

Name : Vedika Adheli

Roll No : 01

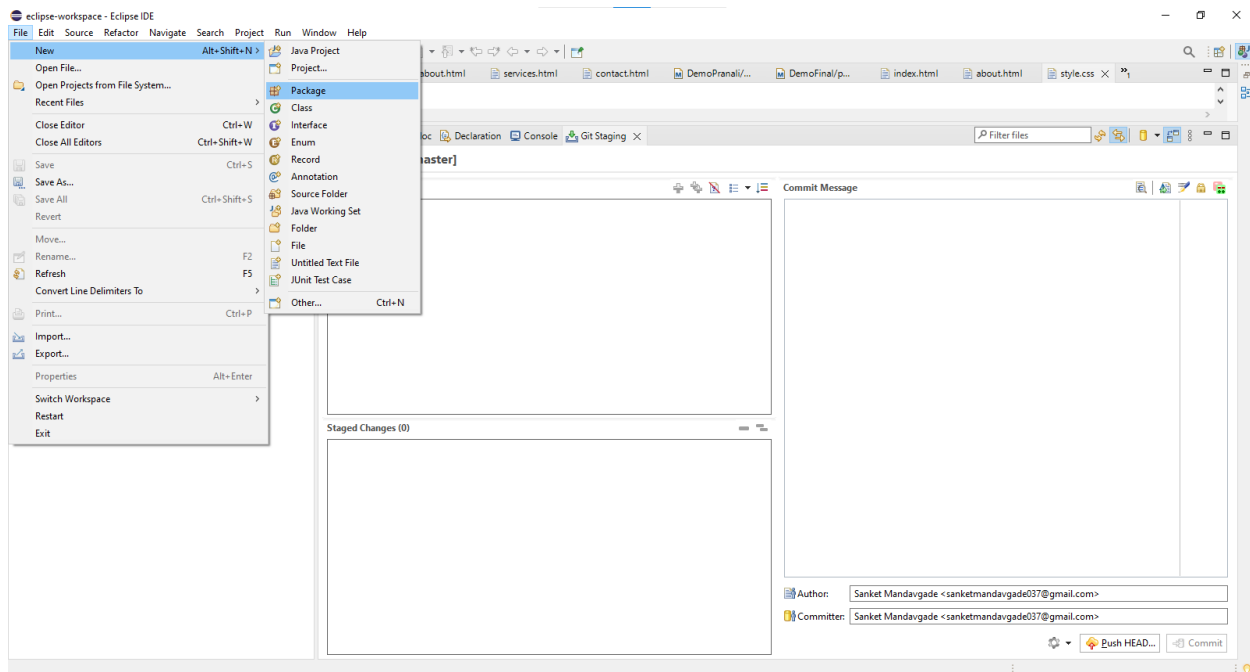
Devops Assignment : 06

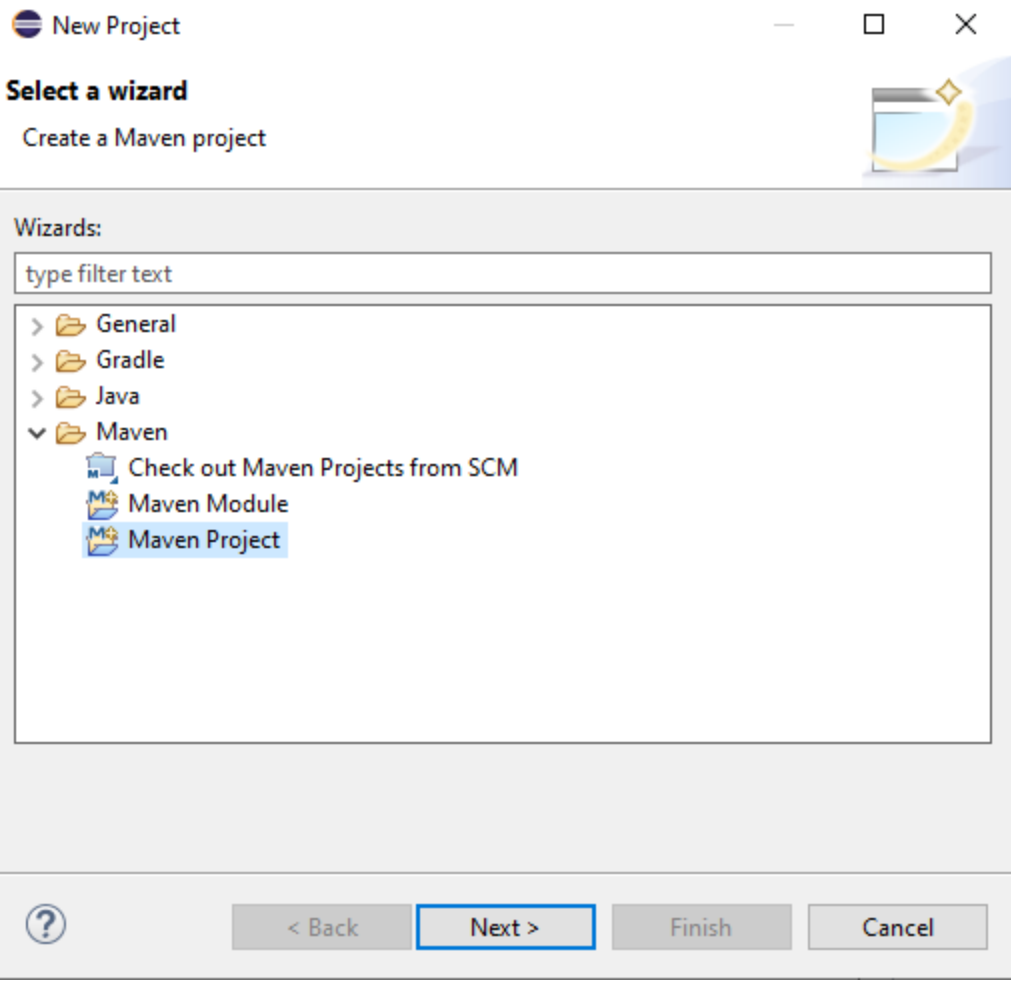
Create a web application & Commit changes to GitHub/GitLab repository. Use GitHub/GitLab, Maven, Tomcat with Jenkins. Automate Deployment of the above-mentioned application to a container using Jenkins plugin,"Deploy to container".

- Download Apache Maven stable version as zip and extract it into 'C:/program files' folder
- Download Apache Tomcat installer and install it,change the port to 8081 because jenkins is running on port 8080

To create a web application,

- 1.Open eclipse ide for java developers
- 2.Go to files,click on project
- 3.Click on Maven Project as shown below





New Maven Project

New Maven project
Select project name and location

☐ Create a simple project (skip archetype selection)

☒ Use default Workspace location

Location: Browse...

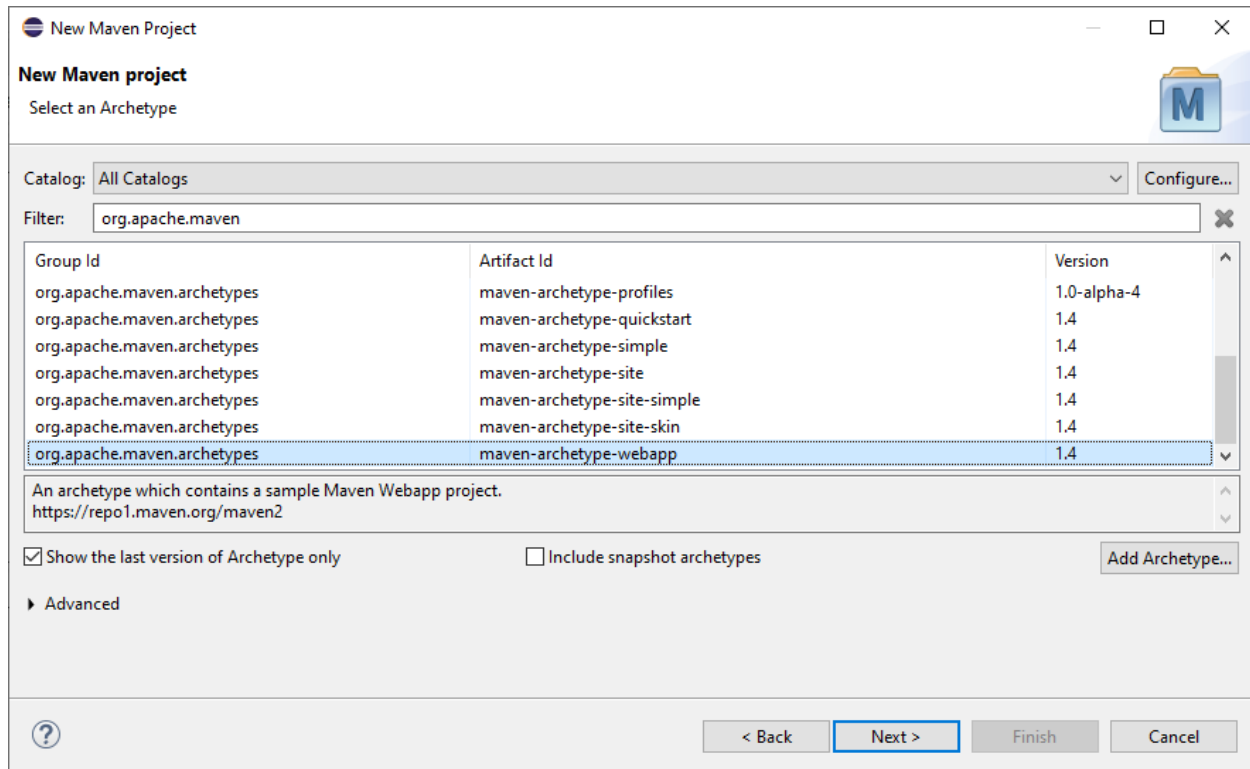
☐ Add project(s) to working set

Working set: More...

► Advanced

< Back Next > Finish Cancel

Use default workspace location and click on 'Next' You will find window as shown below In filter search filter as 'org.apache.maven.archetypes' select webapp option,version 1.4



Click on next

Given group id as Demo,artifact id as 'PranotiSample'

Click on finish

New Maven Project

New Maven project
Specify Archetype parameters

Group Id:

Artifact Id:

Version:

Package:

☒ run archetype generation interactively

Properties available from archetype:

Name	Value

► Advanced

You will find project building in console,
When it terminates click 'Y' and press Enter

```

eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help

Package Explorer X
  Demo2 [Demo2 master]
  DemoFinal [DemoFinal master]
  DemoPranali [DemoPranali master]
  ISE Project
  new1 [new1 master]

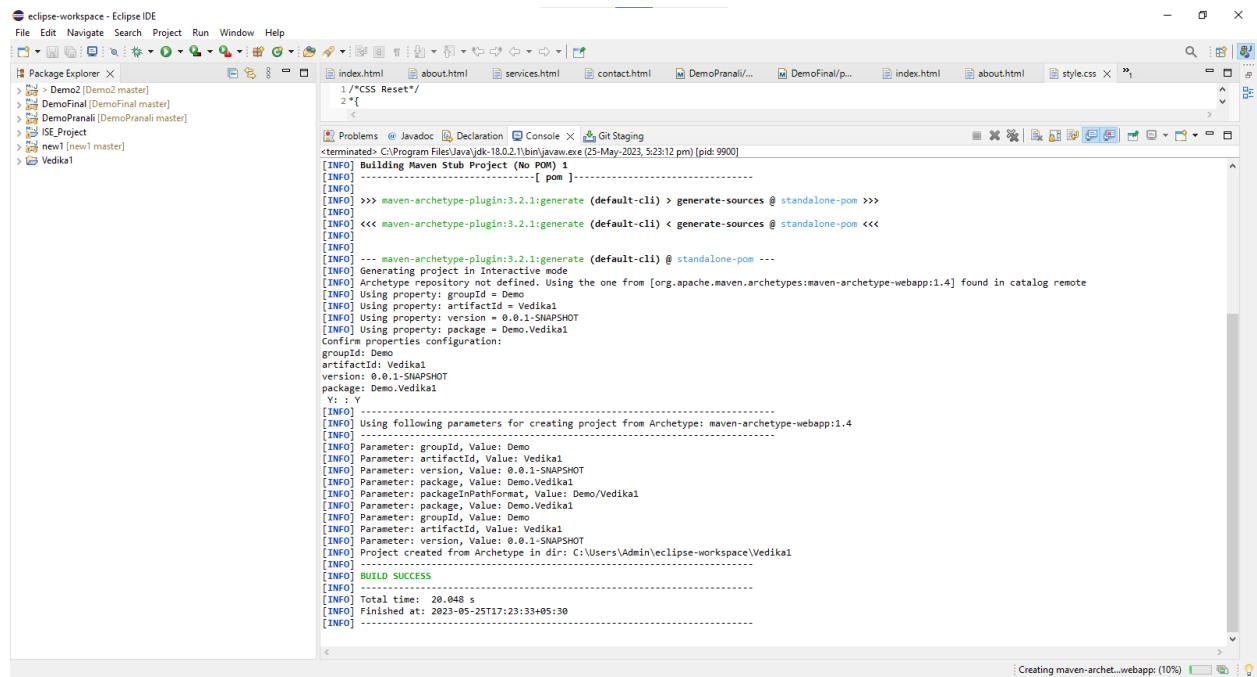
index.html about.html services.html contact.html DemoPranali/... DemoFinal/p... index.html about.html style.css X

Problems Javadoc Declaration Console X Git Staging
C:\Program Files\Java\jdk-18.0.2\bin\javaw.exe (25-May-2023, 5:23:12 pm) [pid: 9900]
[INFO] Scanning for projects...
Downloading from : https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-metadata.xml
Downloading from : https://repo.maven.apache.org/maven2/org/codehaus/mojo/maven-metadata.xml
Progress (1): 5.8/21 kB
Progress (2): 5.8/21 kB | 8.2/14 kB
Progress (2): 2.4/21 kB | 8.2/14 kB
Progress (2): 2.4/21 kB | 6.1/14 kB
Progress (2): 4.4/21 kB | 6.1/14 kB
Progress (2): 3.6/21 kB | 6.1/14 kB
Progress (2): 4.2/21 kB | 6.1/14 kB
Downloaded from : https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-metadata.xml (14 kB at 31 kB/s)
Downloaded from : https://repo.maven.apache.org/maven2/org/codehaus/mojo/maven-metadata.xml (21 kB at 43 kB/s)
Progress (1): 980 B
Downloaded from : https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-archetype-plugin/maven-metadata.xml (980 B at 17 kB/s)
[INFO] -----< org.apache.maven:standalone-pom -----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----[ pom ]-----
[INFO] >>> maven-archetype-plugin:3.2.1:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO] <<< maven-archetype-plugin:3.2.1:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO] --- maven-archetype-plugin:3.2.1:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Interactive mode
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archetypes:maven-archetype-webapp:1.4] found in catalog remote
[INFO] Using property: groupId = Demo
[INFO] Using property: artifactId = Vedika1
[INFO] Using property: version = 0.0.1-SNAPSHOT
[INFO] Using property: package = Demo.Vedika1
Confirm properties configuration:
groupId: Demo
artifactId: Vedika1
version: 0.0.1-SNAPSHOT
package: Demo.Vedika1
Y: : Y
Creating maven-archet...webapp: (33%)

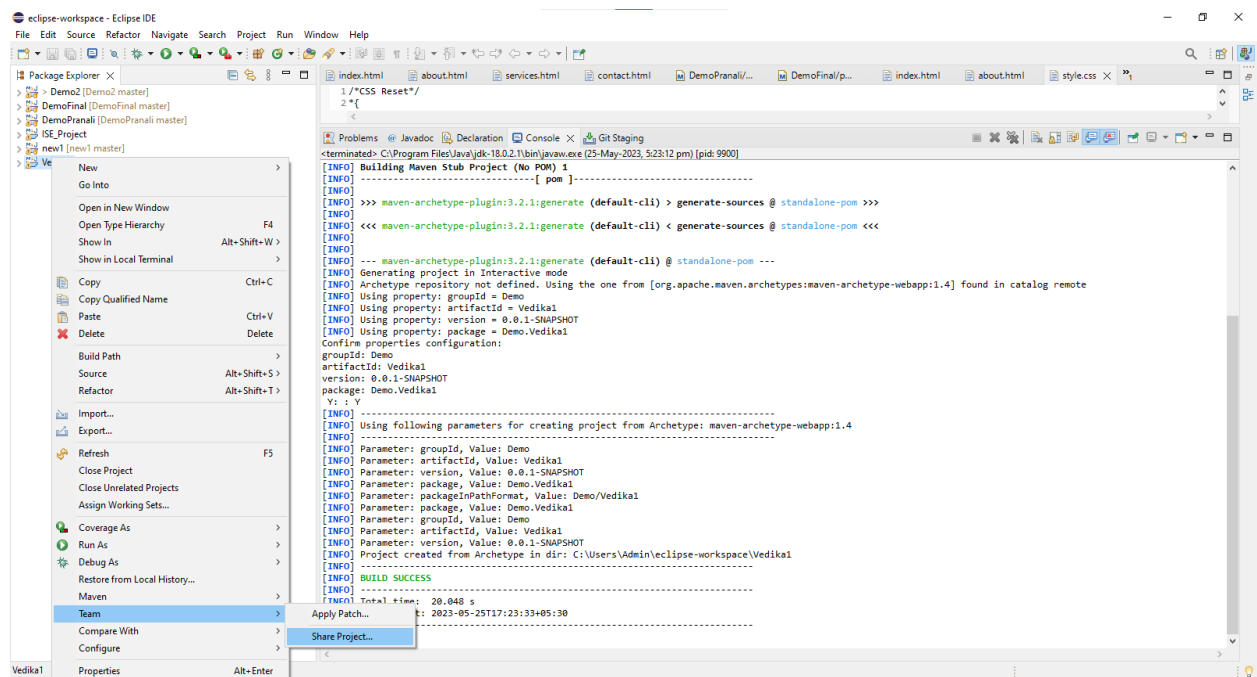
```

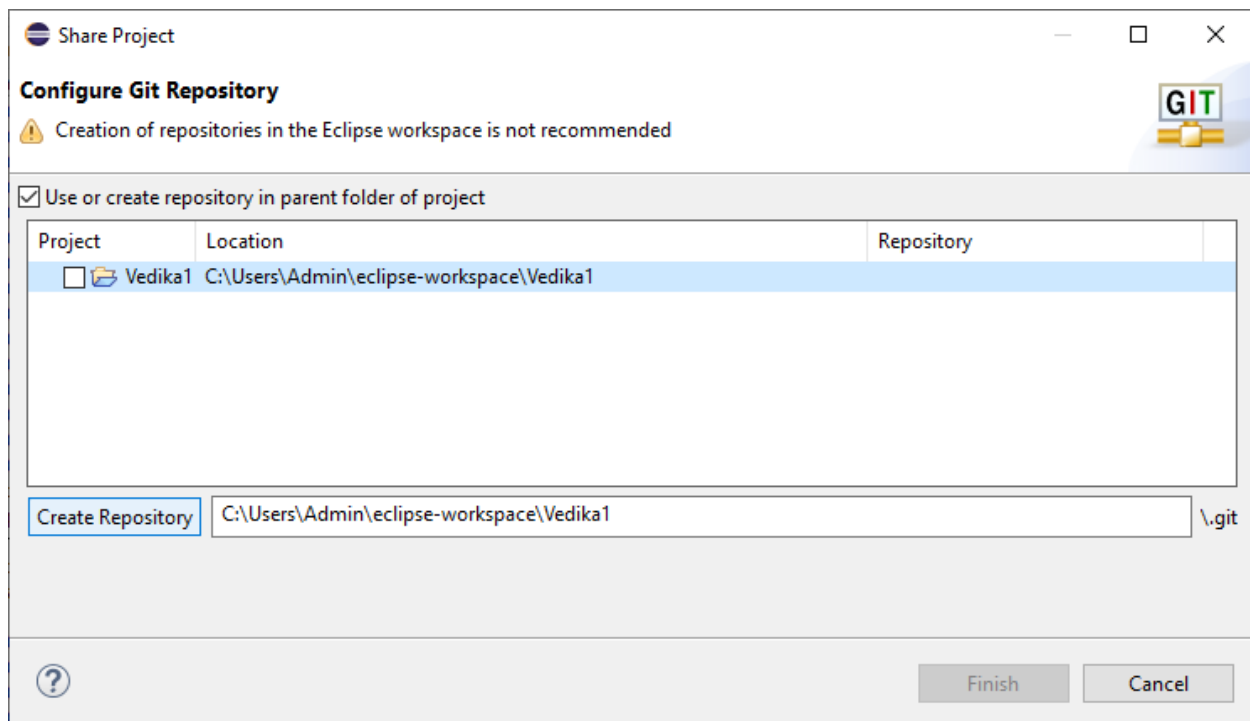
You will see below window with build success

This project contains 'pom.xml' which contains all the dependencies required for application

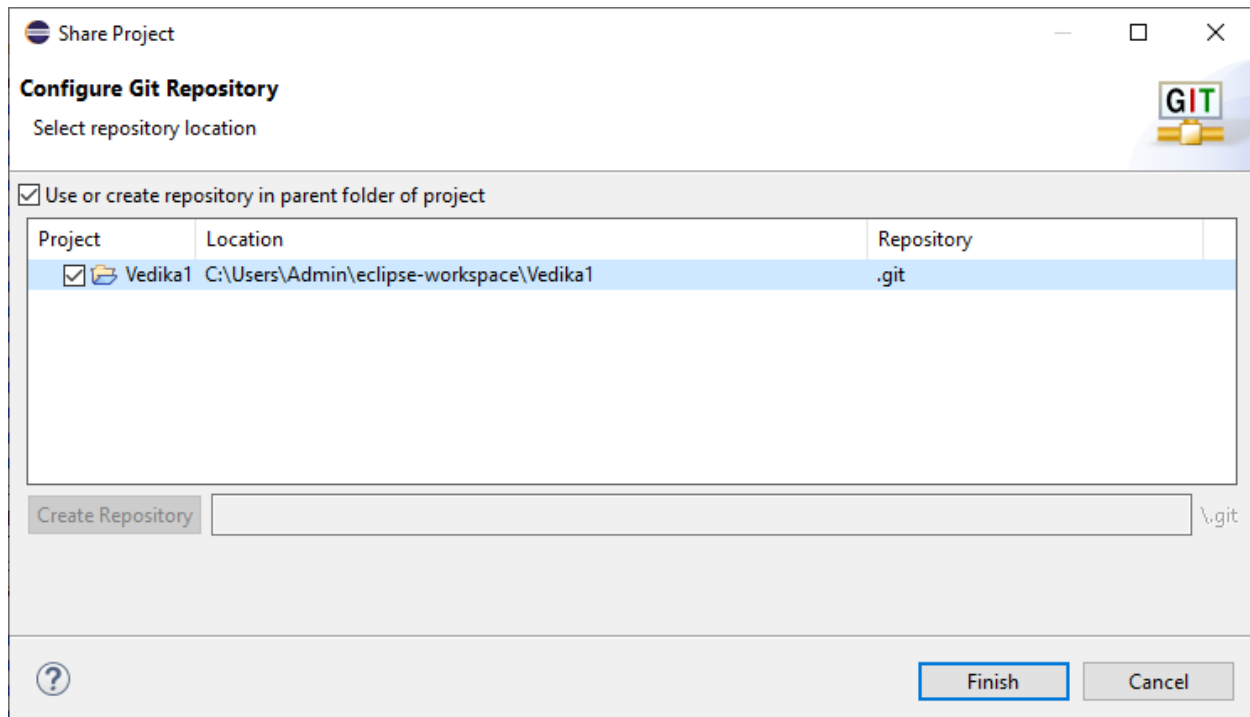


Right click on your created maven project and click on Team-share project In a below window click the check box as 'use or create repository in parent folder of project'



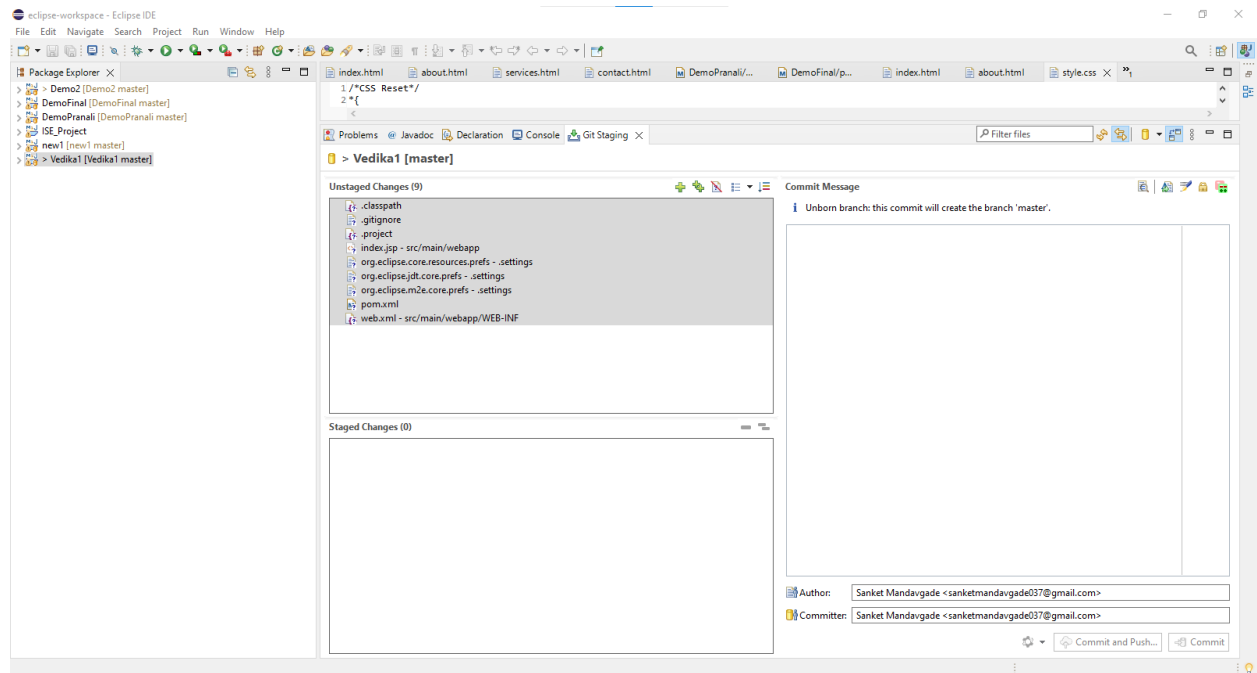
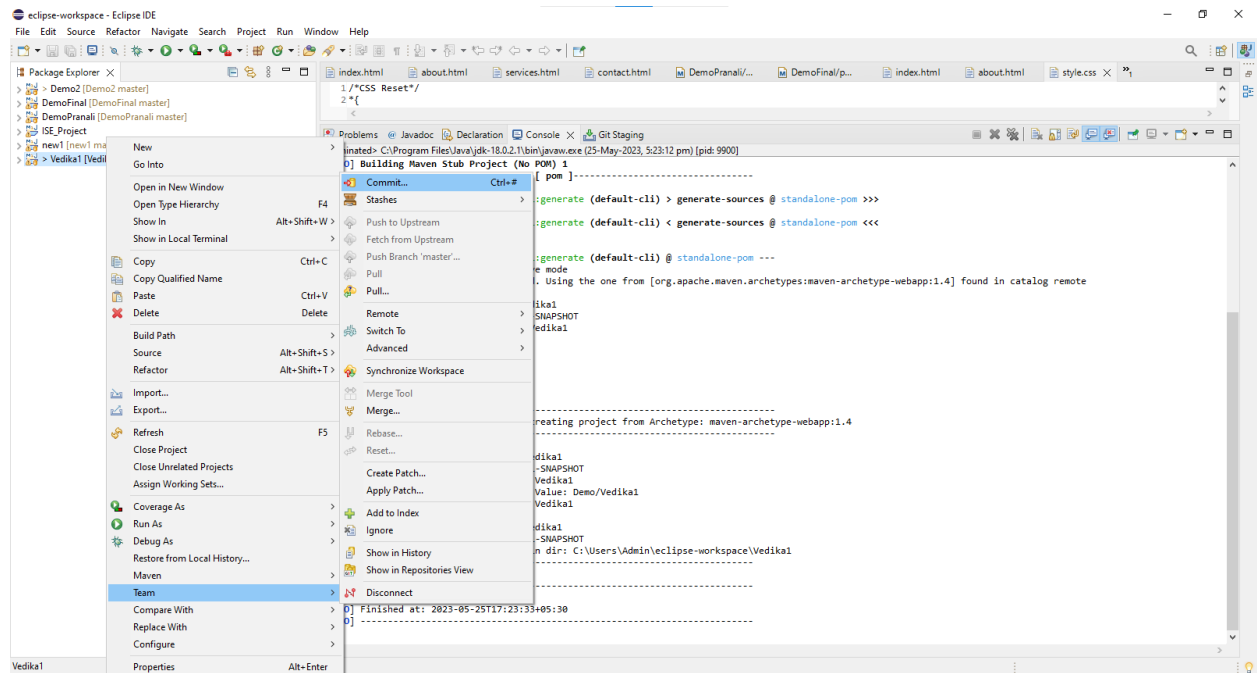


Click on create repository

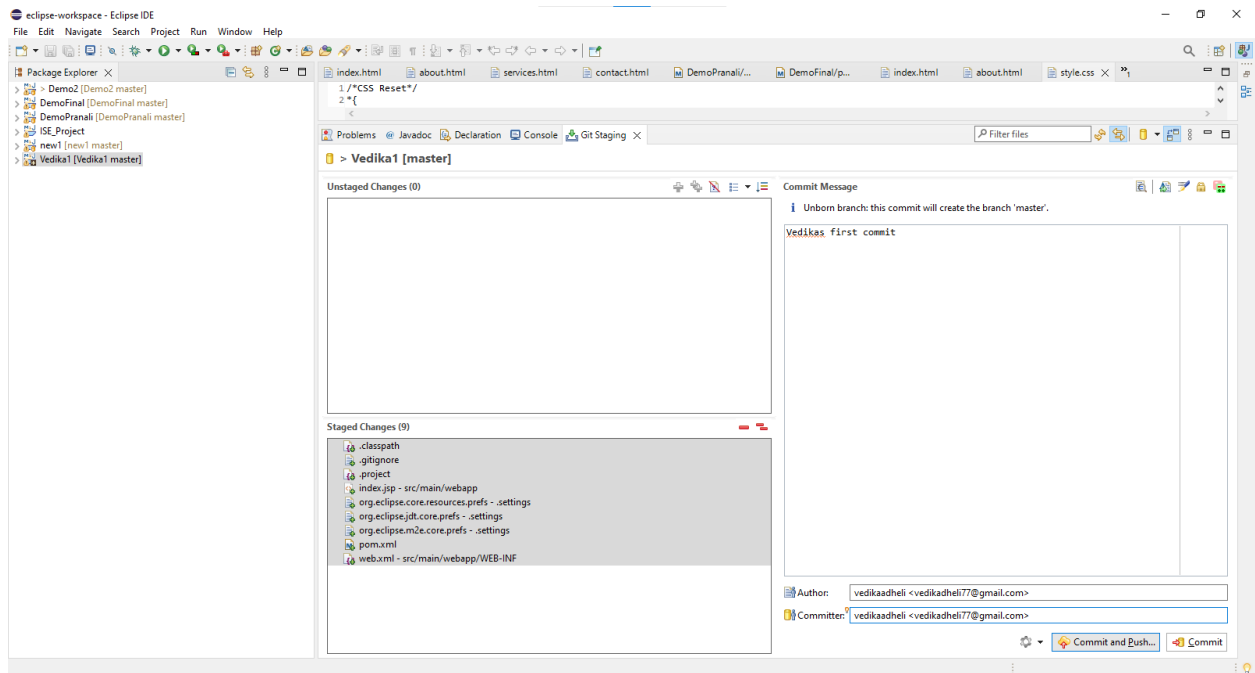
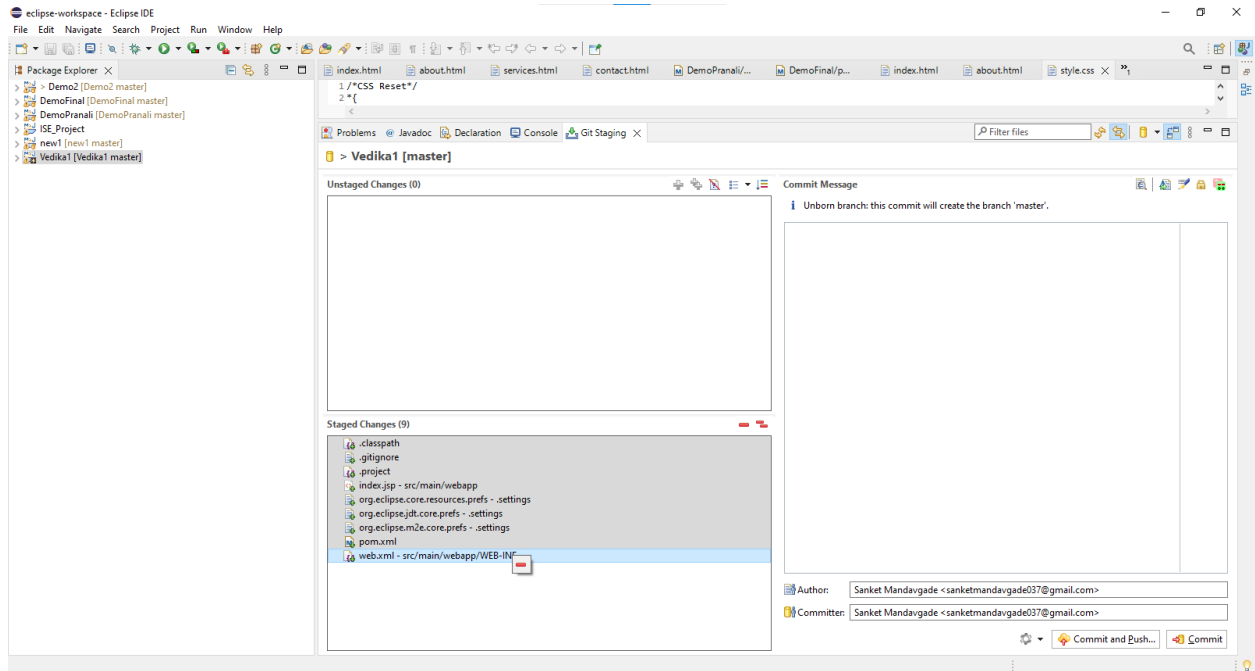


Click on finish

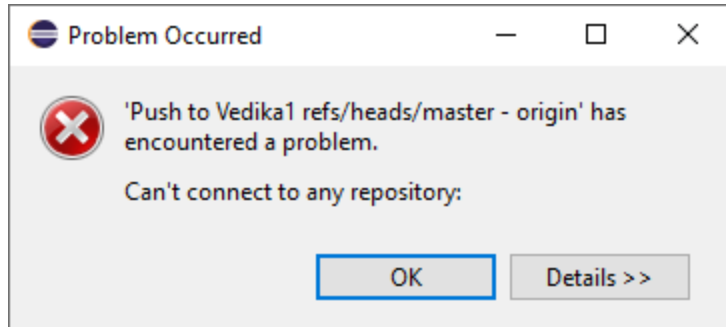
Right click on PranotiSampleProject and click on Team-commit



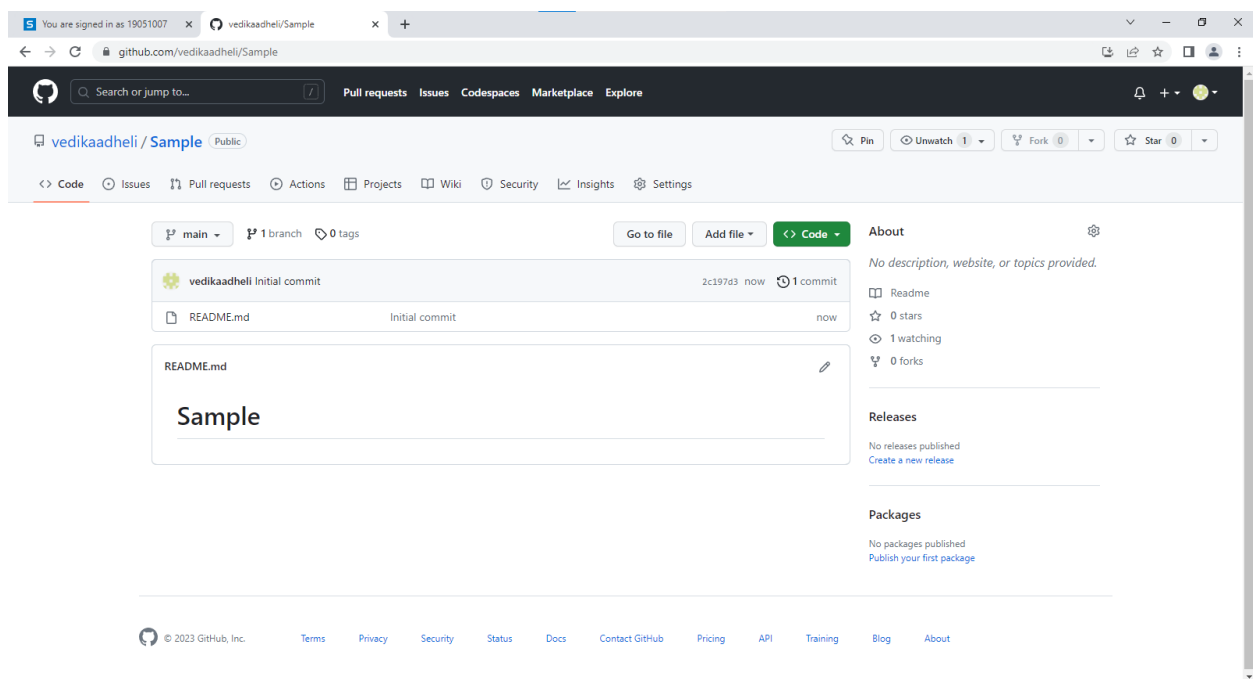
Select all files in Unstaged changes and drag and drop them into staged changes
 Enter commit message
 Edit author and creator name as username <email>
 i.e vedikaadheli <vedikaadheli77@gmail.com>



You will see below window with message 'can't connect to any repository' Click on 'OK'



Login to your Github account and create repository as 'Sample'



After commit click on 'Push Head' button and paste your GitHub repository url
Enter username and password Click on Preview

Push Branch master

Destination Git Repository

Enter the location of the destination repository.

Remote name:

Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

Authentication

User:

Password:

☐ Store in Secure Store

Click on preview in below window

Push Branch master

Push to branch in remote

Select a remote and the name the branch should have in the remote.

Source:

master

068b3a5 Vedikas first commit

Destination:

Remote: origin: https://github.com/vedikaadheli/Sample.git

Branch: master

☒ Configure upstream for push and pull

When pulling: Merge

☐ Force overwrite branch in remote if it exists and has diverged

Show [advanced push](#) dialog

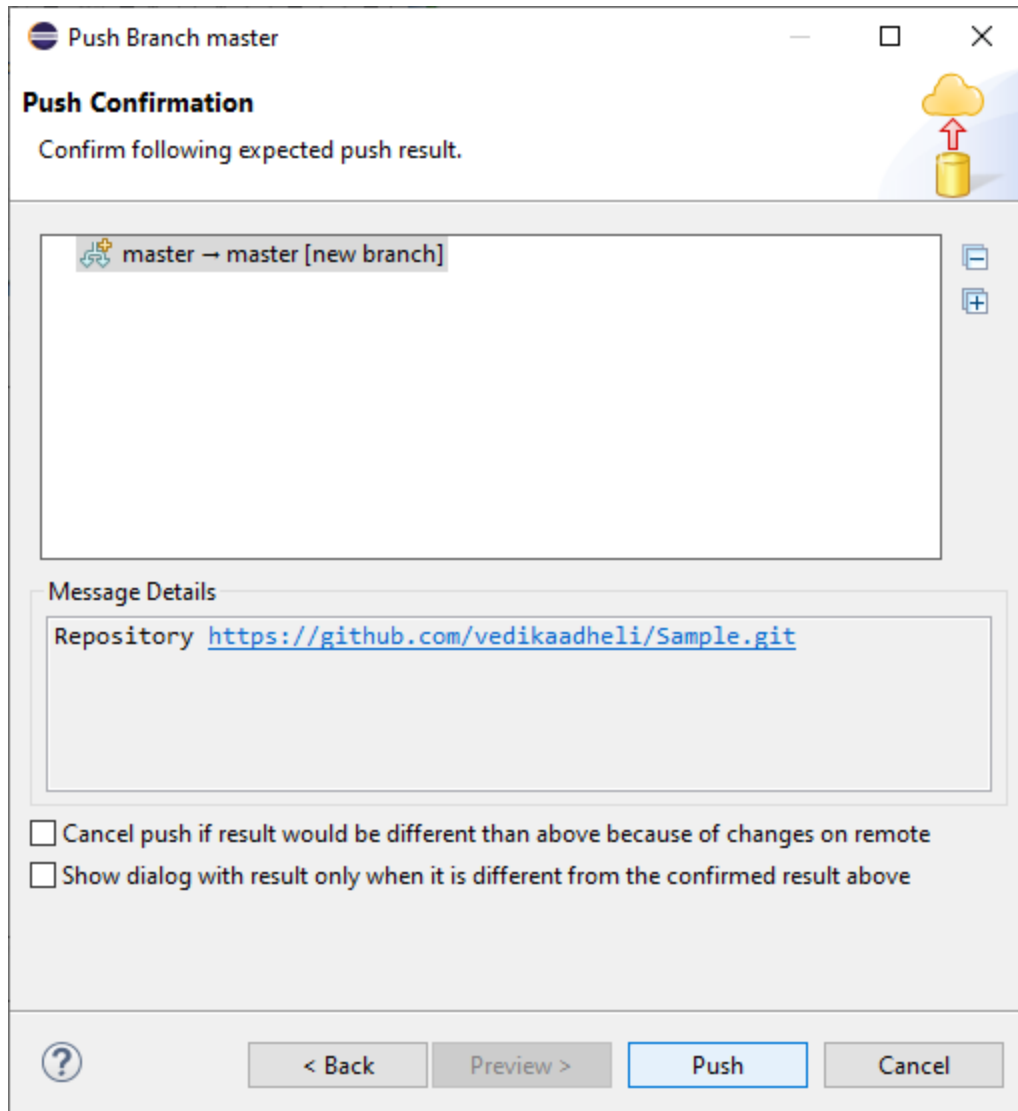
< Back

Preview >

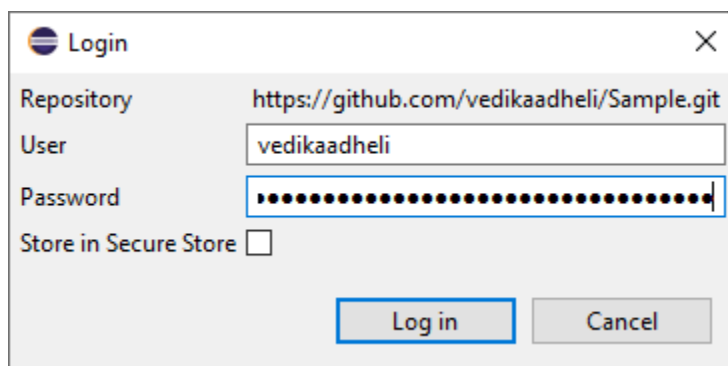
Push

Cancel

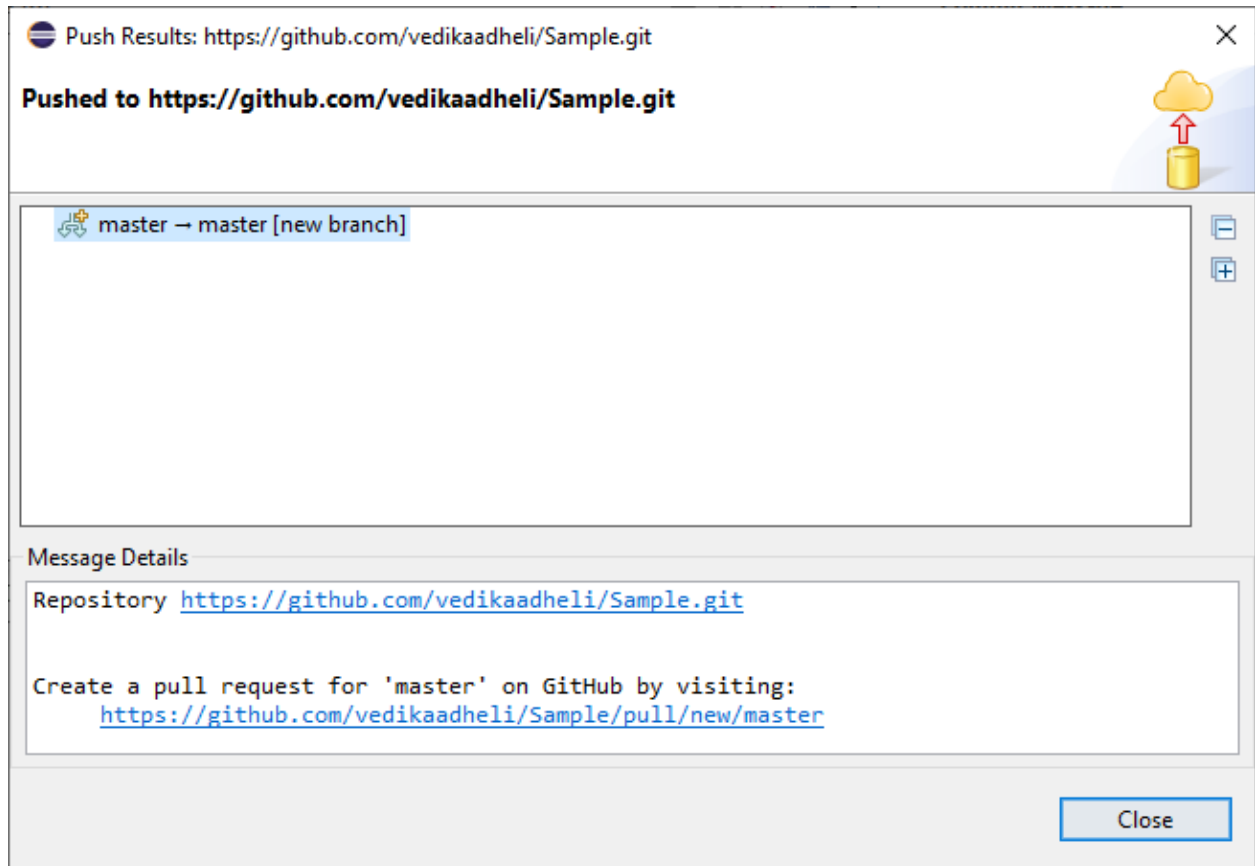
Click on push button



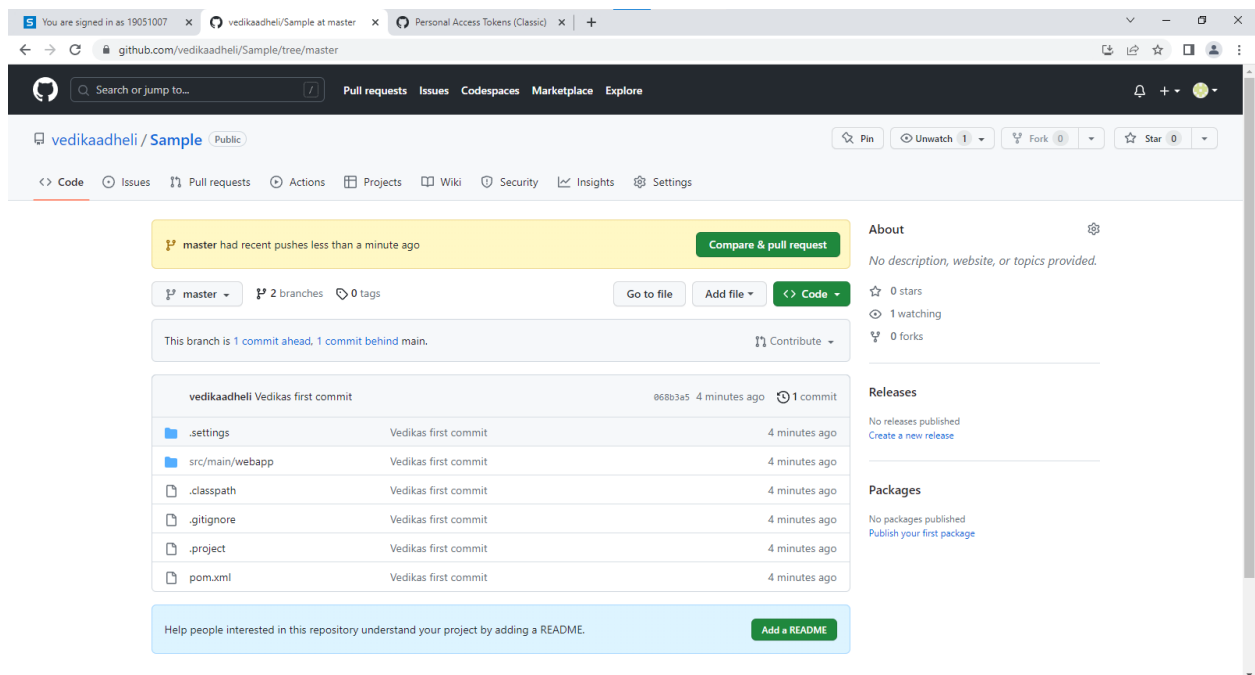
You will see below login window Enter appropriate credentials and Click in 'Log in'



After Log in you will see below window Click on 'Close'



You can see that the maven project is pushed into GitHub in master branch



Now Go to Environment variables,in system variables

- Add JAVA_HOME variable with path 'C:\Program Files\Java\jdk-18.0.1.1\bin'
- Add JRE_HOME variable with path 'C:\Program Files\Java\jdk-18.0.1.1'
- Add M2_HOME variable with path 'C:\apache-maven-3.8.5-bin'
- Add MAVEN_HOME variable with path 'C:\apache-maven-3.8.5-bin\bin'

In User variables add 'C:\apache-maven-3.8.5-bin\bin' path

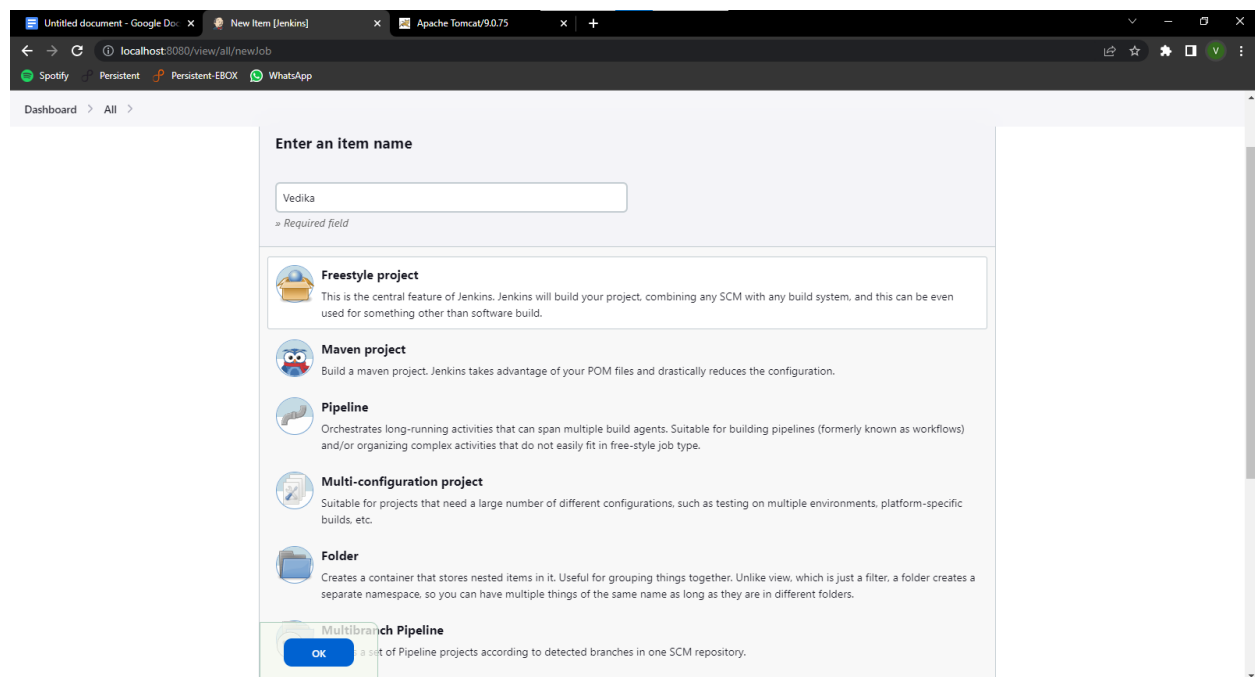
Now Go to Jenkins and Log in Go to Manage Jenkins and download plugins as

- Deploy to container
- Maven project

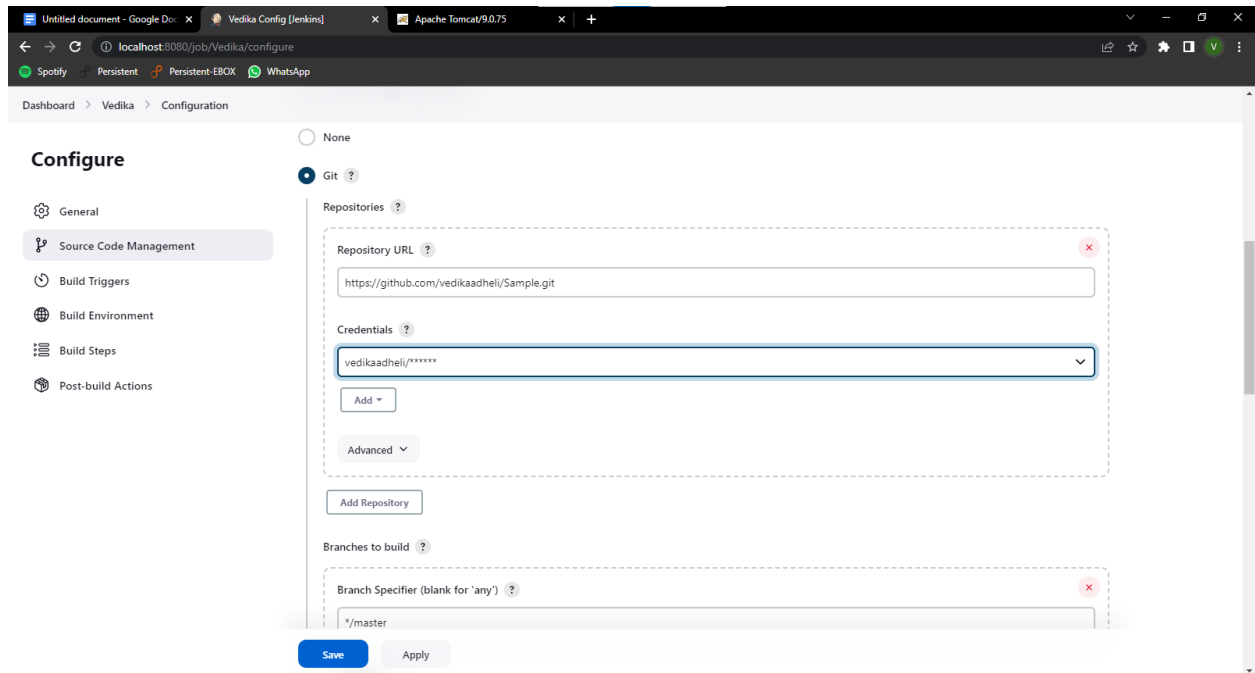
In Manage Jenkins go to 'Global tool configuration'

- Go to Maven
 - Enter name as 'MAVEN_HOME'
 - Enter path as 'C:\apache-maven-3.8.5-bin'
- Add maven and apply and save

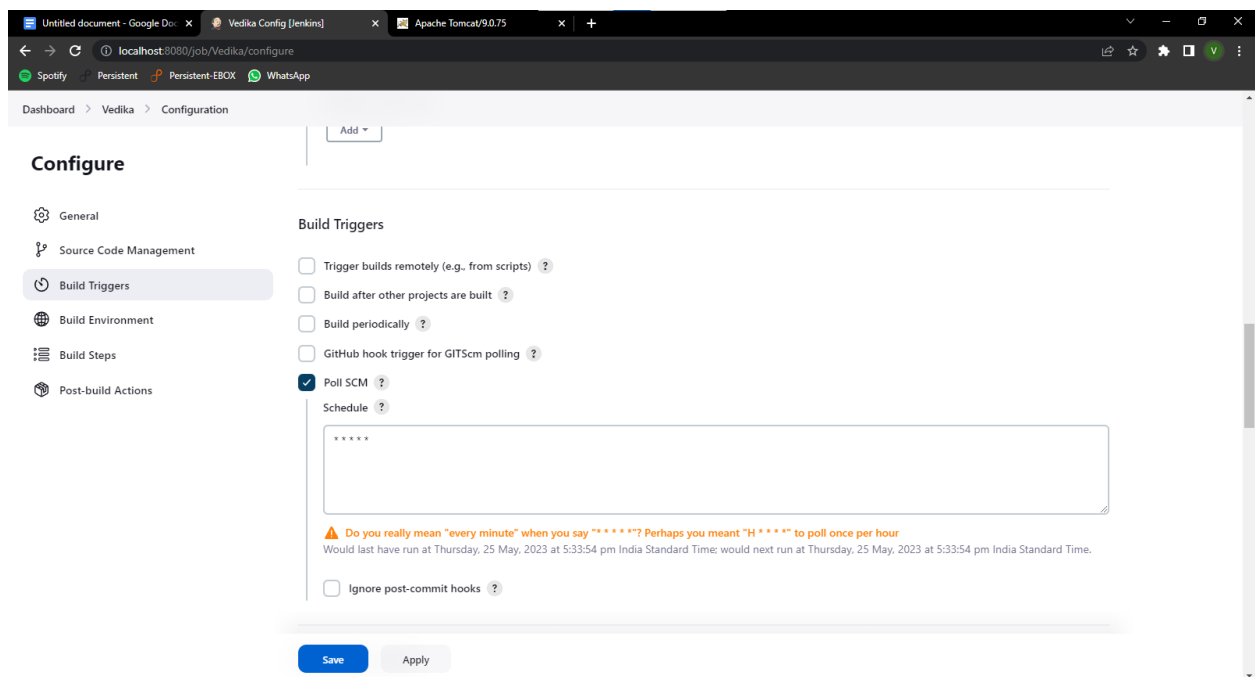
Go to Dashboard Click on New item Enter Item name i.e.'Pranoti1' Select Freestyle project Click on 'OK'



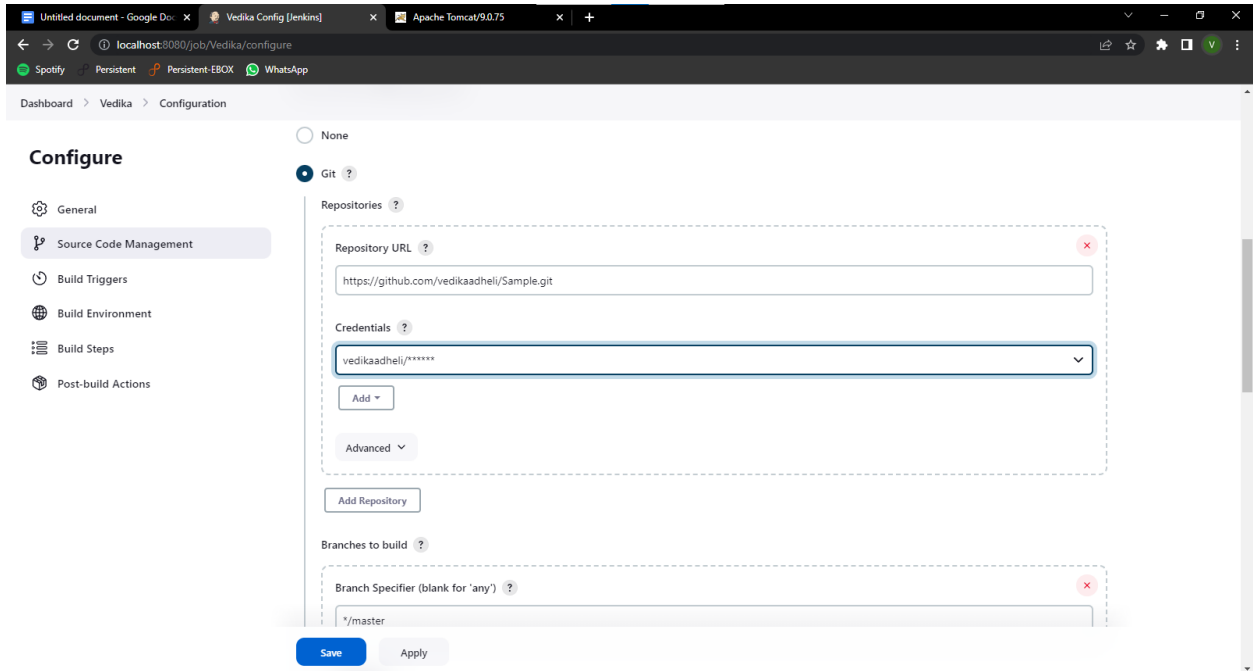
Select Git Option and paste your GitHub repository url



Enter credentials Let the branch be master

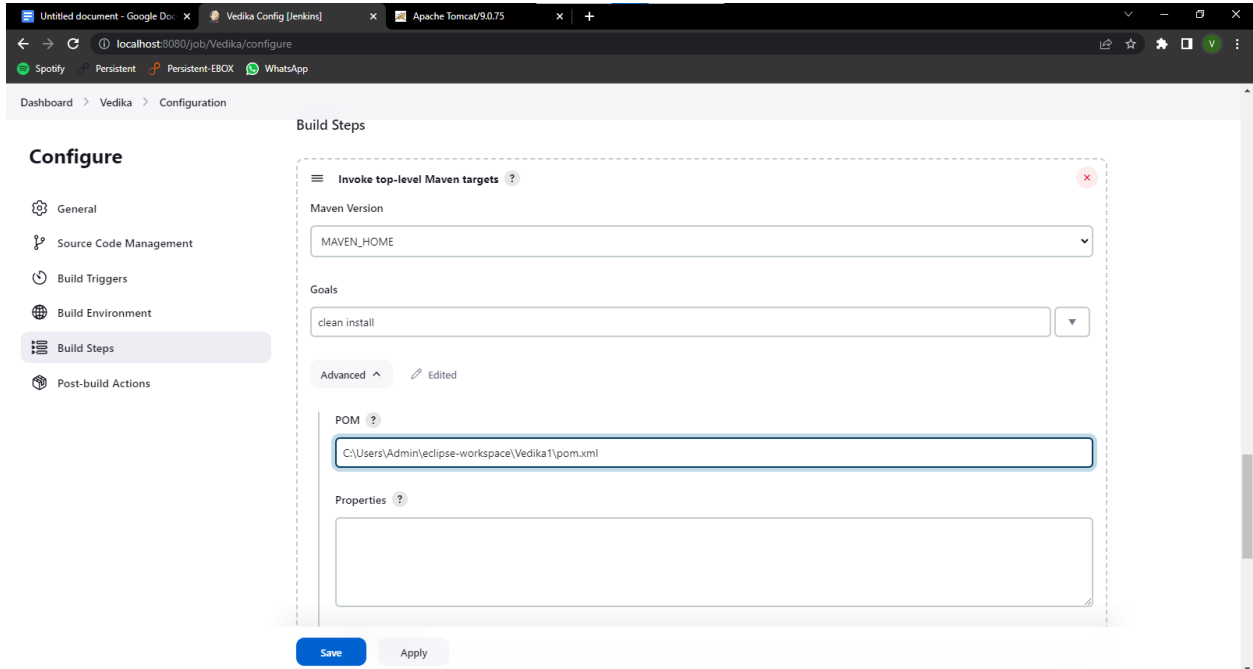


Click on Poll SCM checkbox Enter '* * * * *'



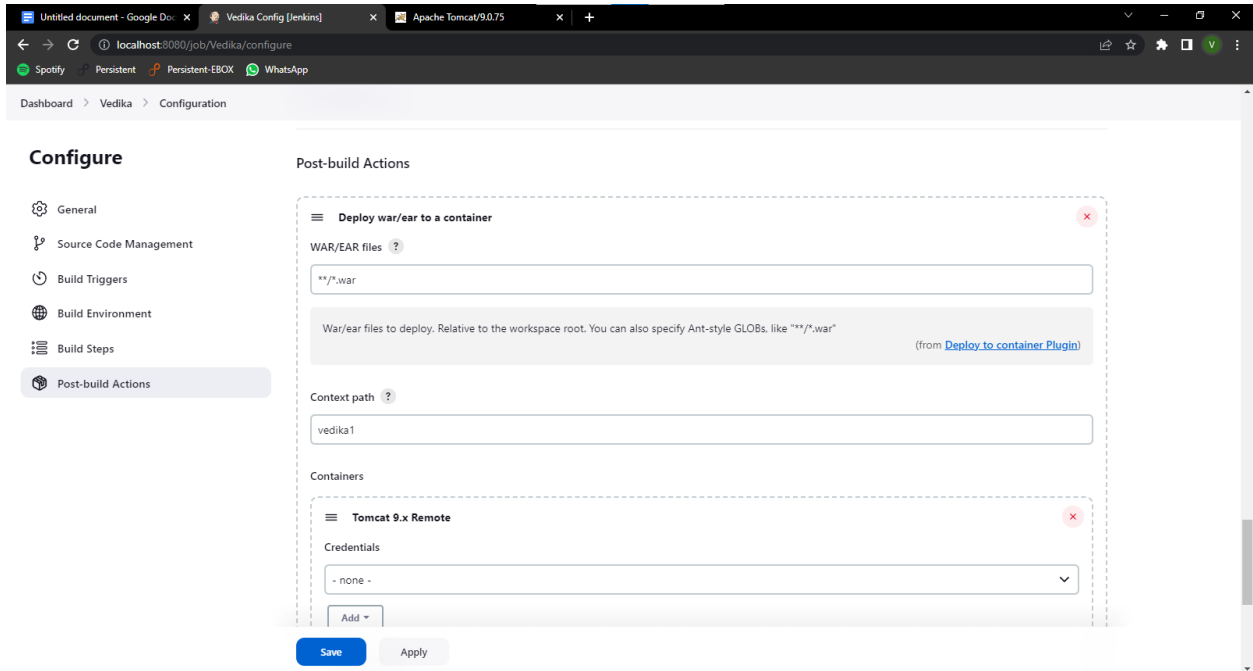
In build steps select 'top level maven targets'

In Maven version, select MAVEN_HOME in dropdown Enter 'clean install' in Goals

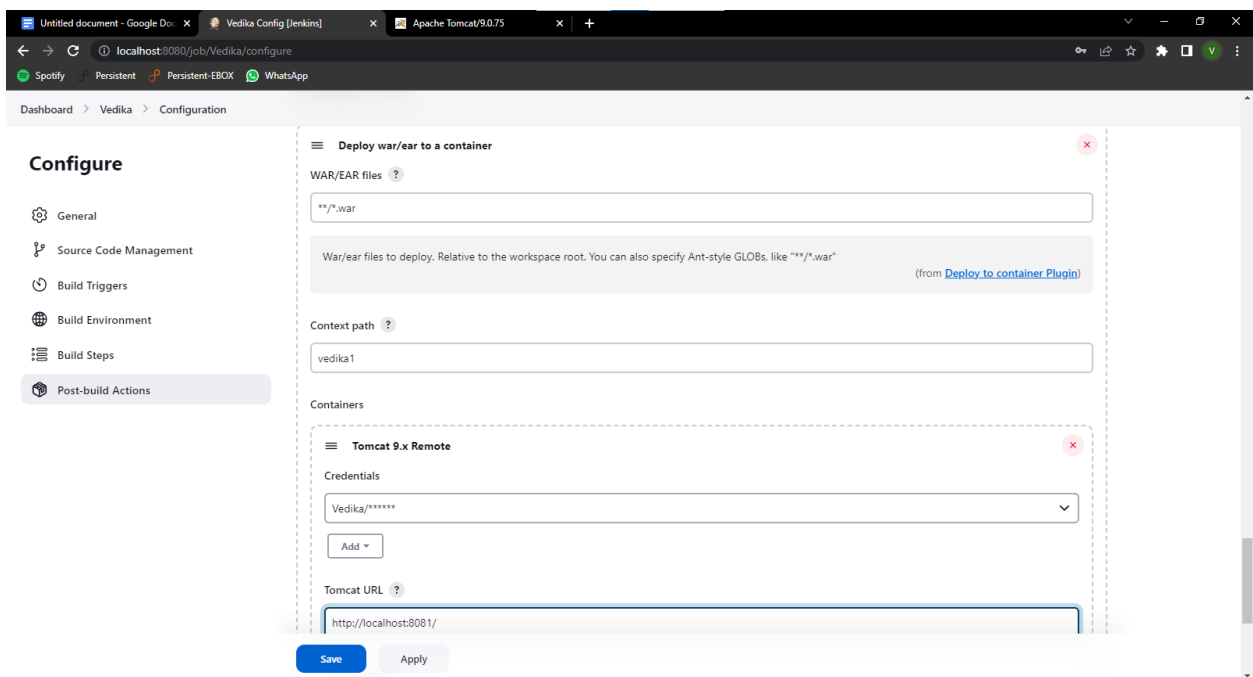


In Post build steps, click on 'deploy war/ear to a container'

Enter '**/*.*.war' in war/ear files field Enter context path as 'vedika1'



In containers select 'Tomcat 9x Remote'
Edit username and password in war file of Tomcat and Enter the credentials
Enter Tomcat URL Then Apply and save

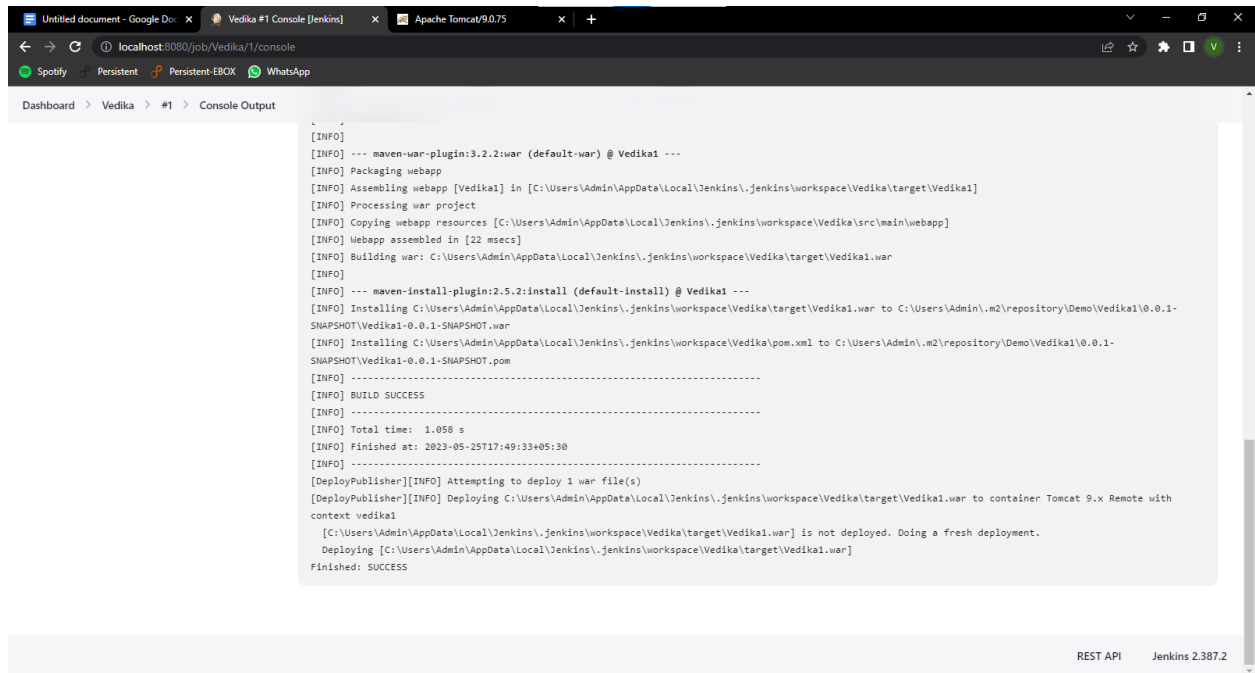


Click on Build now

The screenshot shows the Jenkins web interface for a project named 'Vedika'. The top navigation bar includes the Jenkins logo, a search bar, and user information (Admin, log out). The left sidebar contains a list of project actions: Status, Changes, Workspace, Build Now (highlighted), Configure, Delete Project, Git Polling Log, and Rename. The main content area displays the 'Project Vedika' title, a 'Permalinks' section, and a 'Build History' table. The build history shows a single successful build (#1) on May 25, 2023, at 5:49 PM. The bottom right corner of the dashboard indicates the REST API and Jenkins version 2.387.2.

After build success you can see Console Output as shown below

The screenshot displays the 'Console Output' for build #1 of the 'Vedika' project. The left sidebar shows the 'Console Output' tab selected. The main content area shows the build log, which starts with 'Started by user Admin' and 'Running as SYSTEM'. The log details the process of cloning a repository from GitHub, checking out a specific revision, and building the project using Maven. The build is successful, and the console output is displayed in a scrollable area.



Open Tomcat by entering url i.e. 'localhost:8081/contextpath/' Entered 'localhost:8081/Sample/' Finally we can see web app will be deployed into Tomcat

