



# **KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY**

**(AN AUTONOMOUS INSTITUTION)**



**Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH,  
Narayanguda, Hyderabad, Telangana – 500029**



## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

### **LAB RECORD**

### **SOFTWARE ENGINEERING LAB**

**B. Tech. III YEAR I SEM (KR23)  
ACADEMIC YEAR  
2025-26**



**KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY  
(AN AUTONOMOUS INSTITUTION)**



**Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad  
Narayanguda, Hyderabad, Telangana – 500029**

## **Certificate**

This is to certify that following is a Bonafide Record of the workbook task done by

\_\_\_\_\_ bearing Roll No \_\_\_\_\_ of \_\_\_\_\_

Branch of \_\_\_\_\_ year B. Tech. Course in the \_\_\_\_\_

Subject during the Academic year \_\_\_\_\_ & \_\_\_\_\_ under our supervision.

Number of week tasks completed: \_\_\_\_\_

Signature of Staff Member Incharge

Signature of Head of the Dept.

Signature of Internal Examiner

Signature of External Examiner



# **KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY (AN AUTONOMOUS INSTITUTION)**



**Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad  
Narayanguda, Hyderabad, Telangana – 500029**

## **Daily Laboratory Assessment Sheet**

Name of the Lab:  
Branch & Section:

Student Name:  
HT. No:

## **Faculty Incharge**

## INDEX

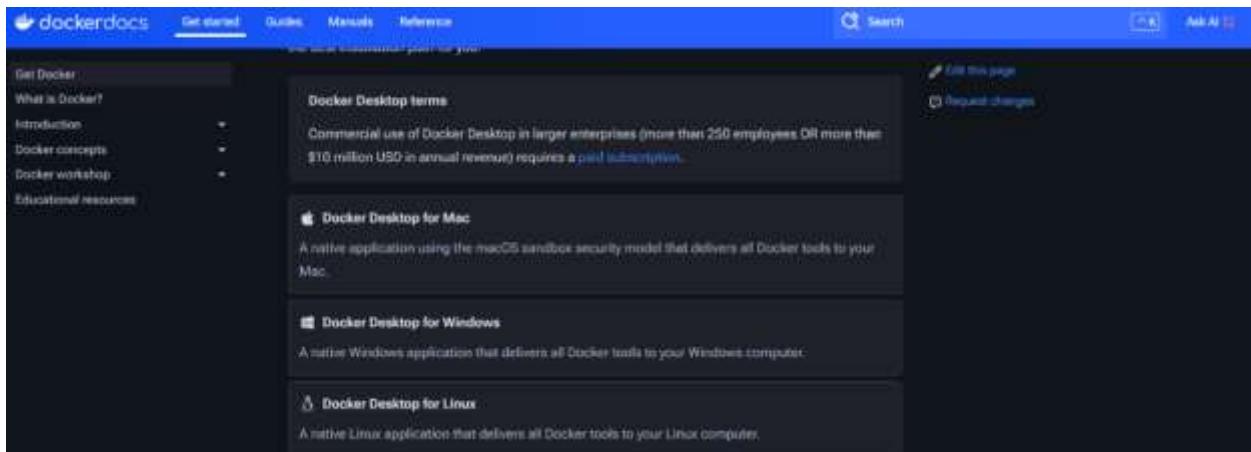
<b>Sr. NO.</b>	<b>CONTENTS</b>	<b>PAGE NO.</b>
1.	<p><b>Software Installation &amp; SRS Document</b></p> <ul style="list-style-type: none"> <li>a. Abstract</li> <li>b. Functional Requirements (FR)</li> <li>c. Non-Functional Requirements (NFR)</li> <li>d. User Identification</li> <li>e. Workflow of Each User</li> <li>f. Use Cases</li> </ul>	
2.	<p><b>Exploring git local and remote commands on the multi-folder project</b></p> <ul style="list-style-type: none"> <li>a. Pushing multi-folder project into private repository (by student).</li> <li>b. Students must explore all listed git commands on the multi-folder project in local and remote repository.</li> <li>c. Students must explore all git commands on given scenario-based question</li> </ul>	
3.	<p><b>Collaborative coding using git</b></p> <ul style="list-style-type: none"> <li>a. To work on collaborative coding by:</li> <li>b. Creating Organization.</li> <li>c. Coordinating with others through a shared repository</li> <li>d. To resolve conflicts when collaborating on same part of code.</li> <li>e. To create and apply patch.</li> </ul>	
4.	<p><b>Build and package Java and Web applications using Maven</b></p> <ul style="list-style-type: none"> <li>a. Understand the structure and lifecycle of a Maven project.</li> <li>b. Build and package Java and Web applications using Maven.</li> <li>c. Add dependencies using <b>pom.xml</b>, compile and test using plugins.</li> <li>d. Resolve errors and conflicts arising from dependency mismatches.</li> <li>e. Work with parent and multi-module Maven projects.</li> <li>f. Generate executable JARs and deployable WARs using Maven.</li> </ul>	
5.	<p><b>Docker CLI commands</b></p> <ul style="list-style-type: none"> <li>a. Learn how to pull, run, stop, start, remove, and inspect containers and images.</li> <li>b. Gain the ability to create, monitor, and troubleshoot running containers.</li> <li>c. Configure and manage networks for container communication.</li> <li>d. Create and manage persistent storage for containers.</li> <li>e. Learn how to list, remove, and manage images efficiently.</li> </ul>	
6.	<p><b>Docker</b></p> <ul style="list-style-type: none"> <li>a. Learn how to define and run multiple interdependent services (e.g., web server, database) in a single configuration file.</li> </ul>	

	<ul style="list-style-type: none"> <li>b. Gain skills in writing and interpreting docker-compose.yml files for service setup.</li> <li>c. Deploy the same setup across different machines without manual configuration.</li> <li>d. Configure container networking and persistent storage within Compose.</li> <li>e. Reduce setup time and enable faster iteration during application development.</li> </ul>	
7.	<p><b><u>Creating a Multi-Module Maven Project</u></b></p> <ul style="list-style-type: none"> <li>a. Build and package Java and Web applications using Maven.</li> <li>b. Add dependencies using <b>pom.xml</b>, compile and test using plugins.</li> <li>c. Resolve errors and conflicts arising from dependency mismatches.</li> <li>d. Work with parent and multi-module Maven projects.</li> <li>e. Generate executable JARs and deployable WARs using Maven</li> </ul>	
8.	<p><b><u>Jenkins Automation</u></b></p> <ul style="list-style-type: none"> <li>a. Hands-on practice on manual creation of Jenkins pipeline using Maven projects from Github</li> <li>b. Create the job and build the pipeline for maven-java and maven-web project.</li> </ul>	
9.	<p><b><u>Pipeline Creation using script</u></b></p> <ul style="list-style-type: none"> <li>a. Evaluation of Jenkins pipeline.</li> <li>b. WORKING ON BUILD TRIGGERS FOR LAST JENKINS PIPILINE</li> <li>c. Hands-on practice on creation of scripted Jenkins pipeline.</li> <li>d. Take the screenshots for above task</li> </ul>	
10.	<p><b><u>Working with minikube and Nagios</u></b></p> <ul style="list-style-type: none"> <li>a. Hands-on practice of creating, running and scaling pods in minikube.</li> <li>b. Running Nginx server on specified port number by explaining the Nginx monitoring tool</li> <li>c. Running Nagios server and Understanding the Monitoring tool using Docker.</li> <li>d. AWS-free Trier account Creation steps</li> </ul>	
11.	<p><b><u>Jenkins-CI/CD</u></b></p> <ul style="list-style-type: none"> <li>a. CI-Continuous Integration using Webhooks.</li> <li>b. Sending E-mail Notification on Build Failure or success</li> </ul>	
12.	<p><b><u>Creation of virtual machine for Ubuntu OS and Deploying the web application</u></b></p> <ol style="list-style-type: none"> <li>1. Creation of virtual machine</li> <li>2. Deploying the web application</li> <li>3. Accessing it publicly</li> </ol>	

## **1. Software Installation & SRS Document:**

### **DOCKER- INSTALLATION**

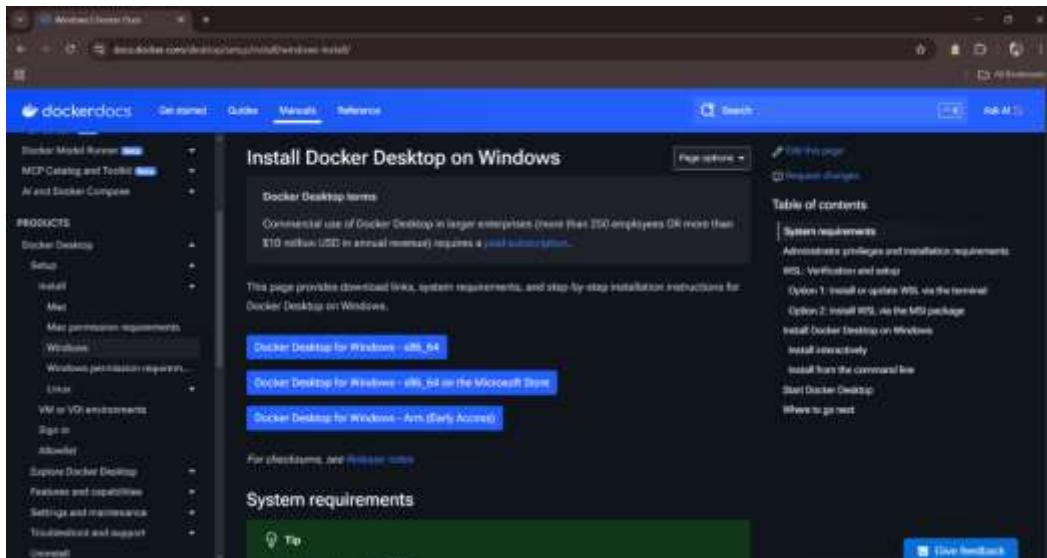
Step-1: Go to docker website



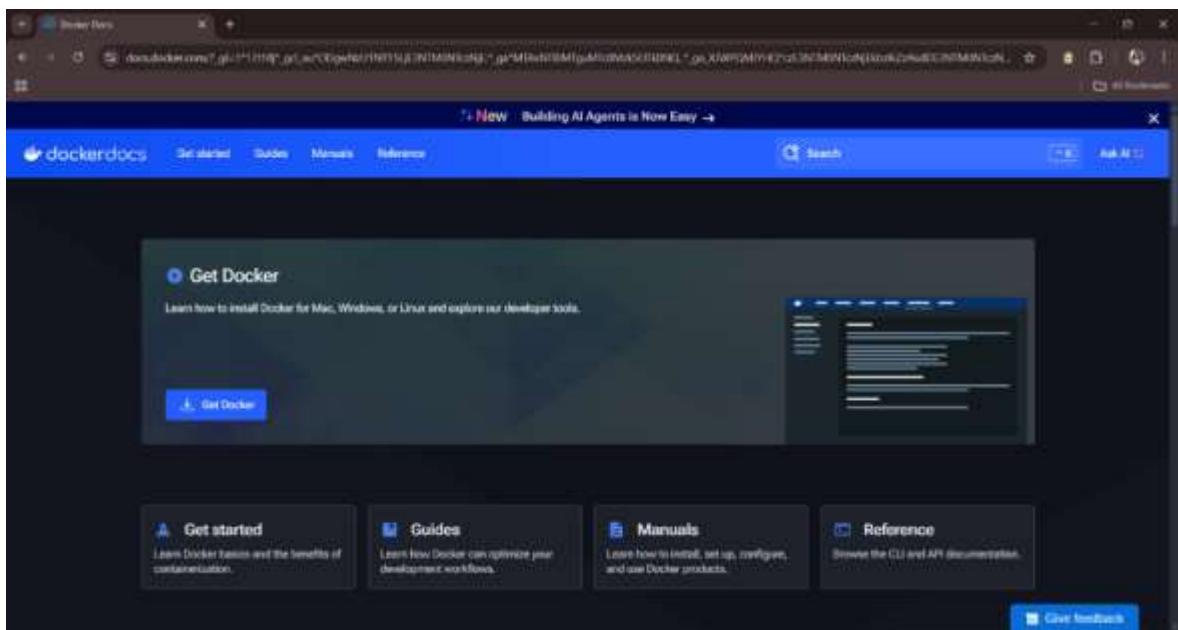
The screenshot shows the Docker documentation website with the 'Get Docker' section selected. It lists three options: 'Docker Desktop for Mac', 'Docker Desktop for Windows', and 'Docker Desktop for Linux'. Each option has a brief description and a link to its respective page.

- Docker Desktop for Mac**: A native application using the macOS sandbox security model that delivers all Docker tools to your Mac.
- Docker Desktop for Windows**: A native Windows application that delivers all Docker tools to your Windows computer.
- Docker Desktop for Linux**: A native Linux application that delivers all Docker tools to your Linux computer.

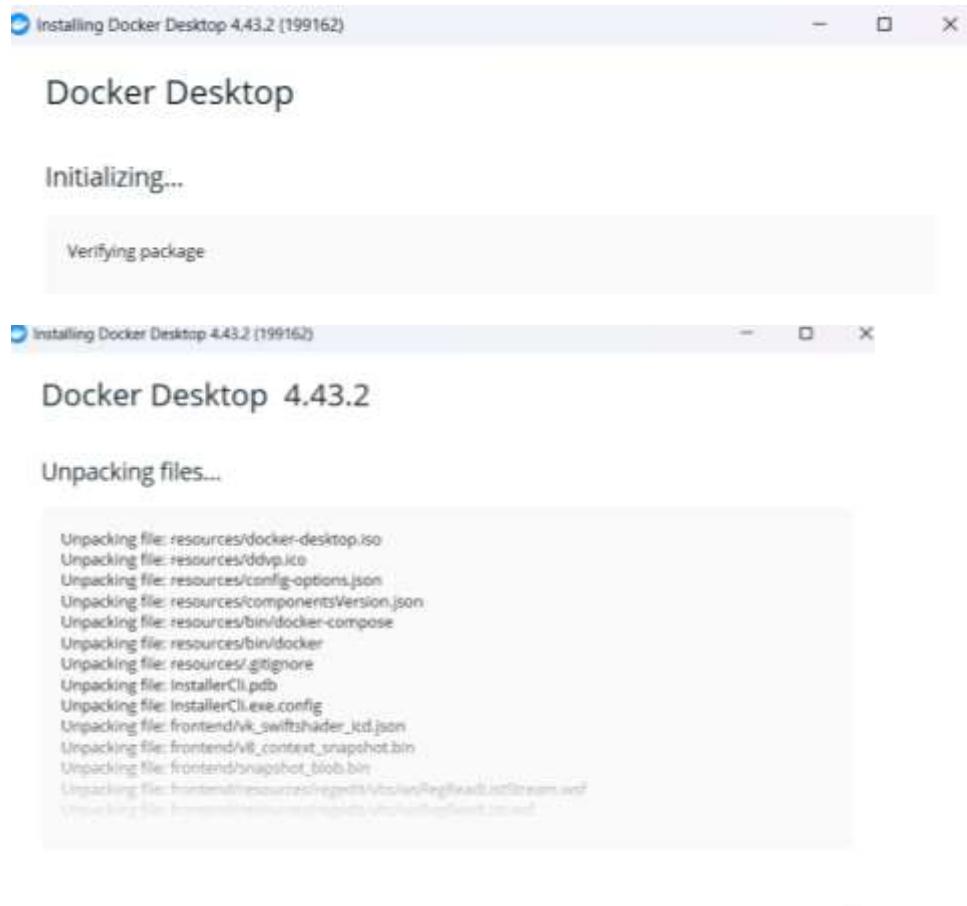
Step-2:Select the suitable one for your system



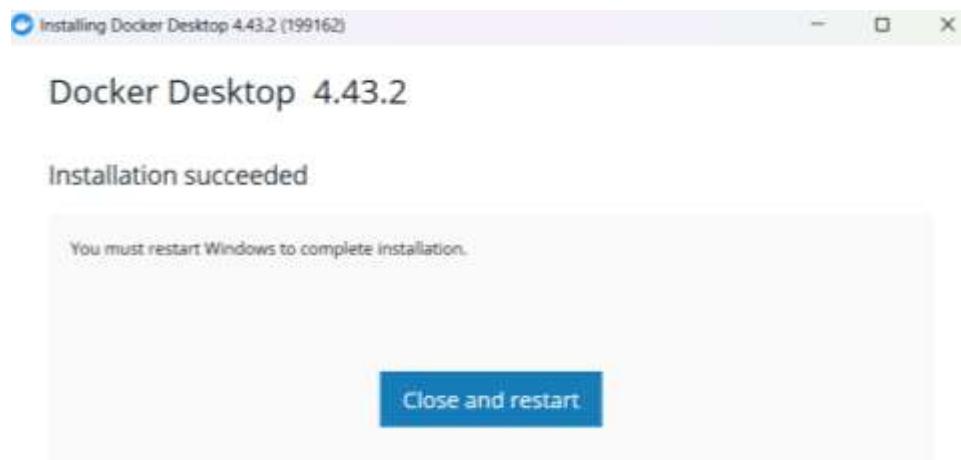
The screenshot shows the 'Install Docker Desktop on Windows' page. It includes sections for 'Docker Desktop terms', 'System requirements', and download links for 'Docker Desktop for Windows - x86\_64', 'Docker Desktop for Windows - x86\_64 on the Microsoft Store', and 'Docker Desktop for Windows - Arm (Dark Access)'. There is also a 'System requirements' section and a 'Tip' button.



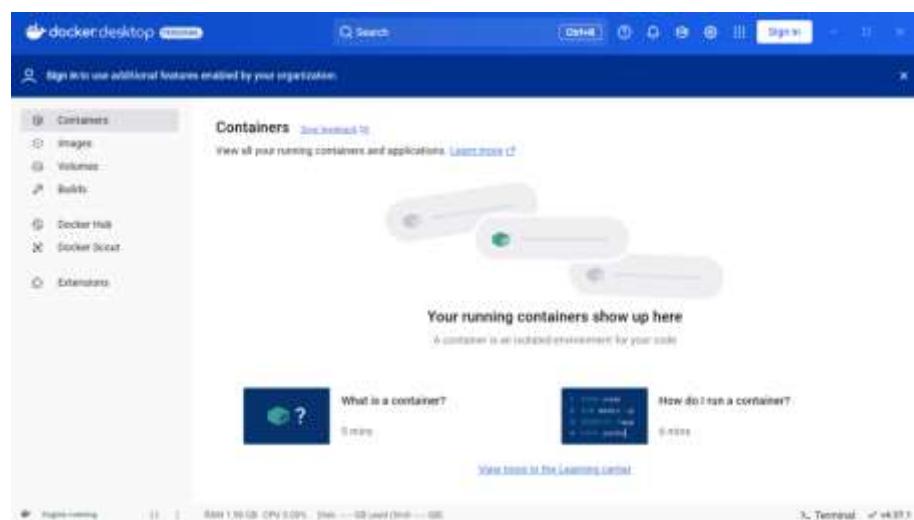
Step-3: After clicking on get docker it starts initializing



Step-4: Installation successful



Step-5: Docker interface



Step-6: docker version

```
Command Prompt
Microsoft Windows [Version 10.0.26100.4652]
(c) Microsoft Corporation. All rights reserved.

C:\Users\NekshaSrinivas>docker --version
Docker version 28.3.2, build 578ccf6

C:\Users\NekshaSrinivas>
```

## GIT – INSTALLATION:

### Step-1: Go to Git website



Step-2: click on downloads and options will be displayed



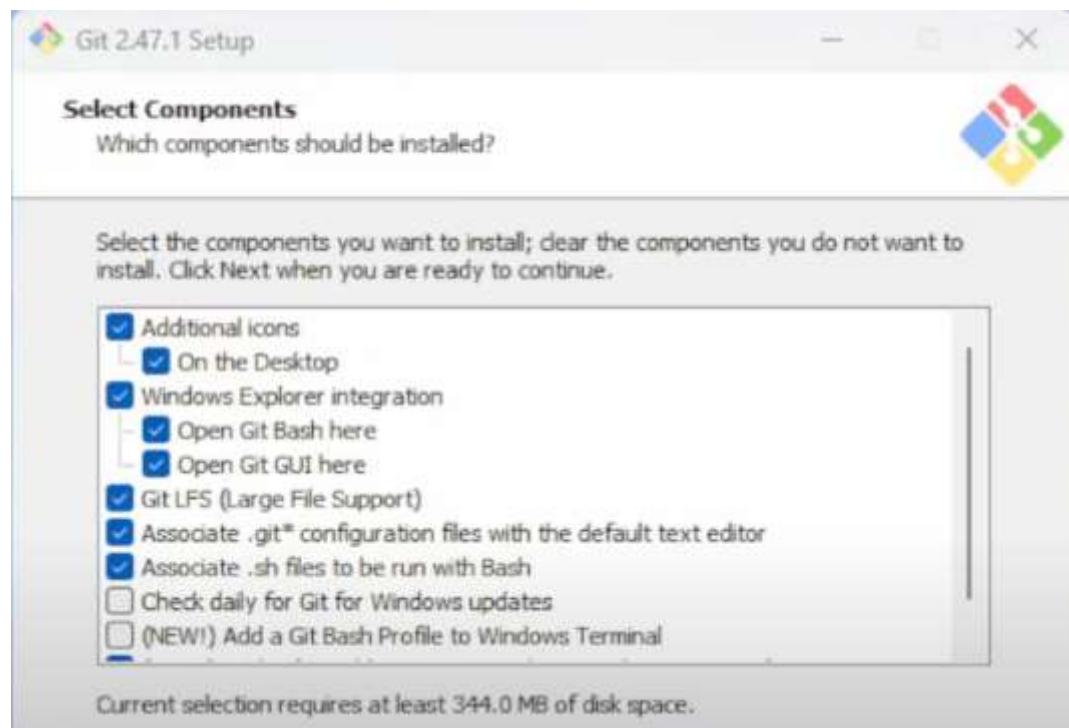
Step-3: Download for windows(suitable one for your system)



Step-4: License will be displayed click on next

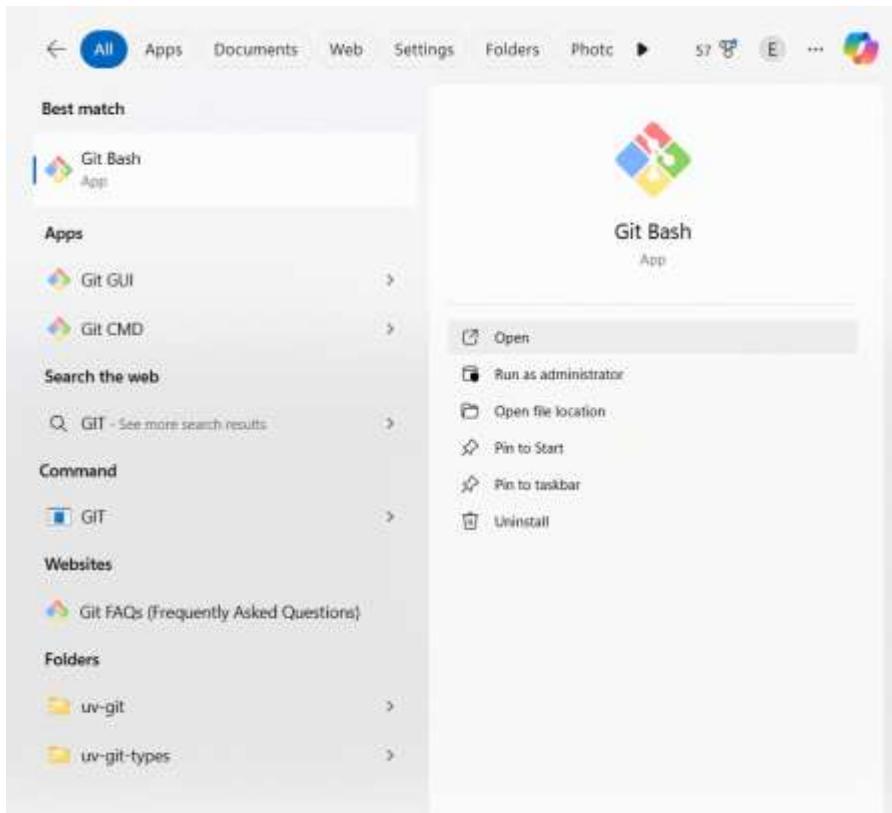


Step-5: Select the components and click next



Git bash:

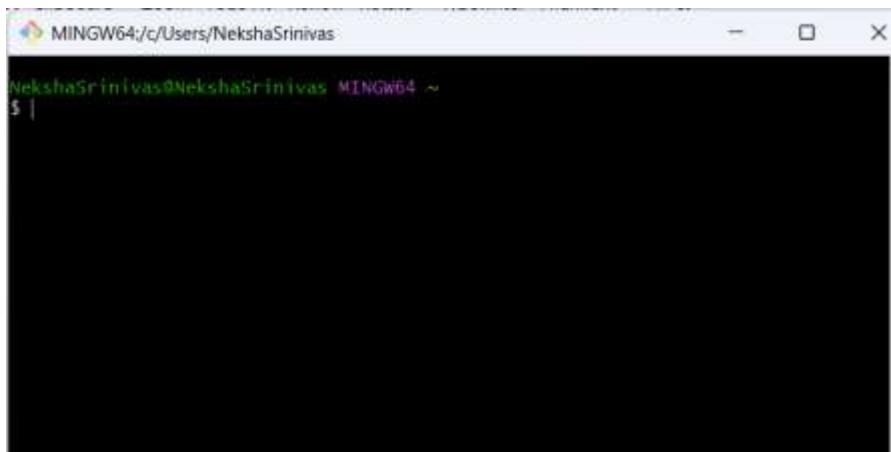
Step-1: Go to search bar and click git bash



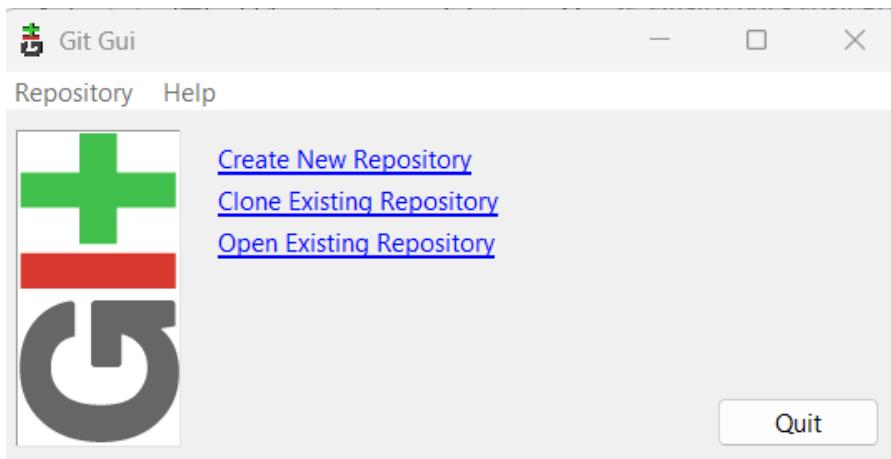
Step-2: Click on finish



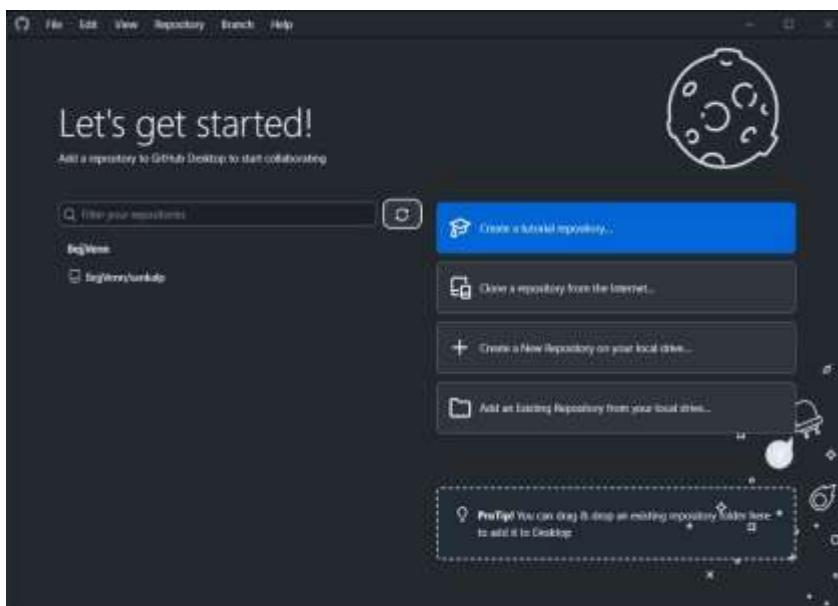
### Step-3: git bash interface



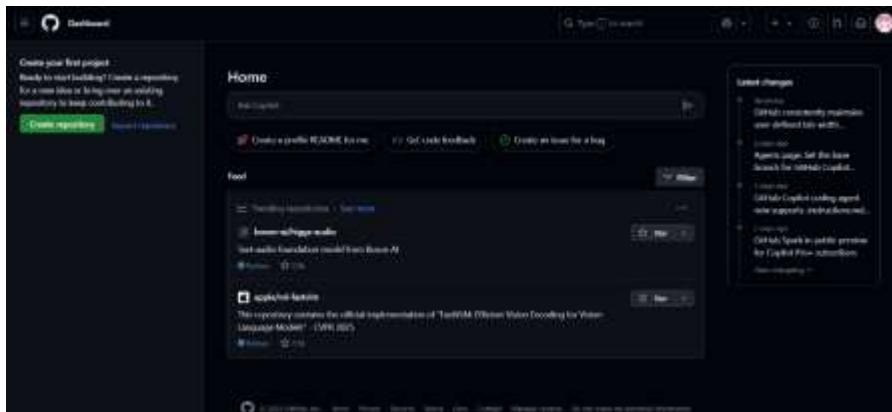
A screenshot of a terminal window titled "MINGW64". The window shows a black command line interface with the prompt "\$ |". The title bar also displays the path "c:/Users/NekshaSrinivas".



### GIT-ACCOUNT



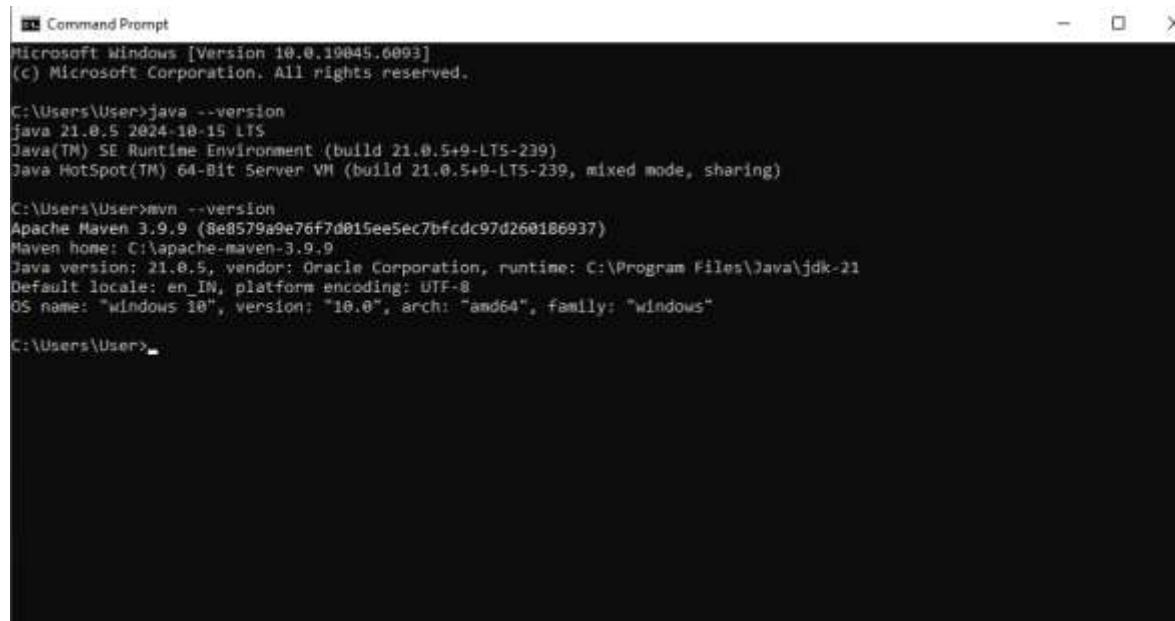
## GIT-ACCOUNT



## Tomcat

A screenshot of a web browser displaying the Apache Tomcat 9.0.98 welcome page. The URL is 'localhost:8080'. The page features a large green banner at the top stating 'If you're seeing this, you've successfully installed Tomcat. Congratulations!'. Below the banner is a cartoon cat logo and a list of recommended reading links: 'Security Considerations How-To', 'Manager Application How-To', and 'Clustering/Session Replication How-To'. To the right are buttons for 'Server Status', 'Manager App', and 'Host Manager'. The main content area is divided into three sections: 'Developer Quick Start' (with links to 'Tomcat Setup', 'First Web Application', 'Rekins A.M.A.', 'JDBC DataSources', 'Examples', 'Servlet Specifications', and 'Tomcat Versions'), 'Managing Tomcat' (with information about manager webapps and the location of the configuration file), 'Documentation' (links to 'Tomcat 9.0 Documentation', 'Tomcat 9.0 Configuration', and 'Tomcat Wiki'), and 'Getting Help' (links to 'FAQ and Mailing Lists' and descriptions of various mailing lists). At the bottom, there are links for 'Other Downloads', 'Other Documentation', 'Get Involved', 'Miscellaneous', and 'Apache Software Foundation'.

## Java and maven versions



```
Microsoft Windows [Version 10.0.19045.6093]
(c) Microsoft Corporation. All rights reserved.

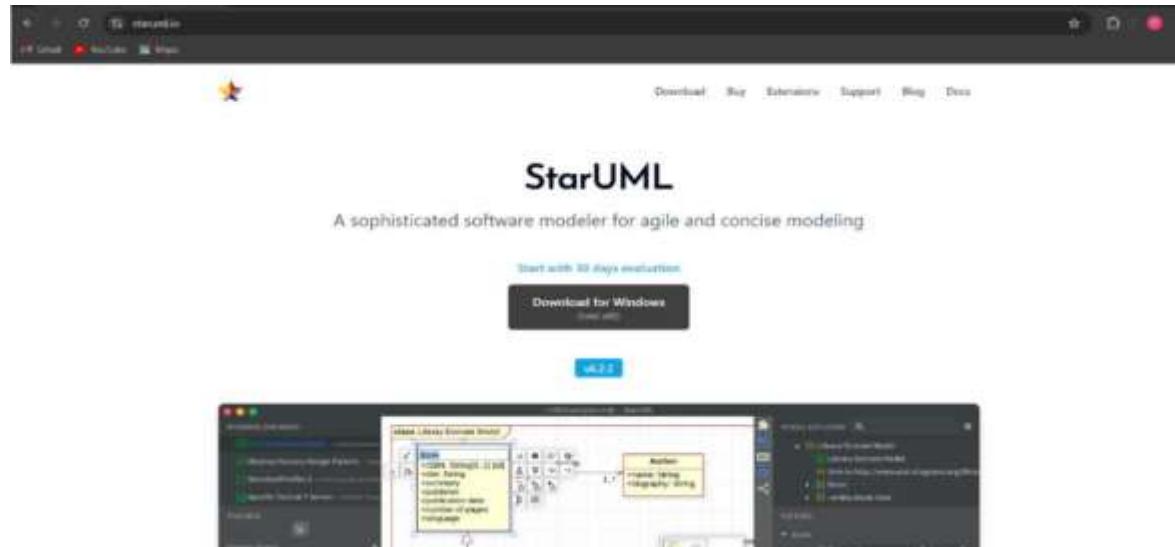
C:\Users\User>java --version
java 21.0.5 2024-10-15 LTS
Java(TM) SE Runtime Environment (build 21.0.5+9-LTS-239)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.5+9-LTS-239, mixed mode, sharing)

C:\Users\User>maven --version
Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfcdc97d260186937)
Maven home: C:\Apache-Maven-3.9.9
Java version: 21.0.5, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-21
Default locale: en_IN, platform encoding: UTF-8
OS name: "Windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\Users\User>
```

## StarUML INSTALLATION

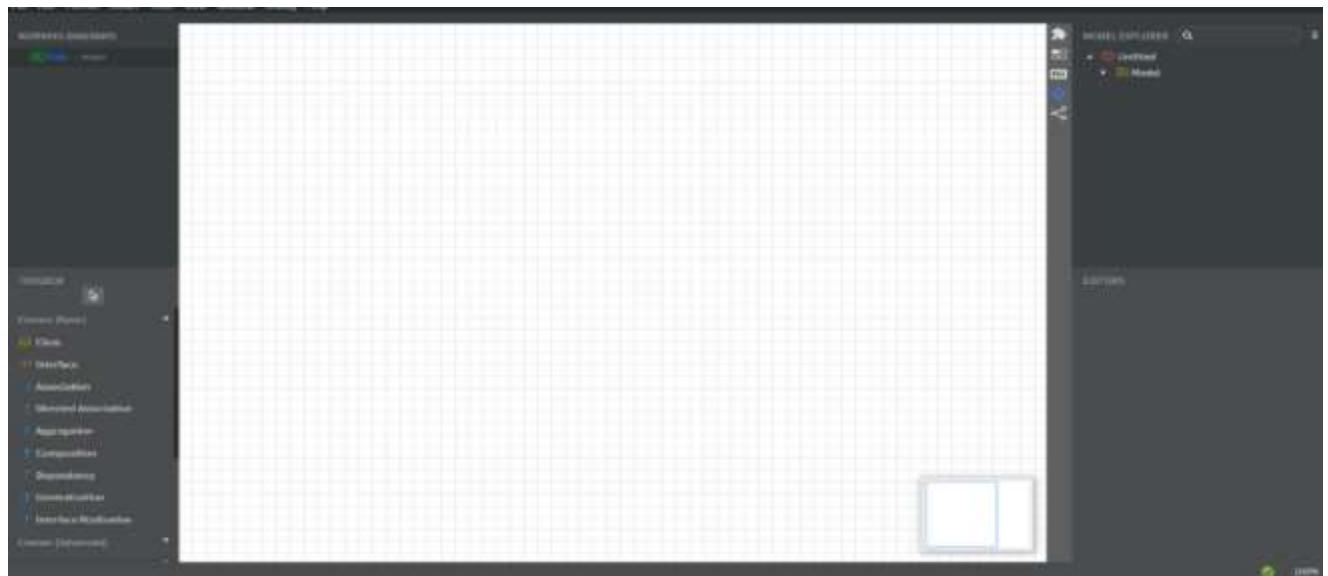
### Step-1: Go to startuml website



Step-2: from the given options select the suitable one for your system



Step-3: Interface



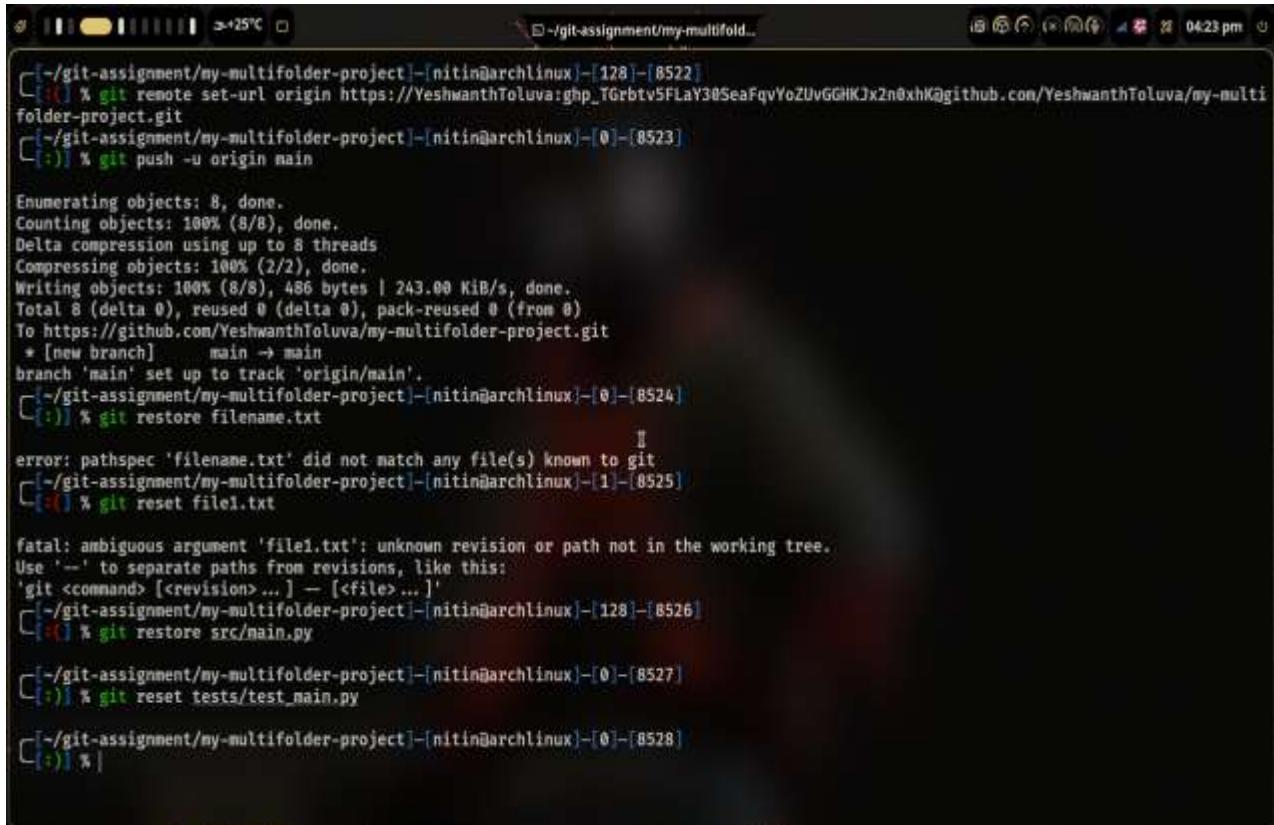
## **2. Exploring git local and remote commands on the multi-folder project**

Github Global Configuration:

```
C:\Users\NekshaSrinivas>git config --global --list
core.editor="C:\Users\NekshaSrinivas\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait
user.name=Edigirala-Neksha
user.email=edigiralaneksha@gmail.com

C:\Users\NekshaSrinivas>
```

Git Push to GitHub Public Repository with Remote Set



The screenshot shows a terminal window on an Arch Linux desktop environment. The title bar indicates the window is for a git session. The terminal content shows the following sequence of commands and their output:

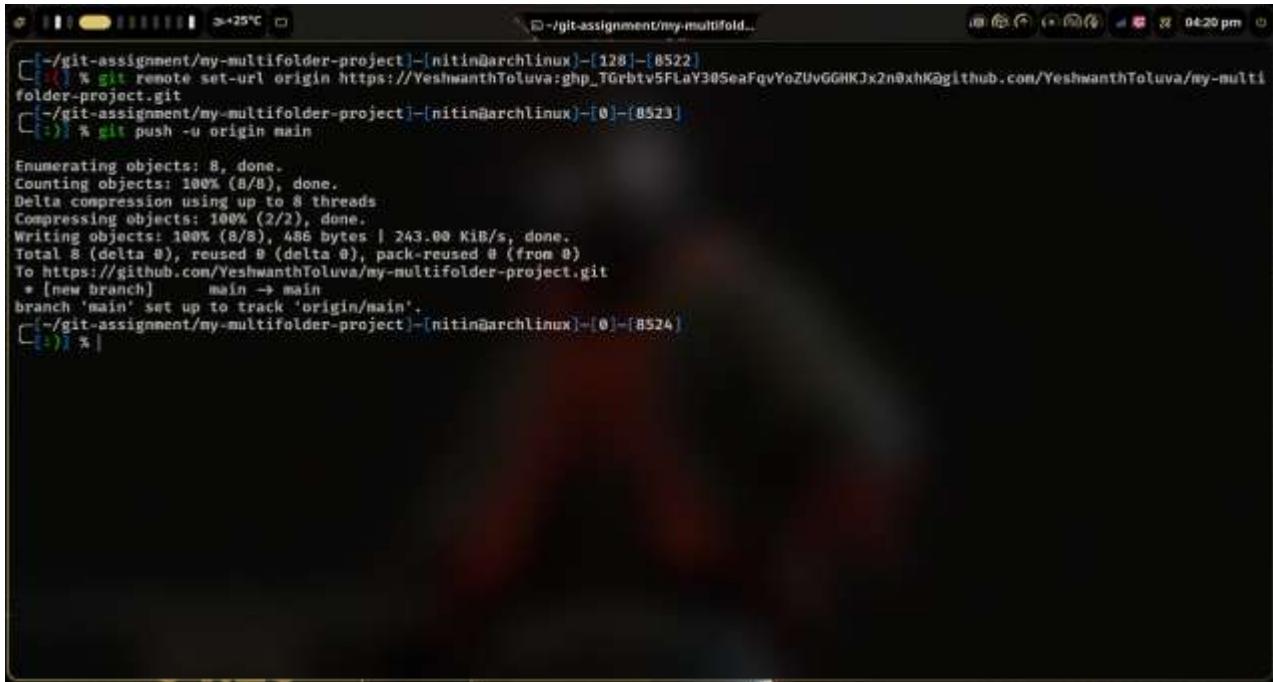
```
[~/git-assignment/my-multifolder-project]-(nitin@archlinux)-[128]-[8522]
[::] % git remote set-url origin https://YeshwanthToluva:ghp_TGrbtv5FLaY30SeafqvYoZUvGGHKJx2n0xhK@github.com/YeshwanthToluva/my-multi
[~/git-assignment/my-multifolder-project]-(nitin@archlinux)-[0]-[8523]
[::] % git push -u origin main

Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (8/8), 486 bytes | 243.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/YeshwanthToluva/my-multifolder-project.git
 * [new branch]      main → main
branch 'main' set up to track 'origin/main'.
[~/git-assignment/my-multifolder-project]-(nitin@archlinux)-[0]-[8524]
[::] % git restore filename.txt
error: pathspec 'filename.txt' did not match any file(s) known to git
[~/git-assignment/my-multifolder-project]-(nitin@archlinux)-[1]-[8525]
[::] % git reset file1.txt

fatal: ambiguous argument 'file1.txt': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision> ...] -- [<file> ... ]'
[~/git-assignment/my-multifolder-project]-(nitin@archlinux)-[128]-[8526]
[::] % git restore src/main.py

[~/git-assignment/my-multifolder-project]-(nitin@archlinux)-[0]-[8527]
[::] % git reset tests/test_main.py
[~/git-assignment/my-multifolder-project]-(nitin@archlinux)-[0]-[8528]
[::] %
```

## Scenario-Based Git Commands: Discarding and Unstaging Changes

A screenshot of a terminal window titled "git-assignment/my-multifolder...". The terminal shows the following command sequence:

```
[~/git-assignment/my-multifolder-project]-(nitin@archlinux)-[128]-[8522]
[::] % git remote set-url origin https://YeshwanthToluva:ghp_TGrbtv5FLaY30SeaFqvYeZUvGGHKJx2n0xhKa@github.com/YeshwanthToluva/my-multi
[~/git-assignment/my-multifolder-project]-(nitin@archlinux)-[0]-[8523]
[::] % git push -u origin main

Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (8/8), 486 bytes | 243.00 KiB/s, done.
total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/YeshwanthToluva/my-multifolder-project.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
[~/git-assignment/my-multifolder-project]-(nitin@archlinux)-[0]-[8524]
[::] %
```

1. You've cloned a repository and made some changes to a local branch. Now you want to push these changes to the remote repository, but you're getting an error saying "rejected - non-fast-forward." How would you resolve this?

This error occurs when the remote branch has changes that your local branch doesn't. To resolve it:

**git pull --rebase origin <branch-name>**

This rebases your local changes on top of the latest remote changes. After resolving any conflicts, push your changes:

2. You've been working on a feature branch, and now you need to push it to the remote repository. However, the remote repository already has a main branch. How do you push your feature branch without affecting the main branch?

You can push your feature branch independently:

**git push origin feature/feat-1**

This creates a new remote branch and does not affect the main branch.

3. You cloned a remote repository, but after a while, the repository's structure changed and new branches were added. How would you keep your local repository updated with the latest changes from the remote repository?

Use the following commands:

**git fetch origin**

This updates your local copy with all branches and changes from the remote. You can then check out new branches using:

**git checkout branch-name**

4. A colleague has pushed some changes to the main branch, but you have local changes in the same branch. You want to pull their changes, but you want to avoid merge conflicts. What steps would you take?

Use rebase to integrate their changes on top of your work:

**git stash # Temporarily store your changes**

**git pull --rebase origin main**

**git stash pop # Apply your changes on top**

This reduces the chance of conflicts and keeps history clean.

5. You accidentally pushed a sensitive file (e.g., API keys) to the remote repository. How would you fix this situation?

Steps to remove the sensitive data:

**Remove the file and commit:**

**git rm --cached path/to/file**

**git commit -m "Remove sensitive file"**

**git push origin main**

If the secret is in history, use git filter-branch or BFG Repo-Cleaner to rewrite history:

**git filter-branch --force --index-filter \**

**"git rm --cached --ignore-unmatch path/to/file" \**

```
--prune-empty --tag-name-filter cat -- --all
```

Force push and rotate the secret.

6. You're working on a feature branch, and your manager requests that you integrate the latest changes from main into your feature branch. What steps would you take?

Use rebase or merge:

Rebase:

```
git checkout feature/your-feature
```

```
git fetch origin
```

```
git rebase origin/main
```

7. You cloned a remote repository, but later you find that you need to push your changes to a different remote repository. How do you configure your local repository to push to this new remote?

Then push your changes:

```
git push origin branch-name
```

8. After running git pull, you notice that your local branch is behind the remote branch. How would you proceed to bring your local branch up to date without losing your local changes?

Use stash or rebase:

```
git stash
```

```
git pull --rebase origin branch-name
```

```
git stash pop
```

This ensures a clean rebase and retains your changes.

9. You're working on a project with multiple collaborators, and you notice that your local changes conflict with changes that have been pushed by others. How would you resolve the conflicts?

Pull the latest changes:

**git pull origin branch-name**

Git will highlight conflicts. Open the files, manually resolve the <<<<<, =====, and >>>>> markers.

Mark as resolved and commit:

**git add .**

**git commit**

10. You've pushed a feature branch to a remote repository, but now you need to delete the branch from the remote. How would you do that?

Use the following command:

**git push origin --delete feature/branch-name**

This will remove the branch from the remote repository.

### 3. Collaborative coding using git

#### GitHub Organization Members Page - se-lab-kmit Team Overview

The screenshot shows the GitHub Organization Members Page for the 'se-lab-kmit' organization. The page lists three members:

- DigitalHatha**: Member, Joined 3 hours ago
- YashwanthToluva**: Member, Joined 3 hours ago
- Vamsith-666**: Member, Joined 3 hours ago

#### GitHub Repository Overview - LocalHunt-01 Private Repository

The screenshot shows the GitHub Repository Overview for the 'LocalHunt-01' repository. The repository is private and contains the following details:

- Code**: main · Branch · Tags
- Issues**: 0
- Pull requests**: 0
- Actions**: 1 Commit
- Projects**: 0
- Security**: 0
- Insights**: 0
- Settings**: 0

The repository has one commit from **YashwanthToluva** titled "Initial commit". The README file contains the text "TEsting the private repo of the organization". The repository has 0 stars, 0 forks, and 0 releases.

## Terminal Git Clone Operations - LocalHunt-01 Repository Setup

The terminal window shows the following command being run:

```
[~] % cd ~/Documents/3rd yr/se lab  
[~] % git clone https://YeshwanthToluva:ghp_GUcawTFSufiiXUbmrjRjHG781FPA2b4gYu8h@github.com/se-lab-kmit/LocalHunt-01.git
```

The output of the command is:

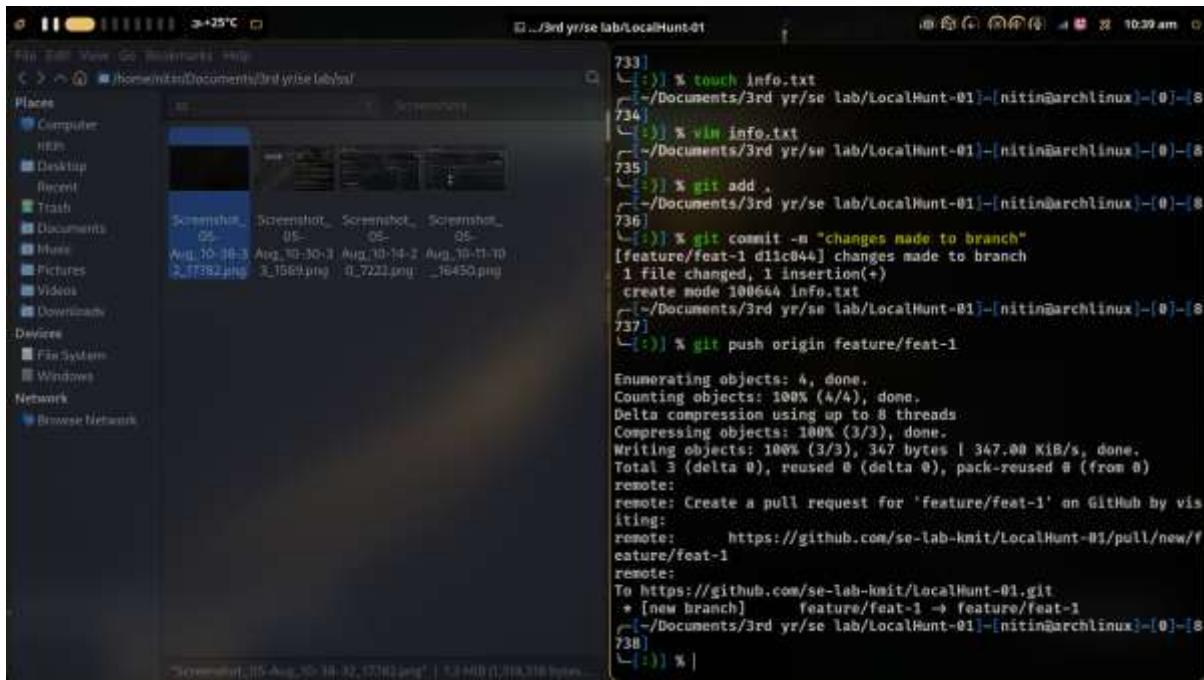
```
zsh: no such file or directory: @github.com/se-lab-kmit/LocalHunt-01.git  
[~] % git clone https://YeshwanthToluva:ghp_GUcawTFSufiiXUbmrjRjHG781FPA2b4gYu8h@github.com/se-lab-kmit/LocalHunt-01.git  
Cloning into 'LocalHunt-01'...  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Compressing objects: 100% (2/2), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
Receiving objects: 100% (3/3), done.  
[~] %
```

## Git Branch Operations - Feature Branch Creation and File Management

The terminal window shows the following sequence of commands:

```
[~] % git checkout -b feature/feat-1  
Switched to a new branch 'feature/feat-1'  
[~] % touch info.txt  
[~] % vim info.txt  
[~] % git add .  
[~] % git commit -m "changes made to branch"  
[feature/feat-1 d12c044] changes made to branch  
1 file changed, 1 insertion(+)  
create mode 100644 info.txt  
[~] %
```

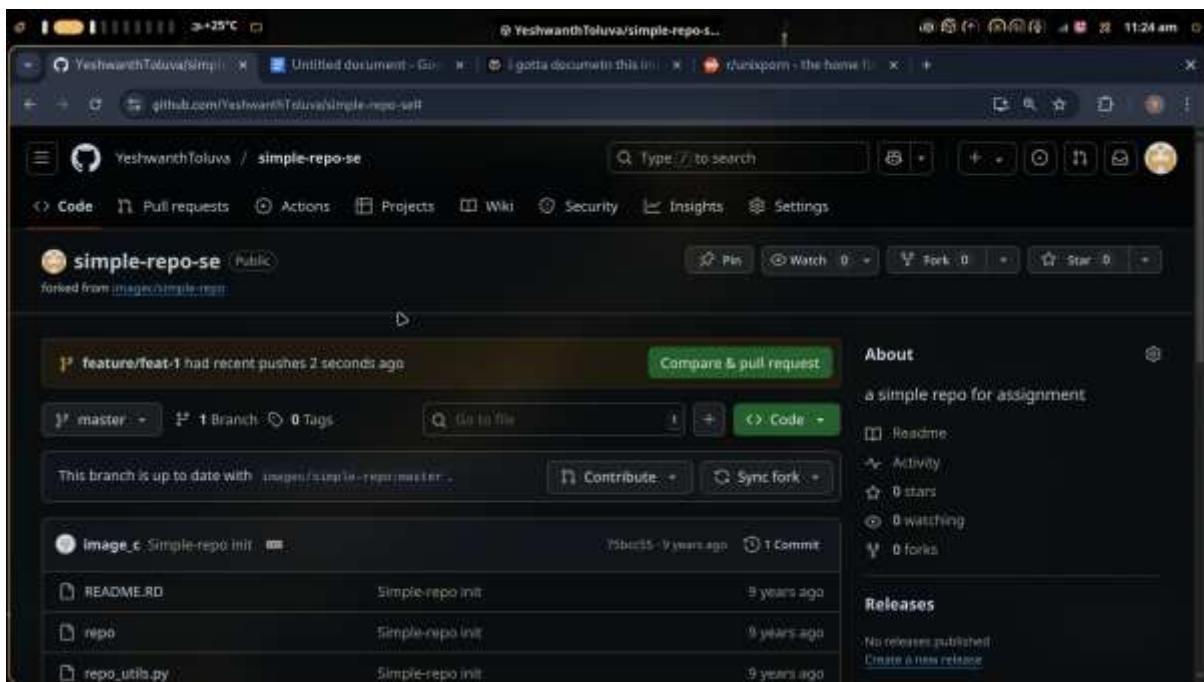
## Git Push and Pull Request Creation - Feature Branch Workflow



```
733] % touch info.txt
734] % vim info.txt
735] % git add .
736] % git commit -m "changes made to branch"
[feature/feat-1 d11c844] changes made to branch
 1 file changed, 1 insertion(+)
   create mode 100644 info.txt
737] % git push origin feature/feat-1

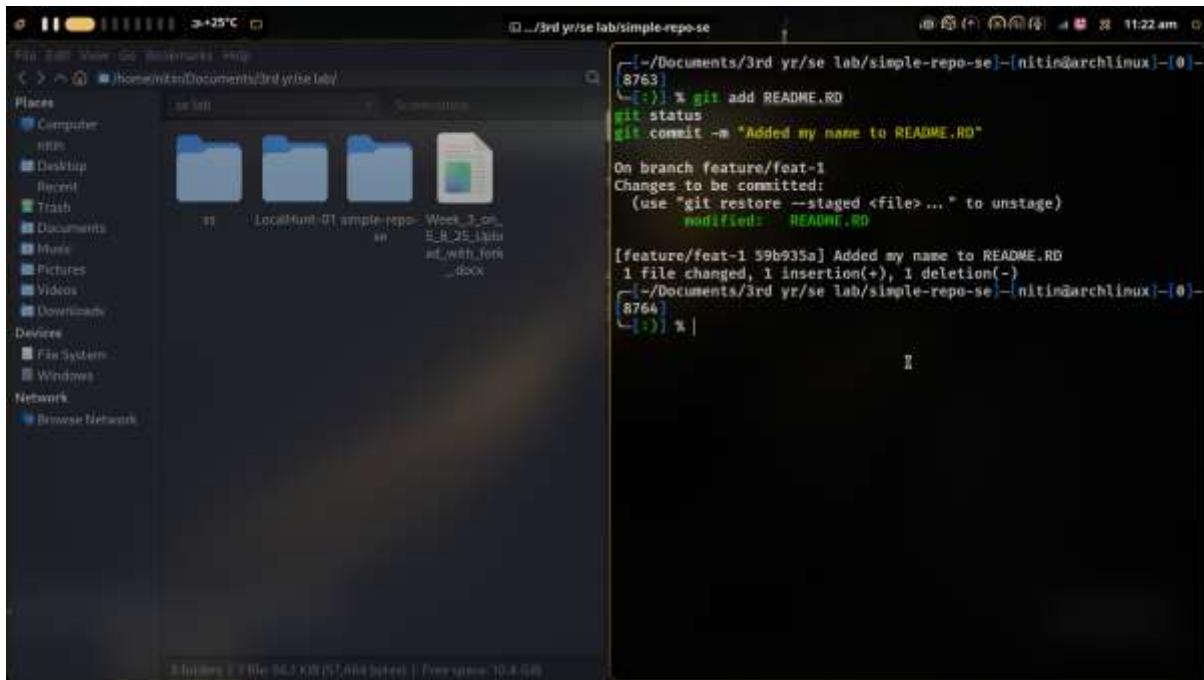
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 347 bytes | 347.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'Feature/feat-1' on GitHub by vis
iting:
remote:   https://github.com/se-lab-kmit/LocalHunt-01/pull/new/f
eature/feat-1
remote:
remote: To https://github.com/se-lab-kmit/LocalHunt-01.git
 * [new branch]      feature/feat-1 -> feature/feat-1
738] %
```

## GitHub Repository Fork - simple-repo-se Overview and Setup



The screenshot shows a GitHub repository page for `simple-repo-se`. The repository was forked from `image_c/simple-repo`. The `master` branch is selected. A green banner at the top indicates that the `feature/feat-1` branch has recent pushes. The repository has 0 stars, 0 forks, and 0 releases. The code listing shows four files: `image_c`, `README.RD`, `repo`, and `repo_utils.py`, all updated 9 years ago.

## Git Commit and Status - README.RD File Modifications in Feature Branch

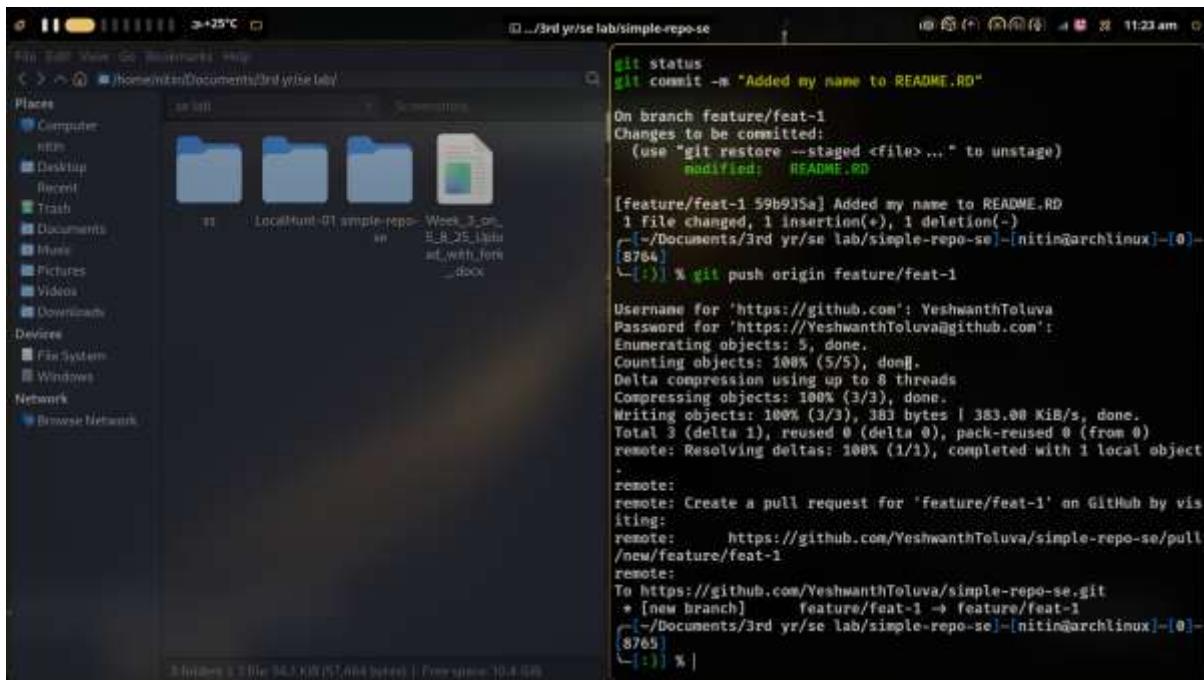


```
[~/Documents/3rd yr/se lab/simple-repo-se] [niting@archlinux] [8763]
[::]:% git add README.RD
git status
git commit -m "Added my name to README.RD"

On branch feature/feat-1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified: README.RD

[feature/feat-1 59b935a] Added my name to README.RD
1 file changed, 1 insertion(+), 1 deletion(-)
[~/Documents/3rd yr/se lab/simple-repo-se] [niting@archlinux] [8764]
[::]:%
```

## Git Push to Forked Repository - Feature Branch Upload and Pull Request Creation



```
[~/Documents/3rd yr/se lab/simple-repo-se] [niting@archlinux] [8764]
[::]:% git status
git commit -m "Added my name to README.RD"

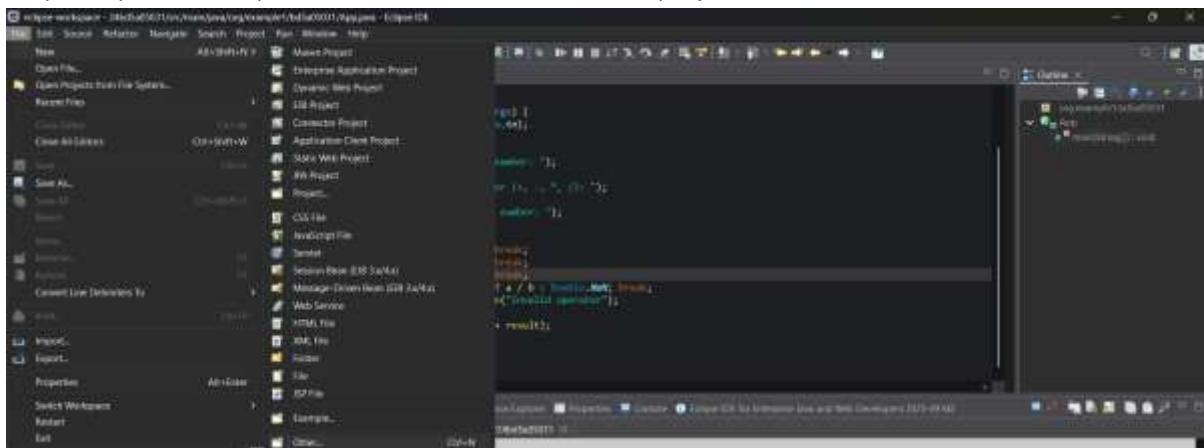
On branch feature/feat-1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified: README.RD

[feature/feat-1 59b935a] Added my name to README.RD
1 file changed, 1 insertion(+), 1 deletion(-)
[~/Documents/3rd yr/se lab/simple-repo-se] [niting@archlinux] [8764]
[::]:% git push origin feature/feat-1

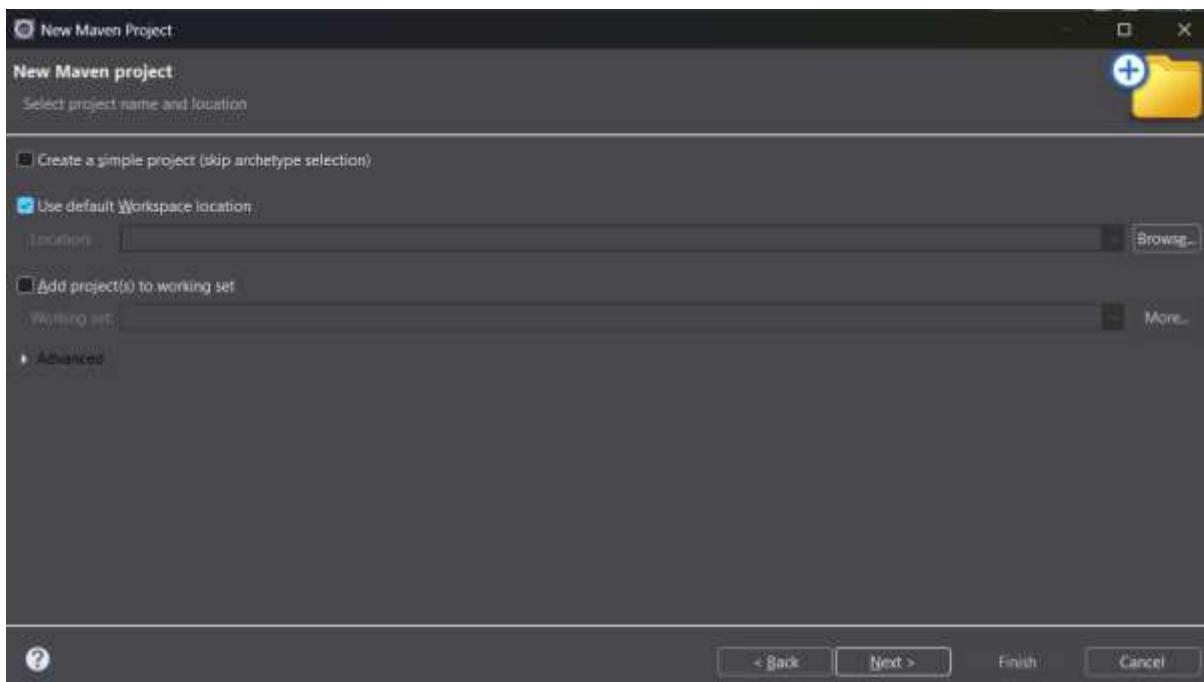
Username for 'https://github.com': YeshwanthToluva
Password for 'https://YeshwanthToluva@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 383 bytes | 383.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object
.
remote:
remote: Create a pull request for 'feature/feat-1' on GitHub by vis
iting:
remote:   https://github.com/YeshwanthToluva/simple-repo-se/pull
remote:   /new/feature/feat-1
remote:
To https://github.com/YeshwanthToluva/simple-repo-se.git
 * [new branch]      feature/feat-1 -> feature/feat-1
[~/Documents/3rd yr/se lab/simple-repo-se] [niting@archlinux] [8765]
[::]:%
```

#### **4. Build and package Java and Web applications using Maven**

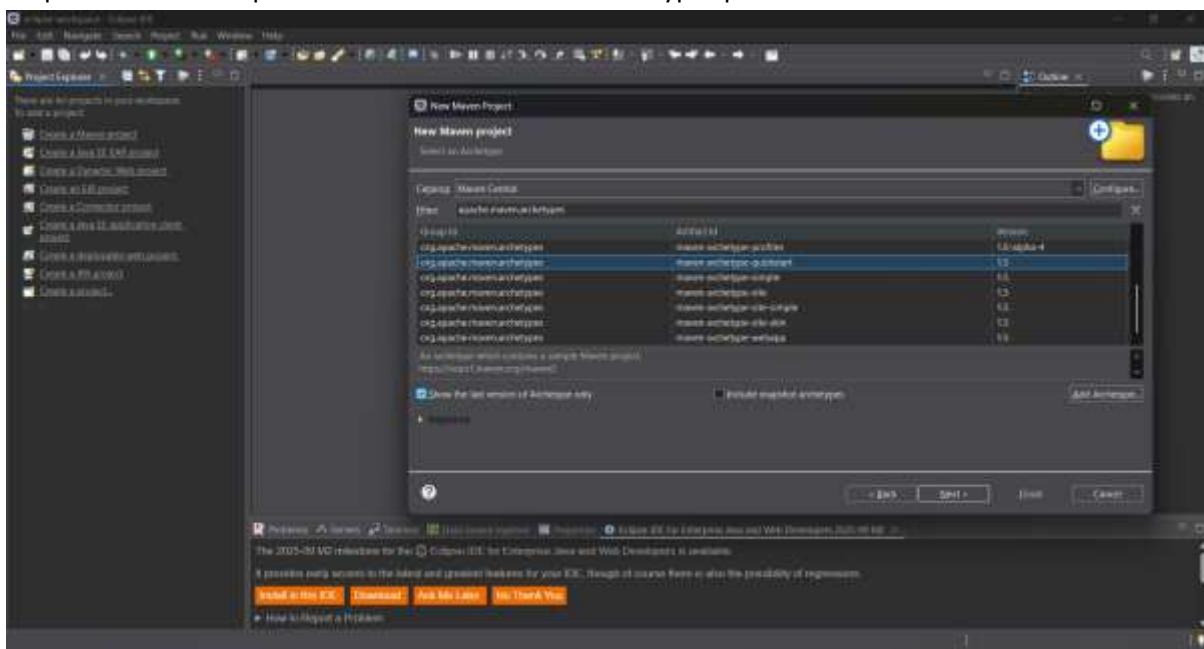
Step-1: Open the eclipse and click on file>new>Maven project



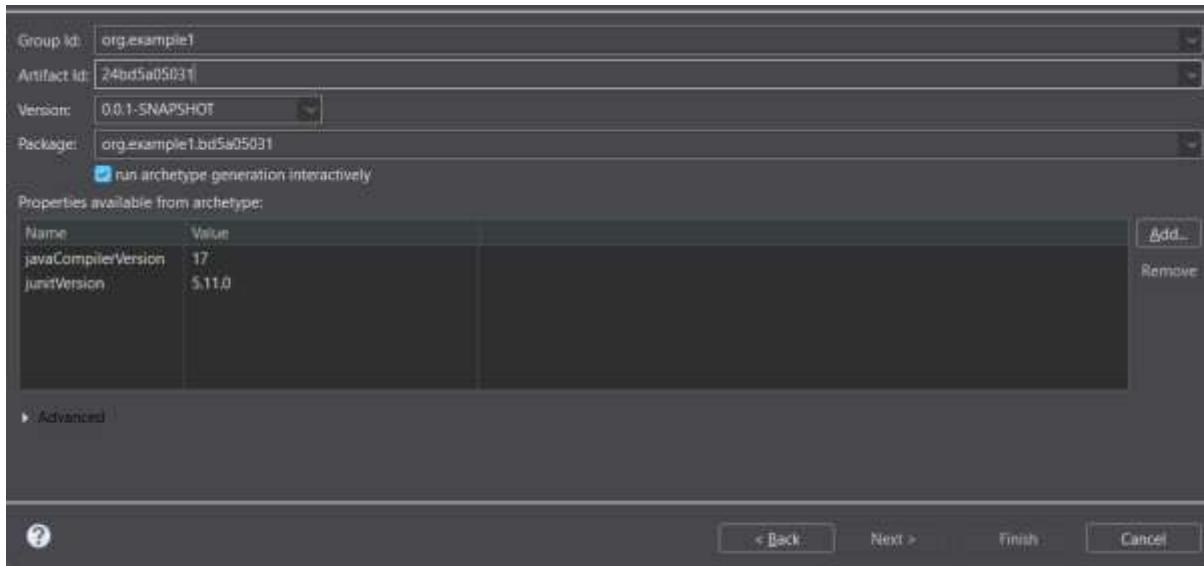
Step-2: select the default workspace and click on next



Step-3: in the filter option select the one maven-archetype-quickstart



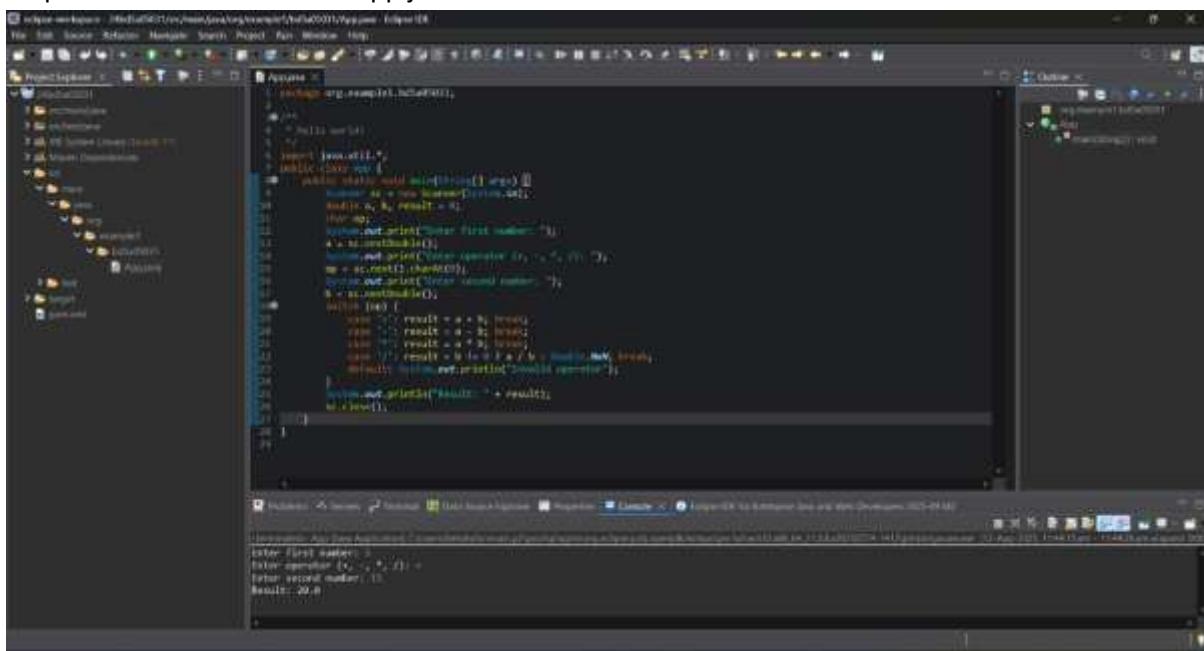
Step-4: give the Group Id and Artifact Id and click on next



Step-5: In the console the progress will be showed type y (refers to yes) and press enter

Step-6: BUILD SUCCESS will be shown

Step-6: write the code in the App.java file

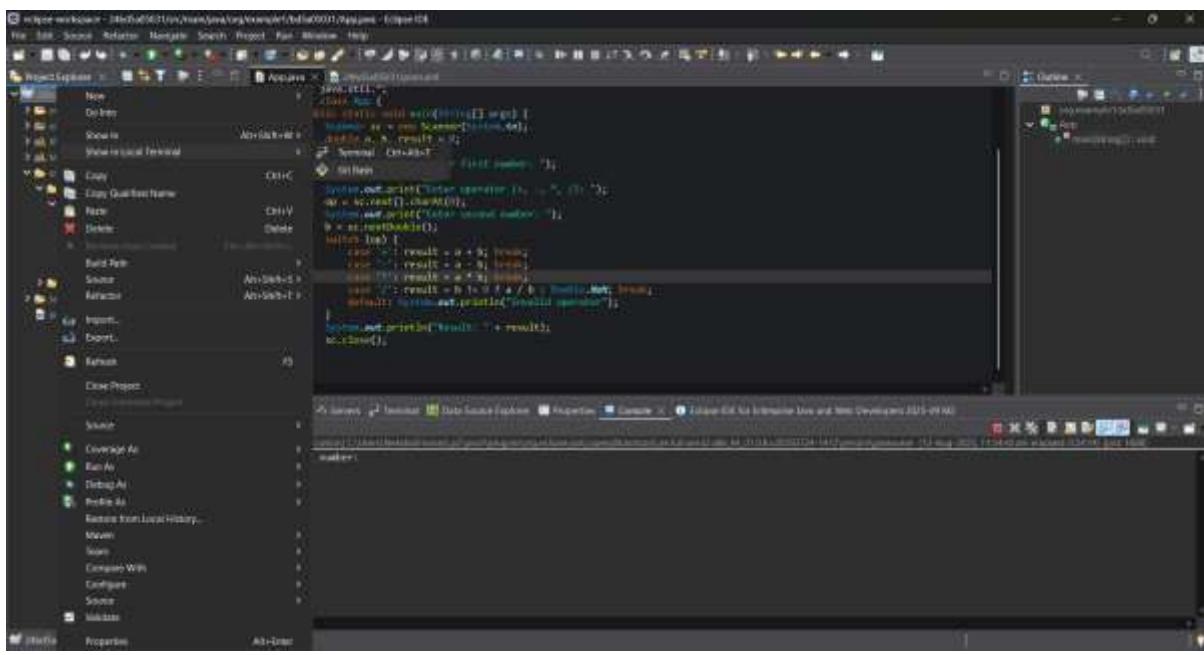


The screenshot shows the Eclipse IDE interface with the code editor open to the `App.java` file. The code implements a calculator with addition, subtraction, multiplication, division, and modulus operations. It includes a main method to demonstrate its usage.

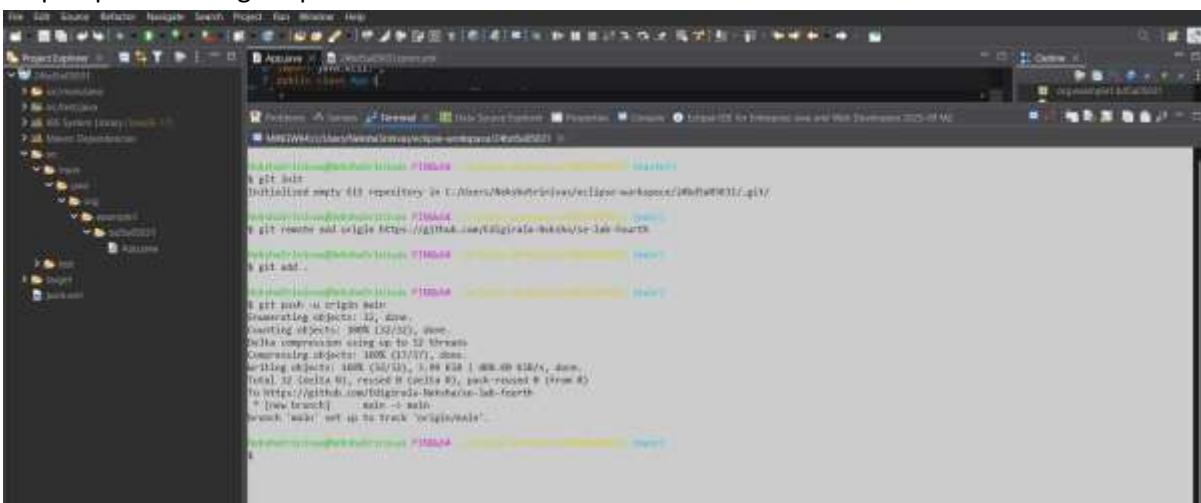
```
import java.util.Scanner;
public class App {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a = sc.nextInt();
        int b = sc.nextInt();
        System.out.println("First number: " + a);
        System.out.println("Second number: " + b);
        int result = a + b;
        System.out.println("Addition: " + result);
        result = a - b;
        System.out.println("Subtraction: " + result);
        result = a * b;
        System.out.println("Multiplication: " + result);
        result = a / b;
        System.out.println("Division: " + result);
        result = a % b;
        System.out.println("Modulus: " + result);
    }
}
```

The Eclipse interface includes a Project Explorer on the left, a code editor in the center, and a terminal or output window at the bottom showing sample input and output.

Step-7: right click on the root folder and select show in git bash



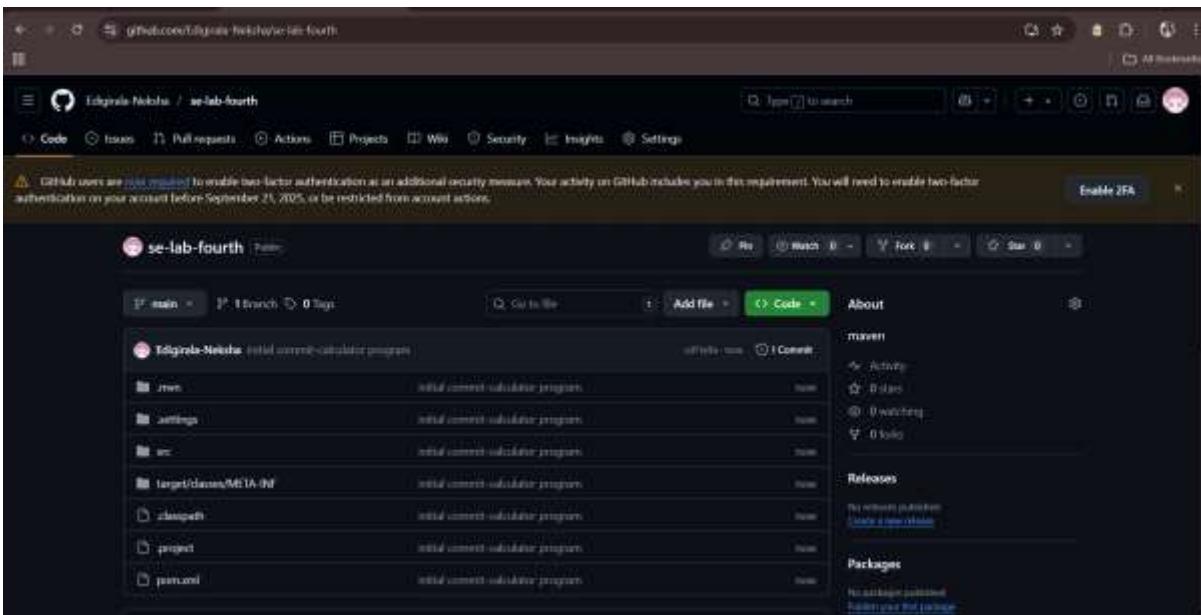
## Step-8: push to the git repo



```
git push -u origin main
Counting objects: 12, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), done.
Total 10 (delta 0), reused 0 deltas, pack-reused 0 (from 0)
remote: Resolving deltas: 100% (10/10), done.
To https://github.com/Edigirala-Neksha/se-lab-fourth
 * [new branch] main -> main
Branch 'main' set up to track 'origin/main'.
```

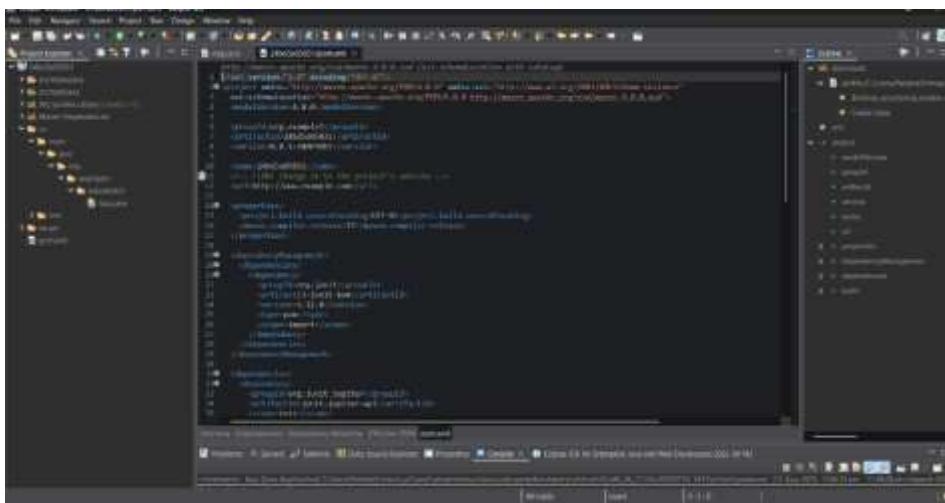
Git repo:

Git repo link: <https://github.com/Edigirala-Neksha/se-lab-fourth>



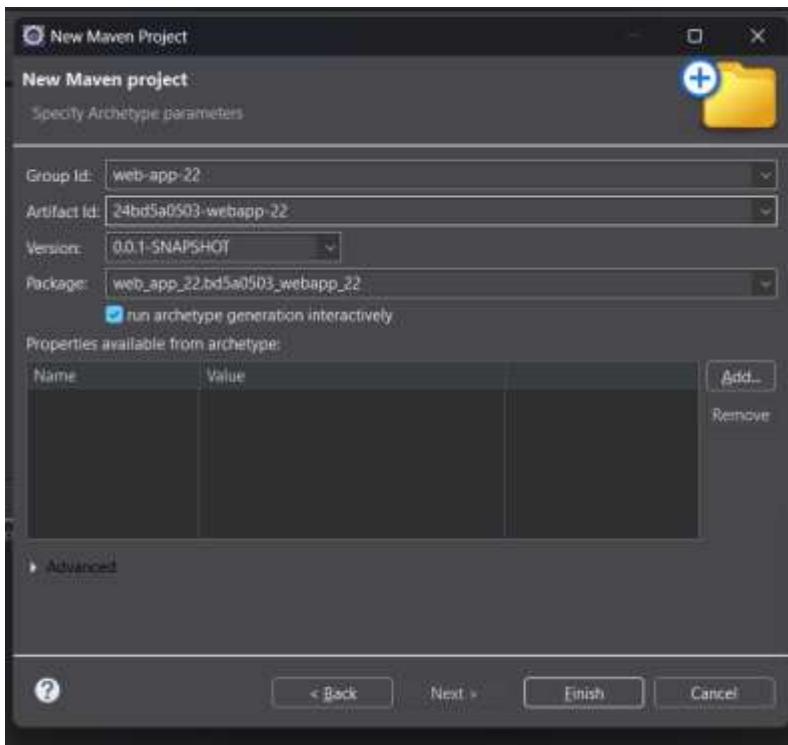
pom.xml file:

Shows the structure-



Creating maven-web project:

Step 1: Create a new maven project and give the details



Step 2: Click y to continue the creation of project

```
C:\Users\NekshaSrinivas\Downloads\maven-archetype-webapp-1.5\src\main\java\org\apache\maven\archetype\remote\archetype-catalog.xml (3370) at %3(M2)
Progress (1): 37/37 MB

Downloaded from central: https://repo.maven.apache.org/maven2/archetype-catalog.xml (3370) at %3(M2)
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archetypes:maven-archetype-webapp:1.5] found in catalog remote
[INFO] Using property: groupId = web-app-22
[INFO] Using property: artifactId = 24bd5a0503-webapp-22
[INFO] Using property: version = 0.0.1-SNAPSHOT
[INFO] Using property: package = web_app_22.bd5a0503_webapp_22
Confirm properties configuration:
groupId: web-app-22
artifactId: 24bd5a0503-webapp-22
version: 0.0.1-SNAPSHOT
package: web_app_22.bd5a0503_webapp_22
Y: y
```

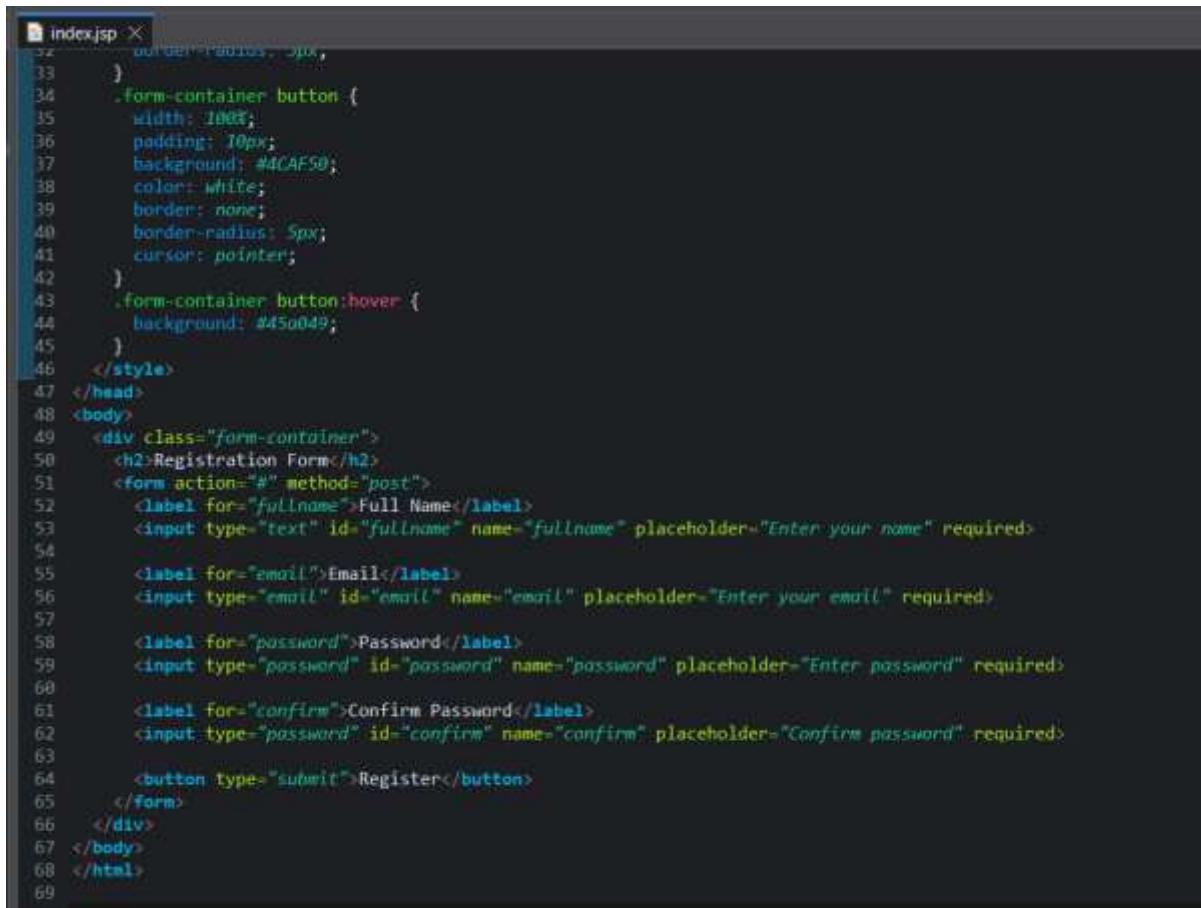
Step 3: If the build is success it will show the message

```
package: web_app_22.bd5a0503_webapp_22
Y: y
[INFO] -----
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-webapp:1.0
[INFO] -----
[INFO] Parameter: basedir, Value: C:\Users\NekshaSrinivas\eclipse-workspace
[INFO] Parameter: package, Value: web_app_22.bd5a0503_webapp_22
[INFO] Parameter: groupId, Value: web-app-22
[INFO] Parameter: artifactId, Value: 24bd5a0503-webapp-22
[INFO] Parameter: packageName, Value: web_app_22.bd5a0503_webapp_22
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a0503-webapp-22
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 43.500 s
[INFO] Finished at: 2025-09-02T19:20:41+05:30
[INFO] -----
```

Step 4: write the html code for the web page:

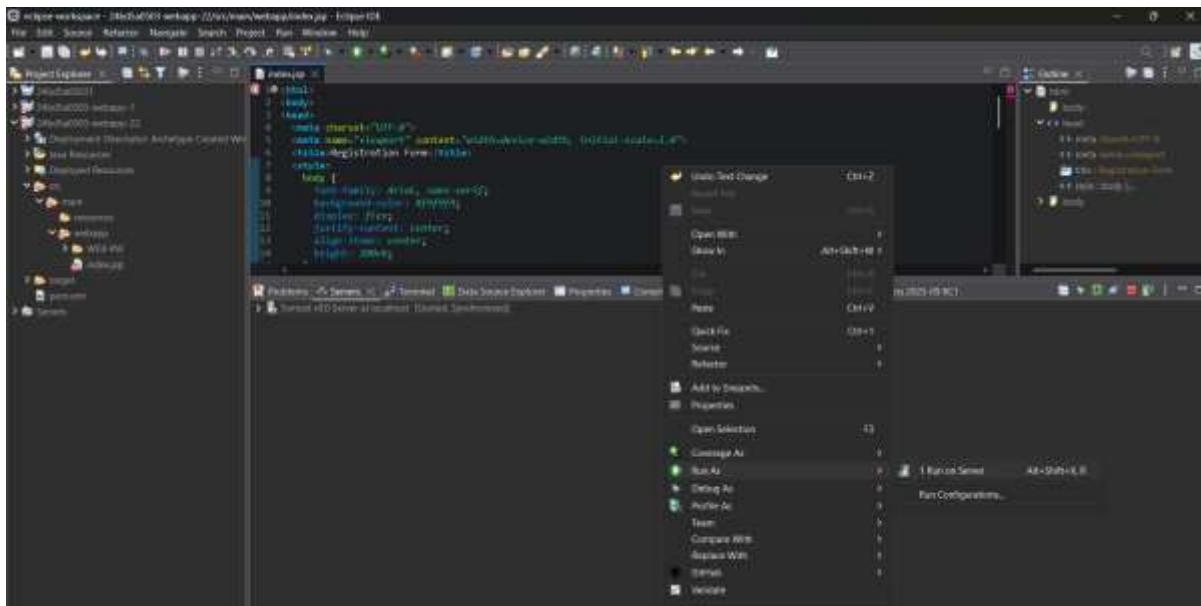
```
index.jsp: X
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <head>
5       <meta charset="UTF-8">
6       <meta name="viewport" content="width=device-width, initial-scale=1.0">
7       <title>Registration Form</title>
8       <style>
9         body {
10           font-family: Arial, sans-serif;
11           background-color: #f0f0f0;
12           display: flex;
13           justify-content: center;
14           align-items: center;
15           height: 100vh;
16         }
17         .form-container {
18           background: #fff;
19           padding: 20px 30px;
20           border-radius: 10px;
21           box-shadow: 0 4px 16px rgba(0,0,0,0.1);
22           width: 300px;
23         }
24         .form-container h2 {
25           text-align: center;
26           margin-bottom: 20px;
27         }
28         .form-container input {
29           width: 100%;
30           padding: 10px;
31           margin: 5px 0;
32           border: 1px solid #ccc;
33           border-radius: 5px;
34         }
35         .form-container button {
36           width: 100%;
37           padding: 10px;
38           background: #4CAF50;
39           color: white;
40           border: none;
41           cursor: pointer;
42           font-size: 1em;
43         }
44         .form-container button:hover {
45           background-color: #45B7D1;
46         }
```

Web-page:



```
index.jsp X
1 <head>
2     <style>
3         .form-container {
4             width: 100px;
5             padding: 10px;
6             background: #4CAF50;
7             color: white;
8             border: none;
9             border-radius: 5px;
10            cursor: pointer;
11        }
12        .form-container button:hover {
13            background: #450049;
14        }
15    </style>
16 </head>
17 <body>
18     <div class="form-container">
19         <h2>Registration Form</h2>
20         <form action="#" method="post">
21             <label for="fullname">Full Name</label>
22             <input type="text" id="fullname" name="fullname" placeholder="Enter your name" required>
23
24             <label for="email">Email</label>
25             <input type="email" id="email" name="email" placeholder="Enter your email" required>
26
27             <label for="password">Password</label>
28             <input type="password" id="password" name="password" placeholder="Enter password" required>
29
30             <label for="confirm">Confirm Password</label>
31             <input type="password" id="confirm" name="confirm" placeholder="Confirm password" required>
32
33             <button type="submit">Register</button>
34         </form>
35     </div>
36 </body>
37 </html>
38
```

Step 5: Select run on server

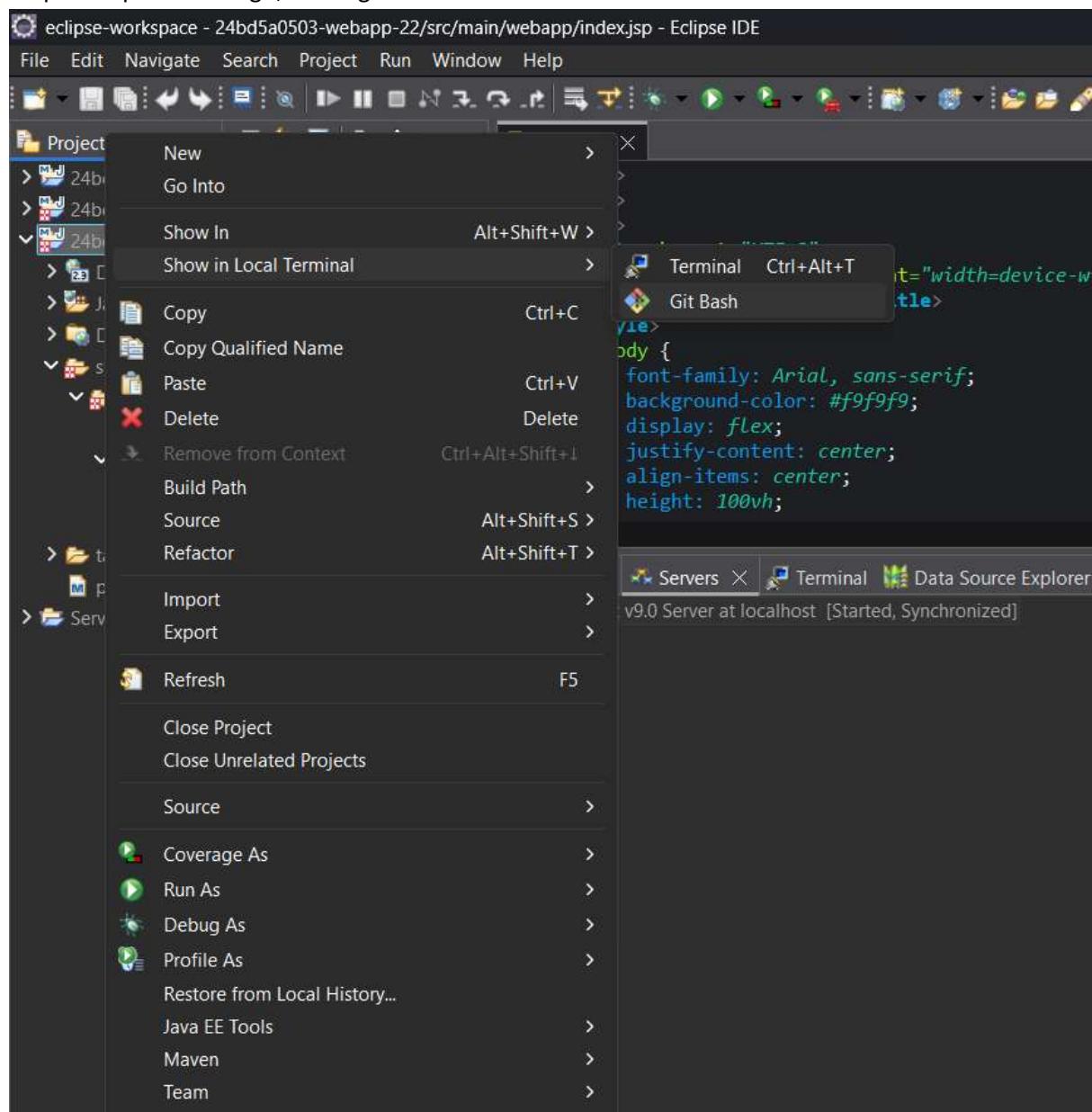


Step 6: It will show the following output:

A screenshot of a web browser window displaying a registration form. The browser's address bar shows the URL: `localhost:8080/24bd5a0503-webapp-22/index.jsp`. The page title is "Registration Form". The form consists of five input fields: "Full Name" (placeholder: "Enter your name"), "Email" (placeholder: "Enter your email"), "Password" (placeholder: "Enter password"), and "Confirm Password" (placeholder: "Confirm password"). Below the inputs is a green "Register" button.

Registration Form	
Full Name	<input type="text" value="Enter your name"/>
Email	<input type="text" value="Enter your email"/>
Password	<input type="password" value="Enter password"/>
Confirm Password	<input type="password" value="Confirm password"/>
<input type="button" value="Register"/>	

Step 7: To push it into git, select git bash from show in local terminal



## Step 8: use the command of git to push the maven web project

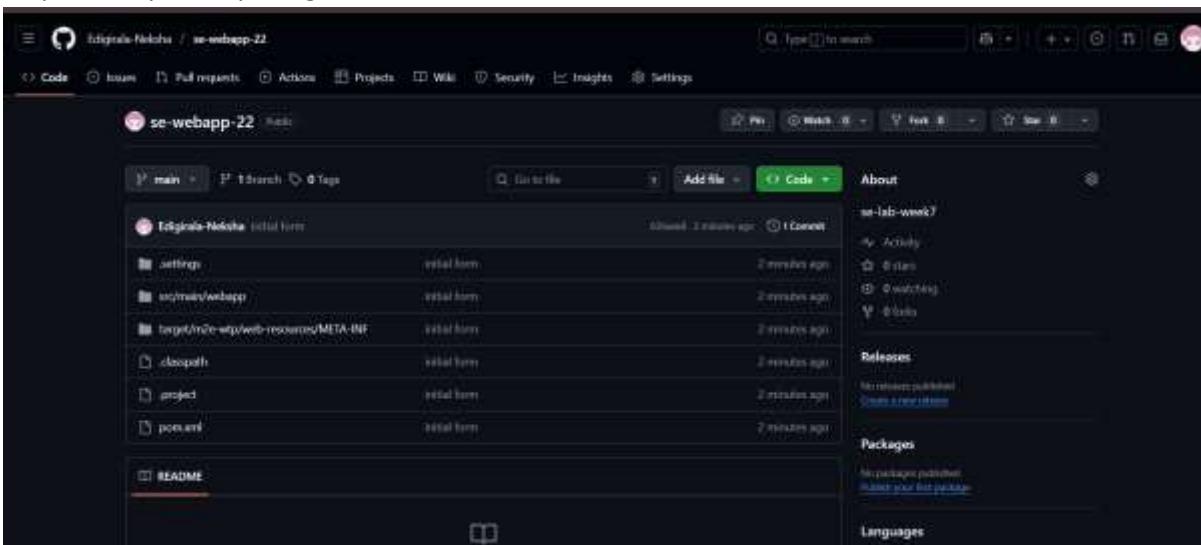
```
MINGW64:/c/Users/NekshaSrinivas/eclipse-workspace/24bd5a05.. NekshaSrinivas$ git init
Initialized empty Git repository in C:/Users/NekshaSrinivas/eclipse-workspace/24bd5a053-webapp-22/.git/
MINGW64:/c/Users/NekshaSrinivas/eclipse-workspace/24bd5a053-webapp-22 (main) NekshaSrinivas$ git add .
MINGW64:/c/Users/NekshaSrinivas/eclipse-workspace/24bd5a053-webapp-22 (main) NekshaSrinivas$ git commit -m "initial form"
[main (root-commit) 63eaeef] initial form
 16 files changed, 254 insertions(+)
 create mode 100644 .classpath
 create mode 100644 .project
 create mode 100644 .settings/.jsdtscope
 create mode 100644 .settings/crg.eclipse.jdt.core.prefs
 create mode 100644 .settings/crg.eclipse.wst.core.prefs
 create mode 100644 .settings/crg.eclipse.wst.commonn.component
 create mode 100644 .settings/crg.eclipse.wst.commonn.project.facet.core.xml
 create mode 100644 .settings/crg.eclipse.wst.jsdt.ui.superType.container
 create mode 100644 .settings/crg.eclipse.wst.jsdt.ui.superType.name
 create mode 100644 .settings/crg.eclipse.wst.validation.prefs
 create mode 100644 pom.xml
 create mode 100644 src/main/webapp/META-INF/web.xml
```

```
MINGW64:/c/Users/NekshaSrinivas/eclipse-workspace/24bd5a053-webapp-22 (main) NekshaSrinivas$ git branch
* main

MINGW64:/c/Users/NekshaSrinivas/eclipse-workspace/24bd5a053-webapp-22 (main) NekshaSrinivas$ git push origin main
Enumerating objects: 29, done.
Counting objects: 100% (29/29), done.
Delta compression using up to 12 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (29/29), 4.43 KiB | 283.00 KiB/s, done.
Total 29 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Edigirala-Neksha/se-webapp-22.git
 * [new branch]      main -> main

MINGW64:/c/Users/NekshaSrinivas/eclipse-workspace/24bd5a053-webapp-22 (main) NekshaSrinivas$
```

## Step 9: verify the repo in git hub



## **5. Docker CLI commands**

### **Installing Docker and Setting up Nginx**

#### **Introduction**

**Docker** is a platform that allows us to run applications inside lightweight containers. Containers are isolated environments that include everything needed to run an application. This makes it easy to set up and deploy software without worrying about dependencies or configurations on the host system.

In this task, we used Docker to run an **Ubuntu container**, install **nginx** inside it, and serve a customized homepage

#### **Step 1: Pulling the Ubuntu Image**

First, we pulled the latest Ubuntu image from Docker Hub.

```
PS C:\Users\NekshaSrinivas> docker --version
Docker version 28.3.2, build 578ccf6
PS C:\Users\NekshaSrinivas> cd SE-1
PS C:\Users\NekshaSrinivas\SE-1> docker --version
Docker version 28.3.2, build 578ccf6
PS C:\Users\NekshaSrinivas\SE-1> docker pull ubuntu:latest
latest: Pulling from library/ubuntu
b71466b94f26: Pull complete
Digest: sha256:7c06e91f61fa88c08cc74f7e1b7c69ae24910d745357e0dfe1d2c0322aaf2
0-f9
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

#### **Step 2: Running the Container**

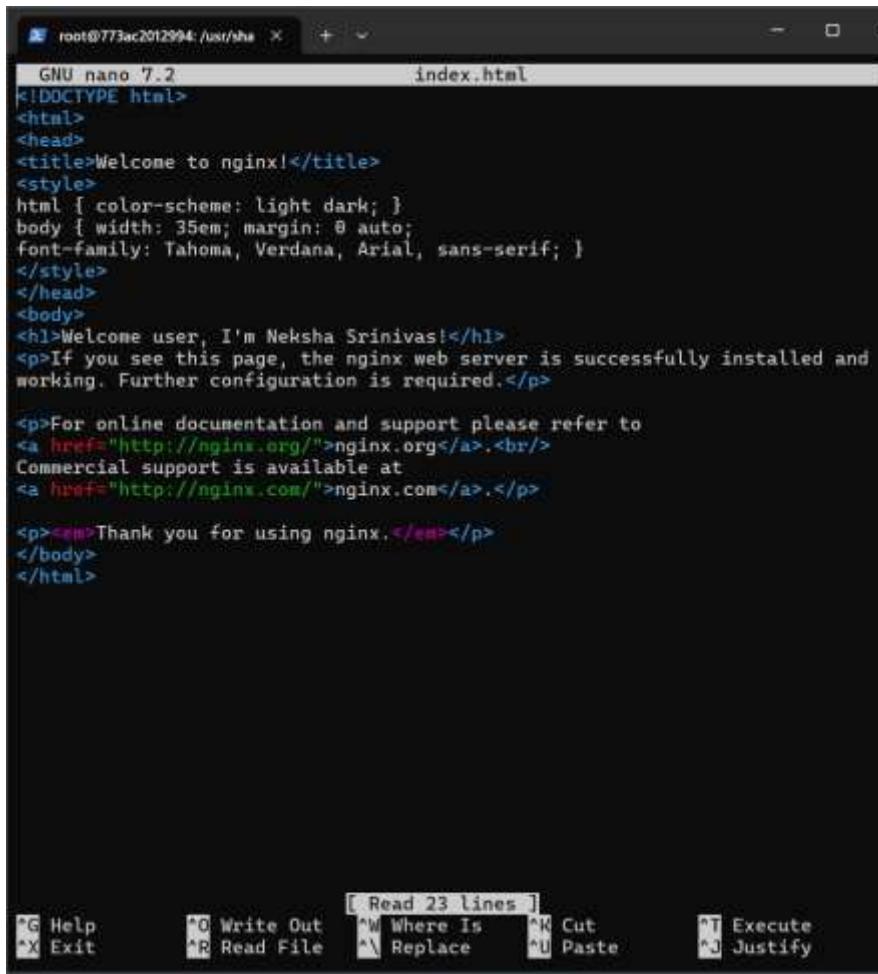
We created and started a new container named **myubuntu**, mapping port **3000** on the host to port **80** inside the container.

```
PS C:\Users\NekshaSrinivas\SE-1> docker run -it -p 9090:80 --name myubuntu1
ubuntu:latest
root@773ac2012994:/# apt update
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1135 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:5 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1355 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2047 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [23.0 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
```

Step 3: Installing Nginx and redirecting to index.html page to edit the content

```
Processing triggers for libc-bin (2.39-0ubuntu8.5) ...
root@773ac2012994:/# ls
bin          dev    lib     mnt   root  sbin usr-is-merged  /tmp
bin usr-is-merged etc    lib64  opt    run    srv      /usr
boot        home   media  proc   sbin   sys           var
root@773ac2012994:/# cd usr
root@773ac2012994:/usr# ls
bin games include lib lib64 libexec local sbin share src
root@773ac2012994:/usr# cd share
root@773ac2012994:/usr/share# ls
apport       gcc      pam
base-files   gdb      pam-configs
base-passwd  info     perl5
bash-completion info.dir pixmaps
bug          keyrings polkit-1
common-licenses libc-bin profile
debconf      libgcrypt20 profile.md5sums
debianutils  lintian  sensible-utils
dict         locale   staff-group-for-usr-local
doc          man      tabset
doc-base     menu    terminfo
dot.bashrc   misc    util-linux
dot.profile  motd    vim
dot.profile.md5sums networks
dpkg         nginx
root@773ac2012994:/usr/share# cd nginx
root@773ac2012994:/usr/share/nginx# ls
html  modules
root@773ac2012994:/usr/share/nginx# cd html
root@773ac2012994:/usr/share/nginx/html# ls
index.html
root@773ac2012994:/usr/share/nginx/html# nano index.html
bash: nano: command not found
root@773ac2012994:/usr/share/nginx/html# apt install nano
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  hunspell
The following NEW packages will be installed:
  nano
```

Step 4: navigate to index.html using command –“nano index.html” Changed the content of h1 tag

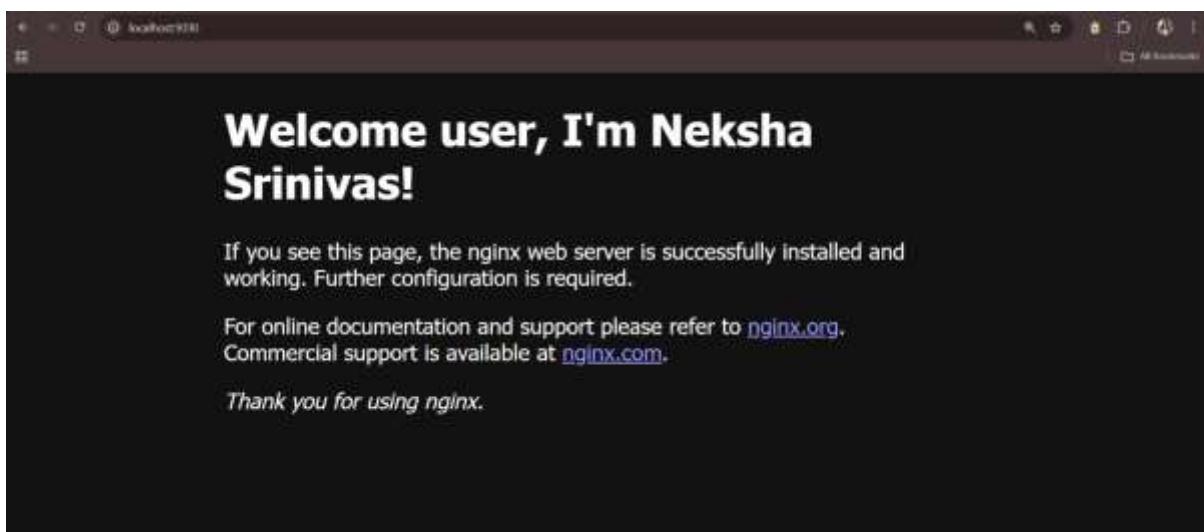


The screenshot shows a terminal window titled "root@773ac2012994: /usr/share/nginx/html". The file being edited is "index.html". The content of the file is as follows:

```
GNU nano 7.2          index.html
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome user, I'm Neksha Srinivas!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><strong>Thank you for using nginx.</strong></p>
</body>
</html>
```

The status bar at the bottom of the terminal window shows the following options: [ Read 23 lines ]. The keyboard shortcuts listed are: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^X Exit, ^R Read File, ^V Replace, ^U Paste, and ^J Justify.

Step 5: Viewing the page from local host



## 6. Docker

DOCKER IMAGE CREATION:

Image can be created in two ways:

1. Using Docker commit
2. Using docker file

Step 1: Created a new container of ubuntu so image can be created on that container

```
PS C:\Users\NekshaSrinivas\SE-1> docker run -it -p 9090:80 --name ubuntu-cont-1 ubuntu:latest
root@74098c332e58:/# apt update
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1137 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:5 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2066 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1363 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [23.0 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:11 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packa
```

Step 2: Using commit the image is being created:

```
PS C:\Users\NekshaSrinivas\SE-1> docker commit ubuntu-cont-1 img-commit-1
sha256:153126502820131f25f36cc59f7c4557275621bcd7a54b48c8ffd4409685efd
PS C:\Users\NekshaSrinivas\SE-1> docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
img-commit-1    latest    153126502820    7 seconds ago   326MB
mynginx         latest    de77ca8d52cb    30 hours ago   279MB
mypythonapp     latest    8a39b6d82115    30 hours ago   1.63GB
nginx           latest    33e0bbc7ca9e    12 days ago    279MB
ubuntu          latest    7c06e91f61fa    3 weeks ago    117MB
PS C:\Users\NekshaSrinivas\SE-1> docker run -it img-commit-1
root@909ab066a51f:/# git --version
git version 2.43.0
root@909ab066a51f:/# docker tag img-commit-1 nekshasrinivas/img-commit-1
bash: docker: command not found
root@909ab066a51f:/# exit
exit
PS C:\Users\NekshaSrinivas\SE-1> docker tag img-commit-1 nekshasrinivas/img-commit-1
PS C:\Users\NekshaSrinivas\SE-1> docker push nekshasrinivas/img-commmit-1
Using default tag: latest
The push refers to repository [docker.io/nekshasrinivas/img-commmit-1]
4024494ad21b: Pushed
b71466b94f26: Mounted from library/ubuntu
latest: digest: sha256:153126502820131f25f36cc59f7c4557275621bcd7a54b48c8ffd4409685efd size: 751
```

### Step 3: Image creation using docker file

```
PS C:\Users\NekshaSrinivas\SE-1> mkdir image-creation

Directory: C:\Users\NekshaSrinivas\SE-1

Mode                LastWriteTime         Length Name
----                -----           -----
d----        26-08-2025      18:26                 image-creation

PS C:\Users\NekshaSrinivas\SE-1> ls

Directory: C:\Users\NekshaSrinivas\SE-1

Mode                LastWriteTime         Length Name
----                -----           -----
d----        26-08-2025      18:26                 image-creation
d----        25-08-2025      12:25                 static_site
-a---        25-08-2025      12:15                  36 app.py
-a---        25-08-2025      12:15                  100 Dockerfile
-a---        23-08-2025      16:28    29739088 myapi.tar
-a---        05-08-2025      16:33                  38 README.md

PS C:\Users\NekshaSrinivas\SE-1> cd image-creation
PS C:\Users\NekshaSrinivas\SE-1\image-creation> notepad Dockerfile
PS C:\Users\NekshaSrinivas\SE-1\image-creation> ls

Directory: C:\Users\NekshaSrinivas\SE-1\image-creation

Mode                LastWriteTime         Length Name
----                -----           -----
-a---        26-08-2025      18:27                  59 Dockerfile.txt

PS C:\Users\NekshaSrinivas\SE-1\image-creation> ren Dockerfile.txt Dockerfile
```

### Step 4: after writing the content in docker file use the command docker build

```
PS C:\Users\NekshaSrinivas\SE-1\image-creation> docker build -t img-dockerfile-1
[*] Building #4.0s (7/7) FINISHED
          docker:desktop-linux
=> [internal] load build definition from Dockerfile          0.1s
=> => transferring dockerfile: 96B                          0.0s
=> [internal] load metadata for docker.io/library/ubuntu:latest 0.1s
=> [internal] load .dockerrcignore                         0.1s
=> => transferring context: 2B                           0.0s
=> [1/3] FROM docker.io/library/ubuntu:latest@sha256:7c06e91f61fa88c 0.1s
=> => resolve docker.io/library/ubuntu:latest@sha256:7c06e91f61fa88c 0.0s
=> [2/3] RUN apt-get update                                12.5s
=> [3/3] RUN apt-get install git -y                      24.0s
=> exporting to image                                     6.0s
=> => exporting layers                                    4.0s
=> => exporting manifest sha256:99d816a6b717e709d838937a995f24d8121e 0.0s
=> => exporting config sha256:f021a40f65d4b684b65cd403292af90ec68210 0.0s
=> => exporting attestation manifest sha256:84dc38eee96b84e155b083e5 0.1s
=> => exporting manifest list sha256:9868ecb2df510b52e539c55076bf63c 0.0s
=> => naming to docker.io/library/img-dockerfile-1:latest 0.0s
=> => unpacking to docker.io/library/img-dockerfile-1:latest 1.8s
PS C:\Users\NekshaSrinivas\SE-1\image-creation> docker run -it img-dockerfile-1
root@adfe97a50685:/# docker --version
bash: docker: command not found
root@adfe97a50685:/# git --version
git version 2.43.0
root@adfe97a50685:/# exit
exit
PS C:\Users\NekshaSrinivas\SE-1\image-creation> docker tag img-dockerfile-1 nekshasrinivas/img-dockerfile-1
PS C:\Users\NekshaSrinivas\SE-1\image-creation> docker push nekshasrinivas/img-dockerfile-1
Using default tag: latest
The push refers to repository [docker.io/nekshasrinivas/img-dockerfile-1]
6a5ccfd4db931: Pushed
edd67216c21: Pushed
b71466b94f26: Mounted from nekshasrinivas/img-commit-1
804a734bd8b1: Pushed
latest: digest: sha256:9868ecb2df510b52e539c55076bf63ccae47b54ab67e29de352ddbc3cb33b109 size: 855
```

## Step 5: checking the images

```
PS C:\Users\NekshaSrinivas\SE-1> docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
img-dockerfile-1   latest   9868ecb2df51  5 minutes ago  326MB
nekshasrinivas/img-dockerfile-1   latest   9868ecb2df51  5 minutes ago  326MB
img-commit-1        latest   153126502820  12 minutes ago  326MB
nekshasrinivas/img-commmit-1    latest   153126502820  12 minutes ago  326MB
mynginx             latest   de77ca8d52cb  30 hours ago   279MB
mypythonapp         latest   8a39b6d82115  30 hours ago  1.63GB
nginx               latest   33e0bbc7ca9e  12 days ago    279MB
ubuntu              latest   7c06e91f61fa  3 weeks ago   117MB
PS C:\Users\NekshaSrinivas\SE-1> |
```

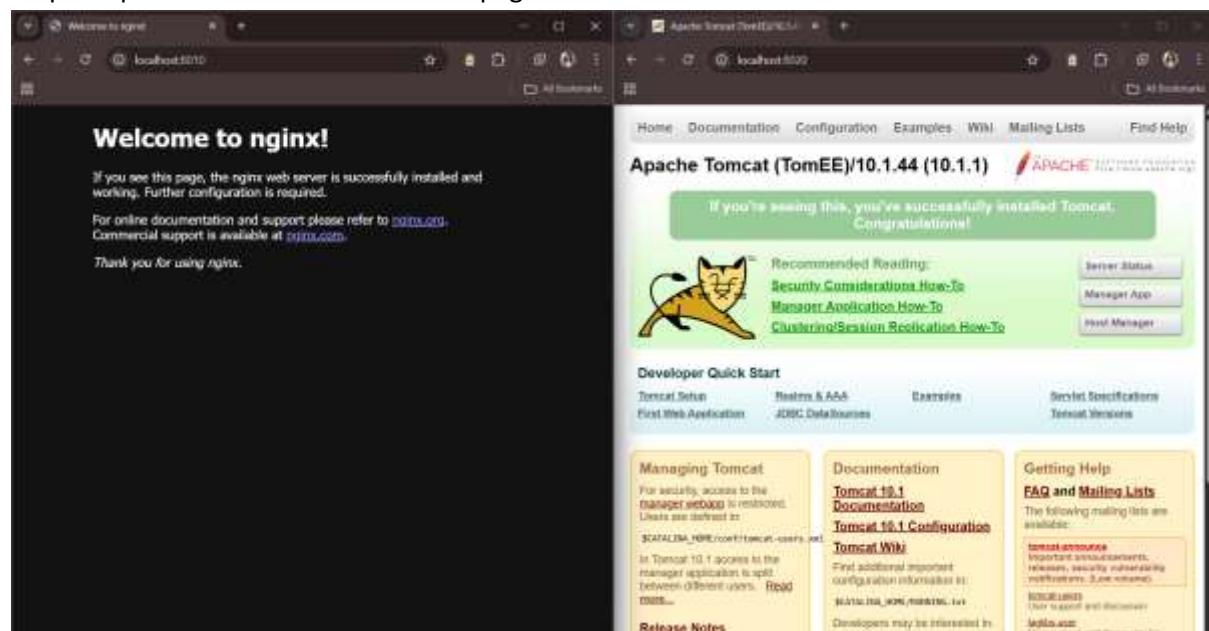
## DOCKER COMPOSE FILE:

Docker Compose is a tool used to define and run multi-container Docker applications. It allows you to define services, networks, and volumes that your application needs, all in a single file. This makes it easier to manage complex applications that require multiple containers (e.g., a web server and a database).

## Step 1: Running two servers at the same time on different ports

```
PS C:\Users\NekshaSrinivas\SE-1> docker run -d -p 8010:80 nginx
2ea4a201f197b93276310a7d23f2a46060ba9c7387f869e8a2a804931b66b2d9
PS C:\Users\NekshaSrinivas\SE-1> docker run -d -p 8020:8080 tomee
3a524036f6b212843be468585f80fb029aed07715a8e33a38e4eb306044765a2
PS C:\Users\NekshaSrinivas\SE-1> |
```

## Step 2: Open the local host to view the pages



Step 3: Using docker file to run two servers parallelly

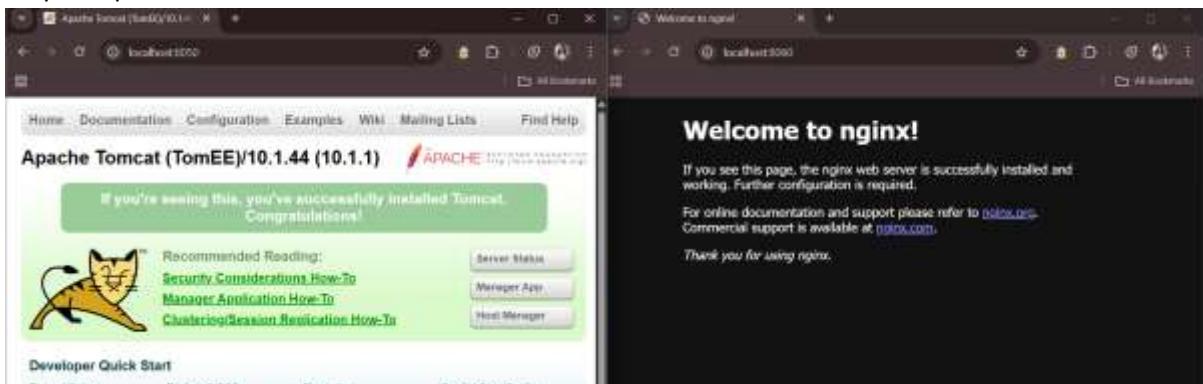


```
docker-compose.yml X
C: > Users > NekshaSrinivas > SE-1 > comp-1-server > docker-compose.yml
1 services:
2   web:
3     image: nginx
4     ports:
5       - "8060:80"
6   db:
7     image: tomee
8     ports:
9       - "8050:8080"
10
```

Step 4: Use the docker-compose up -d command to execute the docker file

```
No configuration file provided, not found
PS C:\Users\NekshaSrinivas\SE-1\comp-1-server> ren Dockerfile docker-compose
.yml
PS C:\Users\NekshaSrinivas\SE-1\comp-1-server> docker-compose up -d
[+] Running 3/3
✓ Network comp-1-server_default  C...          0.1s
✓ Container comp-1-server-db-1    St...         0.6s
✓ Container comp-1-server-web-1  S...          0.7s
PS C:\Users\NekshaSrinivas\SE-1\comp-1-server> |
```

Step 5: Open the localhost to view the servers



## WORD-PRESS:

Step 1: Create a docker-compose file and write the content for wordpress and mysql

```
PS C:\Users\NekshaSrinivas\SE-1> cd mysql
PS C:\Users\NekshaSrinivas\SE-1\mysql> notepad docker-compose
PS C:\Users\NekshaSrinivas\SE-1\mysql> ls

Directory: C:\Users\NekshaSrinivas\SE-1\mysql

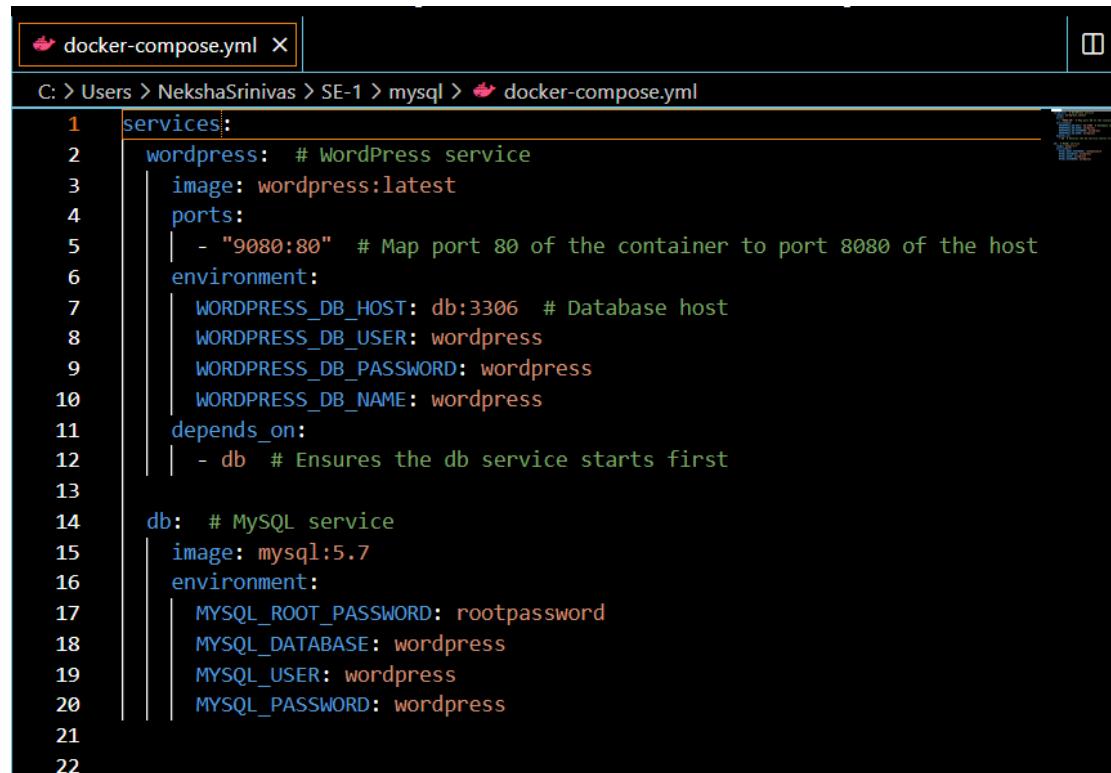
Mode                LastWriteTime        Length Name
----                -----          -----
-a----    26-08-2025     18:48           672 docker-compose.txt

PS C:\Users\NekshaSrinivas\SE-1\mysql>
PS C:\Users\NekshaSrinivas\SE-1\mysql> ren docker-compose.txt docker-compose
.yml
PS C:\Users\NekshaSrinivas\SE-1\mysql> ls

Directory: C:\Users\NekshaSrinivas\SE-1\mysql

Mode                LastWriteTime        Length Name
----                -----          -----
-a----    26-08-2025     18:48           672 docker-compose.yml
```

Step 2: docker-compose.yml file:



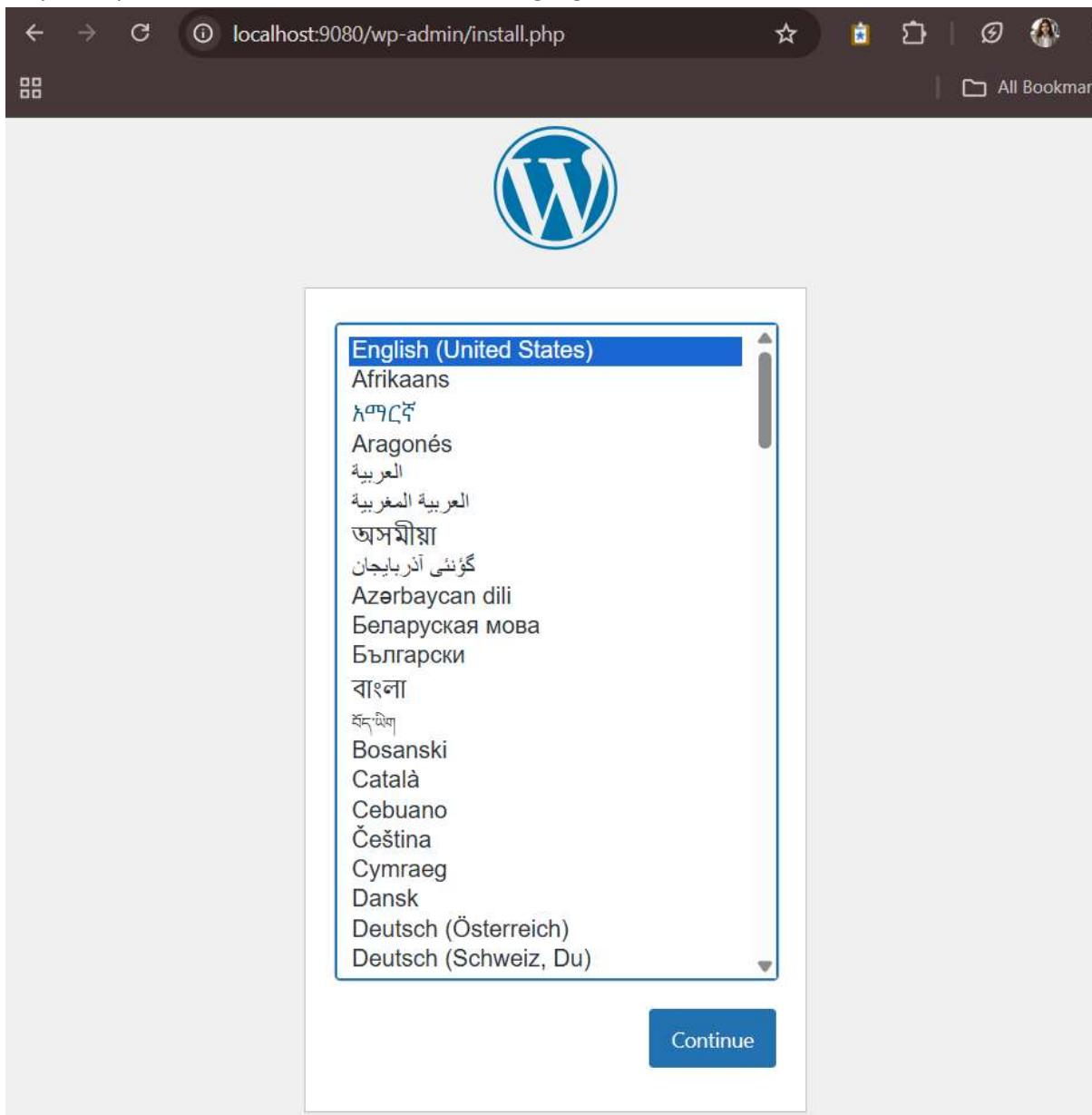
The screenshot shows a code editor window with the title "docker-compose.yml". The file path is "C: > Users > NekshaSrinivas > SE-1 > mysql > docker-compose.yml". The code itself is a Docker Compose configuration file:

```
1 services:
2   wordpress: # WordPress service
3     image: wordpress:latest
4     ports:
5       - "9080:80" # Map port 80 of the container to port 8080 of the host
6     environment:
7       WORDPRESS_DB_HOST: db:3306 # Database host
8       WORDPRESS_DB_USER: wordpress
9       WORDPRESS_DB_PASSWORD: wordpress
10      WORDPRESS_DB_NAME: wordpress
11      depends_on:
12        - db # Ensures the db service starts first
13
14   db: # MySQL service
15     image: mysql:5.7
16     environment:
17       MYSQL_ROOT_PASSWORD: rootpassword
18       MYSQL_DATABASE: wordpress
19       MYSQL_USER: wordpress
20       MYSQL_PASSWORD: wordpress
```

Step 3: Use the docker-compose up -d command to start the compose

```
PS C:\Users\NekshaSrinivas\SE-1\mysql> docker-compose up -d
[+] Running 3/3
  ✓ Network mysql_default          Created              0.1s
  ✓ Container mysql-db-1           Started             0.8s
  ✓ Container mysql-wordpress-1   Start...            1.0s
PS C:\Users\NekshaSrinivas\SE-1\mysql>
```

Step 4: Open in the local host and select the language



Step 5: Fill the details in the welcome page

## Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

### Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title

Hey

Username

Neksha Srinivas

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password

Sri@121318

 Hide

Medium

**Important:** You will need this password to log in. Please store it in a secure location.

Your Email

edigiralaneksha@gmail.com

Double-check your email address before continuing.

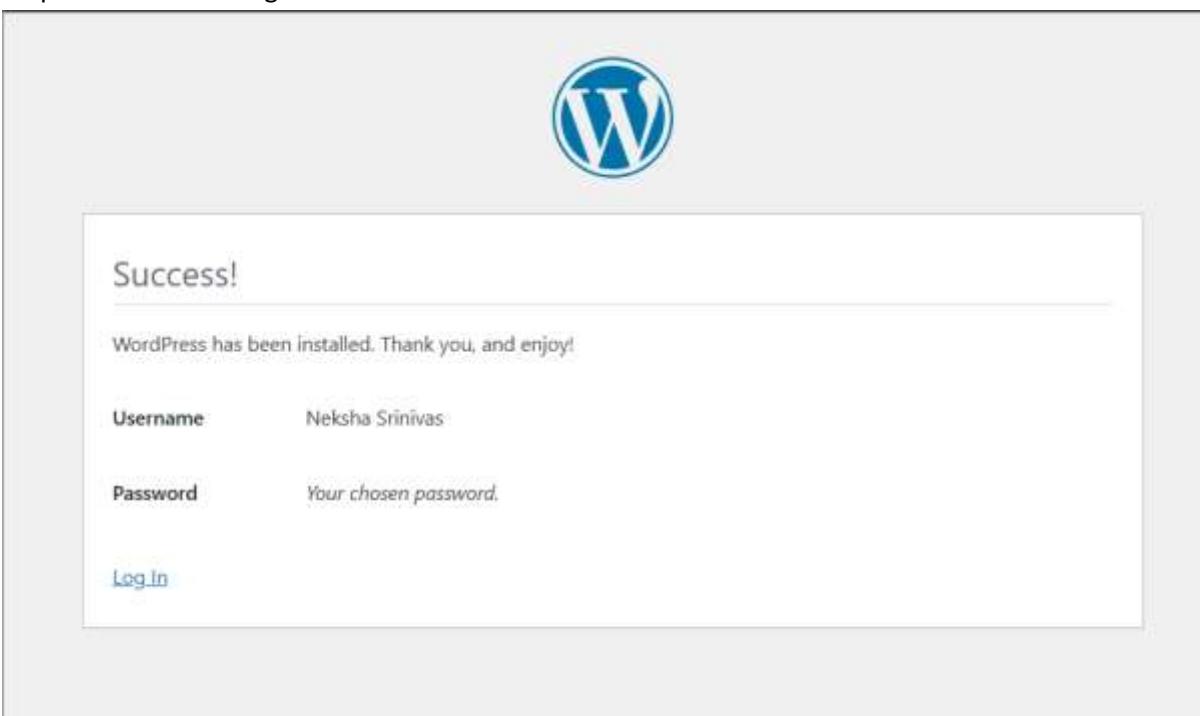
Search engine visibility

Discourage search engines from indexing this site

It is up to search engines to honor this request.

[Install WordPress](#)

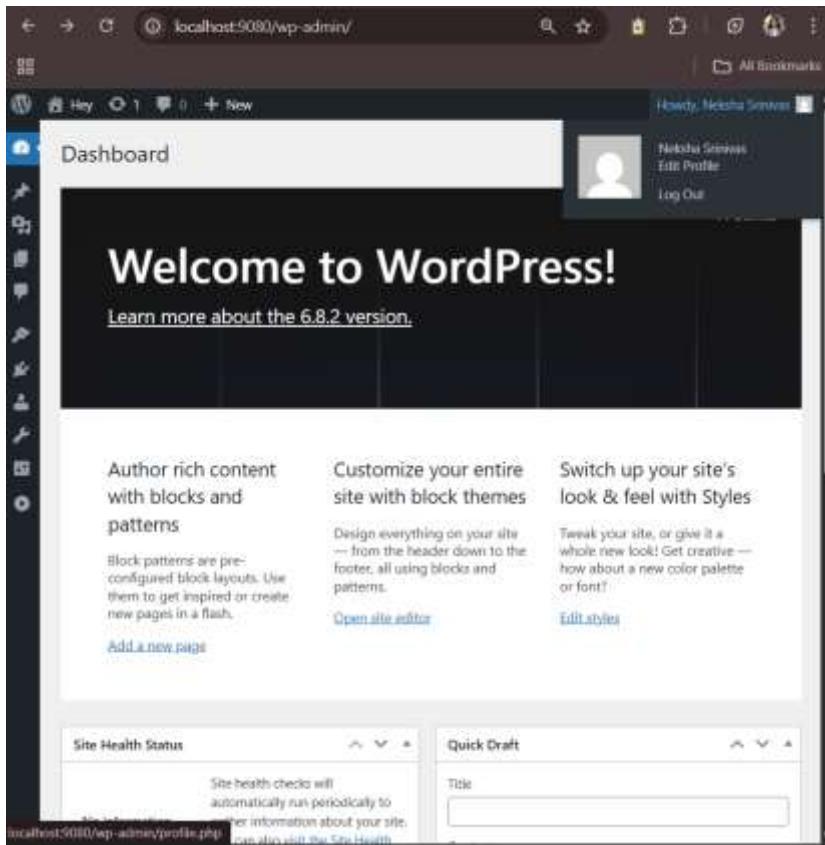
Step 6: Success message will be shown



Step 7: Use your credentials to log in



Step 7: The following page will be shown after login



Task:

Create a simple Flask app in app.py:

Step 1: create a separate folder

```
PS C:\Users\NekshaSrinivas\SE-1> mkdir custom_flask

Directory: C:\Users\NekshaSrinivas\SE-1

Mode                LastWriteTime         Length Name
----                -----          -----    -----
d----        28-08-2025     10:01                 custom_flask

PS C:\Users\NekshaSrinivas\SE-1> cd custom_flask
PS C:\Users\NekshaSrinivas\SE-1\custom_flask> notepad app.py
PS C:\Users\NekshaSrinivas\SE-1\custom_flask> notepad Dockerfile
PS C:\Users\NekshaSrinivas\SE-1\custom_flask> ren Dockerfile.txt Dockerfile
PS C:\Users\NekshaSrinivas\SE-1\custom_flask> ls

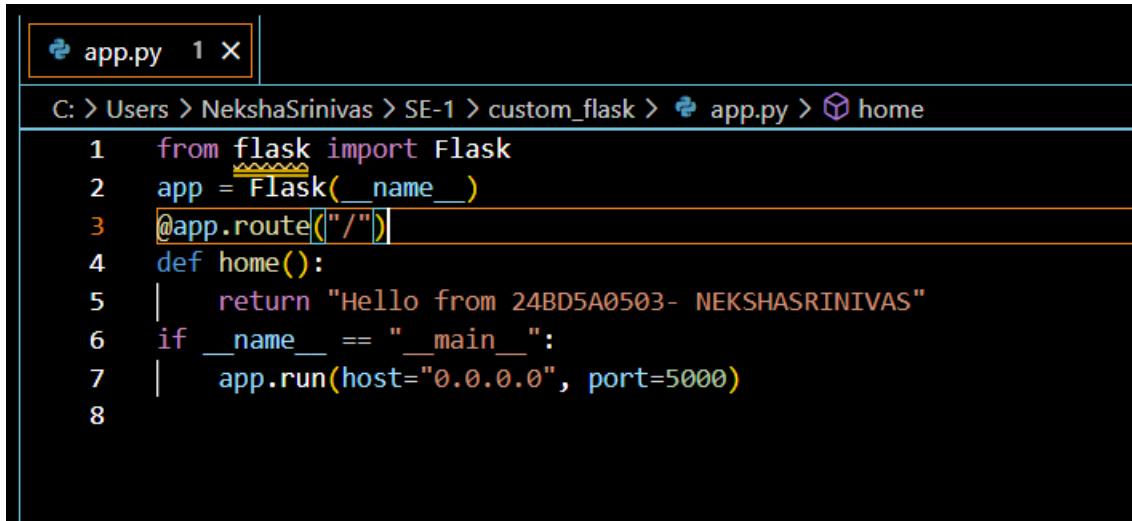
Directory: C:\Users\NekshaSrinivas\SE-1\custom_flask

Mode                LastWriteTime         Length Name
----                -----          -----    -----
-a----        28-08-2025     10:02           187 app.py
-a----        28-08-2025     10:02           105 Dockerfile

PS C:\Users\NekshaSrinivas\SE-1\custom_flask> notepad docker-compose.yml
```

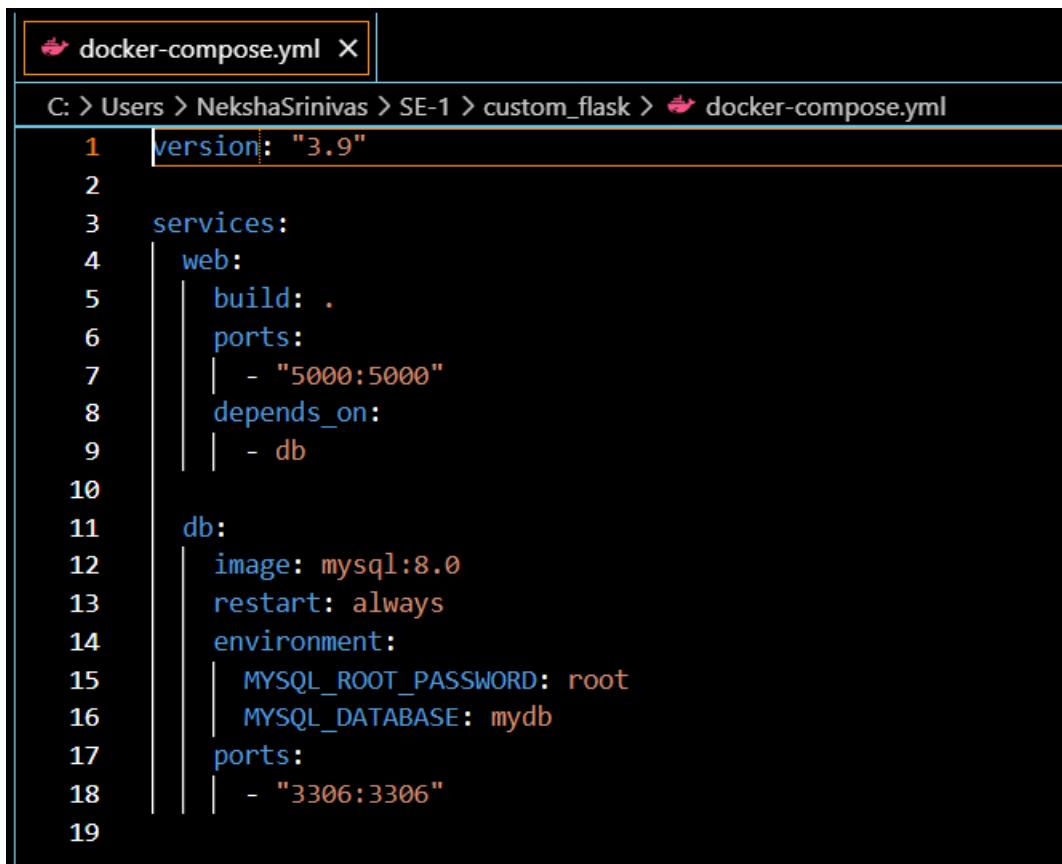
Step 2: write the content of app.py , docker-compose.yml & Dockerfile

app.py:



```
app.py 1 X
C: > Users > NekshaSrinivas > SE-1 > custom_flask > app.py > home
1  from flask import Flask
2  app = Flask(__name__)
3  @app.route("/")
4  def home():
5      return "Hello from 24BD5A0503- NEKSHASRINIVAS"
6  if __name__ == "__main__":
7      app.run(host="0.0.0.0", port=5000)
8
```

docker-compose.yml:



```
docker-compose.yml X
C: > Users > NekshaSrinivas > SE-1 > custom_flask > docker-compose.yml
1  version: "3.9"
2
3  services:
4      web:
5          build: .
6          ports:
7              - "5000:5000"
8          depends_on:
9              - db
10
11     db:
12         image: mysql:8.0
13         restart: always
14         environment:
15             MYSQL_ROOT_PASSWORD: root
16             MYSQL_DATABASE: mydb
17         ports:
18             - "3306:3306"
19
```

Dockerfile:

```
FROM python:3.10-slim
WORKDIR /app
COPY app.py /app/
RUN pip install flask
CMD ["python", "app.py"]
```

Step 3: run the compose using the command docker compose up –build:

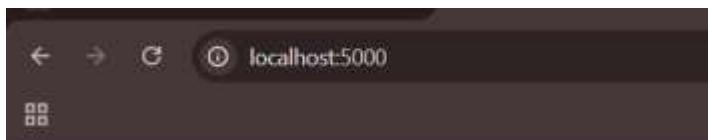
```
Mode           LastWriteTime          Length Name
----           -----          ---- 
-a---       28-08-2025     10:02           187 app.py
-a---       28-08-2025     10:03            82 docker-compose.yml
-a---       28-08-2025     10:02          105 Dockerfile

PS C:\Users\NekshaSrinivas\SE-1\custom_flask> docker compose up --build
validating C:\Users\NekshaSrinivas\SE-1\custom_flask\docker-compose.yml: additional properties 'web' not allowed
PS C:\Users\NekshaSrinivas\SE-1\custom_flask> docker compose up --build
time="2025-08-28T08:24:45+05:30" level=warning msg="C:\\\\Users\\\\NekshaSrinivas\\\\SE-1\\\\custom_flask\\\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 12/12
  ✓ db Pulled
    ✓ 84fa#2a56901 Pull complete
    ✓ 500d7b2546c4 Pull complete
    ✓ ecc6cc933381 Pull complete
    ✓ 5cd63fb67e17 Pull complete
    ✓ 4d3eacc36b14 Pull complete
    ✓ 9476bbfaedba Pull complete
    ✓ 789fa151603e Pull complete
    ✓ 1756a3#72d796 Pull complete
    ✓ bc0f5543b464 Pull complete
    ✓ 131412d69359 Pull complete
    ✓ 03ca01bc78d4 Pull complete
#1 [internal] load local bake definitions
#1 reading from stdin 542B done
#1 DONE 0.0s

#2 [internal] load build definition from Dockerfile
#2 transferring dockerfile: 142B 0.0s done
#2 DONE 0.1s

#3 [internal] load metadata for docker.io/library/python:3.10-slim
```

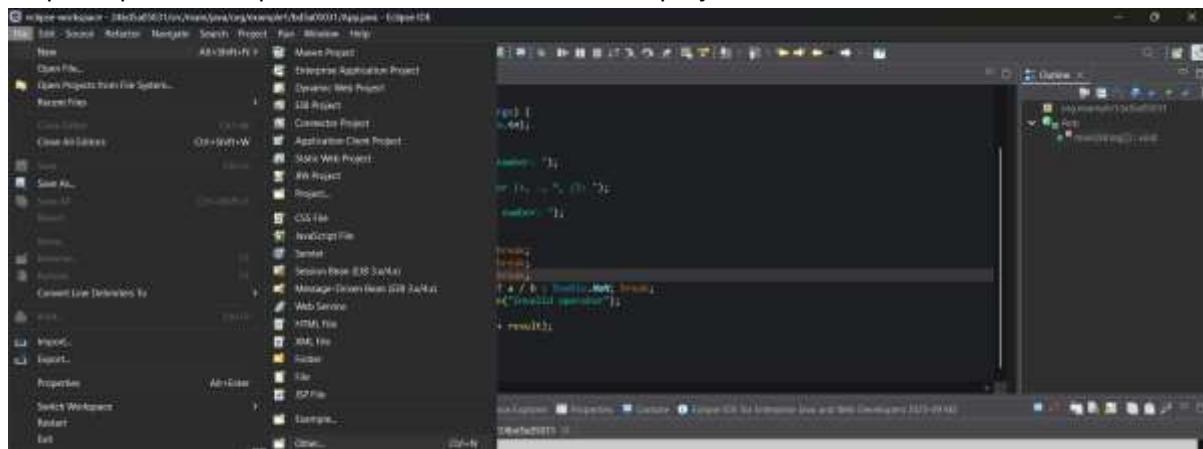
Step 4: Open the local host to view the custom page:



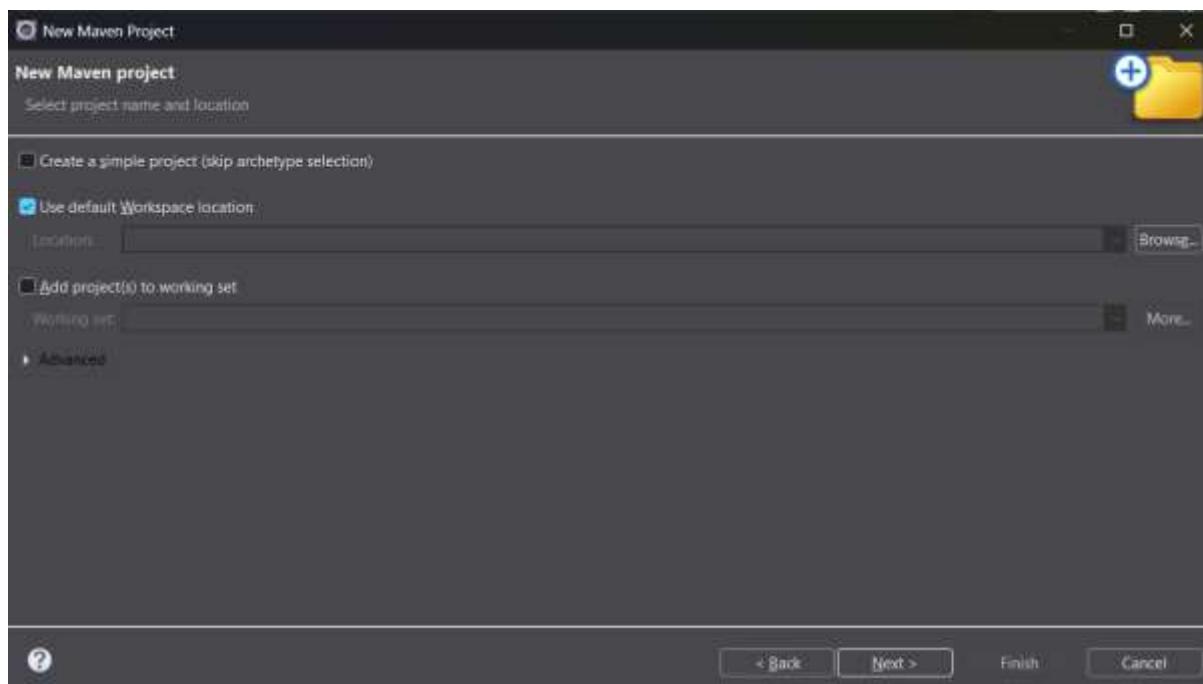
Hello from 24BD5A0503- NEKSHASRINIVAS

## **7. Creating a Multi-Module Maven Project**

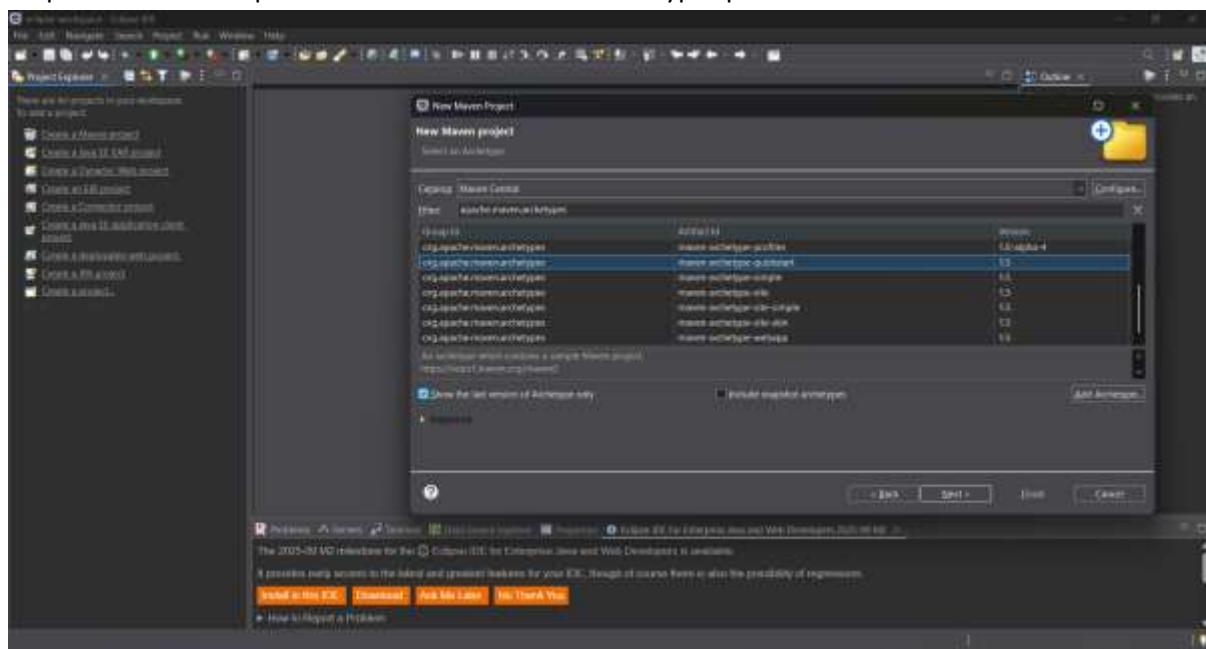
Step-1: Open the eclipse and click on file>new>Maven project



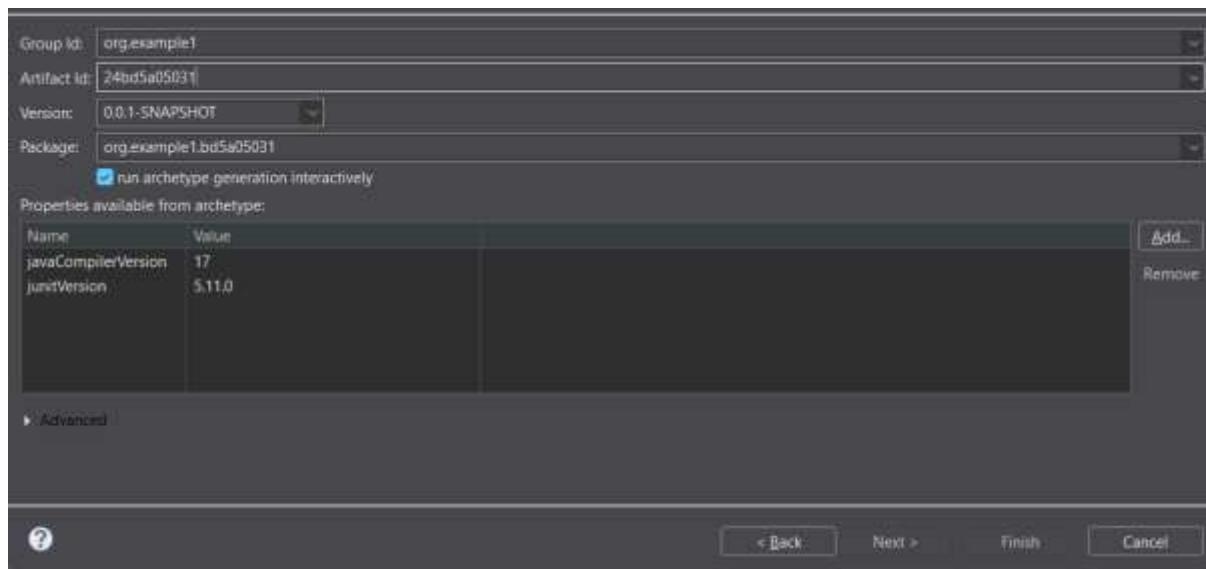
Step-2: select the default workspace and click on next



Step-3: in the filter option select the one maven-archetype-quickstart



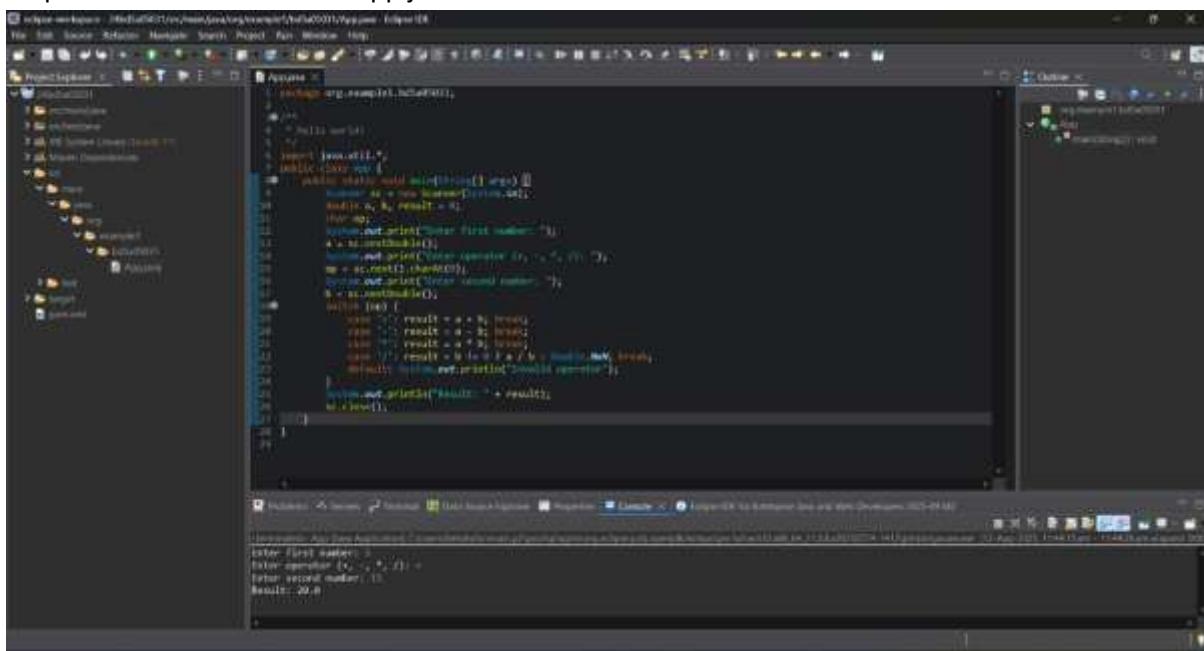
Step-4: give the Group Id and Artifact Id and click on next



Step-5: In the console the progress will be showed type y (refers to yes) and press enter

Step-6: BUILD SUCCESS will be shown

Step-6: write the code in the App.java file

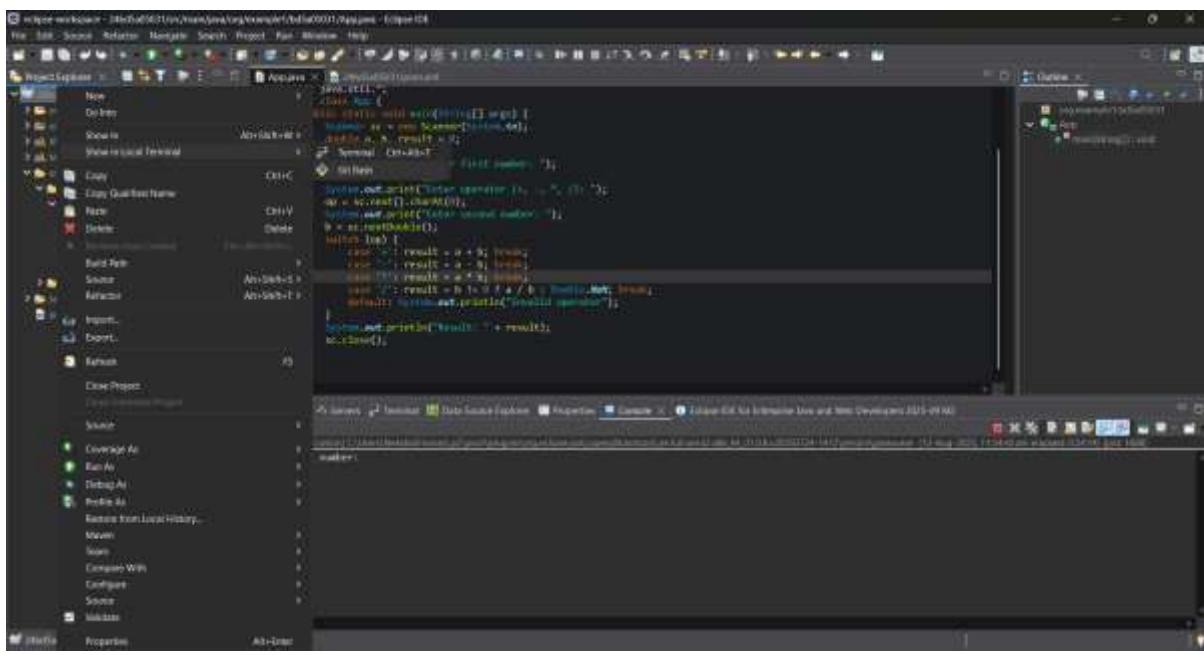


The screenshot shows the Eclipse IDE interface with the code editor open to the `App.java` file. The code implements a calculator with addition, subtraction, multiplication, and division operations. It includes a main method to handle user input and output.

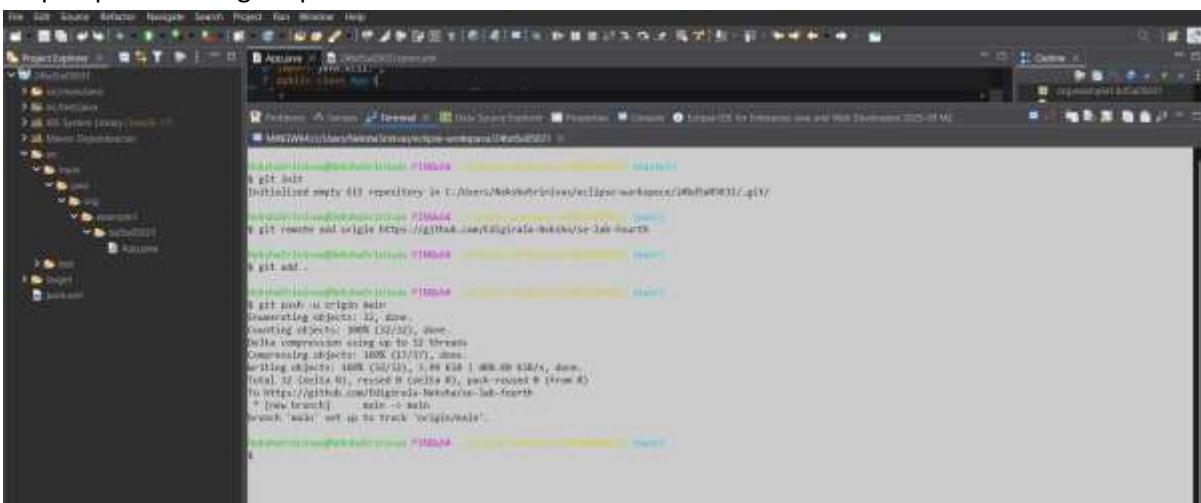
```
import java.util.Scanner;
public class App {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a = sc.nextInt();
        int b = sc.nextInt();
        System.out.println("First number: " + a);
        System.out.println("Second number: " + b);
        char op = sc.next().charAt(0);
        if (op == '+') {
            int result = a + b;
            System.out.println("Result: " + result);
        } else if (op == '-') {
            int result = a - b;
            System.out.println("Result: " + result);
        } else if (op == '*') {
            int result = a * b;
            System.out.println("Result: " + result);
        } else if (op == '/') {
            int result = a / b;
            System.out.println("Result: " + result);
        }
    }
}
```

The Eclipse interface includes a Project Explorer on the left, a code editor in the center, and a terminal or output window at the bottom showing sample console output.

Step-7: right click on the root folder and select show in git bash



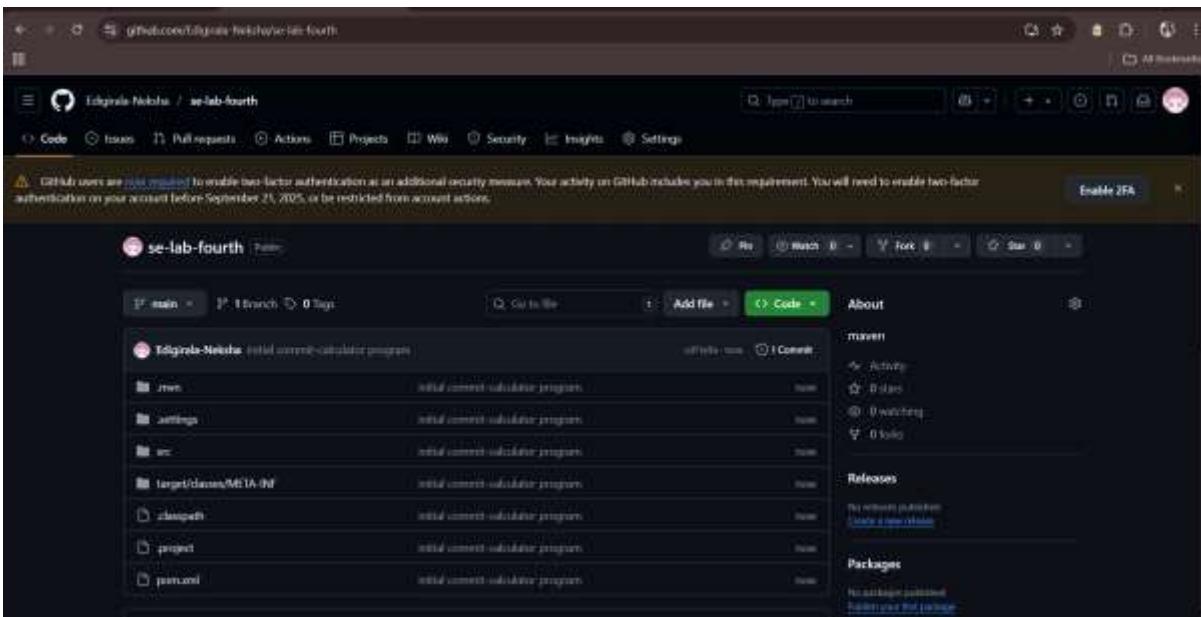
## Step-8: push to the git repo



```
git push -u origin main
Counting objects: 12, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), done.
Total 10 (delta 0), reused 0 deltas, pack-reused 0 (from 0)
remote: Resolving deltas: 100% (10/10), done.
To https://github.com/Edigirala-Neksha/se-lab-fourth
 * [new branch] main -> main
Branch 'main' set up to track 'origin/main'.
```

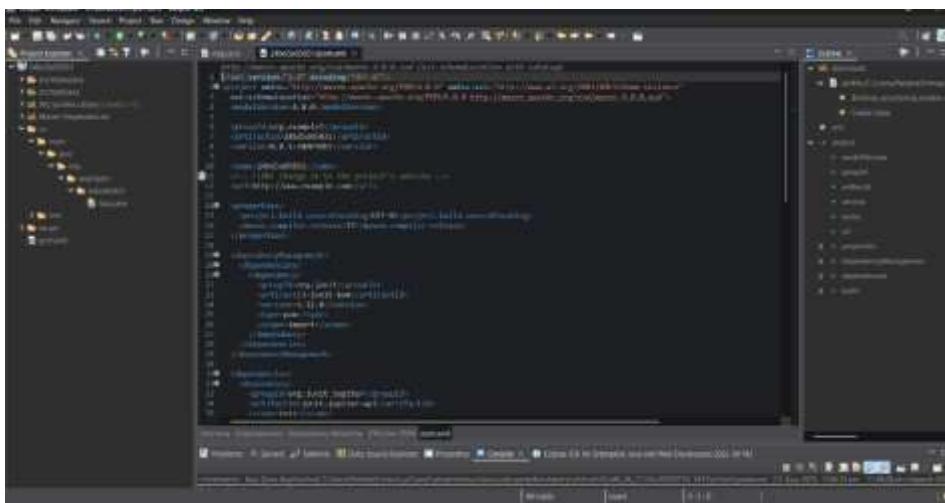
Git repo:

Git repo link: <https://github.com/Edigirala-Neksha/se-lab-fourth>



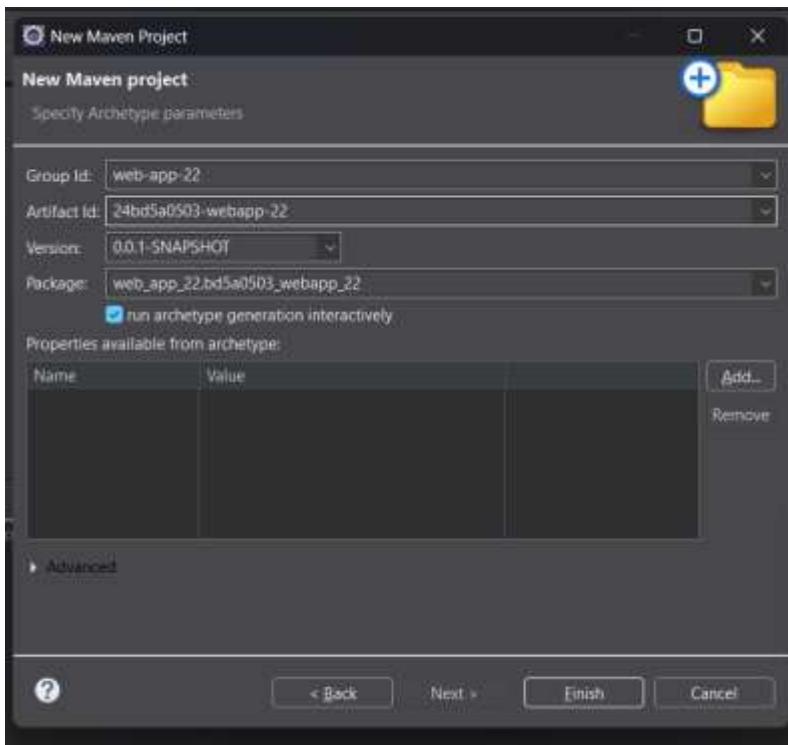
pom.xml file:

Shows the structure-



Creating maven-web project:

Step 1: Create a new maven project and give the details



Step 2: Click y to continue the creation of project

```
C:\Users\NekshaSrinivas\Downloads\maven-archetype-webapp-1.5\src\main\java\org\apache\maven\archetype\remote\archetype-catalog.xml (3370) at %3(M2)
Progress (1): 37/37 MB
Downloaded from central: https://repo.maven.apache.org/maven2/archetype-catalog.xml (3370) at %3(M2)
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archetypes:maven-archetype-webapp:1.5] found in catalog remote
[INFO] Using property: groupId = web-app-22
[INFO] Using property: artifactId = 24bd5a0503-webapp-22
[INFO] Using property: version = 0.0.1-SNAPSHOT
[INFO] Using property: package = web_app_22.bd5a0503_webapp_22
Confirm properties configuration:
groupId: web-app-22
artifactId: 24bd5a0503-webapp-22
version: 0.0.1-SNAPSHOT
package: web_app_22.bd5a0503_webapp_22
Y: y
```

Step 3: If the build is success it will show the message

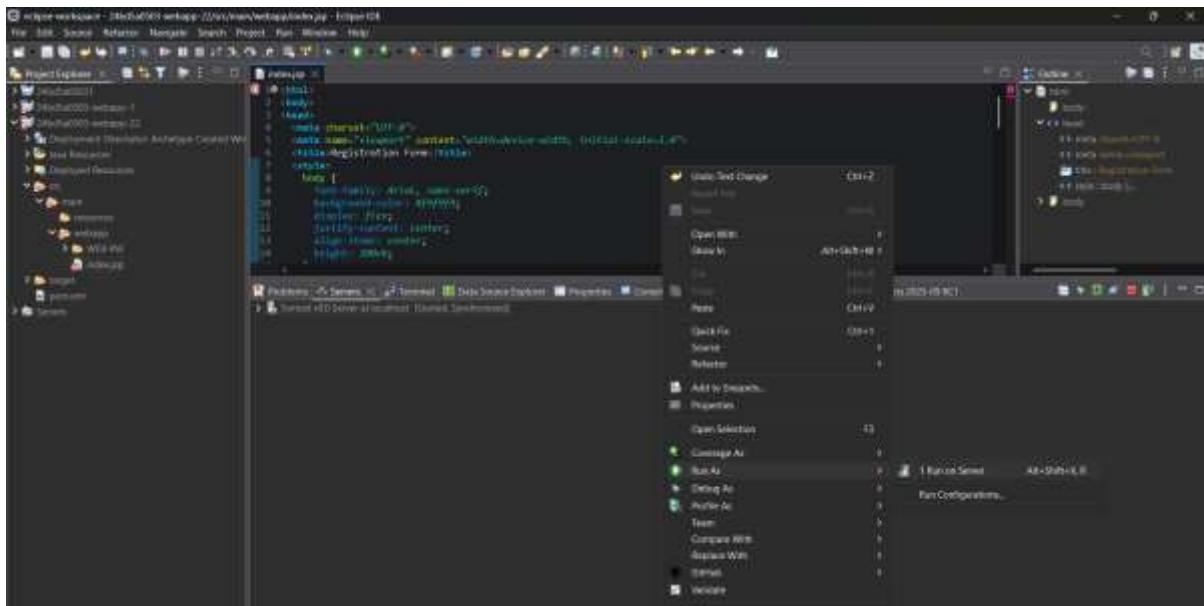
```
package: web_app_22.bd5a0503_webapp_22
Y: y
[INFO] -----
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-webapp:1.0
[INFO] -----
[INFO] Parameter: basedir, Value: C:\Users\NekshaSrinivas\eclipse-workspace
[INFO] Parameter: package, Value: web_app_22.bd5a0503_webapp_22
[INFO] Parameter: groupId, Value: web-app-22
[INFO] Parameter: artifactId, Value: 24bd5a0503-webapp-22
[INFO] Parameter: packageName, Value: web_app_22.bd5a0503_webapp_22
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a0503-webapp-22
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 43.500 s
[INFO] Finished at: 2025-09-02T19:20:41+05:30
[INFO] -----
```

Step 4: write the html code for the web page:

```
index.jsp: X
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <head>
5       <meta charset="UTF-8">
6       <meta name="viewport" content="width=device-width, initial-scale=1.0">
7       <title>Registration Form</title>
8       <style>
9         body {
10           font-family: Arial, sans-serif;
11           background-color: #f0f0f0;
12           display: flex;
13           justify-content: center;
14           align-items: center;
15           height: 100vh;
16         }
17         .form-container {
18           background: #fff;
19           padding: 20px 30px;
20           border-radius: 10px;
21           box-shadow: 0 4px 16px rgba(0,0,0,0.1);
22           width: 300px;
23         }
24         .form-container h2 {
25           text-align: center;
26           margin-bottom: 20px;
27         }
28         .form-container input {
29           width: 100%;
30           padding: 10px;
31           margin: 5px 0;
32           border: 1px solid #ccc;
33           border-radius: 5px;
34         }
35         .form-container button {
36           width: 100%;
37           padding: 10px;
38           background: #4CAF50;
39           color: white;
40           border: none;
41           cursor: pointer;
42           font-size: 1em;
43         }
44         .form-container button:hover {
45           background-color: #45B7D1;
46         }
```

```
index.jsp X
  1 <head>
  2   <title>Registration Form</title>
  3   <style>
  4     .form-container {
  5       width: 100px;
  6       padding: 10px;
  7       background: #4CAF50;
  8       color: white;
  9       border: none;
 10       border-radius: 5px;
 11       cursor: pointer;
 12     }
 13     .form-container button:hover {
 14       background: #450049;
 15     }
 16   </style>
 17 </head>
 18 <body>
 19   <div class="form-container">
 20     <h2>Registration Form</h2>
 21     <form action="#" method="post">
 22       <label for="fullname">Full Name</label>
 23       <input type="text" id="fullname" name="fullname" placeholder="Enter your name" required>
 24
 25       <label for="email">Email</label>
 26       <input type="email" id="email" name="email" placeholder="Enter your email" required>
 27
 28       <label for="password">Password</label>
 29       <input type="password" id="password" name="password" placeholder="Enter password" required>
 30
 31       <label for="confirm">Confirm Password</label>
 32       <input type="password" id="confirm" name="confirm" placeholder="Confirm password" required>
 33
 34       <button type="submit">Register</button>
 35     </form>
 36   </div>
 37 </body>
 38 </html>
 39
```

## Step 5: Select run on server

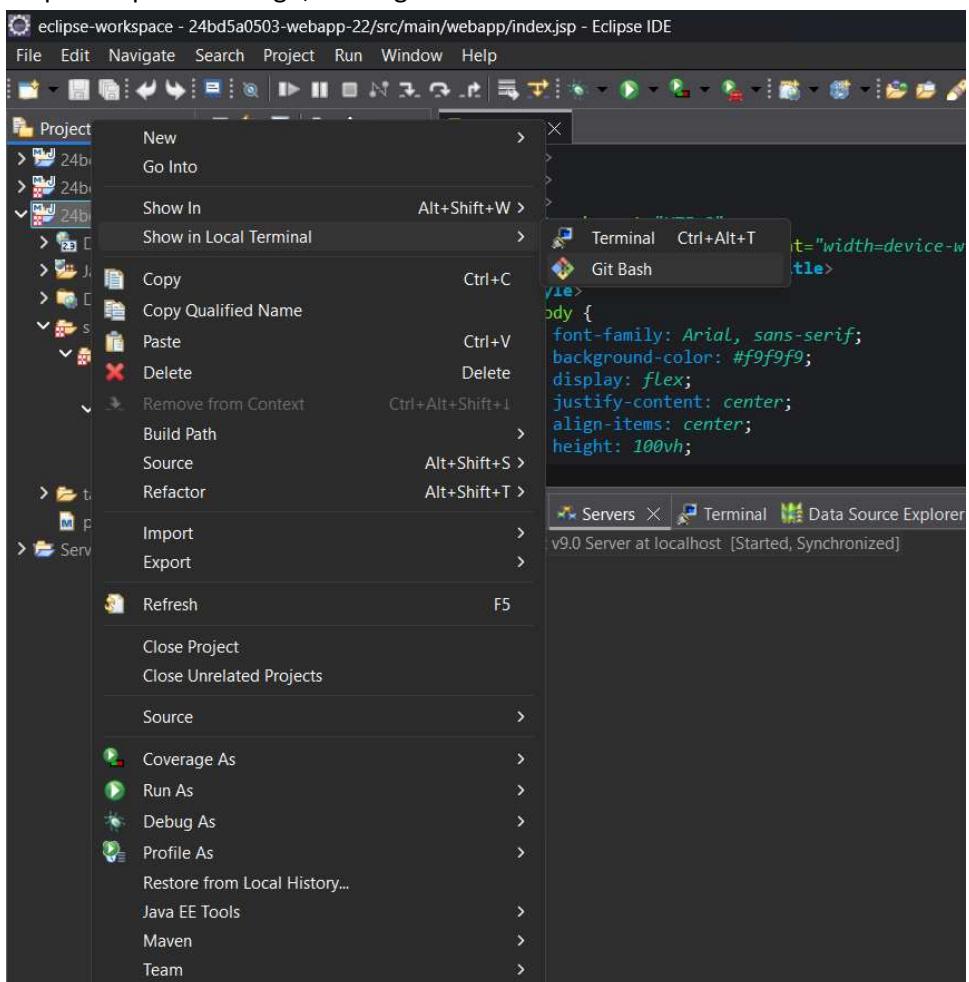


Step 6: It will show the following output:

A screenshot of a web browser window displaying a registration form. The browser's address bar shows the URL: `localhost:8080/24bd5a0503-webapp-22/index.jsp`. The page title is "Registration Form". The form consists of five input fields: "Full Name" (placeholder: "Enter your name"), "Email" (placeholder: "Enter your email"), "Password" (placeholder: "Enter password"), and "Confirm Password" (placeholder: "Confirm password"). Below the inputs is a green "Register" button.

Registration Form	
Full Name	<input type="text" value="Enter your name"/>
Email	<input type="text" value="Enter your email"/>
Password	<input type="password" value="Enter password"/>
Confirm Password	<input type="password" value="Confirm password"/>
<input type="button" value="Register"/>	

Step 7: To push it into git, select git bash from show in local terminal



Step 8: use the command of git to push the maven web project

```
nekshaSrinivas@NekshaSrinivas MINGW64 /c/Users/NekshaSrinivas/eclipse-workspace/24bd5a05... (master)
$ git init
Initialized empty Git repository in C:/Users/NekshaSrinivas/eclipse-workspace/24bd5a0503-webapp-22/.git/
nekshaSrinivas@NekshaSrinivas MINGW64 /c/Users/NekshaSrinivas/eclipse-workspace/24bd5a0503-webapp-22 (main)
$ git add .

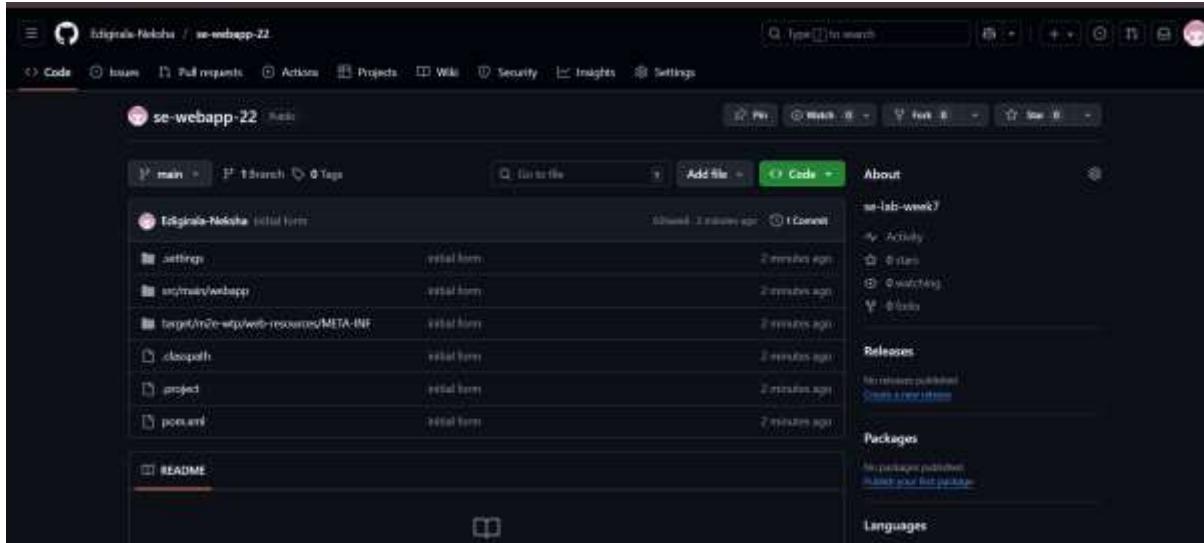
nekshaSrinivas@NekshaSrinivas MINGW64 /c/Users/NekshaSrinivas/eclipse-workspace/24bd5a0503-webapp-22 (main)
$ git commit -m "initial form"
[main (root-commit) 636aeef] initial form
 16 files changed, 254 insertions(+)
 create mode 100644 .classpath
 create mode 100644 .project
 create mode 100644 .settings/.jsdtscope
 create mode 100644 .settings/crg.eclipse.jdt.core.prefs
 create mode 100644 .settings/crg.eclipse.m2e.core.prefs
 create mode 100644 .settings/crg.eclipse.wst.commonn.component
 create mode 100644 .settings/crg.eclipse.wst.commonn.project.facet.core.xml
 create mode 100644 .settings/crg.eclipse.wst.jsdt.ui.superType.container
 create mode 100644 .settings/crg.eclipse.wst.jsdt.ui.superType.name
 create mode 100644 .settings/crg.eclipse.wst.validation.prefs
 create mode 100644 pom.xml
 create mode 100644 src/main/webapp/META-INF/web.xml
```

```
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0503-webapp-22 (main)
$ git branch
* main

NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0503-webapp-22 (main)
$ git push origin main
Enumerating objects: 29, done.
Counting objects: 100% (29/29), done.
Delta compression using up to 12 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (29/29), 4.43 KiB | 283.00 KiB/s, done.
Total 29 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Edigirala-Neksha/se-webapp-22.git
 * [new branch]      main -> main

NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0503-webapp-22 (main)
$
```

### Step 9: verify the repo in git hub



The screenshot shows a GitHub repository page for 'se-webapp-22'. The repository was created by 'Edigirala-Neksha' and has one branch, 'main'. The page displays a list of files in the 'main' branch:

- initial form (2 minutes ago)

On the right side of the page, there are sections for 'About', 'Activity', 'Stats', 'Watching', 'Follow', 'Releases', and 'Languages'.

## 8. Jenkins Automation

### Steps for MavenJava Automation

#### Step 1: Open Jenkins (localhost:8888)

Click on "New Item" (left side menu) and name it as maven\_java > select freestyle project > click on "OK"

New Item

Enter an item name: maven\_java

Select an item type:

- Freestyle project** Classic, general-purpose job type that checks out from up to one SCM, executes build steps locally, triggered by post-build steps like archiving artifacts and sending email notifications.
- Maven project** Build a Maven project; Jenkins takes advantage of your POM file and drastically reduces the configuration.
- Pipeline** Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organising complex activities that do not easily fit as free-style job type.
- Multi-configuration project** Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder** Creates a container that stores nested items in it. Useful for grouping things together. Unlike views, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- Multibranch Pipeline** Creates a set of Pipeline projects according to detected branches in one SCM repository.

OK

#### Step 2: Configuration of maven\_java project

Give the description

Configure

General

Enabled

Description

Java Build demo

Plain text Preview

Discard old builds ?

GitHub project

General

Source Code Management

Triggers

Environment

Build Steps

Post-build Actions

In the source code management select git and give the git repo link

The screenshot shows the Jenkins configuration page for a job named 'Mavenjava'. In the left sidebar, under 'Configure', the 'Source Code Management' section is selected. It displays a 'Repositories' configuration with a 'Repository URL' field containing the value 'https://github.com/SanikaSomishetty/eclipse-maven-projects.git'. Below it, a 'Credentials' dropdown is set to 'none'. There is also a '+ Add' button and an 'Advanced' link.

In the build steps click on add build step > give maven version as MAVEN\_HOME > select invoke top-level maven targets > goals as clean

The screenshot shows the Jenkins configuration page for the same job 'Mavenjava'. In the left sidebar, under 'Configure', the 'Build Steps' section is selected. It lists two 'Invoke top-level Maven targets' steps. Both steps have 'Maven Version' set to 'MAVEN\_HOME' and 'Goals' set to 'clean'. At the bottom of the page, there are 'Save' and 'Apply' buttons.

In the build steps click on add build step > give maven version as MAVEN\_HOME > select invoke top-level maven targets > goals as install

The screenshot shows the Jenkins configuration interface for a job named 'Mavenjava'. The 'Build Steps' section is active, displaying a single step: 'Invoke top-level Maven targets'. The 'Goals' field contains 'install'. The 'Advanced' dropdown is open, showing the 'MAVEN\_HOME' environment variable set to 'MAVEN\_HOME'. Other tabs like General, Source Code Management, Triggers, Environment, Post-build Actions, and Add build step are visible.

Configure

General

Source Code Management

Triggers

Environment

**Build Steps**

Post-build Actions

clean

Advanced

Invoke top-level Maven targets

Maven Version

MAVEN\_HOME

Goals

install

Advanced

Add build step

Post-build Actions

Save Apply

In the post build actions > click on add post build action > select the archive the artifacts > in the file to archive give “\*\*/\*”

For the second post build action,

In the post build actions > click on add post build action > select build other projects > give projects to build as MavenJava\_Test

Click on apply and save

The screenshot shows the Jenkins job configuration page for 'Mavenjava'. The left sidebar lists configuration sections: General, Source Code Management, Triggers, Environment, Build Steps, and Post-build Actions. The 'Post-build Actions' section is currently selected and highlighted.

The main area displays the 'Post-build Actions' configuration. It includes a section titled 'Archive the artifacts' with a 'Files to archive' field containing '\*\*/\*'. Below this is an 'Advanced' dropdown.

Another section titled 'Build other projects' is present, with a 'Projects to build' field containing 'MavenJava\_Test'. Underneath this, there are three trigger options: 'Trigger only if build is stable' (selected), 'Trigger even if the build is unstable', and 'Trigger even if the build fails'.

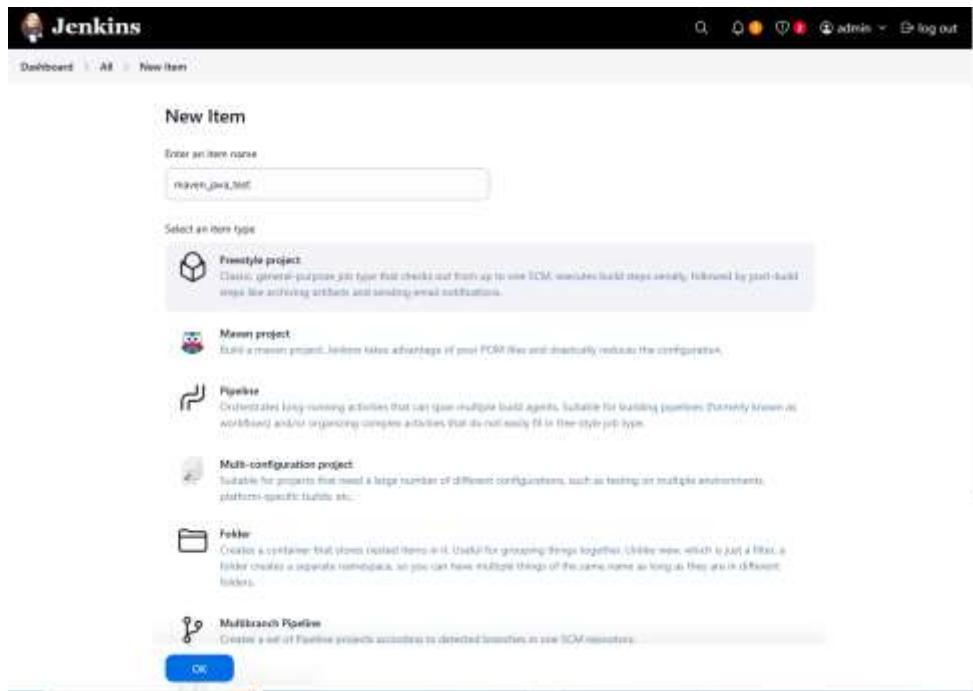
At the bottom of the configuration area are 'Save' and 'Apply' buttons.

If the build is success:

The screenshot shows a Windows desktop environment. At the top, there is a taskbar with several icons, including a search bar labeled "Type here to search". Below the taskbar, a window titled "maven\_web\_build [Jenkins]" is open, displaying the Jenkins job configuration for "maven\_web\_build". The job is marked as "Status: green" and has a checkmark icon. The configuration includes fields for "Changes", "Workspace", "Build Now", "Configure", "Delete Project", and "Rename". On the right side of the Jenkins window, there is a sidebar with "Builds" listed under "Today": "#2 11:43 AM" and "#3 11:42 AM". Other sections visible include "Last Successful Artifacts", "Downstream Projects" (listing "maven\_web\_test"), and "Permalinks". The bottom right corner of the screen shows the Jenkins version "Jenkins 2.489".

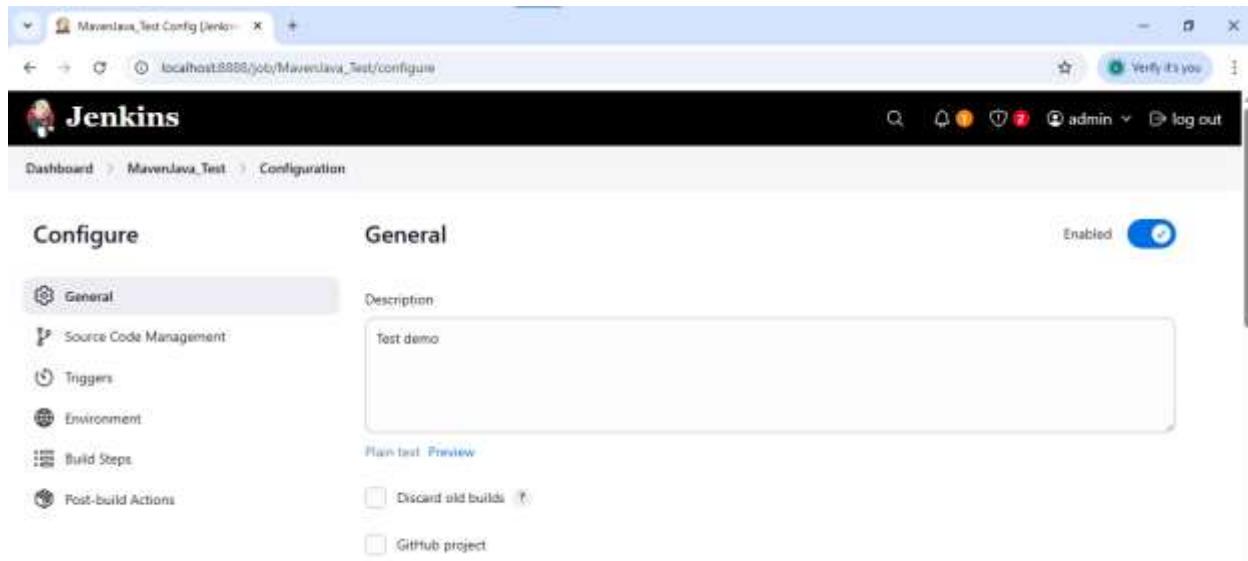
### Step 3: Create Freestyle Project (e.g., MavenJava\_Test)

Click on new item > give item name as mavaen\_java\_test or MavenJava\_Test and select free style project and click ok



### Step 4: Configuration of maven\_java project

Give the description



In the source code management select none and in environment select “delete workspace before build starts”

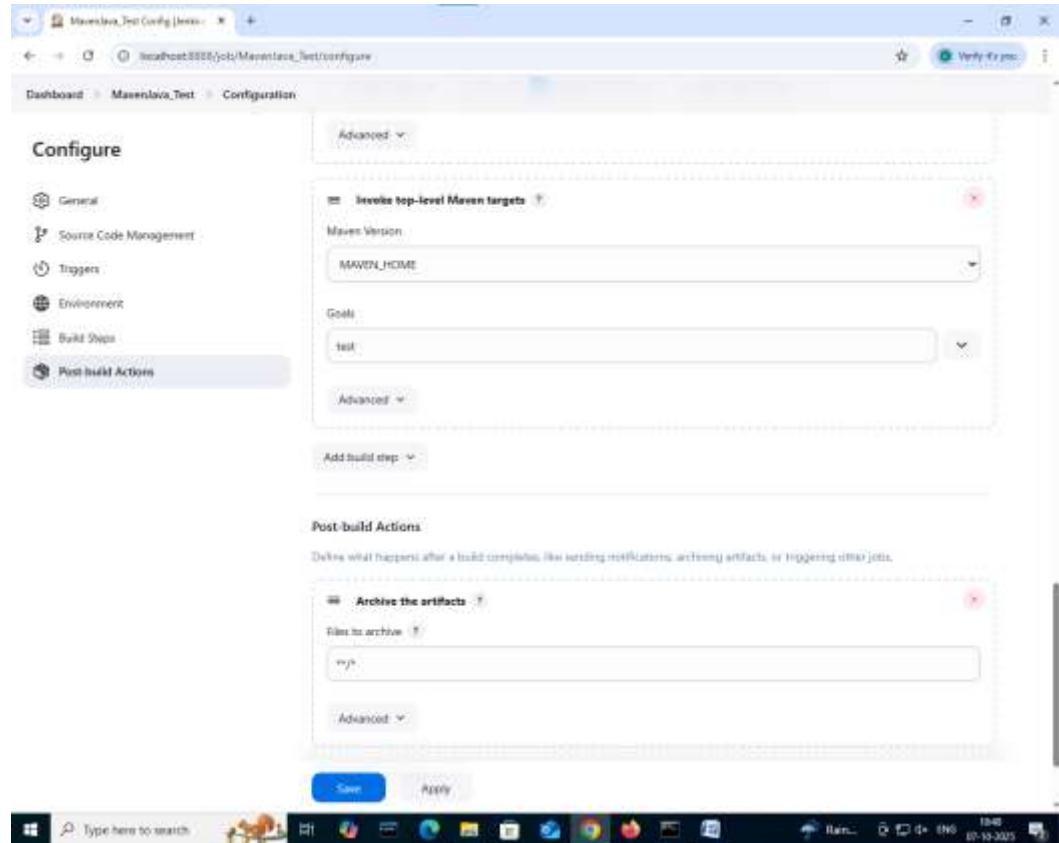
The screenshot shows the Jenkins configuration page for the 'MavenJava\_Test' job. Under 'Source Code Management', the 'None' option is selected. In the 'Environment' section, the checkbox for 'Delete workspace before build starts' is checked. Other options like 'Advanced' and 'Use secret text(s) or file(s)' are available but unchecked. Buttons for 'Save' and 'Apply' are at the bottom.

In the build steps> select add a build step> select “copy artifacts from another project”> give project name as Maven java and artifacts to copy as \*\*/\*

The screenshot shows the Jenkins configuration page for the 'MavenJava\_Test' job. Under 'Build Steps', the 'Copy artifacts from another project' step is selected. It shows the 'Project name' set to 'Mavenjava', 'Which build' set to 'Latest successful build', and 'Artifacts to copy' set to '\*\*/\*'. Other fields like 'Artifacts-not to copy', 'target directory', and 'Parameter filters' are also visible. Buttons for 'Save' and 'Apply' are at the bottom.

In the post build actions> select archive the artifacts and enter files as \*\*/\*

Click on apply and save



In the dashboard you will find MavenJava and MavenJava\_Test

The screenshot shows the Jenkins dashboard. On the left, there are links for 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. A 'Build Queue' section indicates 'No builds in the queue.' Below it, a 'Build Executor Status' table shows 6/2 executors. The main area displays a table of build jobs. The columns are: S (Status), W (Last Build), Name, Last Success, Last Failure, and Last Duration. The jobs listed are: INTERNAL\_JAVA (Status: Red, Last Success: 9 mo 3 days ago, Last Failure: #34, Duration: 0.67 sec), MavenJava (Status: Green, Last Success: 13 days ago, Last Failure: N/A, Duration: 11 sec), MavenJava\_Test (Status: Green, Last Success: 13 days ago, Last Failure: N/A, Duration: 3.4 sec), new (Status: Red, Last Success: 9 mo 3 days ago, Last Failure: #3, Duration: 31 sec), web\_build (Status: Green, Last Success: 9 mo 9 days ago, Last Failure: N/A, Duration: 0.2 sec), web\_deploy (Status: Red, Last Success: N/A, Last Failure: 9 mo 9 days ago, Last Failure: #15, Duration: 0.31 sec), and web\_test (Status: Green, Last Success: 9 mo 9 days ago, Last Failure: N/A, Duration: 3.4 sec).

S	W	Name	Last Success	Last Failure	Last Duration
Red	Cloud	INTERNAL_JAVA	9 mo 3 days ago	#34	0.67 sec
Green	Cloud	MavenJava	13 days ago	#2	N/A
Green	Cloud	MavenJava_Test	13 days ago	#3	N/A
Red	Cloud	new	9 mo 3 days ago	#3	31 sec
Green	Cloud	web_build	9 mo 9 days ago	#8	0.2 sec
Red	Cloud	web_deploy	N/A	9 mo 9 days ago	#15
Green	Cloud	web_test	9 mo 9 days ago	#12	N/A

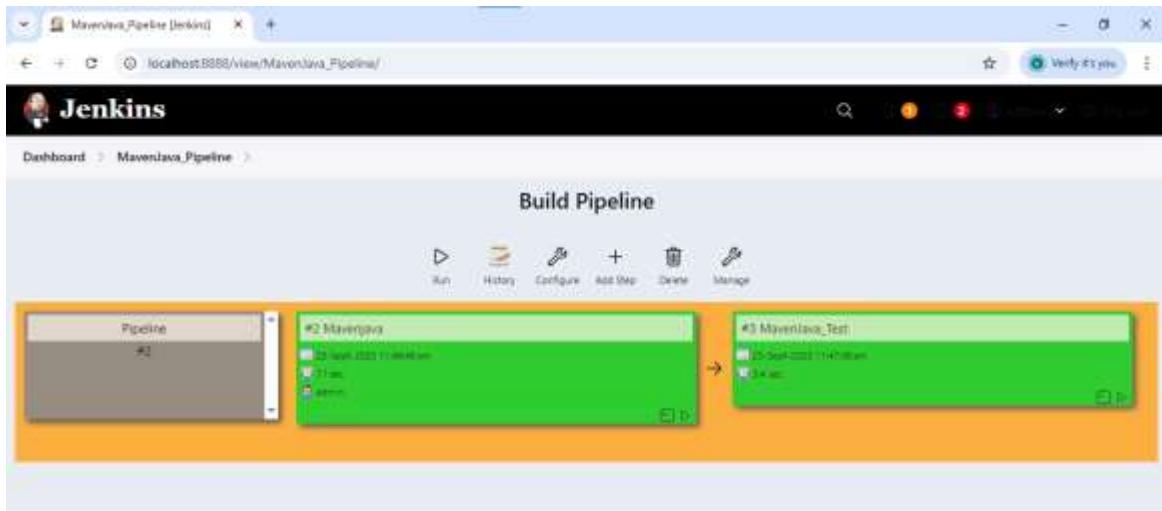
If you open the MavenJava file the following will be shown in case on no errors

The screenshot shows the Jenkins interface for the 'MavenJava' project. The top navigation bar includes links for 'Dashboard', 'Manage Jenkins', and 'Log out'. The main title 'Mavenjava' is displayed with a green checkmark icon. On the left, a sidebar menu lists options like 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', and 'Rename'. Below this is a 'Builds' section with three recent builds listed: 'September 21, 2023' (green), '42 11:49:00' (green), and '41 11:49:00' (green). The central content area is titled 'Last Successful Artifacts' and lists various JAR files with their sizes and download links. At the bottom, sections for 'Downstream Projects' and 'Upstream Projects' are shown, both currently empty. A 'Permalinks' section at the very bottom contains two links: 'Last build (43) 15 days ago' and 'Last stable build (43) 15 days ago'.

If you open the MavenJava\_Test file the following will be shown in case on no errors

The screenshot shows the Jenkins interface for the 'MavenJava\_Test' project. The top navigation bar includes links for 'Dashboard', 'Manage Jenkins', and 'Log out'. The main title 'MavenJava\_Test' is displayed with a green checkmark icon. The left sidebar and build history are identical to the 'MavenJava' project. The central content area is titled 'Last Successful Artifacts' and lists the same set of JAR files as the 'MavenJava' project. At the bottom, sections for 'Upstream Projects' and 'Upstream Projects' are shown, both currently empty. A 'Permalinks' section at the very bottom contains two links: 'Last build (43) 15 days ago' and 'Last stable build (43) 15 days ago'.

## MavenJava\_pipeline

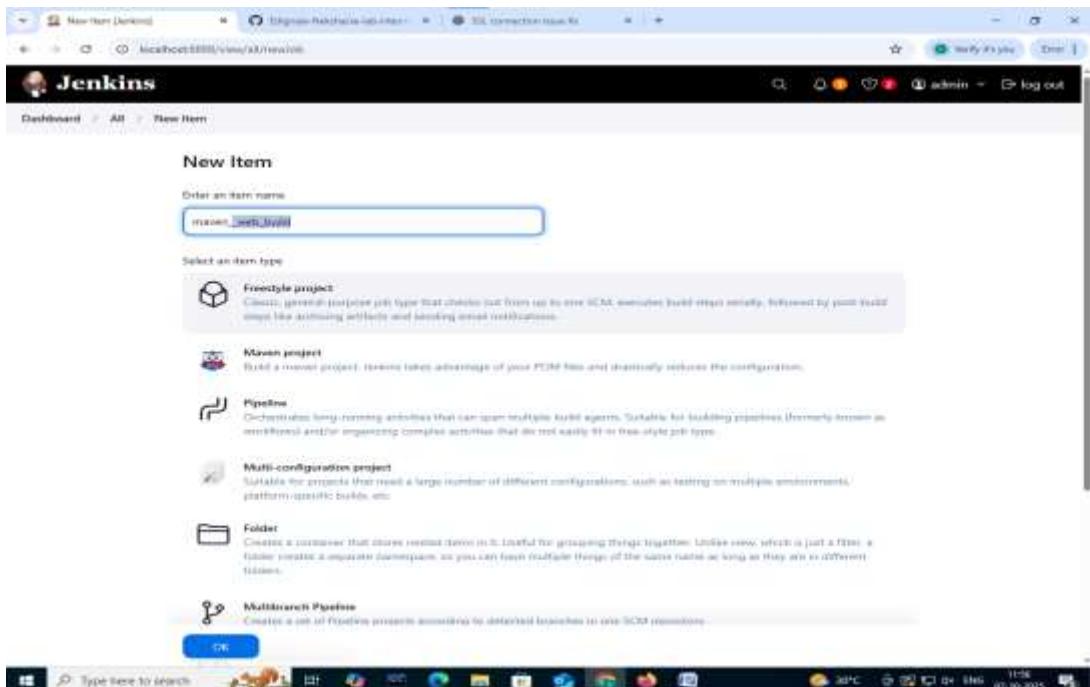


## II. Maven Web Automation Steps:

Create Freestyle Project (e.g., MavenWeb\_Build)

### Step 1: Open Jenkins (localhost:8888)

Click on "New Item" (left side menu) and name it as maven\_web\_build > select freestyle project > click on "OK"



## Step 2: Configuration of maven\_web\_build project

Give the description

The screenshot shows the Jenkins configuration interface for the 'maven\_web\_build' project. The top navigation bar includes tabs for 'Dashboard', 'Jenkins', 'Builds', 'Queue', 'Console', and 'Logs'. Below the navigation is a search bar and a user dropdown for 'admin'. The main content area has a left sidebar with 'Configure' and 'General' tabs, and a right sidebar with sections for 'Source Code Management', 'Triggers', 'Environment', 'Build Steps', and 'Post-build Actions'. The 'General' tab is active, showing a 'Description' field containing 'web build demo'. The 'Enabled' switch is turned on. In the 'Post-build Actions' section, several checkboxes are listed: 'Discard old builds', 'GitHub project', 'Permission to Copy Artifact', 'This project is parameterized', 'Throttle builds', and 'Execute concurrent builds if necessary'. At the bottom of the configuration page are 'Save' and 'Apply' buttons.

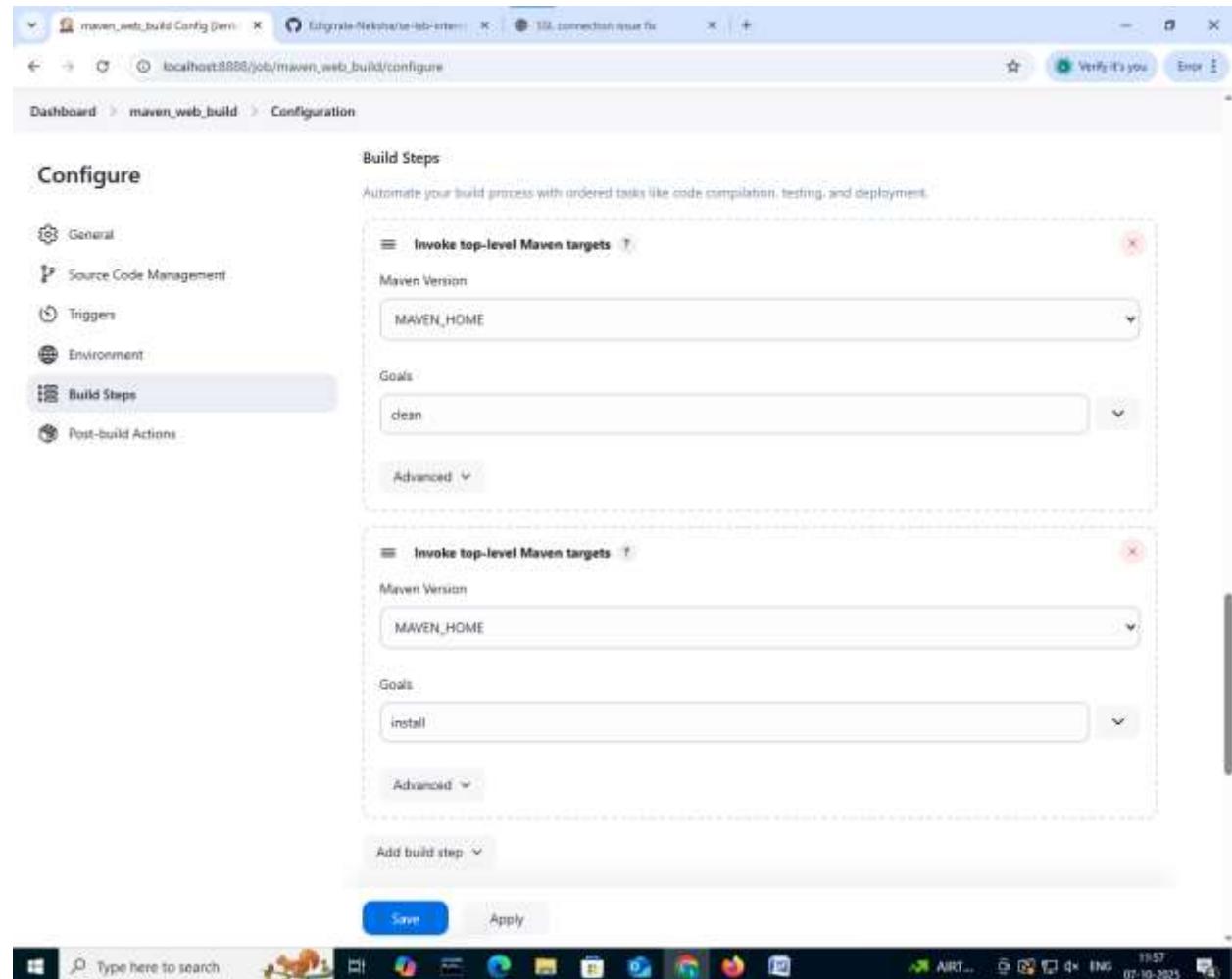
In the source code management select git and give the git repo link

The screenshot shows a Jenkins configuration page for a job named "maven\_web\_build". The left sidebar has tabs for General, Source Code Management (which is selected), Triggers, Environment, Build Steps, and Post-build Actions. The main content area is titled "Source Code Management" with the sub-instruction "Connect and manage your code repository to automatically pull the latest code for your builds." A radio button for "None" is unselected, while a radio button for "Git" is selected. Below the radio buttons is a section titled "Repositories" with a "Repository URL" input field containing "https://github.com/Edigitala-Neksha/se-lab-internal-1.git". Underneath it is a "Credentials" dropdown menu showing "- none -" and a "+ Add" button. There is also an "Advanced" dropdown menu. Below these sections is a "Add Repository" button. Further down is a "Branches to build" section with a "Branch Specifier (blank for 'any')" input field containing "\*main". At the bottom of the page are "Save" and "Apply" buttons, and a status bar at the bottom of the screen.

In the build steps click on add build step > give maven version as MAVEN\_HOME > select invoke top-level maven targets > goals as clean

For the second build step,

In the build steps click on add build step > give maven version as MAVEN\_HOME > select invoke top-level maven targets > goals as install



In the post build actions > click on add post build action > select the archive the artifacts > in the file to archive give “\*\*/\*”

For the second post build action,

In the post build actions > click on add post build action > select build other projects > give projects to build as maven\_web\_test

Click on apply and save

The screenshot shows the Jenkins job configuration page for 'maven\_web\_build'. The 'Post-build Actions' section is active. It contains two entries:

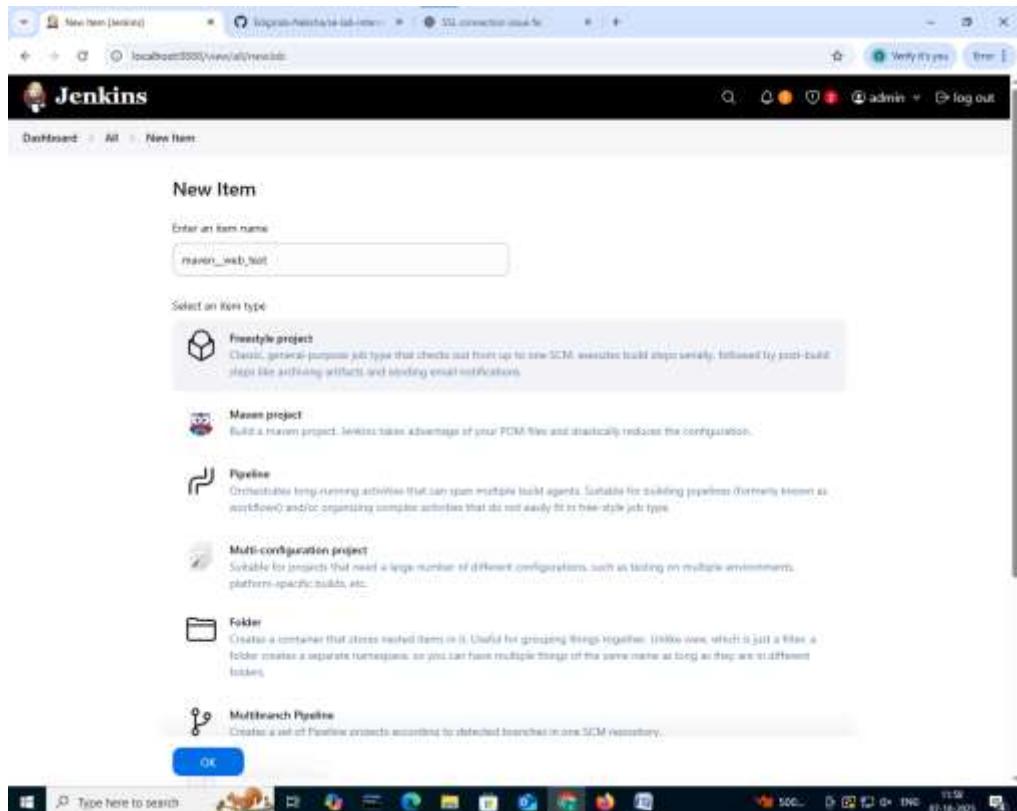
- Archive the artifacts**: Set to archive all files (\*\*/\*).
- Build other projects**: Set to build 'maven\_web\_test'. The trigger option 'Trigger only if build is stable' is selected.

At the bottom, there are 'Save' and 'Apply' buttons. The status bar at the bottom right indicates 'jenkins 2.489'.

Create Freestyle Project (e.g., MavenWeb\_Test):

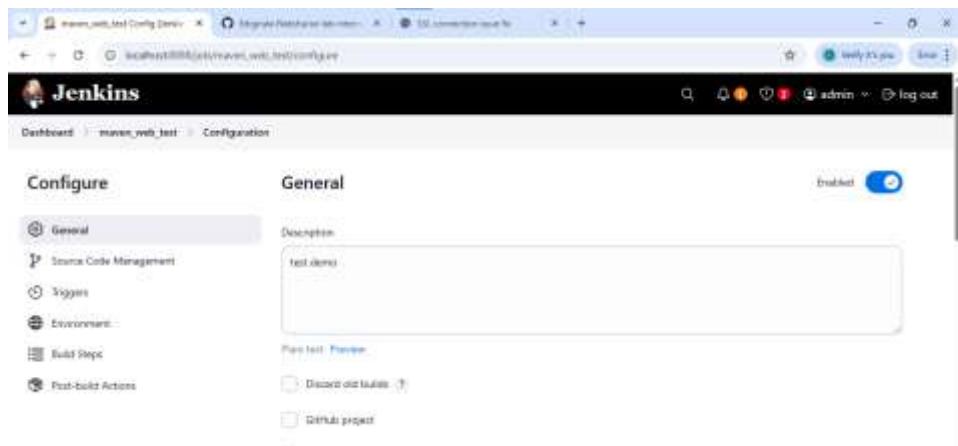
### Step 1: Open Jenkins (localhost:8888)

Click on "New Item" (left side menu) and name it as maven\_web\_test > select freestyle project > click on "OK"



### Step 2: Configuration of maven\_web\_test project

Give the description



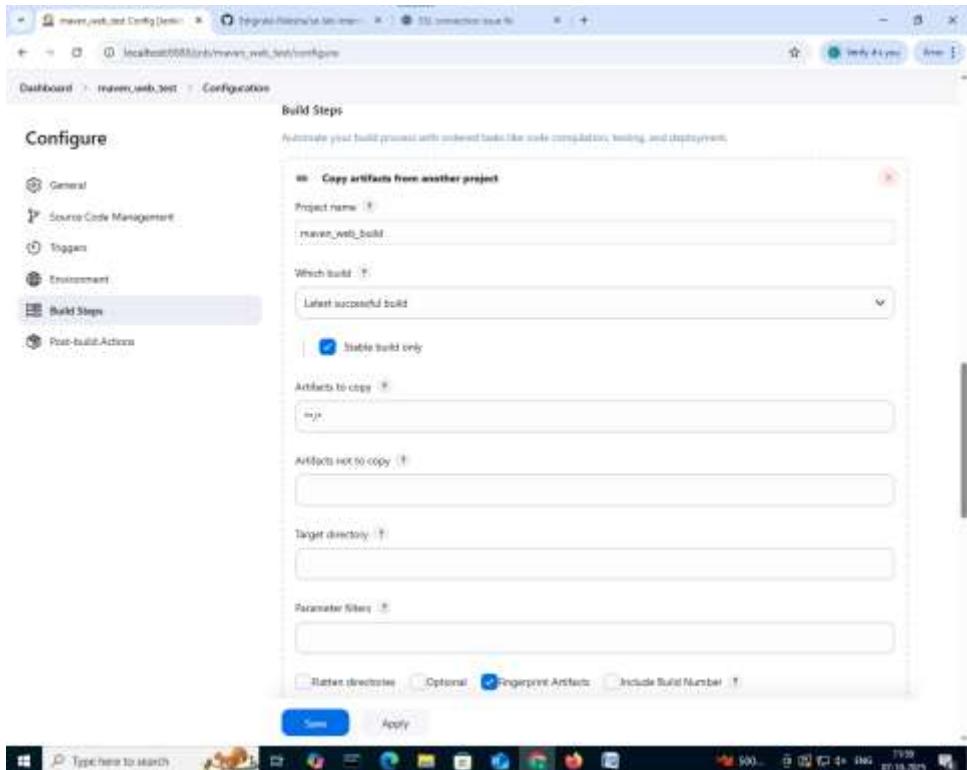
In the source code management select none and in environment select “delete workspace before build starts”

The screenshot shows the Jenkins job configuration interface for a job named "maven\_web\_test". The left sidebar lists sections: General, Source Code Management (selected), Triggers, Environment, Build Steps, and Post-build Actions. The main content area has two tabs: "Source Code Management" and "Environment".

**Source Code Management:** A note says "Connect and manage your code repository to automatically pull the latest code for your builds." Below it, a radio button for "None" is selected, while "Git" is unselected. A "Triggers" section follows, containing several checkboxes for triggering builds based on events like remote triggers, other project builds, periodic polling, GitHub hooks, or SCM polling.

**Environment:** A note says "Configure settings and variables that define the context in which your build runs, like credentials, paths, and global parameters." Below it, a checkbox for "Delete workspace before build starts" is checked. An "Advanced" dropdown menu is open, showing options: "Use secret text(s) or file(s)", "Provide Configuration files", "Add timestamps to the Console Output", and "Inspect build log for published build scans". At the bottom of the configuration page are "Save" and "Apply" buttons.

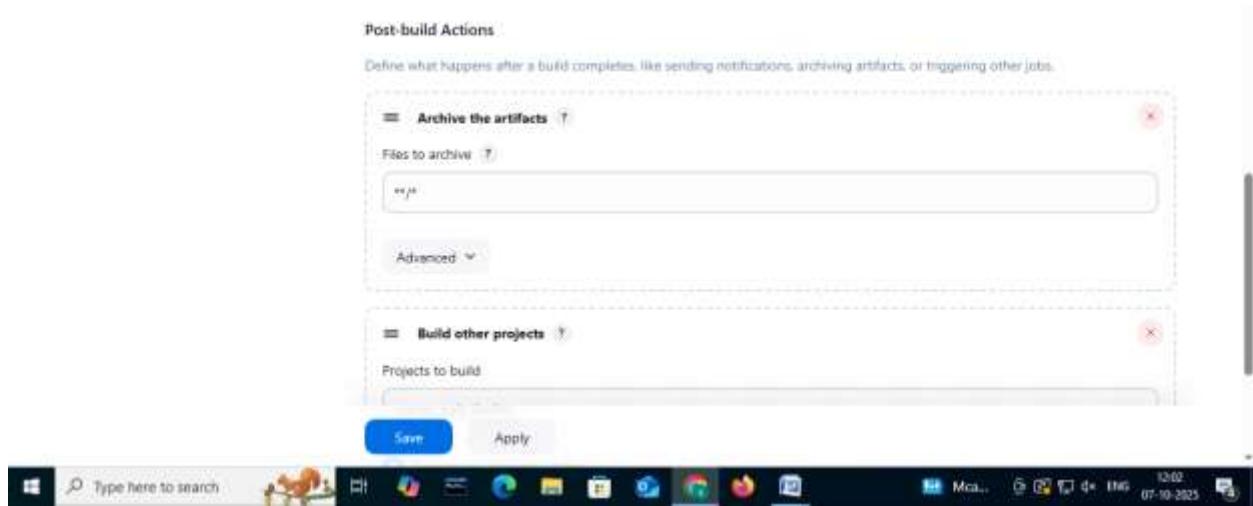
In the build steps click on add build step > select copy artifacts from another project > give project name as maven\_web\_build > give artifacts to copy as \*\*/\*



In the build steps click on add build step > give maven version as MAVEN\_HOME > select invoke top-level maven targets > goals as test



In the post build actions > click on add post build action > select the archive the artifacts > in the file to archive give \*\*/\*



In the post build actions > click on add post build action >select build other projects > give name as maven\_web\_deploy> select “trigger only if build is stable”



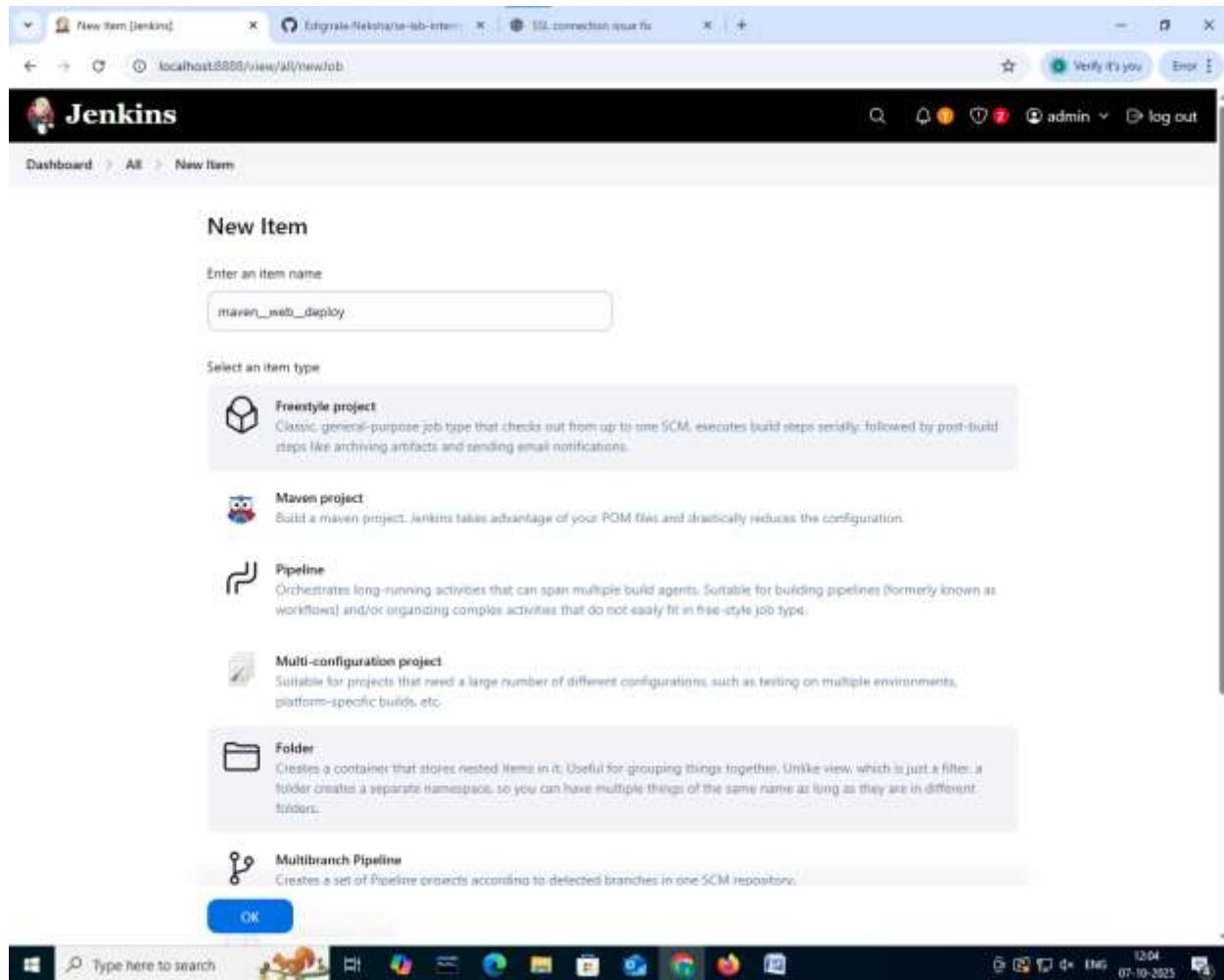
If the build is success:

The screenshot shows a Windows desktop environment. At the top, there is a taskbar with several open windows: 'maven\_web\_test [Jenkins]', 'Integrate-Netsharne-lib-internal...', 'Apache Tomcat/9.0.80', and 'Jenkins support for Java 21'. Below the taskbar is a browser window displaying the Jenkins interface for the 'maven\_web\_test' job. The page shows the job status as 'Success' (green checkmark), the last commit message 'test demo', and the last successful artifacts. It also lists upstream and downstream projects. On the left, there is a sidebar with options like 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', and 'Rename'. Below this is a 'Builds' section showing the history of builds from today, with builds #4, #3, #2, and #1 listed. The desktop background is white, and the system tray at the bottom right shows the date (07-10-2025), time (12:37), battery level (30%), and other system icons.

Create Freestyle Project (e.g., MavenWeb\_Deploy):

### Step 1: Open Jenkins (localhost:8888)

Click on "New Item" (left side menu) and name it as maven\_web\_deploy > select freestyle project > click on "OK"



## Step 2: Configuration of maven\_web\_deploy project

Give the description

The screenshot shows the Jenkins configuration interface for a job named "maven\_web\_deploy". The "General" configuration tab is selected. The "Description" field contains the text "deploy demo". Under the "Post-build Actions" section, several checkboxes are listed: "Discard old builds", "GitHub project", "Permission to Copy Artifact", "This project is parameterized", "Throttle builds", and "Execute concurrent builds if necessary". The "Enabled" switch is turned on. Below the general configuration, there is a "Source Code Management" section with a note to "Connect and manage your code repository to automatically pull the latest code for your builds". At the bottom of the configuration page, there are "Save" and "Apply" buttons. The background shows a Windows taskbar at the bottom.

In the source code management select none and in environment select “delete workspace before build starts”

The screenshot shows the Jenkins configuration interface for a job named "maven\_web\_deploy". The "Source Code Management" section is selected, showing the "None" option is chosen. The "Environment" section is also visible, with the "Delete workspace before build starts" checkbox checked. The Jenkins status bar at the bottom indicates "NIFTY" and the date "07-10-2025".

**Source Code Management**

Select and manage your code repository to automatically pull the latest code for your builds.

None

Git

**Triggers**

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

Trigger builds remotely (e.g., from scripts)

Build after other projects are built

Build periodically

GitHub hook trigger for GITScm polling

Poll SCM

**Environment**

Configure settings and variables that define the context in which your build runs, like credentials, paths, and global parameters.

Delete workspace before build starts

Advanced

Use secret text(s) or file(s)

Provide Configuration files

Add timestamps to the Console Output

Inspect build log for published build scans

**Save** **Apply**

In the build steps click on add build step > select copy artifacts from another project > give project name as maven\_web\_test > give artifacts to copy as \*\*/\*

The screenshot shows a Jenkins configuration page for a job named 'maven\_web\_deploy'. The left sidebar lists configuration sections: General, Source Code Management, Triggers, Environment, Build Steps (which is selected), and Post-build Actions. The main content area is titled 'Build Steps' and contains a single step configuration:

- Project name:** maven\_web\_test
- Which build:** Latest successful build
- Stable build only:** Checked
- Artifacts to copy:** \*\*/\*
- Artifacts not to copy:** (empty)
- Target directory:** (empty)
- Parameter filters:** (empty)
- Advanced options:** Flatten directories (unchecked), Optional (unchecked), Fingerprint Artifacts (checked), Include Build Number (unchecked)

At the bottom are 'Save' and 'Apply' buttons.

In the post build actions > click on add post build actions > select deploy war/ear to a container > enter war/ear files as \*\*/\*.war > context path as webpath > give the credentials and tomcat URL

The screenshot shows the Jenkins configuration interface for a job named 'maven\_web\_deploy'. The 'Post-build Actions' section is selected. A 'Deploy war/ear to a container' action is added. In the 'WAR/EAR files' field, '\*\*/\*.war' is specified. The 'Context path' is set to 'webpath'. Under the 'Containers' section, a 'Tomcat 9.x Remote' container is selected. The 'Credentials' dropdown contains 'admin/\*\*\*\*\*'. The 'Tomcat URL' is set to 'https://localhost:8080/'. There are 'Save' and 'Apply' buttons at the bottom.

If the build is success:

The screenshot shows a Jenkins job page for 'maven\_web\_deploy'. The status is green with a checkmark, indicating a successful build. The build number is #13, which was run at 12:36PM. The build log link is 'deploy demo'. Below the status, there's a section for 'Upstream Projects' with a link to 'maven\_web\_test'. The 'Builds' section lists the last 10 builds, all of which are successful (green circles). The Jenkins interface includes a sidebar with options like Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. The top navigation bar shows tabs for 'Dashboard', 'maven\_web\_deploy', and other Jenkins instances.

## Create Pipeline View for MavenWeb

Click "+" beside "All" on the dashboard and Enter name as maven\_web\_pipeline

Select type as build pipeline view

The screenshot shows the Jenkins interface for creating a new view. The title bar says 'New view [Jenkins]'. The main area is titled 'New view' with a sub-section 'Name' containing the value 'maven\_web\_pipeline'. Below this is a 'Type' section with three options: 'Build Pipeline View' (selected), 'List View', and 'My View'. The 'Build Pipeline View' option is described as showing jobs in a build pipeline view. The 'List View' option is described as showing items in a simple list format. The 'My View' option is described as automatically displaying all jobs the user has access to. At the bottom right of the form is a blue 'Create' button.

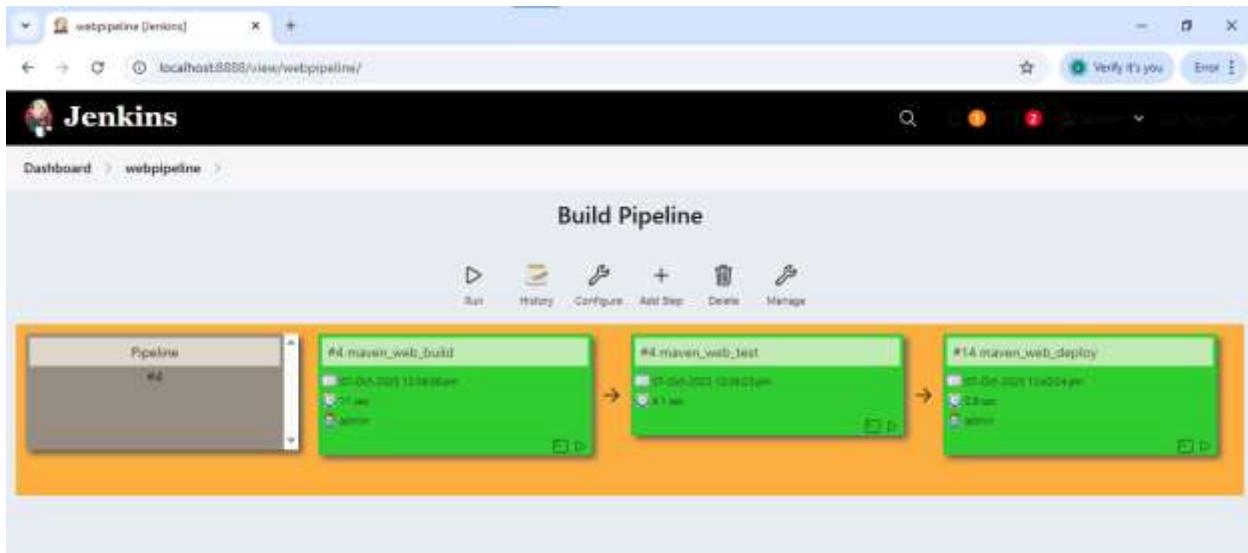
Give the description and in the upstream directly the maven\_web\_build will be shown

The screenshot shows the Jenkins 'Edit View' configuration page for a view named 'maven\_web\_pipeline'. The 'Name' field is filled with 'maven\_web\_pipeline'. The 'Description' field contains the placeholder text 'Describe the purpose of this view.'. In the 'Pipeline Flow' section, the 'Upstream / downstream config' dropdown is set to 'Select Pipeline Job', and 'maven\_web\_build' is listed as the selected option. Below this, the 'Trigger Options' section has 'Save' and 'Apply' buttons.

Click on apply and save

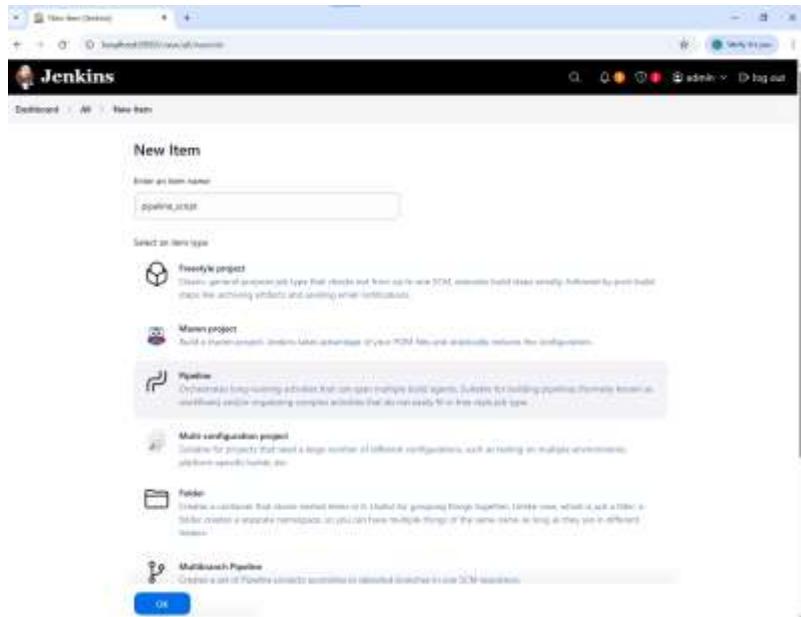
The screenshot shows the Jenkins 'Edit View' configuration page for a view named 'maven\_web\_pipeline'. The 'Column Headers' dropdown is set to 'No header'. The 'Refresh Frequency (in seconds)' input field is set to '3'. The 'Console Output Unit Style' dropdown is set to 'LightBox'. The 'Widgets' section contains two checkboxes: 'Filter build queue' (unchecked) and 'Filter build executors' (unchecked). At the bottom, there are 'Save' and 'Apply' buttons.

In the stage view it we be shown as:

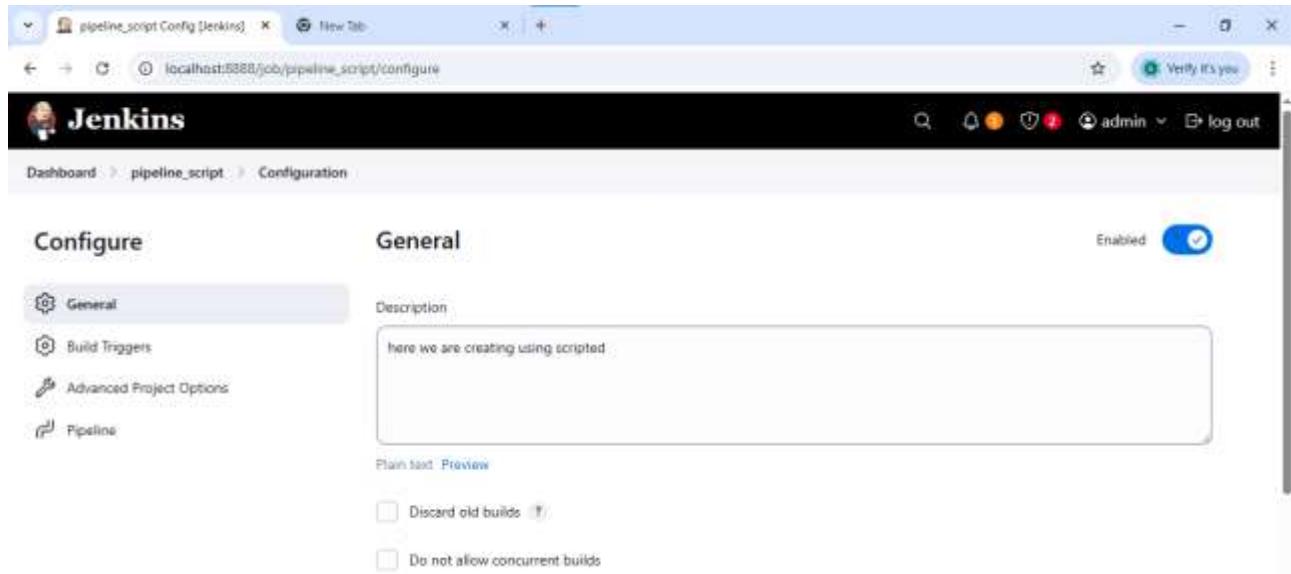


## **9.Pipeline Creation using script**

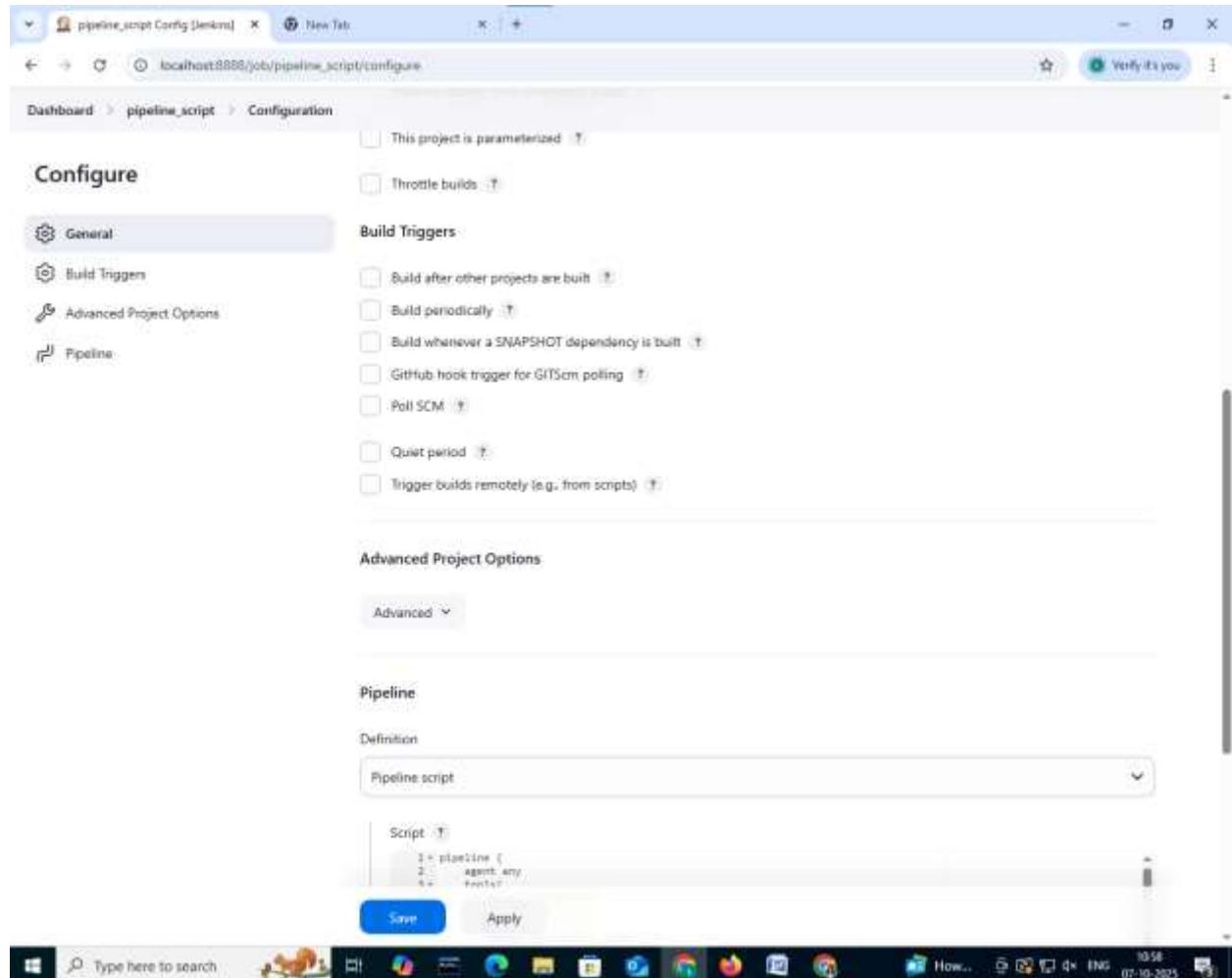
Step 1: In the Jenkins select the new item and give the name as pipeline\_script and select pipeline and click ok



Step 2: In the configuration, give the description



Step 3: In the pipeline section give definition as pipeline script and enter the script with git reop link and project name



The screenshot shows the Jenkins configuration interface for a job named "pipeline\_script". The "General" tab is selected under the "Configure" section. The "Pipeline" tab is also visible. Under "Build Triggers", several options are listed: "Build after other projects are built", "Build periodically", "Build whenever a SNAPSHOT dependency is built", "GitHub hook trigger for GITScm polling", "Poll SCM", "Quiet period", and "Trigger builds remotely (e.g., from scripts)". The "Advanced Project Options" section is collapsed. The "Pipeline" section is expanded, showing the "Definition" dropdown set to "Pipeline script" and a script editor containing the following Groovy code:

```
1+ pipeline {
2+   agent any
3+ }
```

Below the script editor are "Save" and "Apply" buttons.

Step 4: click on apply and then save

The screenshot shows the Jenkins 'Configure' screen for a pipeline job named 'pipeline\_script'. The 'Advanced Project Options' tab is selected. The 'Pipeline' section contains a 'Definition' dropdown set to 'Pipeline script' and a code editor containing the following Groovy script:

```
1<+ pipeline {
2+   agent any
3+   tools{
4+     'maven' 'ECLIPSE-M2'
5+   }
6+   stages {
7+     stage('git repos & clean') {
8+       steps {
9+         bat "cd ..\nmvn clean"
10+        bat "git clone https://github.com/ServitacionM2/https://silicon-maven-projects.git"
11+        bat "mv clean -f silicon-maven-projects"
12+      }
13+    }
14+    stage("Install") {
15+      steps {
16+        bat "mvn install -f silicon-maven-projects"
17+      }
18+    }
19+  }
20+}
```

Below the code editor is a checkbox labeled 'Use Groovy Sandbox' which is checked. At the bottom of the page are two buttons: 'Save' and 'Apply'. The status bar at the bottom right indicates 'REST API' and 'Jenkins 2.409'.

Step 8: Check the stage view. If is successful.

The screenshot shows the Jenkins interface for the 'pipeline\_script' job. The top navigation bar includes the Jenkins logo, a search bar, and user information for 'admin'. The main content area displays the 'Stage View' for the 'pipeline\_script' job. On the left, there's a sidebar with links for Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Stages, Rename, and Pipeline Syntax. The 'Status' link is currently selected, indicated by a highlighted bar. The central part of the screen shows the 'Stage View' with five stages: Declarative: Tool Install, git repo & clean, install, test, and package. Below these stages is a timeline showing the average stage times: 296ms, 5s, 9s, 3s, and 4s respectively. A legend at the bottom indicates the colors: blue for Oct 17, grey for Nov 2, and green for changes. To the right of the stage view, there's a 'Permalinks' section listing four build links: Last build (#2), Last stable build (#2), Last successful build (#2), and Last completed build (#2). The last link is highlighted in blue.

## **10. Kubernetes Using Minikube:**

### **Step -1:**

#### **Start Minikube : Command- minikube start**

- First, you need to start your Kubernetes cluster using Minikube.
- When you start it, Minikube sets up a lightweight virtual machine on your system and runs a local Kubernetes node inside it.

#### **Step-2:Then check for the status Minikube status**

#### **Step-3:Create an image**

```
PS C:\Users\User>
PS C:\Users\User> kubectl delete deployment mynginx
deployment.apps "mynginx" deleted
PS C:\Users\User> kubectl create deployment mynginx --image=nginx
deployment.apps/mynginx created
PS C:\Users\User> kubectl expose deployment mynginx --type=NodePort --port=80
service/mynginx exposed
PS C:\Users\User> kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
mynginx-79bb8756c7-wpslj   1/1     Running   0          34s
```

#### **Step-4: Check the NGINX Service Details**

- After creating the service, check its details to see which port Kubernetes assigned to it.

```
QoS Class:      BestEffort
nodeSelector:   <none>
Tolerations:   node.kubernetes.io/not-ready:NoExecute opExists for 300s
                node.kubernetes.io/unreachable:NoExecute opExists for 300s
Events:         Type  Reason  Age   From            Message
                ----  -----  --   --              --
Normal  Scheduled  68s  default-scheduler  Successfully assigned default/mynginx-79bb8756c7-wpslj to minikube
Normal  Pulling   67s  kubelet        Pulling image "nginx"
Normal  Pulled    65s  kubelet        Successfully pulled image "nginx" in 2.416s (2.416s including waiting). Image size: 199974475 bytes.
Normal  Created   65s  kubelet        Created container nginx
Normal  Started   64s  kubelet        Started container nginx
PS C:\Users\User> kubectl scale deployment mynginx --replicas=4
deployment.apps/mynginx scaled
PS C:\Users\User> kubectl get service mynginx
Error from server (NotFound): services "mynginx" not found
PS C:\Users\User> kubectl port-forward svc/mynginx 8081:80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::]:8081 -> 80
```

### Step-5:check the detail of the kubectl .

```
PS C:\Users\User> kubectl describe pods
Name:           mynginx-79bb8756c7-wpslj
Namespace:      default
Priority:      0
Service Account: default
Node:          minikube/192.168.49.2
Start Time:    Tue, 14 Oct 2025 12:38:19 +0530
Labels:        app=mynginx
               pod-template-hash=79bb8756c7
Annotations:   <none>
Status:        Running
IP:            10.244.0.16
IPs:
  IP:          10.244.0.16
Controlled By: ReplicaSet/mynginx-79bb8756c7
Containers:
  nginx:
    Container ID:  docker://675066efbd98a54ba39177103943b196de2c61f01d820ede859b48578f3e245e
    Image:         nginx
    Image ID:     docker-pullable://nginx@sha256:3b7732505933ca591ce4a6d860cb713ad96a3176b82f7979a8dfa9973486a0d6
    Port:          <none>
    Host Port:    <none>
    State:        Running
      Started:   Tue, 14 Oct 2025 12:38:22 +0530
    Ready:        True
    Restart Count: 0
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-nh2rw (ro)
Conditions:
  Type          Status
  PodReadyToStartContainers  True
  Initialized    True
  Ready          True
  ContainersReady  True
  PodScheduled   True
Volumes:
  kube-api-access-nh2rw:
    Type:       Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:        kube-root-ca.crt
    ConfigMapOptional:    <nil>
    DownwardAPI:          true
    QoS Class:            BestEffort
    Node-Selectors:       <none>
    Tolerations:
      node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
      node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
```

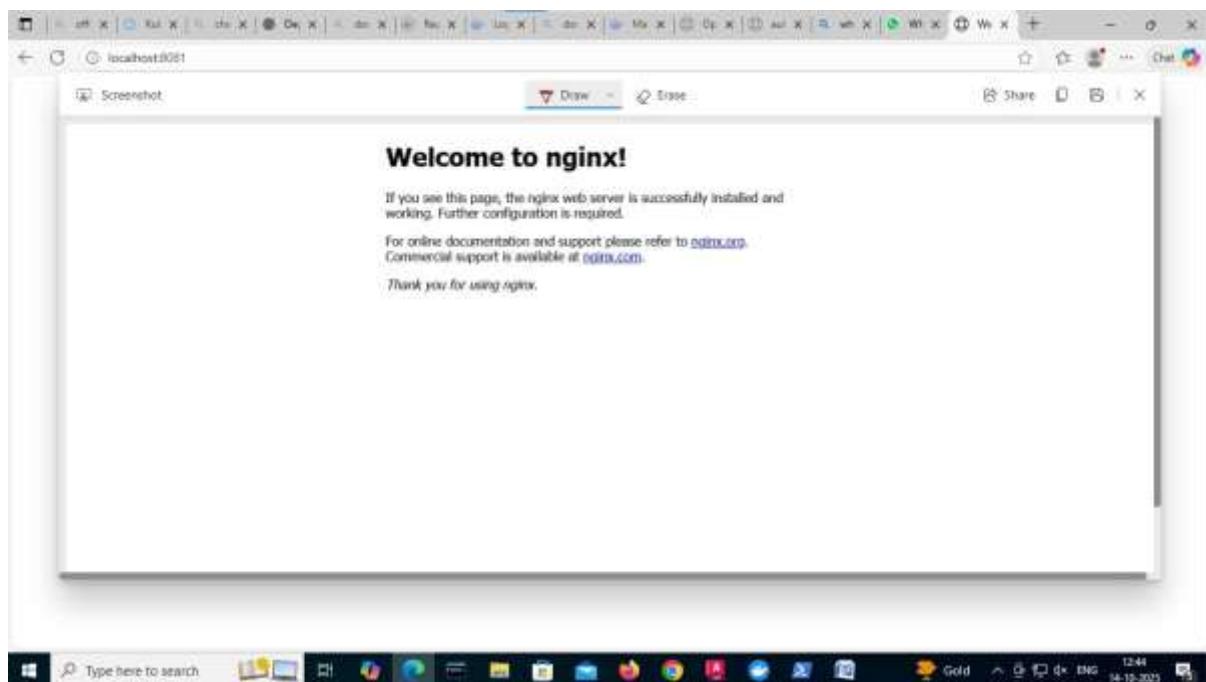
### Step-6:Check the NGINX Service Details

- After creating the service, check its details to see which port Kubernetes assigned to it.

```
PS C:\Users\User> kubectl port-forward svc/mynginx 8081:80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
Handling connection for 8081
Handling connection for 8081
Handling connection for 8081
Handling connection for 8081
```

## Step-7: Open NGINX in the Browser

- Now that your service is exposed, you can open NGINX in your browser.



## 11. Jenkins-CI/CD

### Setting Up Jenkins CI-----using GitHub Webhook with Jenkins

Step 1: Take the authentication key from the ngrok and setup in ngrok terminal

```
tcp          start a TCP tunnel
tls          start a TLS endpoint
update      update ngrok to the latest version
version     print the version string

EXAMPLES:
# forward http traffic from assigned public URL to local port 80
ngrok http 80
# port 8080 available at baz.ngrok.dev
ngrok http --url baz.ngrok.dev 8080
# tunnel arbitrary TCP traffic to port 22
ngrok tcp 22
# secure your app with oauth
ngrok http 80 --oauth=google --oauth-allow-email=foo@foo.com

Paid Features:
ngrok http 80 --url mydomain.com                               # run ngrok with your own custom domain
ngrok http 80 --cidr-allow 2600:8c00::a03c:9le:fe69:9695/32   # run ngrok with IP policy restrictions
Upgrade your account at https://dashboard.ngrok.com/billing/subscription to access paid features

Upgrade your account at https://dashboard.ngrok.com/billing/subscription to access paid features

Flags:
-h, --help      help for ngrok

Use "ngrok [command] --help" for more information about a command.

ngrok is a command line application, try typing 'ngrok.exe http 80'
at this terminal prompt to expose port 80.
C:\Windows\System32>ngrok config add-authtoken 34gKWhQDcoITj34K6eN73XoYG6J_58fBgmPjM5ikZVdKVdYCe|
```

Step-2: Execute the following command using the port number on which Jenkins is running

```
C:\Windows\System32>ngrok.exe http 8888
```

- Following output will be given:

```
ngrok                                         (Ctrl+C to quit)

+ Block threats before they reach your services with our WAF actions - https://ngrok.com/z/waf

Session Status:          online
Account:                 Neksha-Edigirala (Plan: Free)
Update:                  update available (version 3.32.0, Ctrl-U to update)
Version:                 3.24.0-msix
Region:                  India (in)
Latency:                 147ms
Web Interface:           http://127.0.0.1:4040
Forwarding:              https://corkier-darla-handsome.ngrok-free.dev -> http://localhost:8888

Connections:             ttl     opn     rti     rt5     p50     p98
                           2       0       0.00    0.00   38.28   39.47

HTTP Requests:
-----

11:35:59.377 IST POST /github-webhook/          200 OK
11:34:29.479 IST POST /github-webhook/          200 OK
```

Go to Jenkins:

Step-3: Create the Jenkins job in the source code management select the git and enter git repo url and make sure the branch is same (i.e., main)

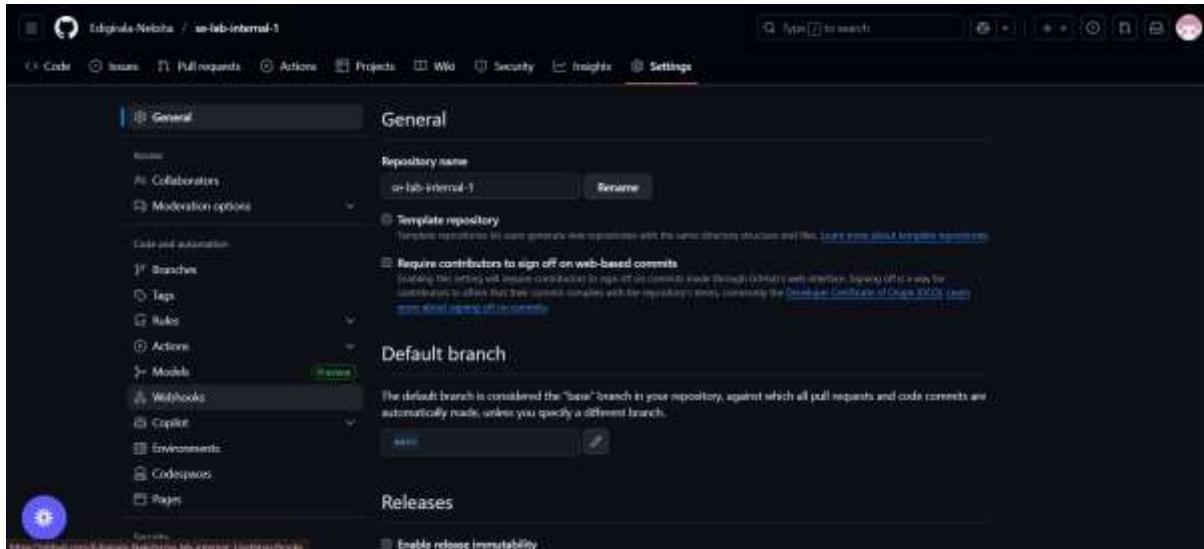
The screenshot shows the Jenkins job configuration page for 'job\_webhook.java'. Under 'Source Code Management', 'Git' is selected. The 'Repository URL' field contains 'https://github.com/digitalo-Nekhaan-lab/internal-1.git'. The 'Branches to build' dropdown is set to 'main'. Other tabs like General, Triggers, Environment, Build Steps, and Post-build Actions are visible on the left.

Step-4: In the triggers section select “Github hook trigger for GITScm polling”

The screenshot shows the Jenkins job configuration page for 'job\_webhook.java'. Under 'Triggers', the 'Github hook trigger for GITScm polling' checkbox is checked. Other options like 'Trigger builds remotely' and 'Build periodically' are unchecked. The 'Save' button is at the bottom.

Click on apply and save

Step-6: open the git hub repo open setting of repo and then go to webhooks

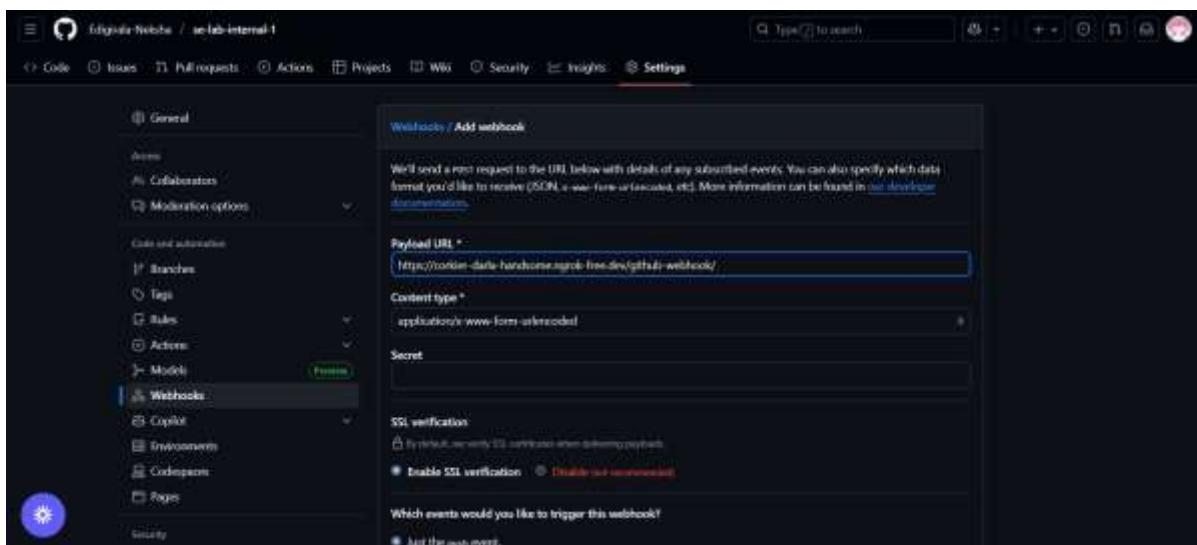


The screenshot shows the GitHub repository settings page for 'eduphia-NestJS / se-lab-intern-1'. The 'General' tab is selected. On the left, there's a sidebar with options like Issues, Collaborators, Moderation options, Branches, Tags, Rules, Actions, Models, Webhooks (which is currently selected), Copilot, Environments, CodeSpaces, and Pages. The main area has sections for 'Repository name' (set to 'se-lab-intern-1'), 'Template repository' (disabled), 'Require contributors to sign off on web-based commits' (disabled), and 'Default branch' (set to 'main'). Below these are sections for 'Releases' and 'Enable release immutability'.

Step-7: Click on add a webhook and take the forwarding URL from ngrok and paste in payload URL and add /github-webhook/ along with the forwarding url

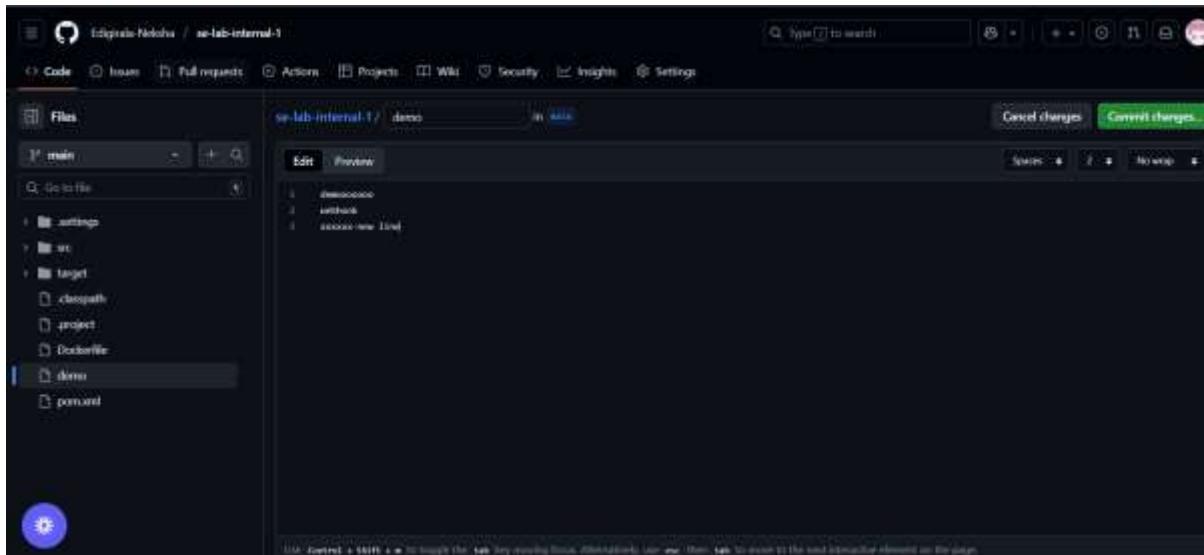
Forwarding URL: <https://corkier-darla-handsome.ngrok-free.dev>

Payload url: <https://corkier-darla-handsome.ngrok-free.dev/github-webhook/>

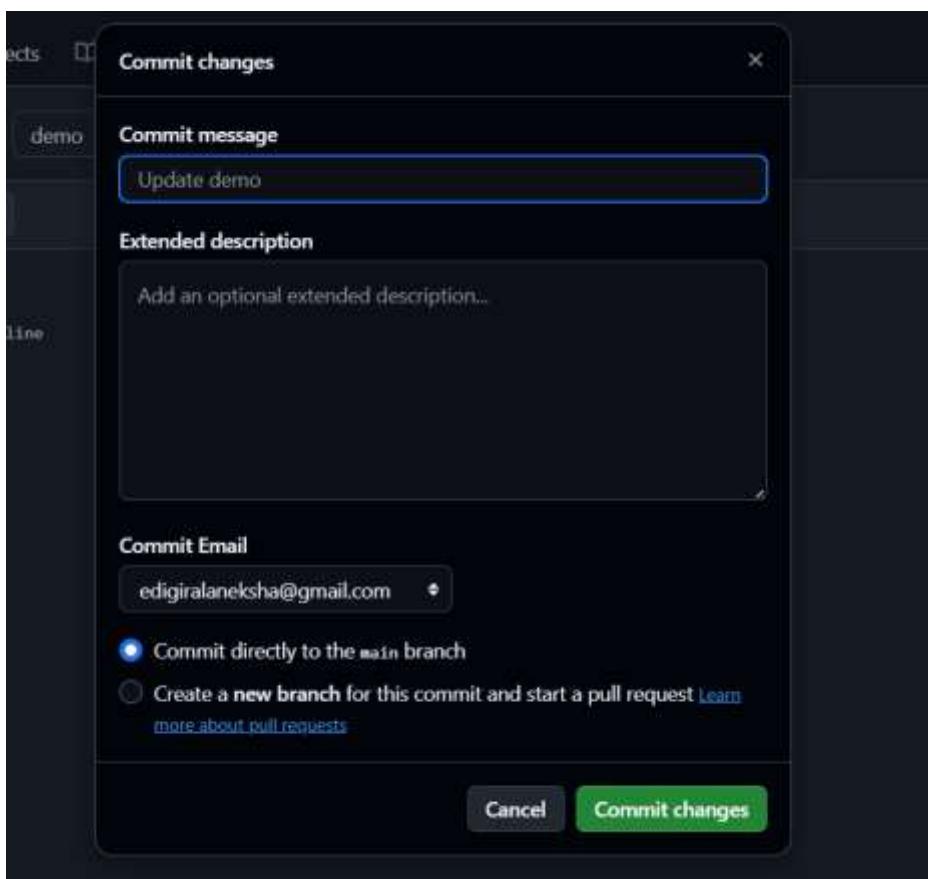


The screenshot shows the 'Webhooks / Add webhook' dialog box. It includes fields for 'Payload URL' (containing 'https://corkier-darla-handsome.ngrok-free.dev/github-webhook/'), 'Content type' (set to 'application/x-www-form-urlencoded'), and 'Secret' (empty). Under 'SSL verification', it says 'By default, servers will verify SSL certificates when delivering payloads' and has two radio button options: 'Enable SSL verification' (selected) and 'Disable SSL verification'. At the bottom, it asks 'Which events would you like to trigger this webhook?' with the option 'Just the push event'.

## Step 8: make changes in the files in github

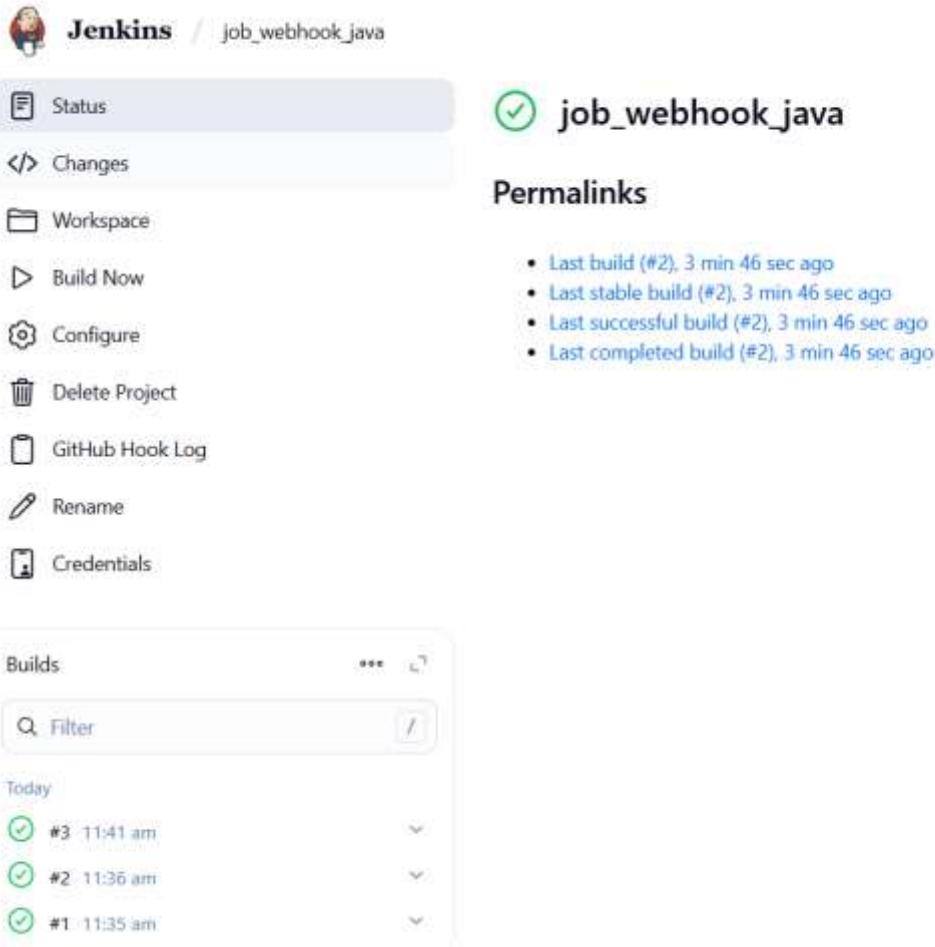


## Step 9: click on commit changes



Step 10: open Jenkins the build will start automatically

The screenshot shows the Jenkins interface for the 'job\_webhook\_java' project. The top navigation bar includes links for 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', 'GitHub Hook Log', 'Rename', and 'Credentials'. The 'Status' tab is currently selected, indicated by a blue background. On the right side, under 'Permalinks', there is a list of four recent builds: 'Last build (#2), 3 min 46 sec ago', 'Last stable build (#2), 3 min 46 sec ago', 'Last successful build (#2), 3 min 46 sec ago', and 'Last completed build (#2), 3 min 46 sec ago'. Below this is a 'Builds' section with a 'Pending' status. It lists a single build entry: '#3' with a pending status icon, followed by the note 'In the quiet period. Expires in 2.9 sec'. Under the 'Today' heading, there is a completed build entry: '#2 11:36 am' with a green checkmark icon.



The screenshot shows the Jenkins interface for the 'job\_webhook\_java' project. At the top, there's a sidebar with various options: Status (highlighted), Changes, Workspace, Build Now, Configure, Delete Project, GitHub Hook Log, Rename, and Credentials. Below this is a 'Builds' section with a 'Filter' input field. It lists three builds from today: #3 (11:41 am), #2 (11:36 am), and #1 (11:35 am). Each build entry has a dropdown arrow next to it.

You can check status : started by git hub push



This screenshot shows the detailed view of build #3. The top bar includes links for Status, Changes, Console Output, Edit Build Information, Delete build #3, Polling Log, Timings, Git Build Data, and Previous build. The main content area displays the build number (#3), timestamp (28-Oct-2025, 11:41:02 am), and a note that it was started by GitHub push by Edigarito-Nekoh. It also shows the time spent (7.8 sec waiting, 2.1 sec build duration, 10 sec total from scheduled to completion), the git revision (bc52a4662c111be24338400d4970f7631687), and the repository URL (https://github.com/123groot-Nekoharu-hub/internal-1.git). A 'Changes' section at the bottom lists one update: '1. Update demo (@ihailo / gitsavant)'.

## Setting Up Jenkins Email Notification Setup (Using Gmail with AppPassword)

### Step-1: Creation of app password

### **Gmail: Enable App Password (for 2-Step Verification)**

#### **ii. Enable 2-Step Verification**

#### **iii. Generate App Password for Jenkins**

- Go to:
  - Security → App passwords
- Select:
  - **App:** Other (Custom name)
  - **Name:** Jenkins-Demo
- Click **Generate**
- Copy the **16-digit app password**
  - Save it in a secure location (e.g., Notepad)

## **2. Jenkins Plugin Installation**

#### **i. Open Jenkins Dashboard**

#### **ii. Navigate to:**

- Manage Jenkins → Manage Plugins

#### **iii. Install Plugin:**

- Search for and install:
  - Email Extension Plugin

The screenshot shows the Jenkins 'Manage Jenkins' interface under the 'Plugins' section. A search bar at the top contains the text 'email'. Below it, a table lists several plugins:

Name	Health	Enabled
Email Extension (1.75.1-1.75.2, 2021)	OK	Enabled
Email Extension Template Plugin (1.1.1.1-1.1.1.5, 2021)	OK	Enabled
Mailer Plugin (1.22.1-1.22.1, 2021)	OK	Enabled
Pipeline	Failed to load: Pipeline (workflow-aggregator: 1000, wf23Workflow, 1) - Failed to load: Pipeline: Basic Steps (workflow-basic-steps: 1000, wf888_wf0f8cfa)	Disabled

### 3. Configure Jenkins Global Email Settings

Go to:

- Manage Jenkins → Configure System

---

#### A. E-mail Notification Section

Field	Value
SMTP Server	smtp.gmail.com
Use SMTP Auth	<input checked="" type="checkbox"/> Enabled
User Name	Your Gmail ID (e.g., archanareddykmit@gmail.com)
Password	Paste the 16-digit App Password
Use SSL	<input checked="" type="checkbox"/> Enabled
SMTP Port	465
Reply-To Address	Your Gmail ID (same as above)

#### ► Test Configuration

- Click: Test configuration by sending test e-mail
- Provide a valid email address to receive a test mail
- Should receive email from Jenkins

Jenkins / Manage Jenkins / System

E-mail Notification

SMTP server

smtp.gmail.com

Default user e-mail suffix ?

Advanced ▾    Edited

Use SMTP Authentication ?

User Name  
edigiralaneksha@gmail.com

Password  
 Concealed    [Change Password](#)

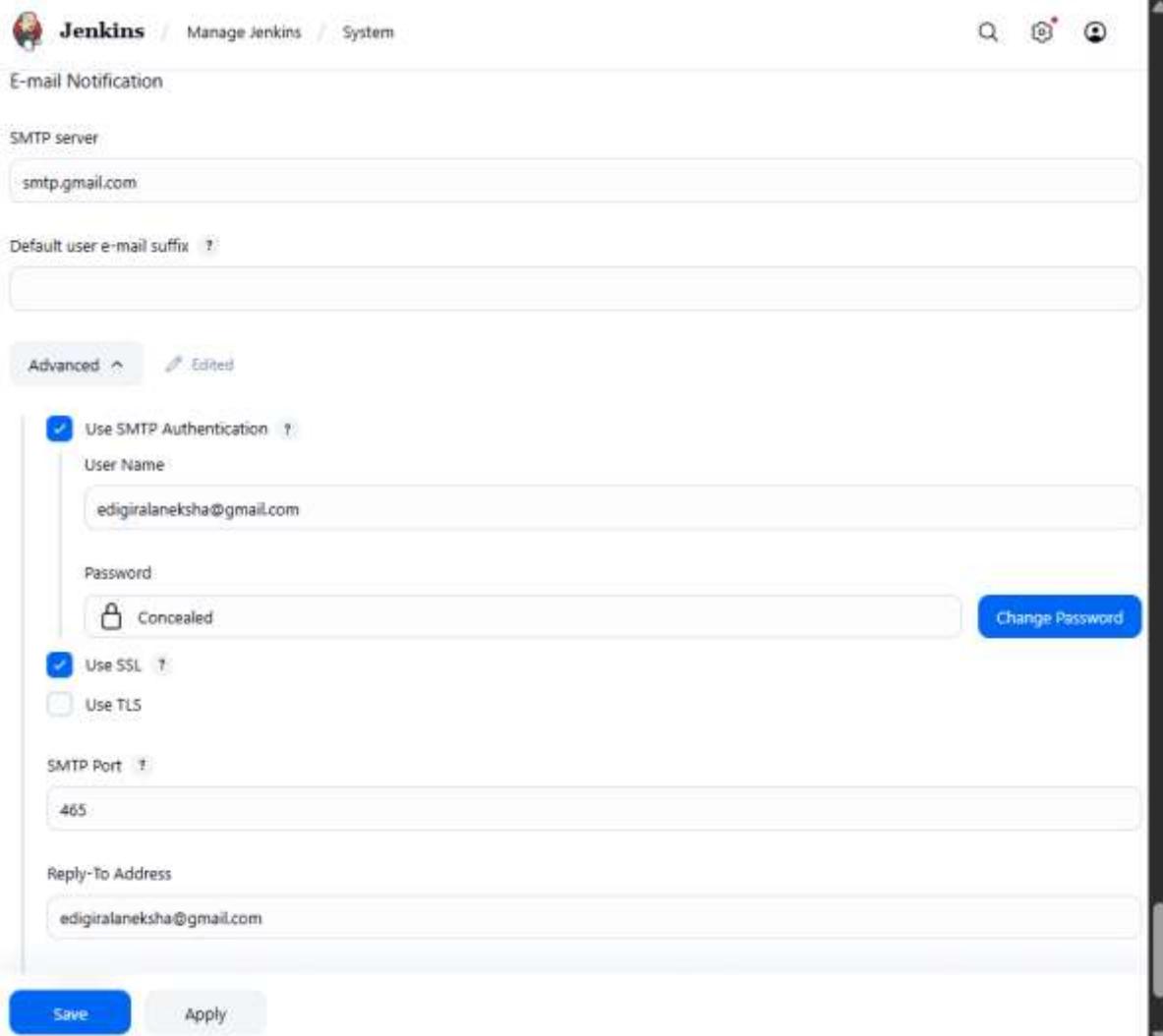
Use SSL ?

Use TLS

SMTP Port ?  
465

Reply-To Address  
edigiralaneksha@gmail.com

[Save](#)    [Apply](#)



## B. Extended E-mail Notification Section

Field	Value
SMTP Server	smtp.gmail.com
SMTP Port	465
Use SSL	<input checked="" type="checkbox"/> Enabled
Credentials	Add Gmail ID and App Password as Jenkins credentials
Default Content Type	text/html or leave default

Field	Value
<b>Default Recipients</b>	Leave empty or provide default emails
<b>Triggers</b>	Select as per needs (e.g., Failure)

Extended E-mail Notification

SMTP server: smtp.gmail.com

SMTP Port: 463

Advanced:  Test

Credentials: edigitalankita@gmail.com/\*\*\*\*\* (test)

Use SSL  
 Use TLS  
 Use OAuth 2.0

Advanced email Properties

**Save** **Apply**

- Default Triggers:
- Default triggers:
- Aborted
  - Always
  - Before Build
  - Failure - 1st
  - Failure - 2nd
  - Failure - Any
  - Failure - Still
  - Failure - X
  - Failure -> Unstable (Test Failures)
  - Fixed
  - Not Built
  - Script - After Build
  - Script - Before Build
  - Status Changed
  - Success
  - Test Improvement
  - Test Regression
  - Unstable (Test Failures)
  - Unstable (Test Failures) - 1st
  - Unstable (Test Failures) - Still
  - Unstable (Test Failures)/Failure -> Success

Content Token Reference:

#### 4. Configure Email Notifications for a Jenkins Job

##### i. Go to:

- Jenkins → Select a Job → Configure

The screenshot shows the Jenkins configuration interface for a job named 'job\_webhook\_java'. The top navigation bar includes the Jenkins logo, the job name, and a 'Configuration' link. On the right side, there are search, settings, and refresh icons. The main area is titled 'Configure' with a 'General' tab selected. The 'General' tab contains a 'Description' field with the value 'java webhook'. Below this are several checkboxes for build triggers and options like 'Discard old builds', 'GitHub project', and 'Notify when Job configuration changes'. A 'Source Code Management' section is present, showing 'Git' is selected as the provider with a 'Repositories' dropdown. At the bottom are 'Save' and 'Apply' buttons.

##### ii. In the Post-build Actions section:

- Click: Add post-build action → **Editable Email Notification**

##### A. Fill in the fields:

Field	Value
<b>Project Recipient List</b>	Add recipient email addresses (comma-separated)
<b>Content Type</b>	Default (text/plain) or text/html
<b>Triggers</b>	Select events (e.g., Failure, Success, etc.)
<b>Attachments</b>	(Optional) Add logs, reports, etc.

### iii. Click Save

#### Post-build Actions

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

≡ **Editable Email Notification** ? ✖

Allows the user to disable the publisher, while maintaining the settings

Disable Extended Email Publisher ?

**Project From**

Project Recipient List ?

Comma-separated list of email address that should receive notifications for this project.

edigiralaneksha@gmail.com,nekshasri99@gmail.com

Project Reply-To List ?

Comma-separated list of email address that should be in the Reply-To header for this project.

\$DEFAULT\_REPLYTO

**Save** **Apply**

Jenkins / job\_webhook\_java

Rename Credentials

Builds

Filter (f)

Today

#4 11:45 am (green circle)

20 October 2025

#3 11:41 am (green circle)

#2 11:36 am (green circle)

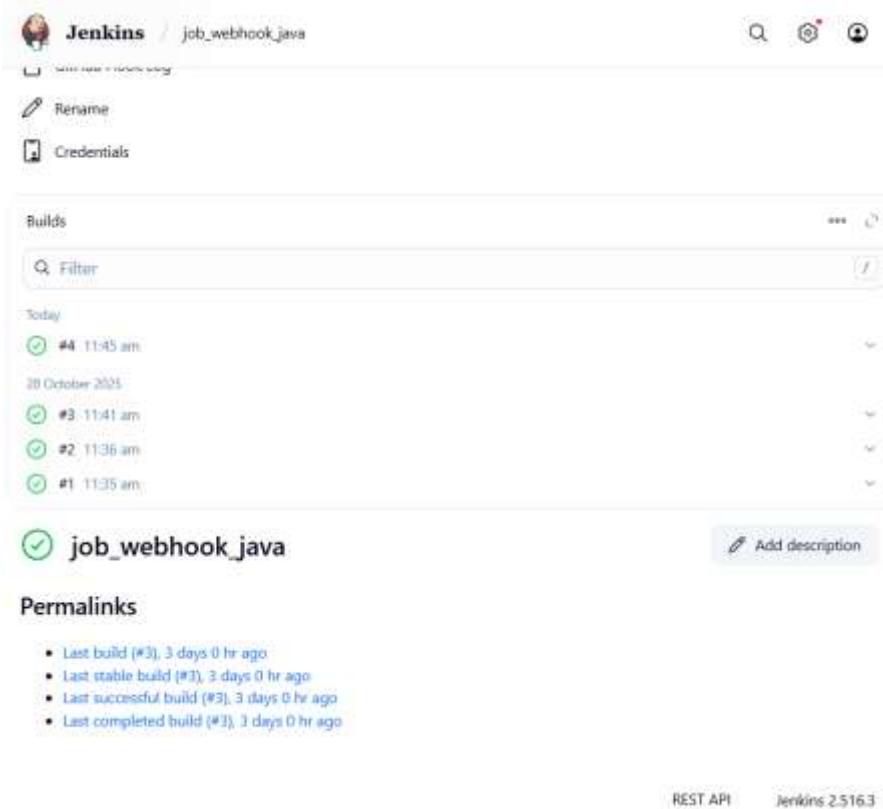
#1 11:35 am (green circle)

job\_webhook\_java Add description

Permalinks

- Last build (#3), 3 days 0 hr ago
- Last stable build (#3), 3 days 0 hr ago
- Last successful build (#3), 3 days 0 hr ago
- Last completed build (#3), 3 days 0 hr ago

REST API Jenkins 2.516.3



Gmail Search mail

Compose

Inbox 84

Starred

Snoozed

Sent

Drafts

Purchases

More

Labels +

job\_webhook\_java - Build # 4 - Successful! Inbox

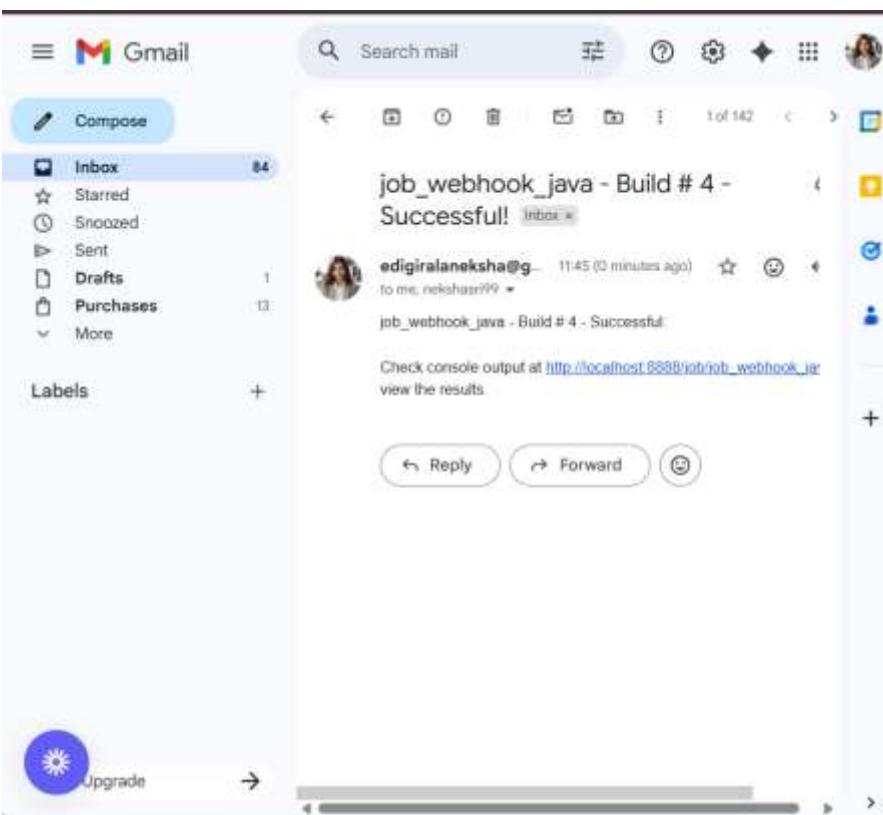
edigiralaneksha@g... 11:45 (0 minutes ago) to me; releshazarri@...

job\_webhook\_java - Build # 4 - Successful!

Check console output at [http://localhost:8080/job/job\\_webhook\\_java](http://localhost:8080/job/job_webhook_java) view the results

Reply Forward

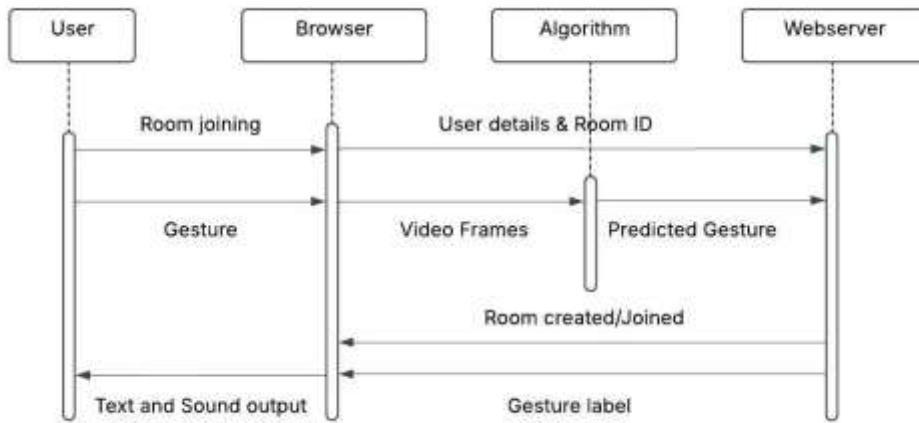
Upgrade



## TUNEORA – A Music Web App

### 1. Sequence Diagram:

A sequence diagram shows how objects interact in a particular scenario of a use case.  
It focuses on the time order of messages exchanged between different components in a system.



### 2. Class Diagram:

A class diagram represents the static structure of a system by showing classes, their attributes, methods, and relationships.

It is mainly used for object-oriented design and modeling data structures.

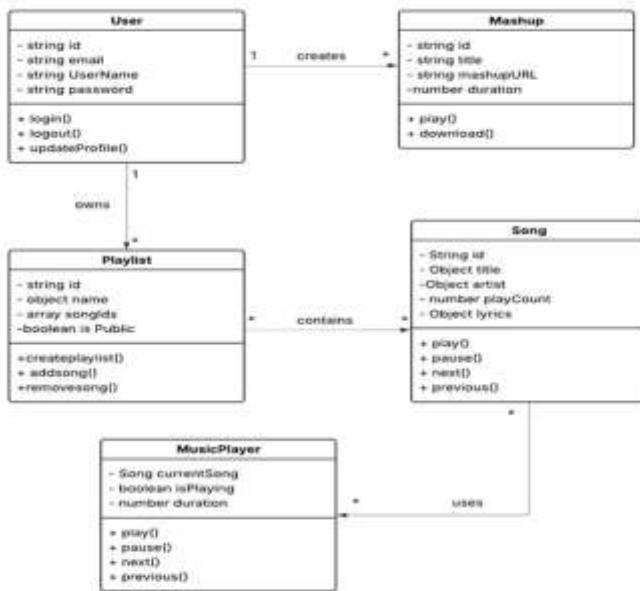
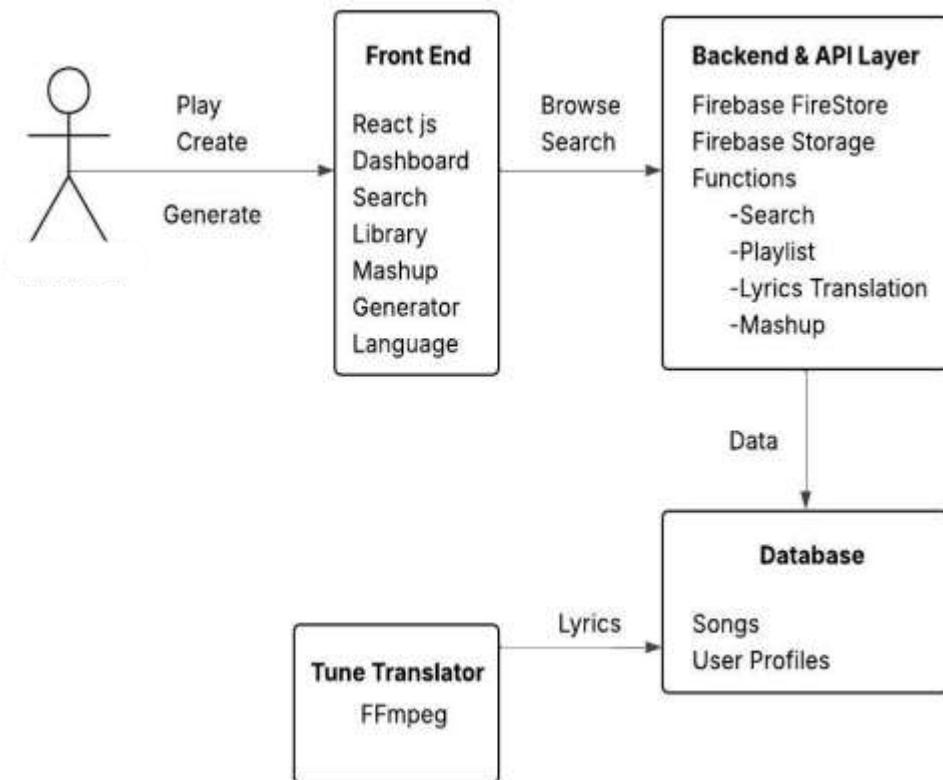


Fig 5: Class Diagram for TuneOra

### 3. Component Diagram:

A component diagram illustrates how different software components are connected and interact to form a complete system.

It helps visualize the organization and dependencies among modules or subsystems.



## **12.Creation of virtual machine for Ubuntu OS and Deploying the web application**

DEPLOYMENT OF INDEX.HTML USING EC2 INSTANCE in AWS

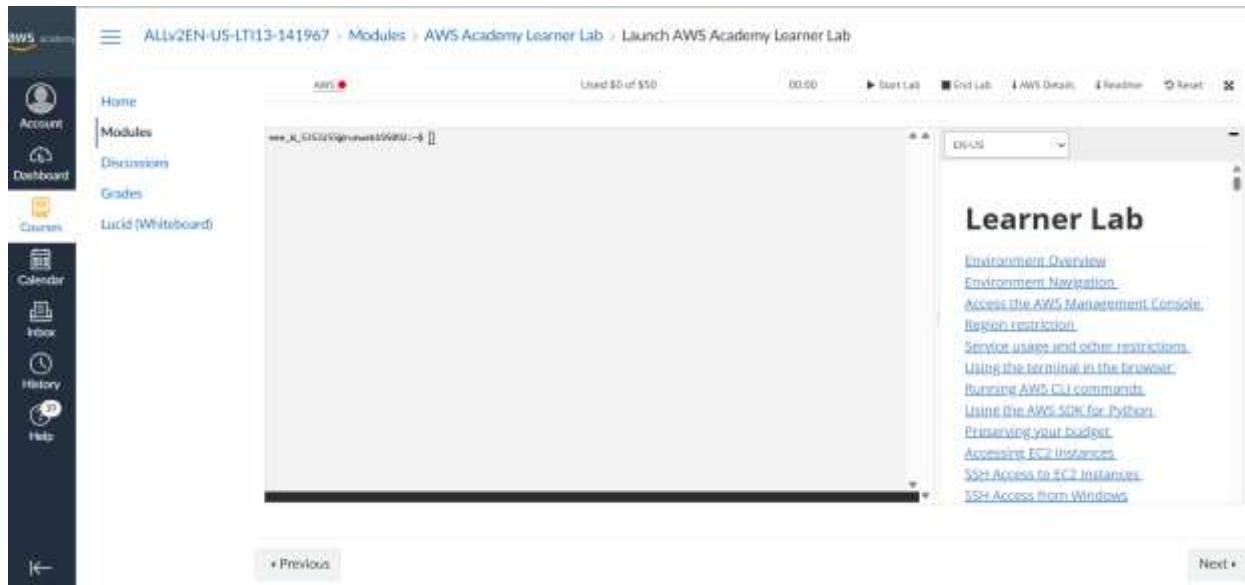
Step 1: Click on Modules.

The screenshot shows the AWS Academy Learner Lab interface. On the left, there's a sidebar with icons for Account, Dashboard, Courses, Calendar, Work, History, and Help. The main navigation bar at the top has 'Allv2EN-US...' and 'Modules' selected. Below the navigation, there's a 'Home' link and a 'Modules' link. The 'Modules' link is underlined, indicating it's the active section. The main content area features a large blue 'V' icon with a red arrow pointing upwards. To the right of the icon, the text 'Learner Lab' is displayed in bold. Below this, there's a list of topics: 'Environment Overview', 'Environment Navigation', 'Access the AWS Management Console', 'Before restrictions', 'Service usage and other restrictions', 'Using the terminal in the browser', 'Running AWS CLI commands', 'Using the AWS SDK for Python', 'Provision your instance', 'Launching EC2 instances', 'SSH access to EC2 instances', and 'Edit Access from Windows'. At the bottom of the page, there are 'Previous' and 'Next' buttons.

Step 2: Scroll down and select Lunch AWS Academy Lab

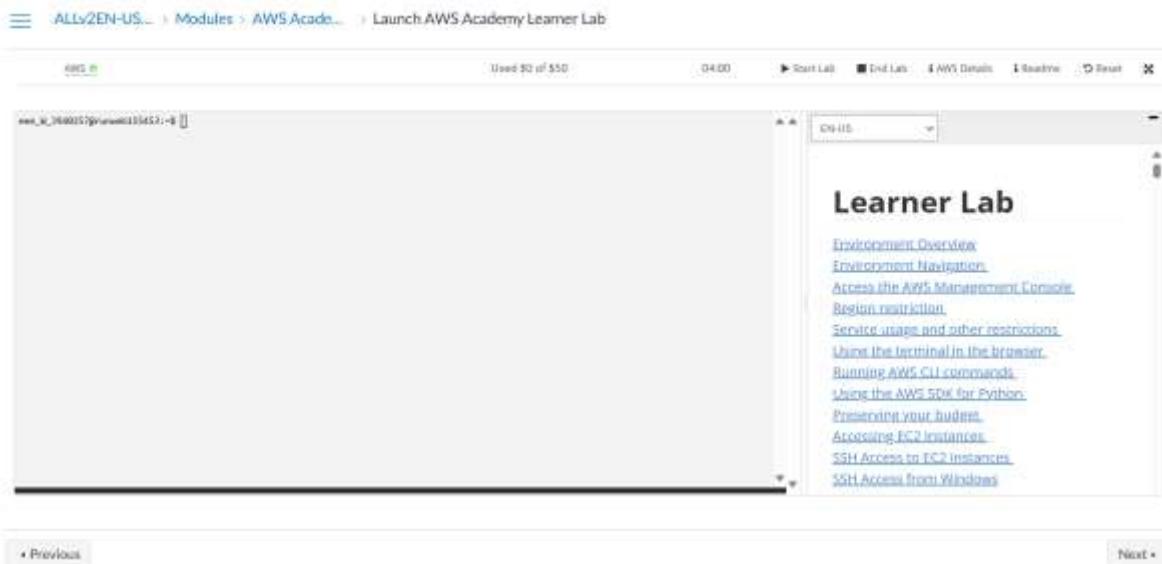
The screenshot shows the same AWS Academy Learner Lab interface as the previous one, but with a different focus. The 'Launch AWS Academy Learner Lab' link under the 'AWS Academy Learner Lab' section is now highlighted with a light blue background. The rest of the interface remains the same, including the sidebar, navigation bar, and the list of topics on the right.

### Step 3: click on start lab



The screenshot shows the AWS Academy Learner Lab interface. On the left, there's a sidebar with navigation links: Home, Modules (which is selected), Discussions, Grades, and Lucid (Whiteboard). Below that are Calendar, Inbox, History, and Help. The main area has tabs for AWS (selected), AWS Lambda, and AWS Lambda. At the top right, there are buttons for Start Lab, End Lab, AWS Details, Reader, and Reset. A progress bar at the top indicates 'Used \$0 of \$50'. The central part of the screen displays a terminal window titled 'aws\_k\_0\_GitHubEnvironment: ~\$'. To the right is a 'Learner Lab' sidebar with a list of links: Environment Overview, Environment Navigation, Access the AWS Management Console, Region restriction, Service usage and other restrictions, Using the terminal in the browser, Running AWS CLI commands, Using the AWS SDK for Python, Preserving your budget, Accessing EC2 instances, SSH Access to EC2 instances, and SSH Access from Windows. At the bottom are 'Previous' and 'Next' buttons.

### Step 4: click on AWS and in the services select EC2



This screenshot is similar to the previous one but shows a different state of the AWS services menu. The 'AWS' tab is still selected, but the 'Services' dropdown menu is now open, revealing options like Lambda, CloudWatch, S3, and EC2. The 'EC2' option is highlighted with a blue selection bar. The rest of the interface, including the learner lab sidebar and navigation, remains the same as in the first screenshot.

## Step 5: select instances and select instance click on launch instance

The screenshot shows the AWS EC2 Dashboard. On the left sidebar, under the 'Instances' section, there is a 'Launch instance' button. Below it, a note says 'Note: Your instances will launch in the US East (N. Virginia) Region'. To the right, there's a 'Service health' section showing 'This service is operating normally' and a 'Zones' section.

## Step 6: Give the name of the machine "week-122"

The screenshot shows the 'Launch an instance' wizard. Step 1: Set instance details. It has sections for 'Name and tags', 'Application and OS Images (Amazon Machine Image)', and 'Storage (volumes)'. The 'Name' field contains 'week-122'. The 'Software image (AMI)' dropdown is set to 'Canonical, Ubuntu, 24/04, ami8...' and the 'Virtual server type (instance type)' dropdown is set to 't3.micro'. The 'Launch instance' button is at the bottom right.

## Step 6: Select the ubuntu server

The screenshot shows the AWS Lambda console with the following details:

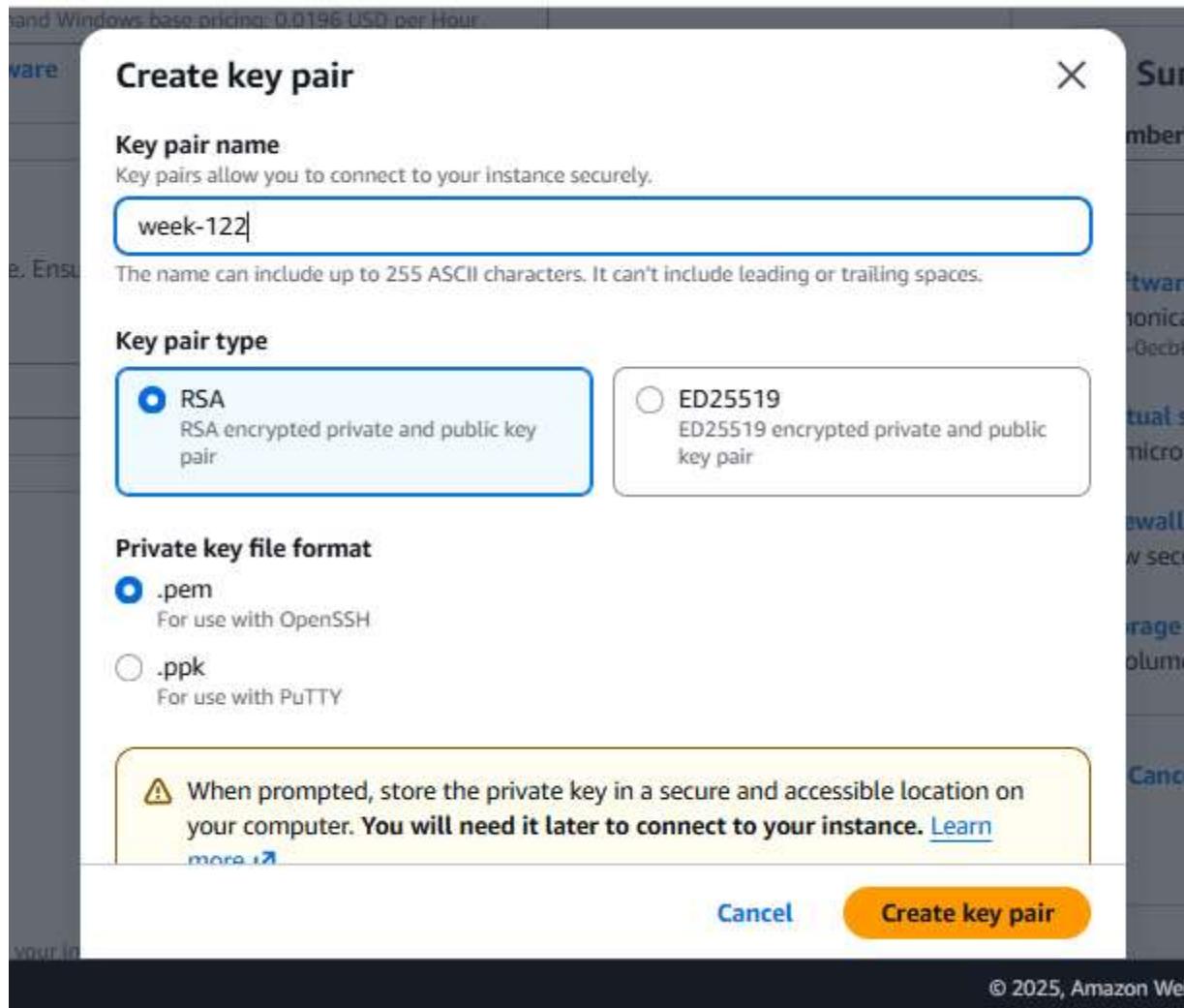
- Function name:** Launch AWS Academy Lambda
- Runtime:** Python 3.9
- Description:** Lambda function created from a template.
- Code entry type:** Lambda@Edge
- Code source:** GitHub (https://github.com/awslabs/lambda-sample-code)
- Environment:** US East (N. Virginia) - us-east-1
- Role:** Lambda execution role - lambda-execution-role
- Logs:** CloudWatch Logs - Lambda logs
- Metrics:** CloudWatch Metrics - Lambda metrics
- Deployment:** Deployment package - zip file
- Test:** Test function - Test function
- Version:** Version 1
- Tags:** None
- Configuration:** Configuration

## Step 7: select architecture as 64-bit and instance type as t3.micro(i.e., they are free)

The screenshot shows the AWS Lambda console with the following details:

- Function name:** Launch AWS Academy Lambda
- Runtime:** Python 3.9
- Description:** Lambda function created from a template.
- Code entry type:** Lambda@Edge
- Code source:** GitHub (https://github.com/awslabs/lambda-sample-code)
- Environment:** US East (N. Virginia) - us-east-1
- Role:** Lambda execution role - lambda-execution-role
- Logs:** CloudWatch Logs - Lambda logs
- Metrics:** CloudWatch Metrics - Lambda metrics
- Deployment:** Deployment package - zip file
- Test:** Test function - Test function
- Version:** Version 1
- Tags:** None
- Configuration:** Configuration

Step 8: Create a new keypair and select type as RSA and .pem option and click on create key pair



Step 9: In network settings select “create security group” and give the security group name

▼ Network settings [Info](#)

VPC - required | [Info](#)

vpc-05a9ef3852073b114 (default) [C](#)

Subnet | [Info](#)

No preference [C](#) Create new subnet [L](#)

Availability Zone | [Info](#)

No preference [C](#) Enable additional zones [L](#)

Auto-assign public IP | [Info](#)

Enable [▼](#)

Firewall (security groups) | [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group  Select existing security group

Security group name - required

week-122

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_-/-/()#,@[]+=&();\$\*

Description - required | [Info](#)

Launch-wizard-1 created 2025-11-11T05:36:49.724Z

Step 10: Click on edit button on the top right corner and select

Type: ssh

Source: anywhere

EC2 > Instances > Launch an instance

week-122

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_-/-/()#,@[]+=&();\$\*

Description - required | [Info](#)

Launch-wizard-1 created 2025-11-11T05:36:49.724Z

Inbound Security Group Rules:

Security group rule 1 (TCP: 22/0.0.0.0/0)

Type	Protocol	Port range	Remove
ssh	TCP	22	<a href="#">Remove</a>

Type | [Info](#) Protocol | [Info](#) Port range | [Info](#)

Source type | [Info](#) Source | [Info](#) Description - optional | [Info](#)

Anywhere [Add rule](#) [Select existing security group](#) e.g. SSH for admin access

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Add security group rule [X](#)

▼ Summary

Number of instances [Info](#)

1

Software image (AMI)

Canonical Ubuntu 24.04 LTS (HVM, SSD)

Virtual server type (instance type)

t2.micro

Firewall security group

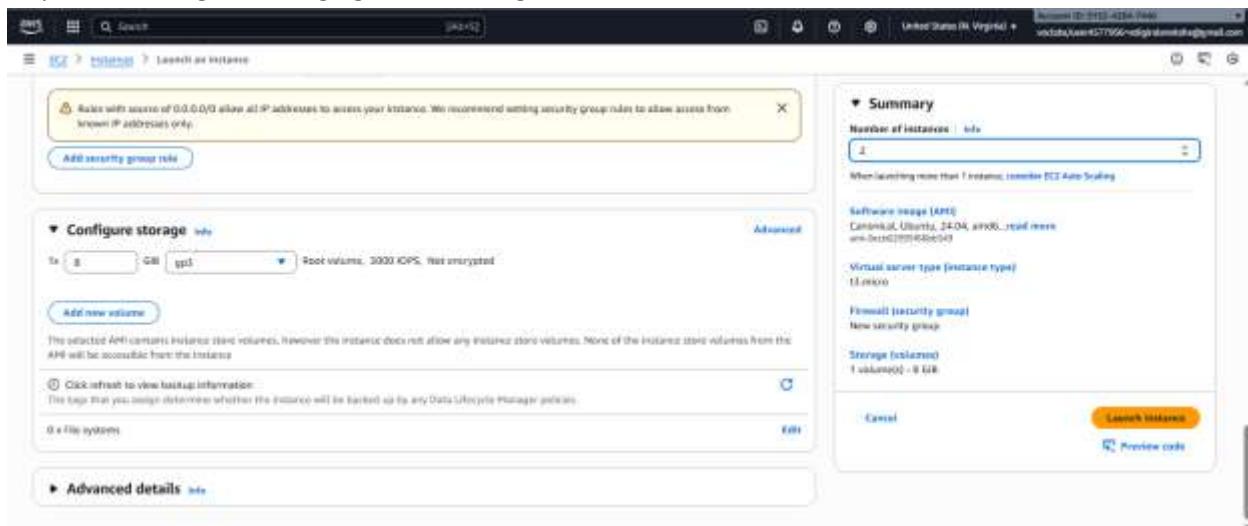
New security group

Storage (volume(s))

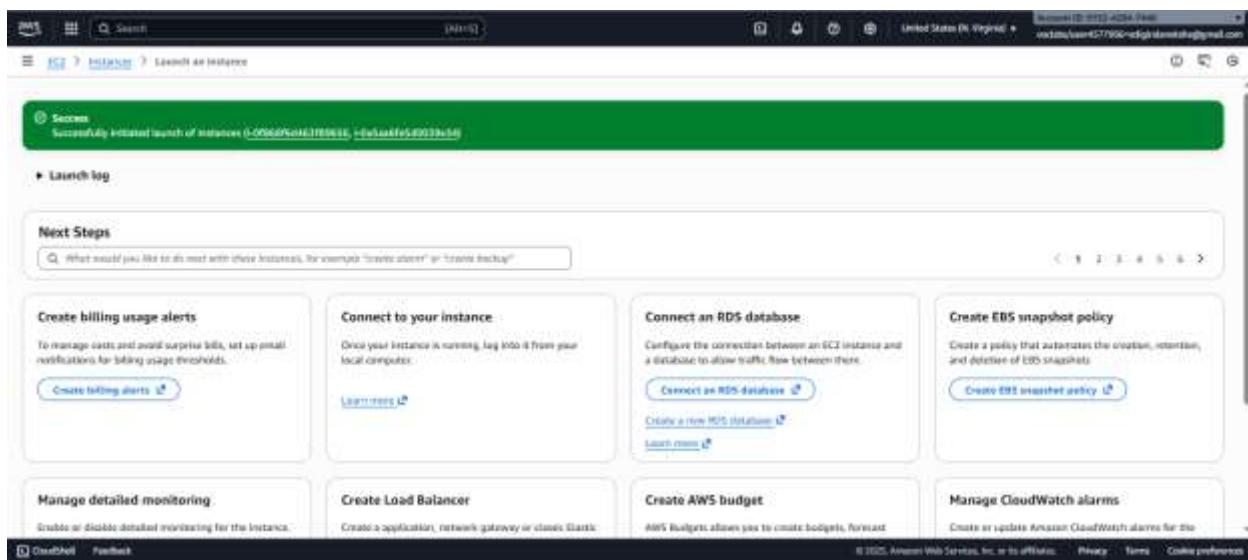
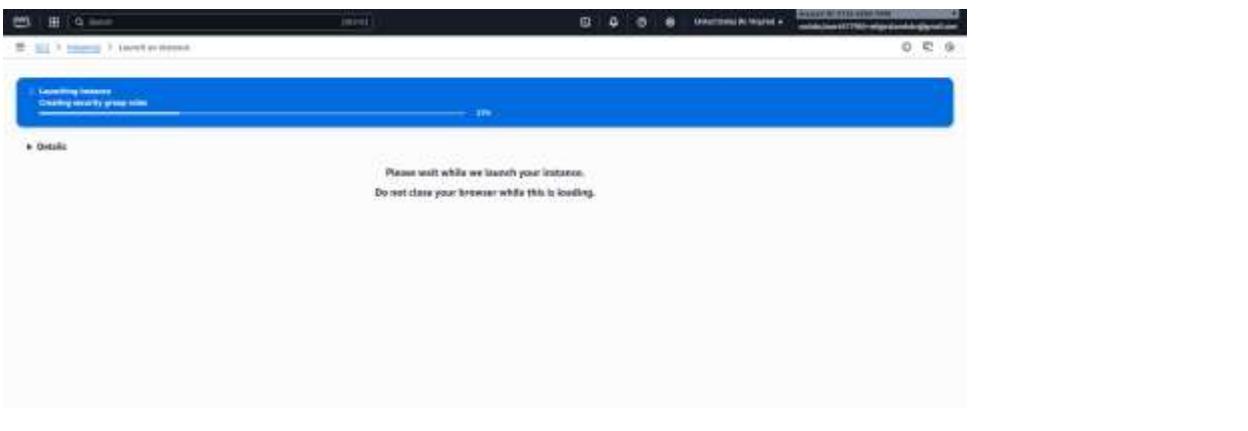
1 volume(s) - 6 GiB

[Launch instance](#) [Preview code](#)

Step 11: in configure storage give 8GB and give number of instances as 2 and click on launch instance



Step 12: The launching of instance will start and successful message will be shown



Step 13: In the instances the created ones will be shown, you can also rename the instance , changed week-1222 to “webapp”

The screenshot shows the AWS EC2 Instances page. On the left sidebar, under the 'Instances' section, there are several options: Instances, Instances Type, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, Images, AMIs, AMI Catalog, and Elastic Block Store. The 'Instances' section is expanded, showing four entries:

Name	Instance ID	Instance State	Instance Type	Status Check	Alarm Status	Availability Zone	Public IPv4 DNS	Public
Datastax003	i-0f868f6d463f89656	Terminated	t2.micro	-	View alarms	us-east-1a	-	-
week-12	i-0f868f6d463f89656	Terminated	t2.micro	-	View alarms	us-east-1a	-	-
week-122	i-0f868f6d463f89656	Running	t2.micro	Reloading	View alarms	us-east-1c	ec2-13-222-21-231.us... 13.222	13.222
<b>webapp</b>	i-0f868f6d463f89656	Running	t2.micro	Reloading	View alarms	us-east-1c	ec2-13-222-21-231.us... 13.222	13.222

Below the table, a detailed view for the 'webapp' instance is shown. The 'Details' tab is selected, displaying the following information:

- Instance ID: i-0f868f6d463f89656
- Public IPv4 address: 172.31.21.231 | open address
- Private IPv4 address: 172.31.0.45
- Public DNS: ec2-13-222-21-231.compute-1.amazonaws.com | open address
- Instance state: Running

Step 14: click on connect and select “SSH Client” and copy the ssh command

The screenshot shows the 'Connect' page for the 'webapp' instance. The 'Connect' tab is selected, with the sub-tab 'SSH client' chosen. The page displays the following information:

Connect to an instance using the browser-based client.

EC2 Instance Connect | Session Manager | **SSH client** | EC2 serial console

Instance ID: i-0f868f6d463f89656 (webapp-122)

1. Open an SSH client.  
2. Locate your private key file. The key used to launch this instance is week-122.pem.  
3. Run this command, if necessary, to ensure your key is not publicly viewable:  
    chmod 400 'week-122.pem'  
4. Connect to your instance using its Public DNS:  
    ssh -i 'week-122.pem' ubuntu@ec2-13-222-21-231.compute-1.amazonaws.com  
**Command copied**

**Note:** In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Step 15: Navigate to the path where the file with .pem extension is present(week-122.pem) and paste the command

```
PS C:\Users\User\downloads> ssh -i "week-122.pem" ubuntu@ec2-13-222-21-231.compute-1.amazonaws.com
The authenticity of host 'ec2-13-222-21-231.compute-1.amazonaws.com (13.222.21.231)' can't be established.
ED25519 key fingerprint is SHA256:NEGegchQjt8om/1AVL5qfmafnMphv5Ad4A1Mwo8qECo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-222-21-231.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Tue Nov 11 05:50:06 UTC 2025

  System load: 0.08      Temperature:      -273.1 C
  Usage of /: 25.9% of 6.71GB  Processes:        118
  Memory usage: 24%          Users logged in:   0
  Swap usage:  0%           IPv4 address for ens5: 172.31.9.83

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-9-83:~$
```

Step 16: check the docker and git version

If they are not present, then go to administrative terminal using command

“sudo su”

Then update using the command “sudo apt-get update”

```

ubuntu@ip-172-31-9-83:~$ docker --version
Command 'docker' not found, but can be installed with:
sudo snap install docker          # version 28.4.0, or
sudo snap install docker          # version 28.1.1+1
sudo apt install docker.io        # version 28.2.2-0ubuntu1~24.04.1
sudo apt install podman-docker   # version 4.9.3+ds1-1ubuntu0.2
See 'snap info <snapname>' for additional versions.
ubuntu@ip-172-31-9-83:~$ git --version
git version 2.43.0
ubuntu@ip-172-31-9-83:~$ sudo su
root@ip-172-31-9-83:/home/ubuntu# sudo apt-get update

```

### Step 17: use command “sudo apt-get install docker.io” to install docker

```

Get:50 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 kB]
Get:51 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [27.4 kB]
Get:52 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [5708 kB]
Get:53 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 kB]
Get:54 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [384 kB]
Fetched 38.6 MB in 6s (6197 kB/s)
Reading package lists... Done
root@ip-172-31-9-83:/home/ubuntu# sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
ifupdown aufs-tools cgroupfs-mount | cgroup-lite debbootstrap docker-buildx docker-compose-v2 docker-doc rinse
ufs-fuse | zfsutils
The following NEW packages will be installed:
bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 0 newly installed, 0 to remove and 10 not upgraded.
Need to get 76.0 kB of archives.
After this operation, 288 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.3.3-0ubuntu1~24.04.2 [8815 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.28-0ubuntu1~24.04.1 [38.4 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 dns-root-data all 2024071801-ubuntu0.24.04.1 [5988 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 dnsmasq-base amd64 2.99-2ubuntu0.1 [576 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 28.2.3-0ubuntu1~24.04.1 [28.3 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 ubuntu-fan all 0.12.16+24.04.1 [34.2 kB]
Fetched 76.0 kB in 0s (81.3 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 71735 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7.1-1ubuntu2_amd64.deb ...
Unpacking bridge-utils (1.7.1-1ubuntu2) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.3.3-0ubuntu1~24.04.2_amd64.deb ...

```

Step 18: Clone the git repo that has maven project and change to that directory

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
root@ip-172-31-9-83:/home/ubuntu# git clone https://github.com/Gayathri2608-hub/maven-practice.git  
Cloning into 'maven-practice'...  
remote: Enumerating objects: 60, done.  
remote: Counting objects: 100% (60/60), done.  
remote: Compressing objects: 100% (40/40), done.  
remote: Total 60 (delta 11), reused 34 (delta 2), pack-reused 0 (from 0)  
Receiving objects: 100% (60/60), 13.39 KiB | 3.35 MiB/s, done.  
Resolving deltas: 100% (11/11), done.  
root@ip-172-31-9-83:/home/ubuntu# ls  
maven-practice  
root@ip-172-31-9-83:/home/ubuntu# cd maven-practice  
root@ip-172-31-9-83:/home/ubuntu/maven-practice# ls  
Dockerfile demo pom.xml readme src target  
root@ip-172-31-9-83:/home/ubuntu/maven-practice#
```

Step 19: change to the project directory and check for Dockerfile, if not present create the Dockerfile – “nano Dockerfile” and then build the image

“sudo docker build -t image\_name .” name of image:img1

```
root@ip-172-31-9-83:/home/ubuntu/maven-practice# ls  
Dockerfile demo pom.xml readme src target  
root@ip-172-31-9-83:/home/ubuntu/maven-practice# sudo docker build -t dep1 .  
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.  
Install the buildx component to build images with BuildKit:  
https://docs.docker.com/go/buildx/  
 Sending build context to Docker daemon 120.8kB  
Step 1/4 : FROM tomcat:9.0  
 9.0: Pulling from library/tomcat  
4b3ffd8ccb52: Pulling fs layer  
b48f960b380d: Pulling fs layer  
58424d8c3e86: Pulling fs layer  
4f4fb700ef54: Pulling fs layer  
37b617836889: Pulling fs layer  
891b6ad931b7: Pulling fs layer  
ac0beccecf50: Pulling fs layer  
4f4fb700ef54: Waiting  
37b617836889: Waiting  
891b6ad931b7: Waiting  
ac0beccecf50: Waiting  
b48f960b380d: Verifying Checksum  
b48f960b380d: Download complete  
4b3ffd8ccb52: Verifying Checksum  
4b3ffd8ccb52: Download complete  
4f4fb700ef54: Verifying Checksum  
4f4fb700ef54: Download complete  
37b617836889: Verifying Checksum  
37b617836889: Download complete  
891b6ad931b7: Verifying Checksum  
891b6ad931b7: Download complete  
ac0beccecf50: Verifying Checksum  
ac0beccecf50: Download complete  
58424d8c3e86: Verifying Checksum  
58424d8c3e86: Download complete  
4b3ffd8ccb52: Pulling fs layer
```

Step 20: Run the image “sudo docker run -d -p 8081:8080 img1”

```
root@ip-172-31-9-83:/home/ubuntu/ar/maven-practice# sudo docker run -d -p 8081:8080 img1
c5fd91cf9a9b4f0625d4d2c0d896406e8da76929ed75a3f9ccc1699fbbb08535
root@ip-172-31-9-83:/home/ubuntu/ar/maven-practice#
```

Step 21: Check the images and the containers

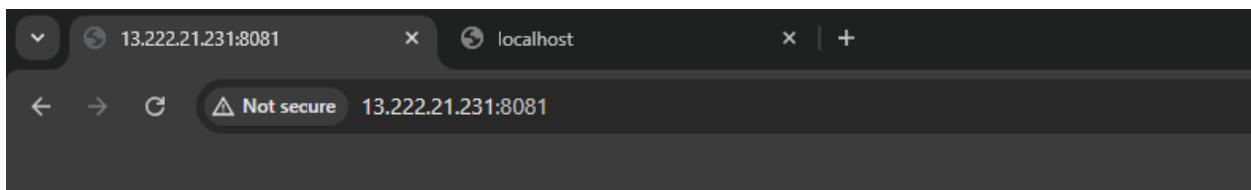
```
root@ip-172-31-9-83:/home/ubuntu/ar/maven-practice# sudo docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
img1            latest    a67a112ce8ac  2 minutes ago  413MB
dep1            latest    28efbe56e633  29 minutes ago  413MB
tomcat          9.0      2e4887a16e43  12 hours ago   413MB
root@ip-172-31-9-83:/home/ubuntu/ar/maven-practice# docker ps
CONTAINER ID   IMAGE      COMMAND      CREATED      STATUS      PORTS
              NAMES
c5fd91cf9a9b  img1      "catalina.sh run"  About a minute ago  Up About a minute  0.0.0.0:8081->8080/tcp, [::]:8081->8080/tcp  charming_banzai
84e7f9ce5ec2  dep1      "catalina.sh run"  9 minutes ago   Up 9 minutes  0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp  angry_shaw
b62aedc8bb3b  dep1      "catalina.sh run"  27 minutes ago  Up 27 minutes  0.0.0.0:7070->8080/tcp, [::]:7070->8080/tcp  sweet_archimedes
root@ip-172-31-9-83:/home/ubuntu/ar/maven-practice#
```

Step 22: Take the public IP address from the instances in AWS and open it in chrome along with the port number mapped.

Public IP- 13.222.21.231

Port used: 8081

Use: 13.222.21.231:8081, you will find your application that is deployed



**Hello World! practice**