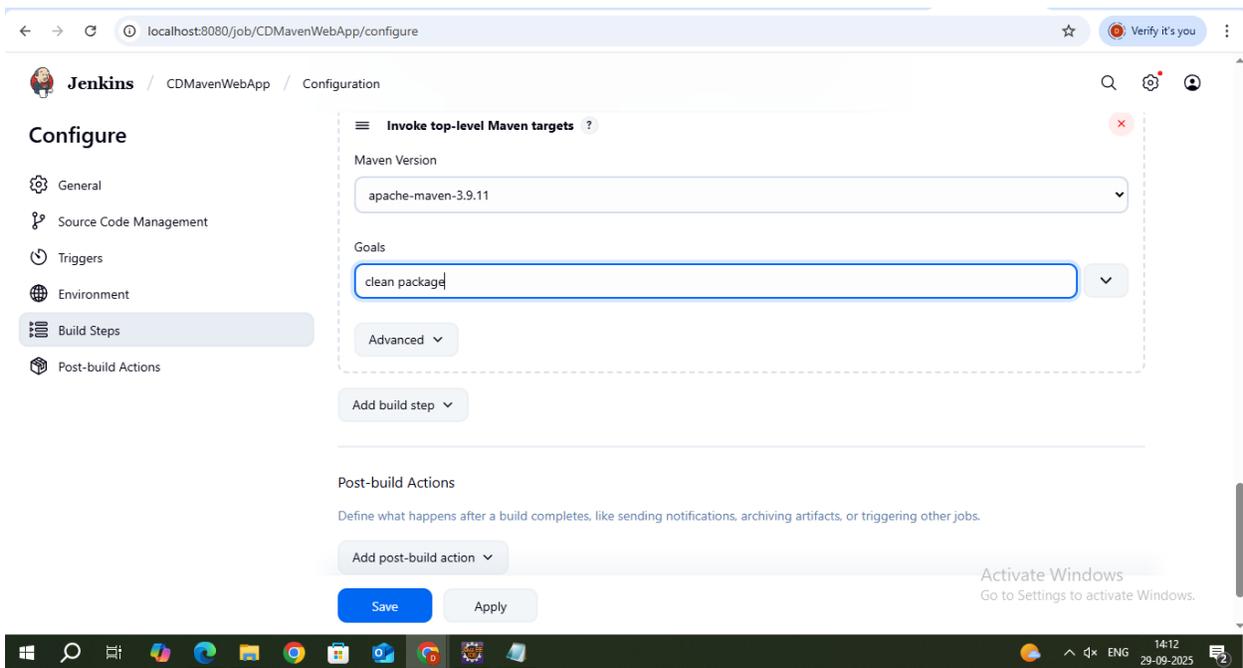
2. Choose the Source Code Management as Git and paste the url of repository and select the master as branch



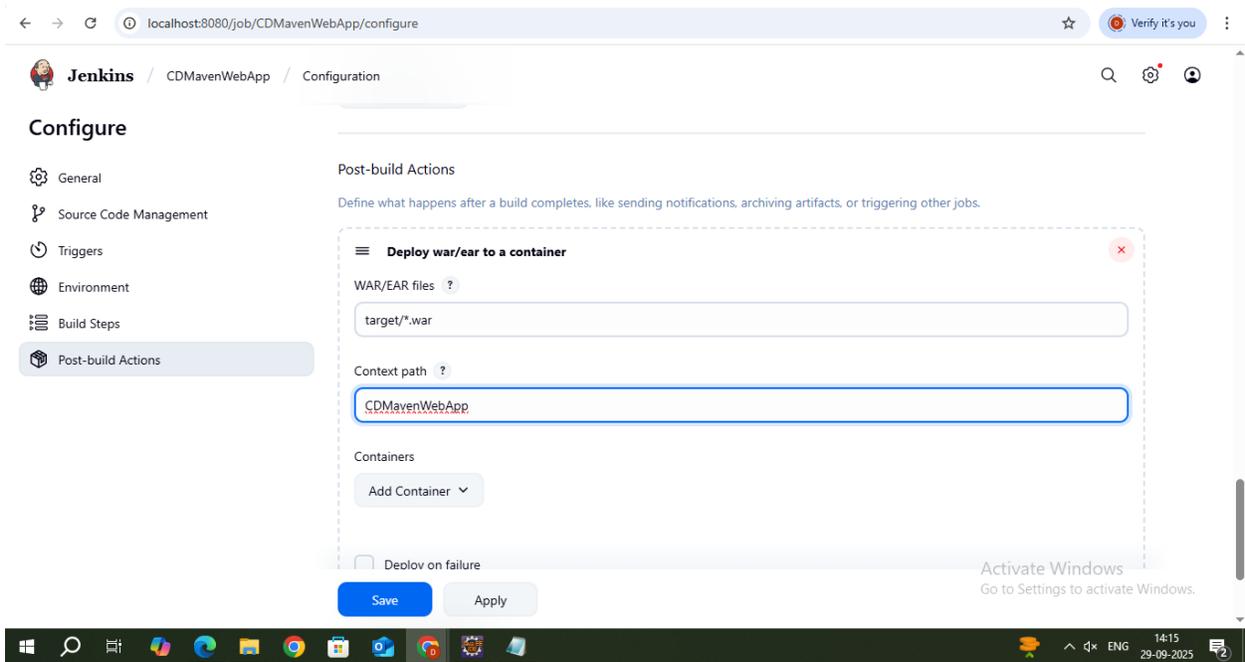
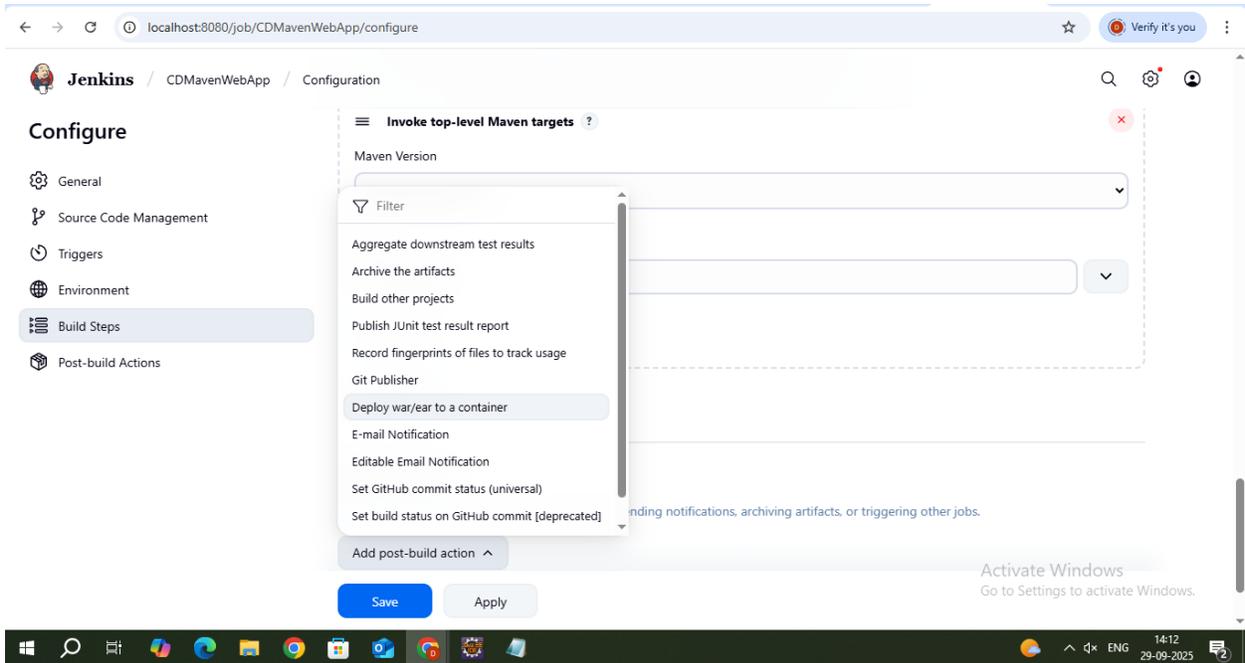
3. In Add Build Steps Choose Invoke top-level Maven targets



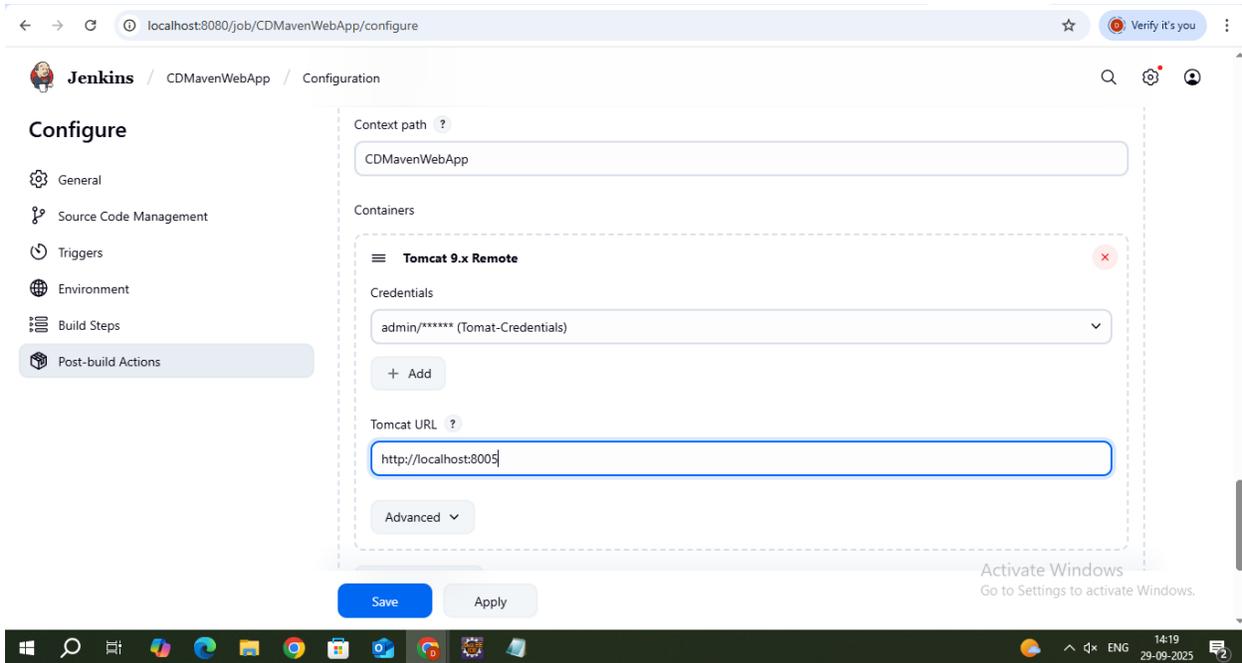
4. Select the maven version and goals as clean and package



5. In post build steps choose Deploy war/ear to a container and give the information about the war file as target/\*.\*war and the ContextPath as CDMavenWebApp

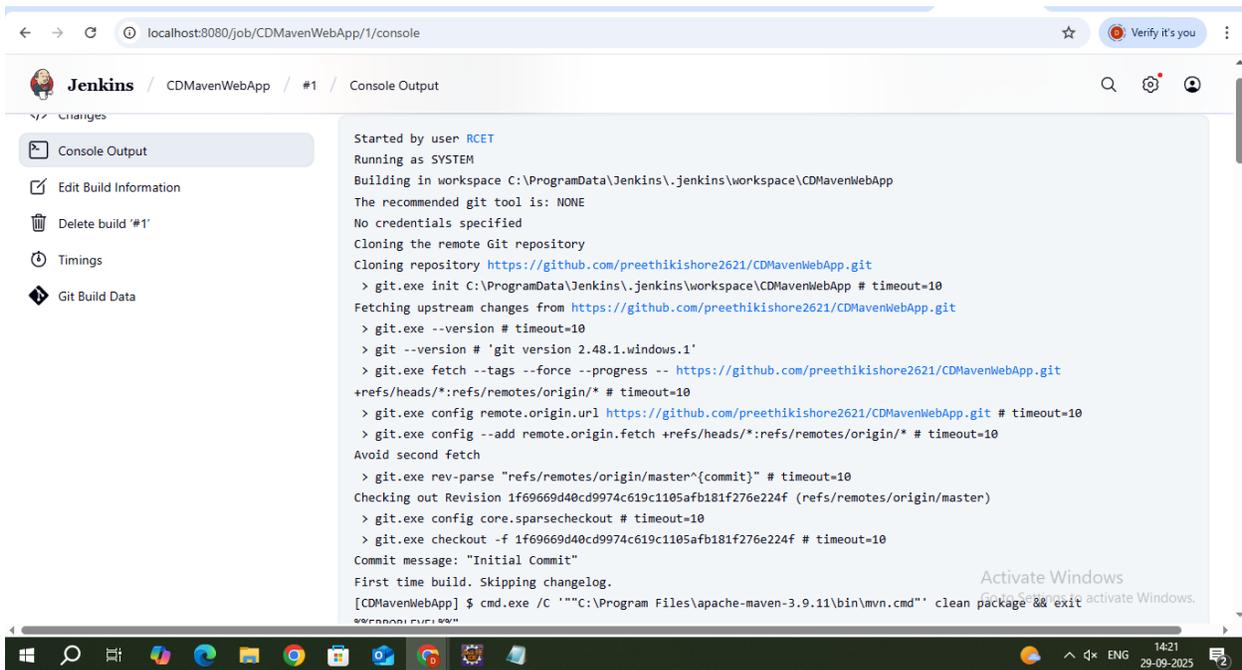


6. Add the container as select the server Tomcat 9.0 and add the credentials of Tomcat and provide the url of Tomcat as <http://localhost:8005>



7. Click Apply and Save

8. Click Build Now for instant build



```

#REPROLEVEL#
[INFO] Scanning for projects...
[INFO]
[INFO] -----< org.sample:CDMavenWebApp >-----
[INFO] Building CDMavenWebApp Maven Webapp 0.0.1-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ war ]-----
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ CDMavenWebApp ---
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ CDMavenWebApp ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\ProgramData\Jenkins\.jenkins\workspace\CDMavenWebApp\src\main\resources
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ CDMavenWebApp ---
[INFO] No sources to compile
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ CDMavenWebApp ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\ProgramData\Jenkins\.jenkins\workspace\CDMavenWebApp\src\test\resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ CDMavenWebApp ---
[INFO] No sources to compile
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ CDMavenWebApp ---

```

```

[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ CDMavenWebApp ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\ProgramData\Jenkins\.jenkins\workspace\CDMavenWebApp\src\test\resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ CDMavenWebApp ---
[INFO] No sources to compile
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ CDMavenWebApp ---
[INFO] No tests to run.
[INFO]
[INFO] --- war:3.4.0:war (default-war) @ CDMavenWebApp ---
[INFO] Packaging webapp
[INFO] Assembling webapp [CDMavenWebApp] in
[C:\ProgramData\Jenkins\.jenkins\workspace\CDMavenWebApp\target\CDMavenWebApp]
[INFO] Processing war project
[INFO] Copying webapp resources [C:\ProgramData\Jenkins\.jenkins\workspace\CDMavenWebApp\src\main\webapp]
[INFO] Building war: C:\ProgramData\Jenkins\.jenkins\workspace\CDMavenWebApp\target\CDMavenWebApp.war
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 6.956 s
[INFO] Finished at: 2025-09-29T14:21:17+05:30
[INFO] -----

```