



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)



NBA
National Board

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

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

	<p><u>Docker</u></p> <ul style="list-style-type: none"> a. Learn how to define and run multiple interdependent services (e.g., web server, database) in a single configuration file. b. Gain skills in writing and interpreting docker-compose.yml files for service setup. c. Deploy the same setup across different machines without manual configuration. d. Configure container networking and persistent storage within Compose. e. Reduce setup time and enable faster iteration during application development. 	
	<p><u>Creating a Multi-Module Maven Project</u></p> <ul style="list-style-type: none"> a. Build and package Java and Web applications using Maven. b. Add dependencies using pom.xml, compile and test using plugins. c. Resolve errors and conflicts arising from dependency mismatches. d. Work with parent and multi-module Maven projects. e. Generate executable JARs and deployable WARs using Maven 	
	<p><u>Jenkins Automation</u></p> <ul style="list-style-type: none"> a. Hands-on practice on manual creation of Jenkins pipeline using Maven projects from Github b. Create the job and build the pipeline for maven-java and maven-web project. 	
	<p><u>Pipeline Creation using script</u></p> <ul style="list-style-type: none"> a. Evaluation of Jenkins pipeline. b. WORKING ON BUILD TRIGGERS FOR LAST JENKINS PIPILINE c. Hands-on practice on creation of scripted Jenkins pipeline. 	
	<p><u>Working with minikube and Nagios</u></p> <ul style="list-style-type: none"> a. Hands-on practice of creating, running and scaling pods in minikube. b. Running Nginx server on specified port number by explaining the Nginx monitoring tool c. Running Nagios server and Understanding the Monitoring tool using Docker. d. AWS-free Trier account Creation steps 	

	<p><u>Jenkins-CI/CD</u></p> <ul style="list-style-type: none"> a. CI-Continuous Integration using Webhooks. b. Sending E-mail Notification on Build Failure or success 	
	<p><u>Creation of virtual machine for Ubuntu OS and Deploying the web application</u></p> <ol style="list-style-type: none"> 1. Creation of virtual machine 2. Deploying the web application 	

1. Software Installation & SRS Document:

DOCKER- INSTALLATION

Step-1: Go to docker website

The screenshot shows the Docker Documentation website with a dark theme. The top navigation bar includes links for 'Get started', 'Guides', 'Manuals', and 'Reference'. A search bar and an AI button are also present. On the left, there's a sidebar titled 'Get Docker' with links for 'What is Docker?', 'Introduction', 'Docker concepts', 'Docker workshop', and 'Educational resources'. The main content area displays information about Docker Desktop, including its terms of use and links for Mac, Windows, and Linux versions.

Step-2: Select the suitable one for your system

Install Docker Desktop on Windows

Docker Desktop terms

Commercial use of Docker Desktop in larger enterprises (more than 250 employees OR more than \$10 million USD in annual revenue) requires a paid subscription.

This page provides download links, system requirements, and step-by-step installation instructions for Docker Desktop on Windows.

Docker Desktop for Windows - x86_64

Docker Desktop for Windows - x86_64 on the Microsoft Store

Docker Desktop for Windows - Arm (Early Access)

For checksums, see [Release notes](#)

System requirements

Tip

Give feedback

: New Building AI Agents is Now Easy →

Get Docker

Learn how to install Docker for Mac, Windows, or Linux and explore our developer tools.

Get Docker

Get started

Learn Docker basics and the benefits of containerization.

Guides

Learn how Docker can optimize your development workflows.

Manuals

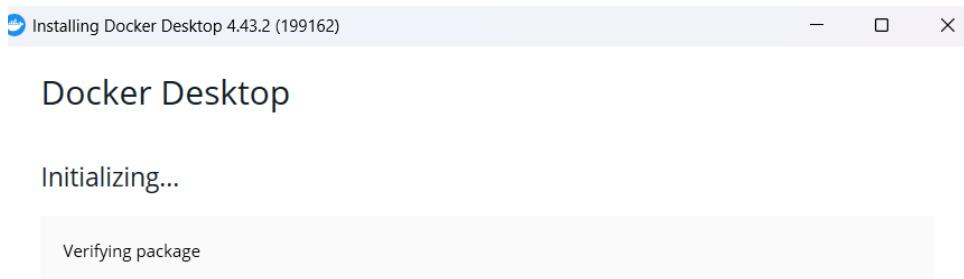
Learn how to install, set up, configure, and use Docker products.

Reference

Browse the CLI and API documentation.

Give feedback

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





Docker Desktop 4.43.2

Unpacking files...

```
Unpacking file: resources/docker-desktop.iso
Unpacking file: resources/ddvp.ico
Unpacking file: resources/config-options.json
Unpacking file: resources/componentsVersion.json
Unpacking file: resources/bin/docker-compose
Unpacking file: resources/bin/docker
Unpacking file: resources/.gitignore
Unpacking file: InstallerCli.pdb
Unpacking file: InstallerCli.exe.config
Unpacking file: frontend/vk_swiftshader_icd.json
Unpacking file: frontend/v8_context_snapshot.bin
Unpacking file: frontend/snapshot_blob.bin
Unpacking file: frontend/resources/regedit/vbs/wsRegReadListStream.wsf
Unpacking file: frontend/resources/regedit/vbs/wsRegReadList.wsf
```

Step-4: Installation successful



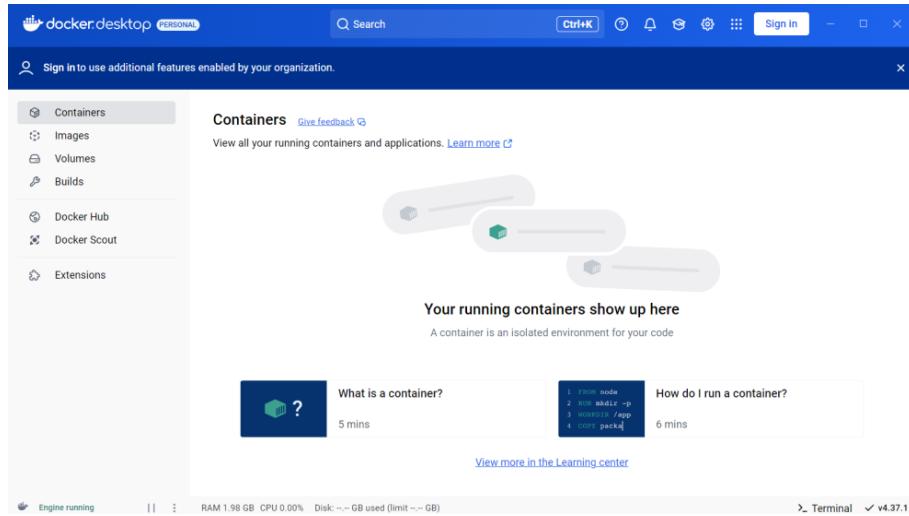
Docker Desktop 4.43.2

Installation succeeded

You must restart Windows to complete installation.

[Close and restart](#)

Step-5: Docker interface



Step-6: docker version

```
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

The screenshot shows the official Git website at git-scm.com. The header features the Git logo and the tagline '--distributed-even-if-your-workflow-isnt'. A search bar is at the top right. The main content area has a light gray background with a grid pattern. It starts with a paragraph about Git being a free and open source distributed version control system. To the right is a diagram showing a network of four computer stacks connected by red arrows, representing a distributed workflow. Below this are several sections: 'About', 'Documentation', 'Downloads', and 'Community'. The 'Downloads' section highlights the latest source release '2.46.2' with a 'Download for Windows' button. At the bottom, there's a note about 'Pro Git' and links for 'Windows GUIs' and 'Tarballs'.

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

The screenshot shows the official Git website at git-scm.com. The main navigation bar includes links for About, Documentation, Downloads, Community, and a search bar. The 'Downloads' section is highlighted. It features a large image of a computer monitor displaying the latest source release (2.46.2) with a 'Download for Windows' button. Below this, there are sections for 'GUI Clients' (listing macOS, Windows, and Linux/Unix options) and 'Logos' (with a 'View Logos' link). A sidebar on the left contains a link to the 'Pro Git book'.

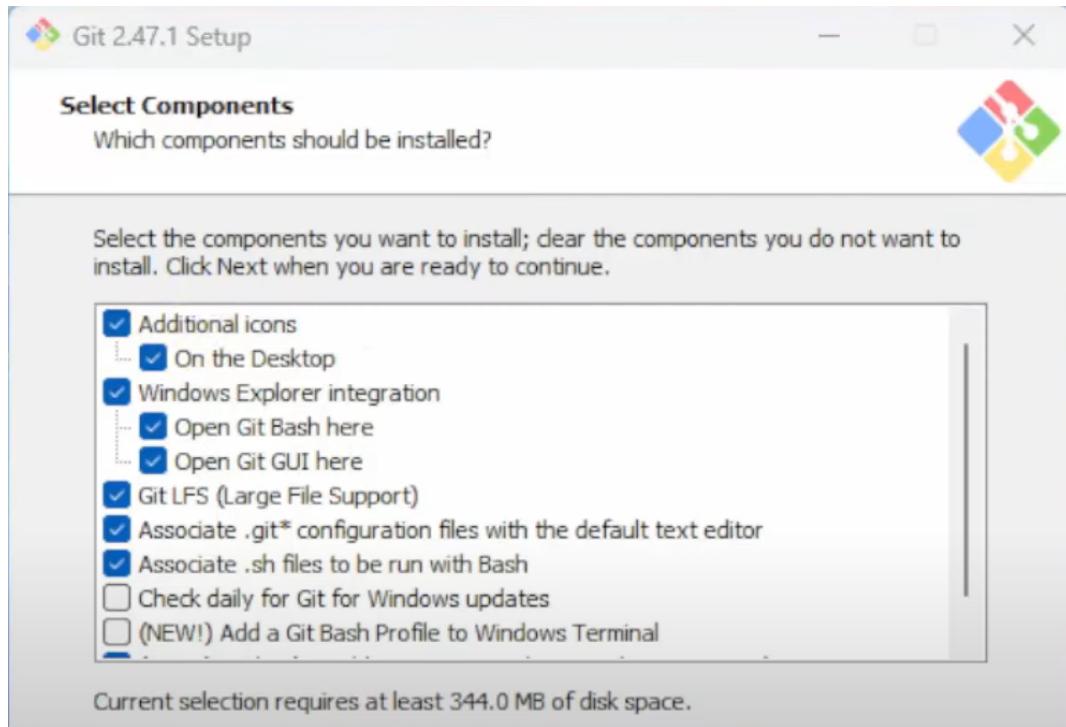
Step-3: Download for windows(suitable one for your system)

This screenshot shows the same Git website as above, but with a mouse cursor hovering over the 'Download for Windows' button. A 'Recent download history' sidebar is overlaid on the right side of the page, listing several recently downloaded files with their file types, sizes, and timestamps. The sidebar has a close button and a minimize/maximize button.

Step-4: License will be displayed click on next

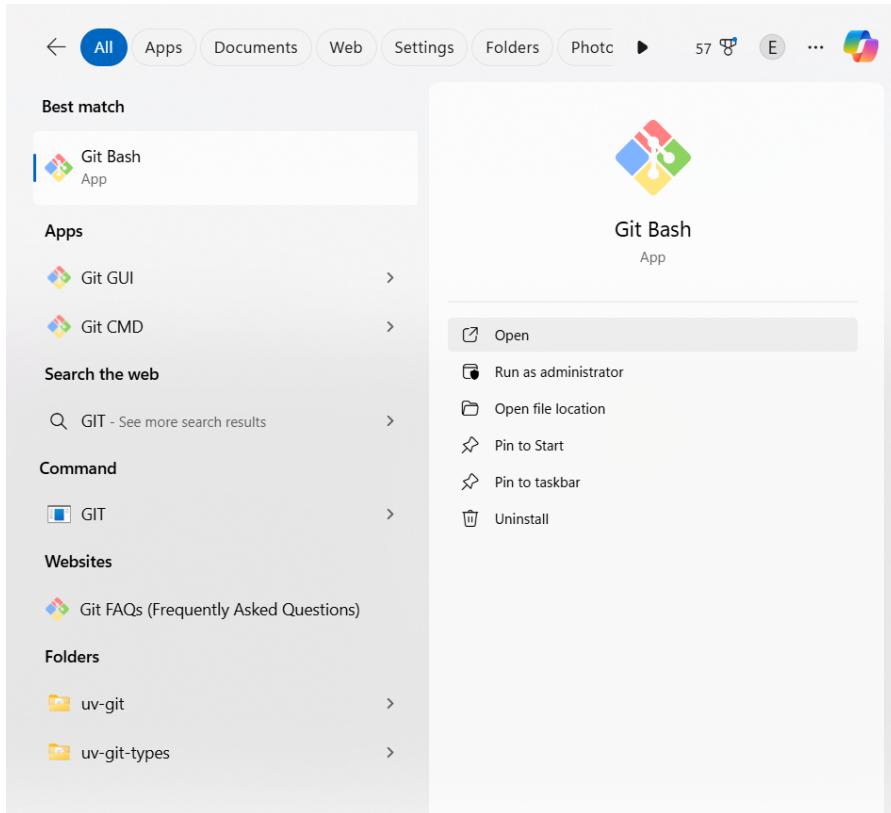


Step-5: Select the components and click next

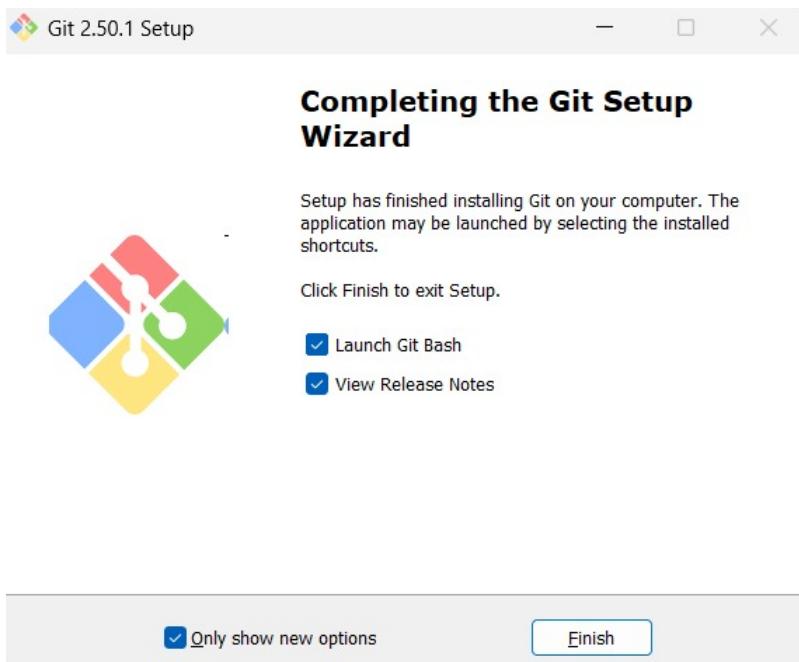


Git bash:

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



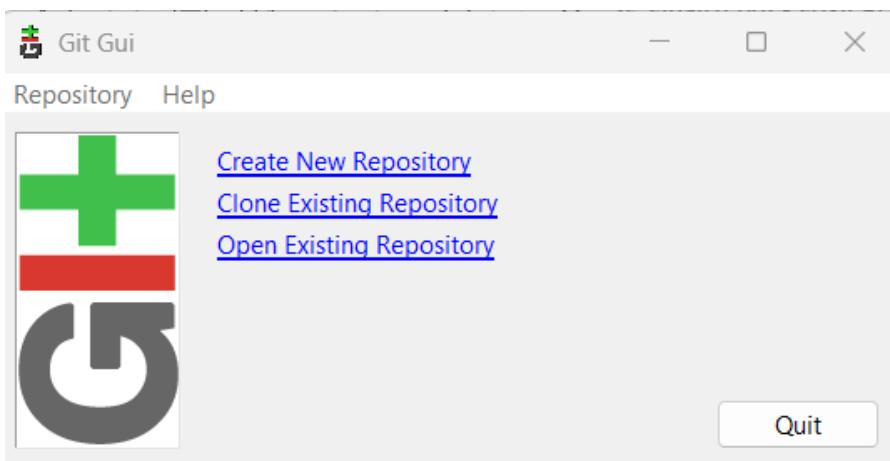
Step-2: Click on finish



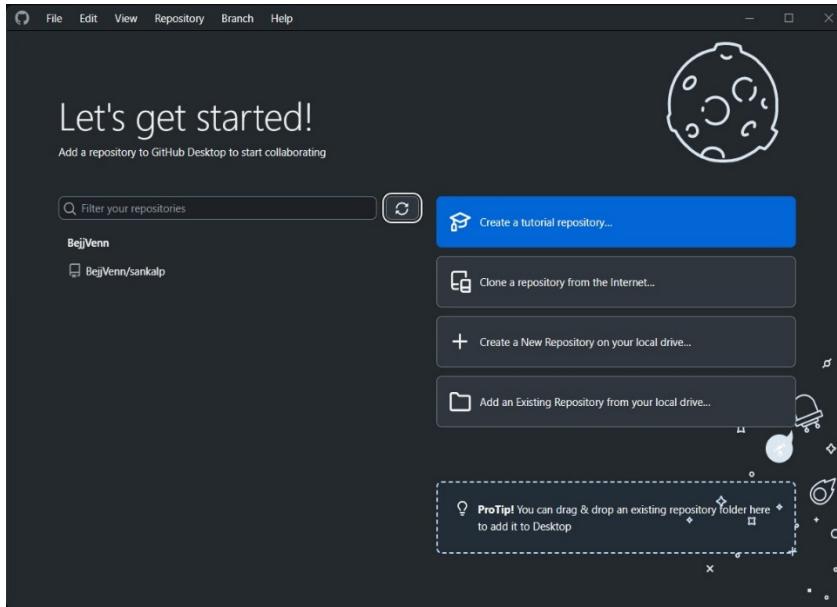
Step-3: git bash interface



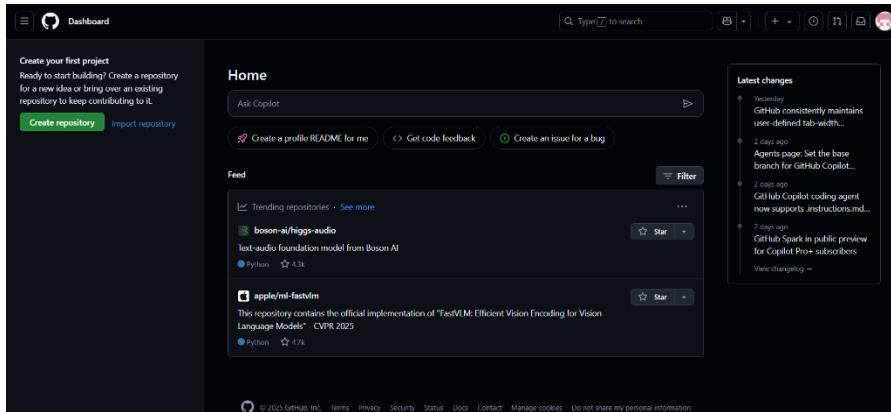
A screenshot of a terminal window titled "MINGW64:/c/Users/NekshaSrinivas". The title bar also shows "NekshaSrinivas@NekshaSrinivas MINGW64 ~". The main area of the terminal is black, indicating it is currently empty or has no output.



GIT-ACCOUNT



GIT-ACCOUNT



Tomcat

The screenshot shows the Apache Tomcat 9.0.98 homepage. At the top, there's a banner with the text "If you're seeing this, you've successfully installed Tomcat. Congratulations!" Below this is a cartoon cat icon. To the right are links for "Server Status", "Manager App", and "Host Manager". The main content area is divided into three columns: "Developer Quick Start" (with links to Tomcat Setup, First Web Application, Realms & AAA, JDBC DataSources, Examples, and Servlet Specifications/Tomcat Versions), "Documentation" (with links to Tomcat 9.0 Documentation, Configuration, and Wiki), and "Getting Help" (with links to FAQ and Mailing Lists, including tomcat-announce, tomcat-users, taglibs-user, and tomcat-dev). At the bottom, there are links for Other Downloads (Tomcat Connectors, Tomcat Native, Taglibs, Deployer), Other Documentation (Tomcat Connectors, mod_ajp Documentation, Tomcat Native, Deployer), Get Involved (Overview, Source Repositories, Mailing Lists, Wiki), Miscellaneous (Contact, Legal, Sponsorship, Thanks), and the Apache Software Foundation (Who We Are, Heritage, Apache Home, Resources). The footer contains the copyright notice "Copyright ©1999-2025 Apache Software Foundation. All Rights Reserved".

Java and maven versions

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

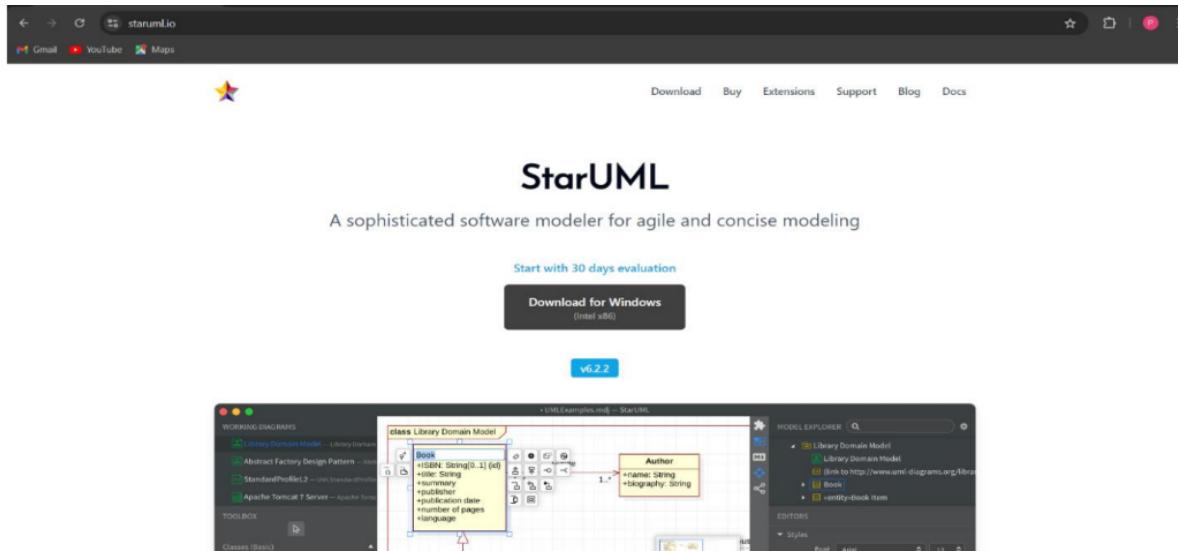
C:\Users\User>java --version
java 21.0.5 2024-10-15 LTS
Java(TM) SE Runtime Environment (build 21.0.5+9-LTS-239)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.5+9-LTS-239, mixed mode, sharing)

C:\Users\User>mvn --version
Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfc97d260186937)
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



Download Buy Extensions Support Blog Docs

Download

Start with 30 days evaluation

v6.2.2



macOS 10.13 or higher



Windows 10 or higher



Ubuntu or Fedora

macOS
(Intel x86)

Windows
(x86-64bit)

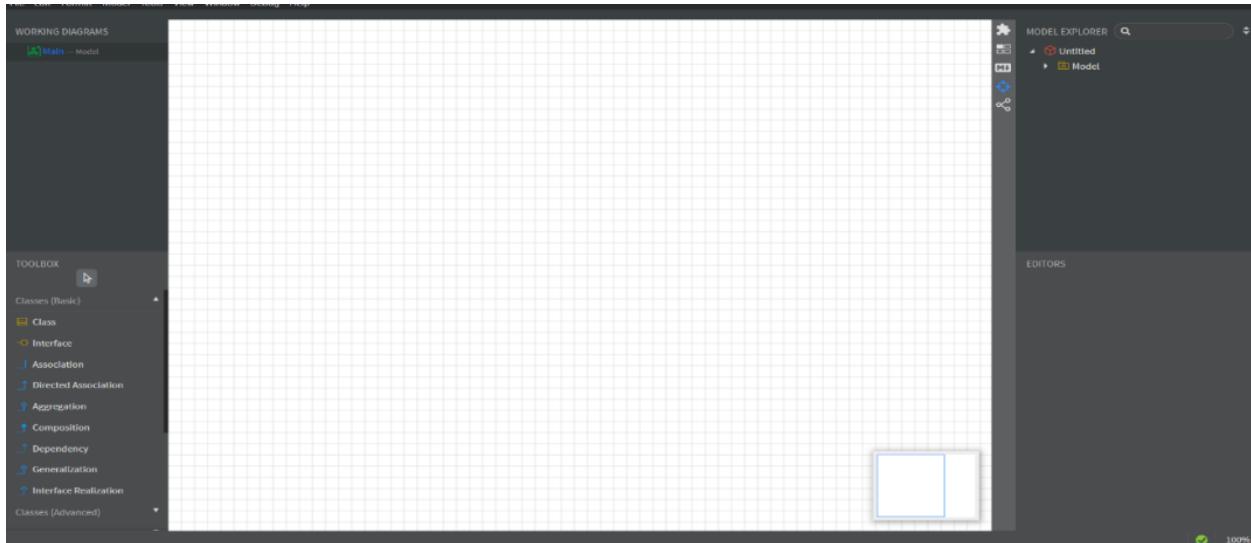
.deb
(x86-64bit)

macOS
(Apple arm64)

.rpm
(x86-64bit)

If you want to download for previous versions, you can get a link for previous versions by [finding your license key](#).

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

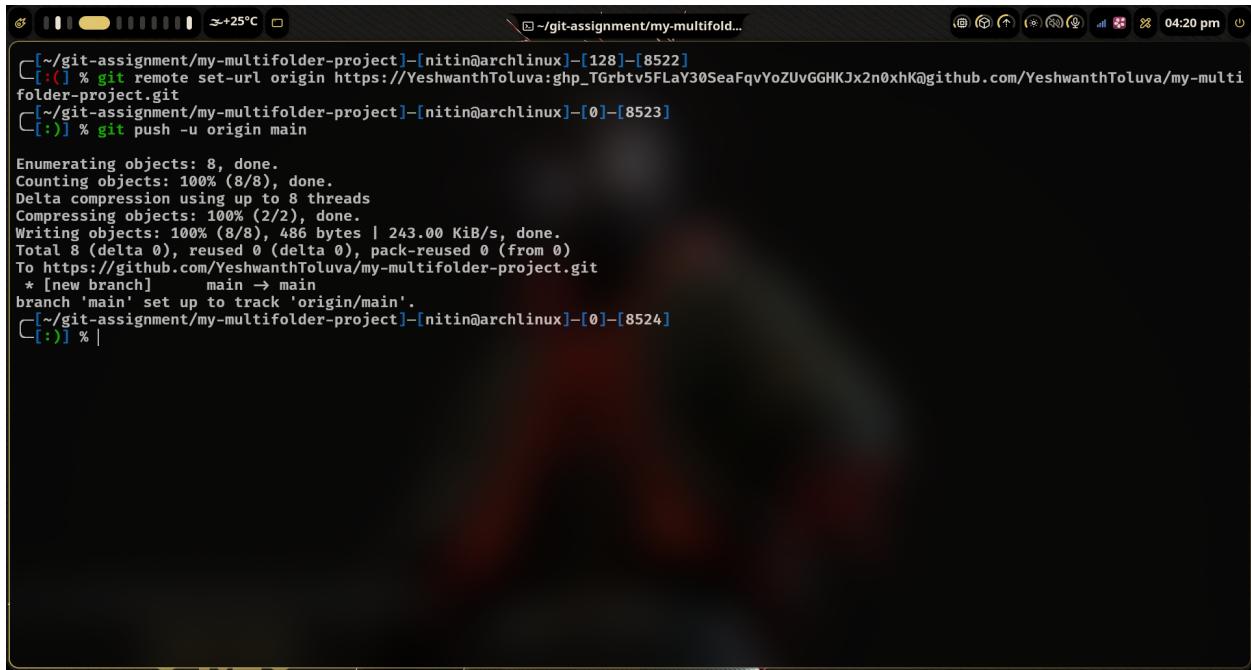
```
[~/git-assignment/my-multiproject]-[nitin@archlinux]-[128]-[8522]
[::] % git remote set-url origin https://YeshwanthToluva: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@github.com/YeshwanthToluva/my-multiproject.git
[~/git-assignment/my-multiproject]-[nitin@archlinux]-[0]-[8523]
[::] % git 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-multiproject.git
 * [new branch]    main → main
branch 'main' set up to track 'origin/main'.
[~/git-assignment/my-multiproject]-[nitin@archlinux]-[0]-[8524]
[::] % git restore filename.txt
          ^
error: pathspec 'filename.txt' did not match any file(s) known to git
[~/git-assignment/my-multiproject]-[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-multiproject]-[nitin@archlinux]-[128]-[8526]
[::] % git restore src/main.py
[~/git-assignment/my-multiproject]-[nitin@archlinux]-[0]-[8527]
[::] % git reset tests/test_main.py
[~/git-assignment/my-multiproject]-[nitin@archlinux]-[0]-[8528]
[::] % |
```

Scenario-Based Git Commands: Discarding and Unstaging Changes



A screenshot of a terminal window on Arch Linux. The title bar shows the path: ~/git-assignment/my-multifolder-project. The terminal output is as follows:

```
[~/git-assignment/my-multifolder-project] [nitin@archlinux] [128]-[8522]
[:::] % git remote set-url origin https://YeshwanthToluva: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@github.com/YeshwanthToluva/my-multifolder-project.git
[~/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?

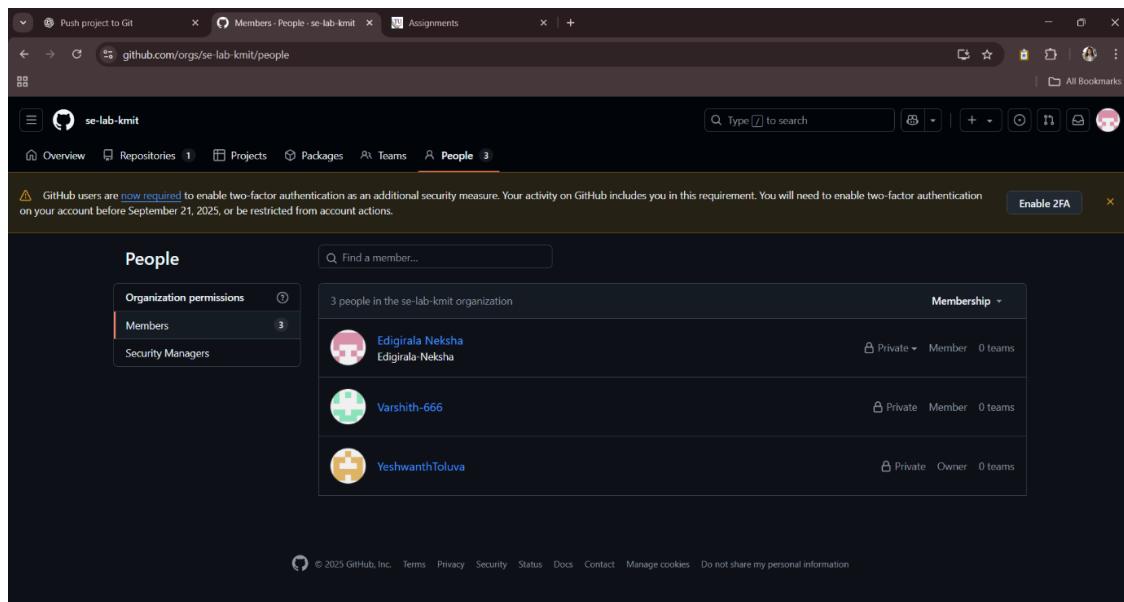
Use the following command:

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

This will remove the branch from the remote repository.

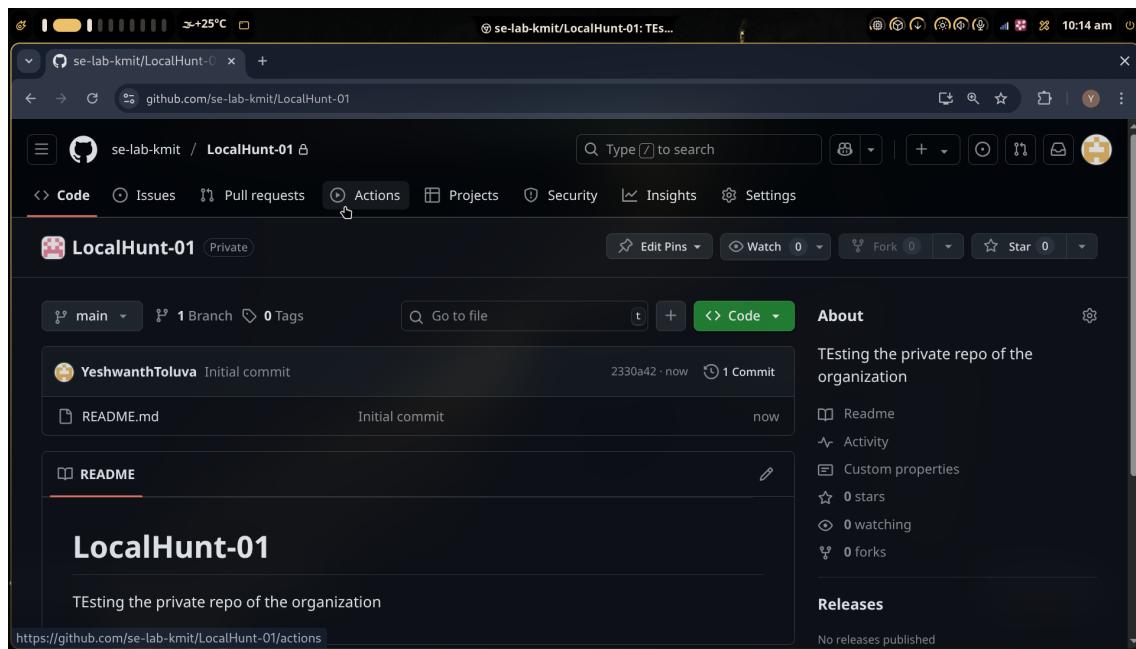
3. Collaborative coding using git

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



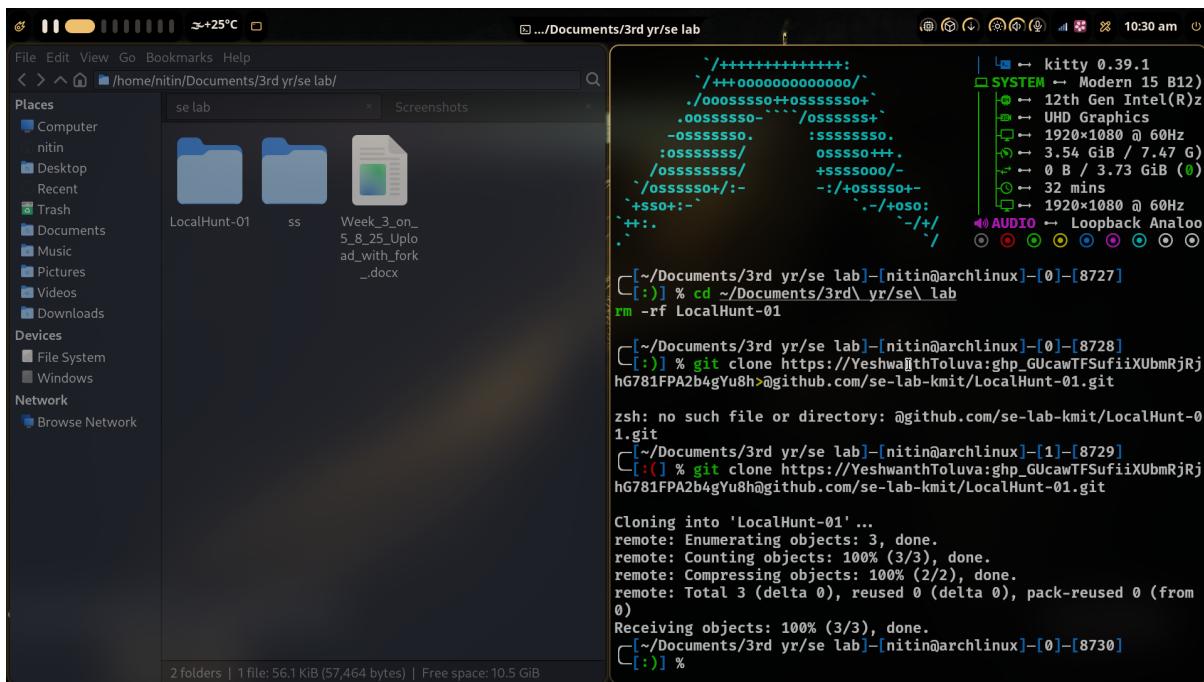
A screenshot of a web browser showing the GitHub organization "se-lab-kmit" page. The URL is github.com/orgs/se-lab-kmit/people. The page displays three members: Edigirala Neksha (Member, 0 teams), Varshith-666 (Member, 0 teams), and YeshwanthToluva (Owner, 0 teams). A sidebar on the left shows organization permissions and member counts. A banner at the top encourages users to enable two-factor authentication.

GitHub Repository Overview - LocalHunt-01 Private Repository

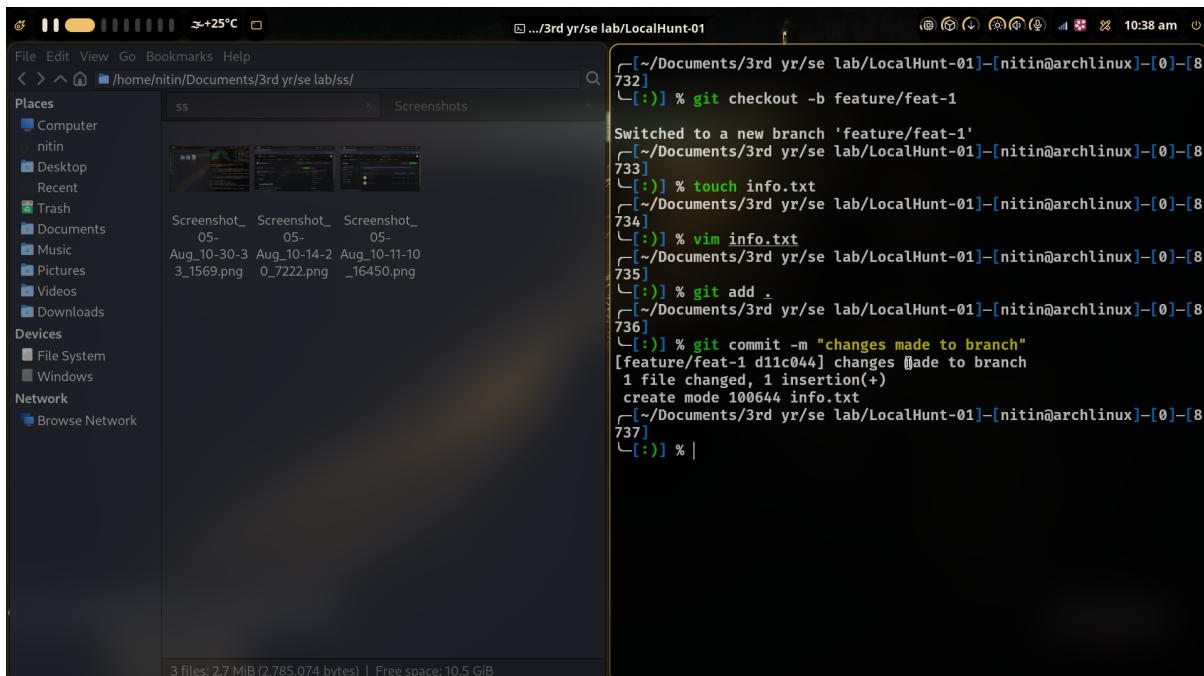


A screenshot of a web browser showing the GitHub repository "LocalHunt-01". The URL is github.com/se-lab-kmit/LocalHunt-01. The repository is private, created by YeshwanthToluva, and has one branch named "main". The README file contains the text "TEsting the private repo of the organization". The repository has 0 stars, 0 forks, and 0 releases published.

Terminal Git Clone Operations - LocalHunt-01 Repository Setup



Git Branch Operations - Feature Branch Creation and File Management



Git Push and Pull Request Creation - Feature Branch Workflow

The terminal window shows a file browser listing several screenshots from the LocalHunt-01 challenge. The command history at the bottom details the creation of an info.txt file, its addition to the git index, a commit message, and a push to the feature/feat-1 branch on GitHub.

```

733] [::] % touch info.txt
[~/Documents/3rd yr/se lab/LocalHunt-01-[nitin@archlinux]-[0]-[8
734] [::] % vim info.txt
[~/Documents/3rd yr/se lab/LocalHunt-01-[nitin@archlinux]-[0]-[8
735] [::] % git add .
[~/Documents/3rd yr/se lab/LocalHunt-01-[nitin@archlinux]-[0]-[8
736] [::] % git commit -m "changes made to branch"
[feature/feat-1 d11c044] changes made to branch
1 file changed, 1 insertion(+)
create mode 100644 info.txt
[~/Documents/3rd yr/se lab/LocalHunt-01-[nitin@archlinux]-[0]-[8
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:
To https://github.com/se-lab-kmit/LocalHunt-01.git
 * [new branch]      feature/feat-1 -> feature/feat-1
[~/Documents/3rd yr/se lab/LocalHunt-01-[nitin@archlinux]-[0]-[8
738] [::] %

```

"Screenshot_05-Aug_10-38-32_17782.png" | 1.3 MiB (1,318,318 bytes...)

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

The GitHub repository page for simple-repo-se shows basic information like forks, stars, and activity. It lists a single commit from image_c, and the README.RD file is shown with its content.

simple-repo-se (Public)
forked from [imagec/simple-repo](#)

About
a simple repo for assignment

Code Pull requests Actions Projects Wiki Security Insights Settings

Compare & pull request

Branches: master, 1 Branch, 0 Tags

This branch is up to date with [imagec/simple-repo:master](#).

Commit: [image_c Simple-repo init](#) · 75bcc55 · 9 years ago · 1 Commit

File	Content	Age
README.RD	Simple-repo init	9 years ago
repo	Simple-repo init	9 years ago
repo_utils.py	Simple-repo init	9 years ago

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

A screenshot of a terminal window on Arch Linux. The terminal shows a file browser window for '/home/nitin/Documents/3rd yr/se lab/'. The terminal history shows the following commands:

```

[~/.Documents/3rd yr/se lab/simple-repo-se]-[nitin@archlinux]-[0]-
[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]-[nitin@archlinux]-[0]-
[8764] [::] %

```

The file browser shows several folders and files in the directory.

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

A screenshot of a terminal window on Arch Linux. The terminal shows a file browser window for '/home/nitin/Documents/3rd yr/se lab/'. The terminal history shows the following commands:

```

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]-[nitin@archlinux]-[0]-
[8764] [::] %

```

Then, the user runs `git push origin feature/feat-1`, which outputs the following:

```

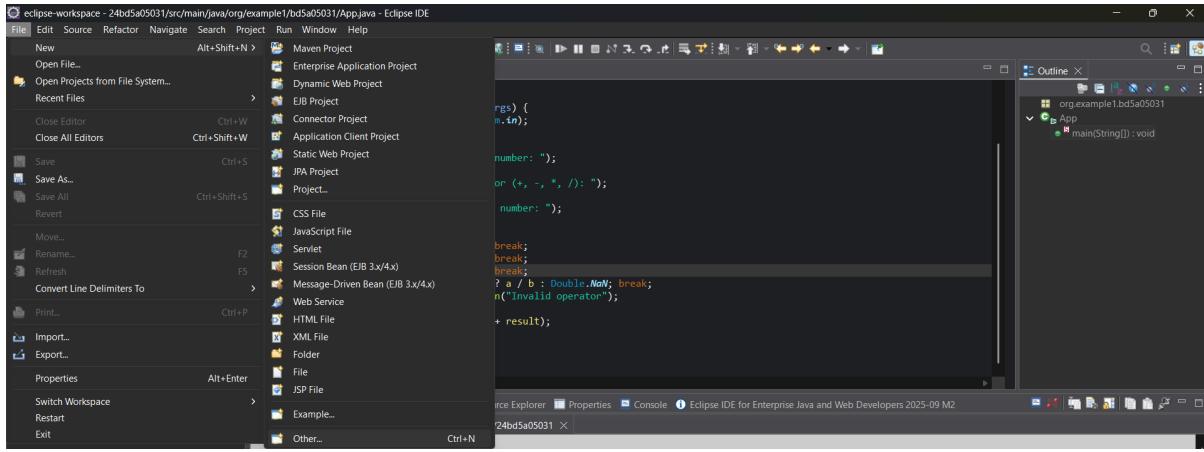
git push origin feature/feat-1

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 visiting:
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]-[nitin@archlinux]-[0]-
[8765] [::] %

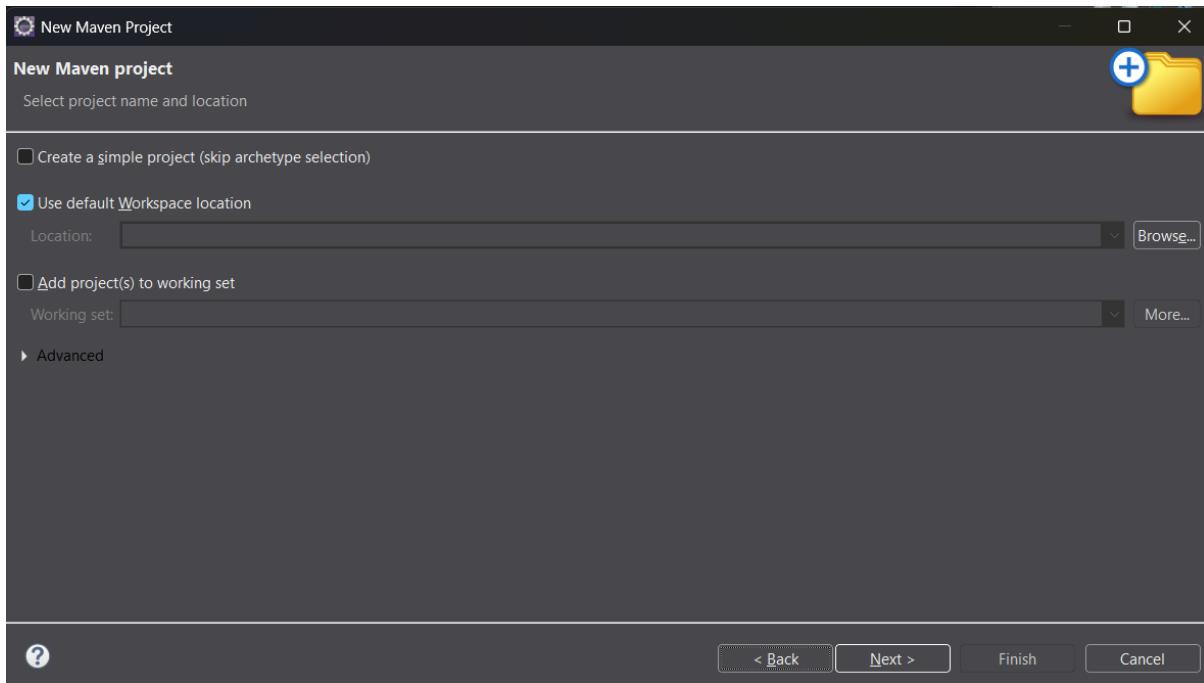
```

4. Build and package Java and Web applications using Maven

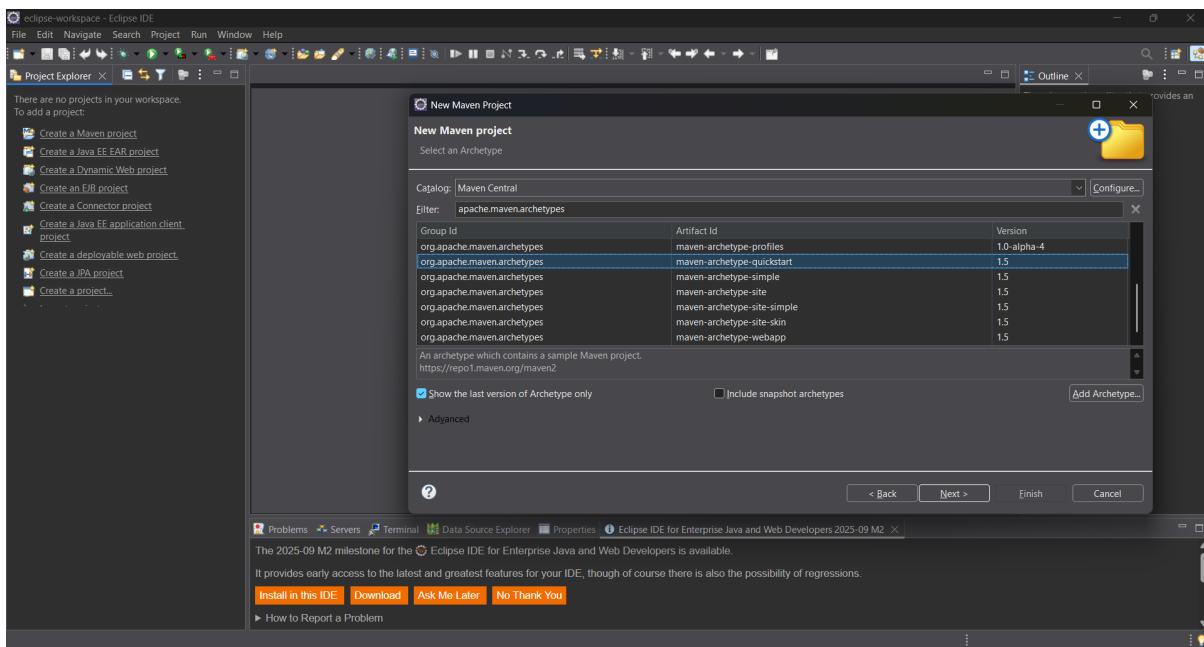
Step-1: Open the eclipse and click on file>new>Maven project



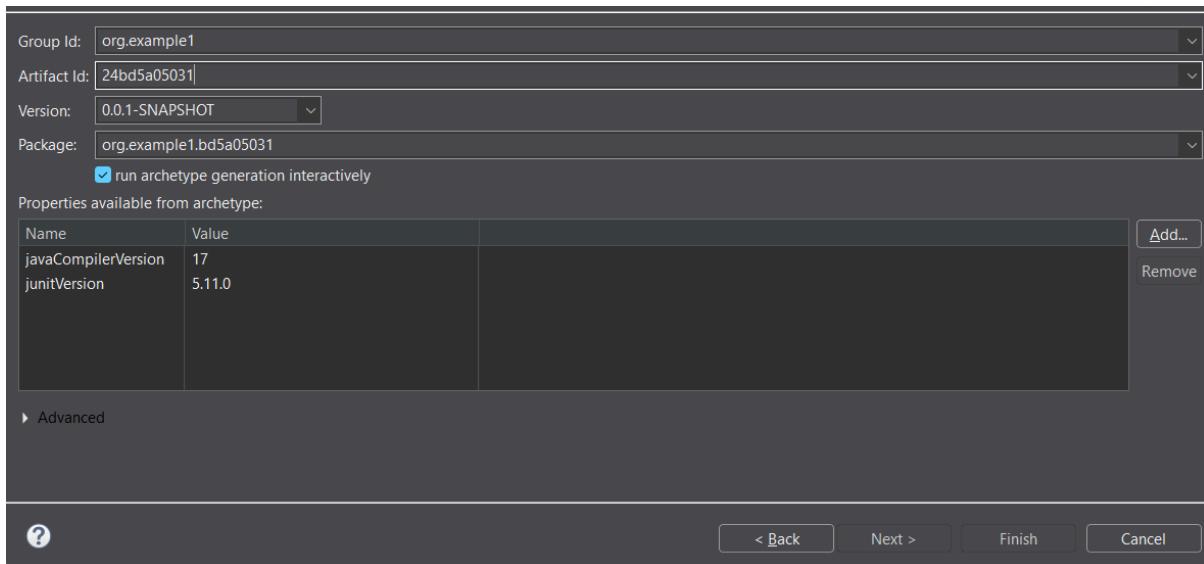
Step-2: select the default workspace and click on next



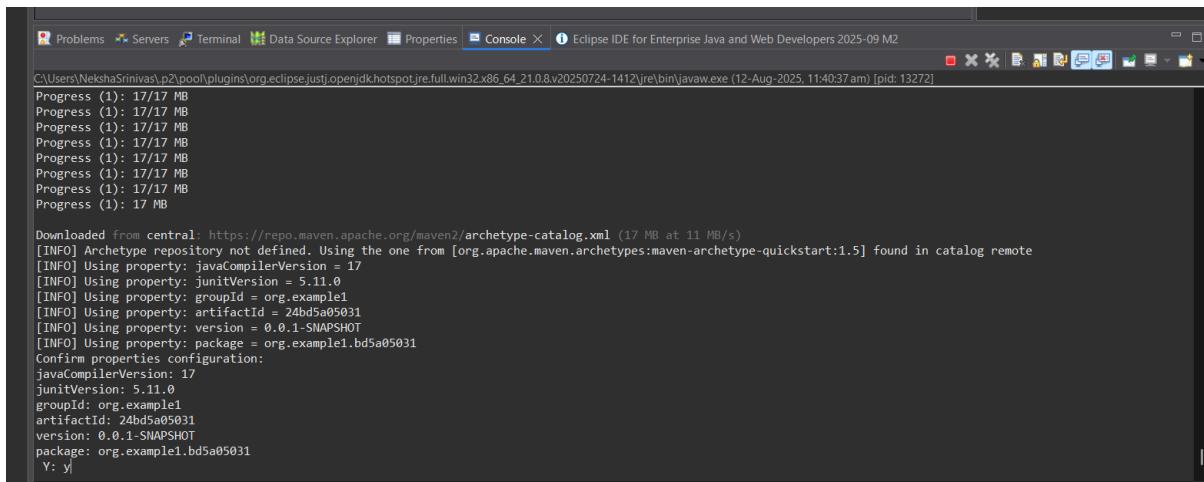
Step-3: in the filter option select the one maven-archetype-quickstart



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

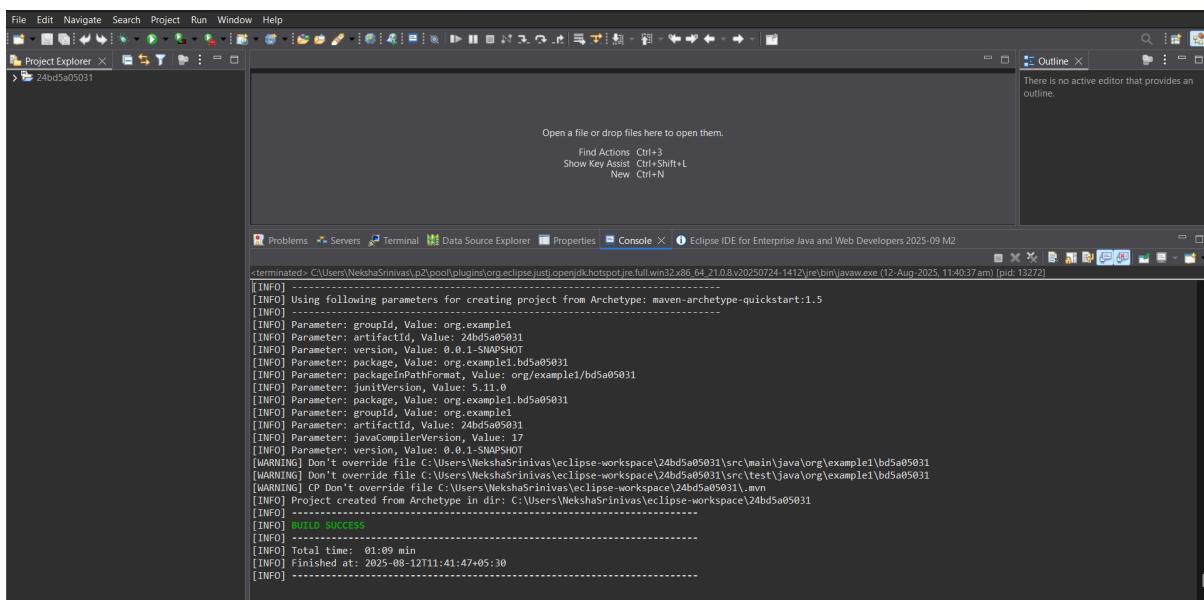


The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the following text:

```
C:\Users\NekshaSrinivas\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (12-Aug-2025, 11:40:37 am) [pid: 13272]
Progress (1): 17/17 MB
Progress (1): 17 MB

Downloaded from central: https://repo.maven.apache.org/maven2/archetype-catalog.xml (17 MB at 11 MB/s)
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archetypes:maven-archetype-quickstart:1.5] found in catalog remote
[INFO] Using property: javaCompilerVersion = 17
[INFO] Using property: junitVersion = 5.11.0
[INFO] Using property: groupId = org.example1
[INFO] Using property: artifactId = 24bd5a05031
[INFO] Using property: version = 0.0.1-SNAPSHOT
[INFO] Using property: package = org.example1.bd5a05031
Confirm properties configuration:
javaCompilerVersion: 17
junitVersion: 5.11.0
groupId: org.example1
artifactId: 24bd5a05031
version: 0.0.1-SNAPSHOT
package: org.example1.bd5a05031
Y: y|
```

Step-6: BUILD SUCCESS will be shown



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the following text:

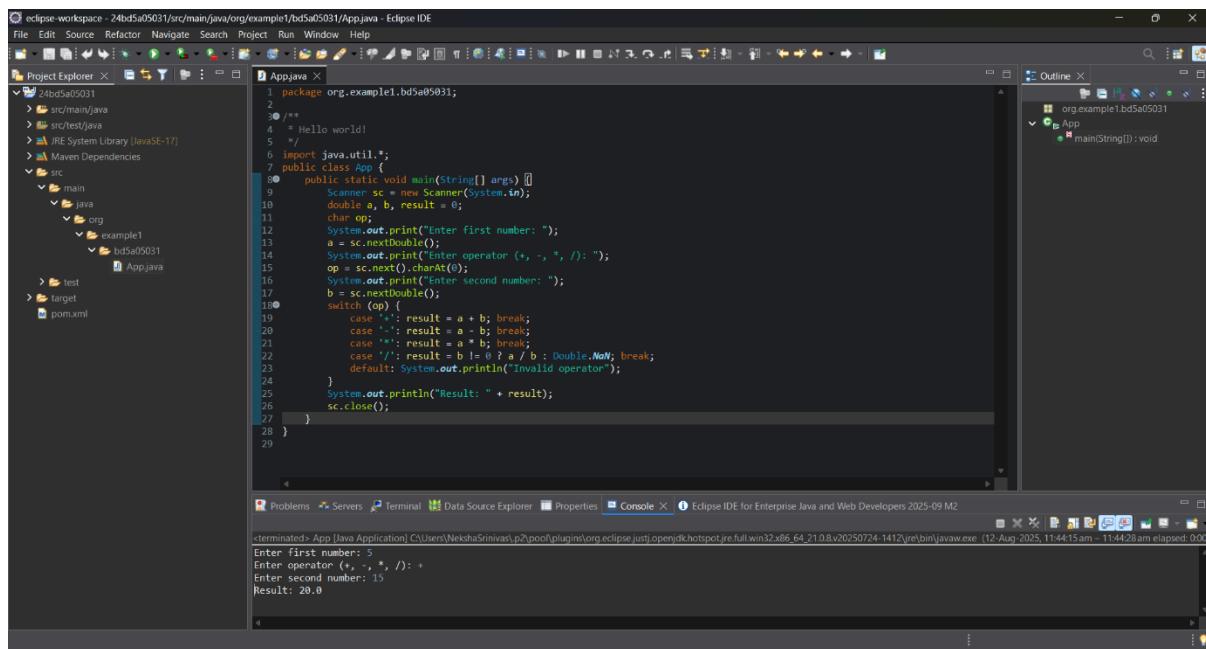
```
File Edit Navigate Search Project Run Window Help
Project Explorer X
> 24bd5a05031

Open a file or drop files here to open them.
Find Actions Ctrl+F
Show Key Assist Ctrl+Shift+N
New Ctrl+N

There is no active editor that provides an outline.

C:\Users\NekshaSrinivas\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (12-Aug-2025, 11:40:37 am) [pid: 13272]
[INFO] -----
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.5
[INFO] -----
[INFO] Parameter: groupId, Value: org.example1
[INFO] Parameter: artifactId, Value: 24bd5a05031
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] Parameter: package, Value: org.example1.bd5a05031
[INFO] Parameter: packageInPathFormat, Value: org.example1.bd5a05031
[INFO] Parameter: junitVersion, Value: 5.11.0
[INFO] Parameter: package, Value: org.example1.bd5a05031
[INFO] Parameter: groupId, Value: org.example1
[INFO] Parameter: artifactId, Value: 24bd5a05031
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] Parameter: package, Value: org.example1.bd5a05031
[WARNING] Don't override file C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a05031\src\main\java\org\example1\bd5a05031
[WARNING] Don't override file C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a05031\src\test\java\org\example1\bd5a05031
[WARNING] CP Don't override file C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a05031\.mvn
[INFO] Project created from Archetype in dir: C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a05031
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:09 min
[INFO] Finished at: 2025-08-12T11:41:47+05:30
[INFO] -----
```

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

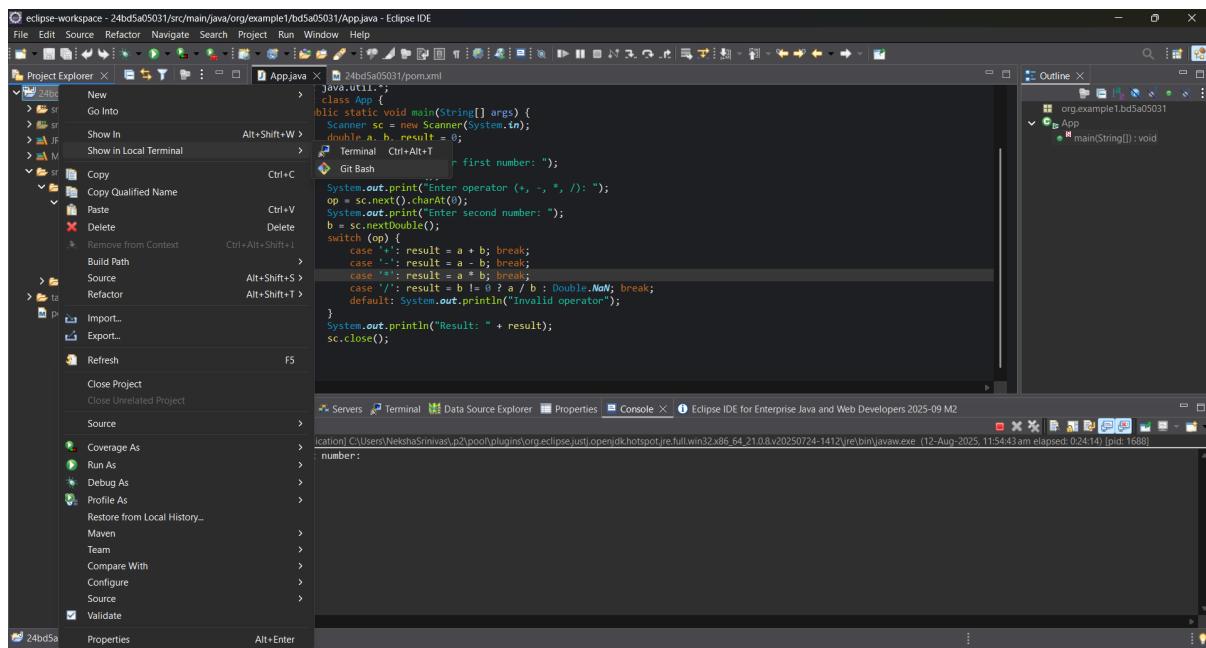


The screenshot shows the Eclipse IDE interface with the code for `App.java` in the center. The code is a simple Java application that takes two numbers from the user and performs arithmetic operations based on the operator entered. The code uses `Scanner` to read input and `System.out.println` to display results. The terminal window at the bottom shows the execution of the program and its output.

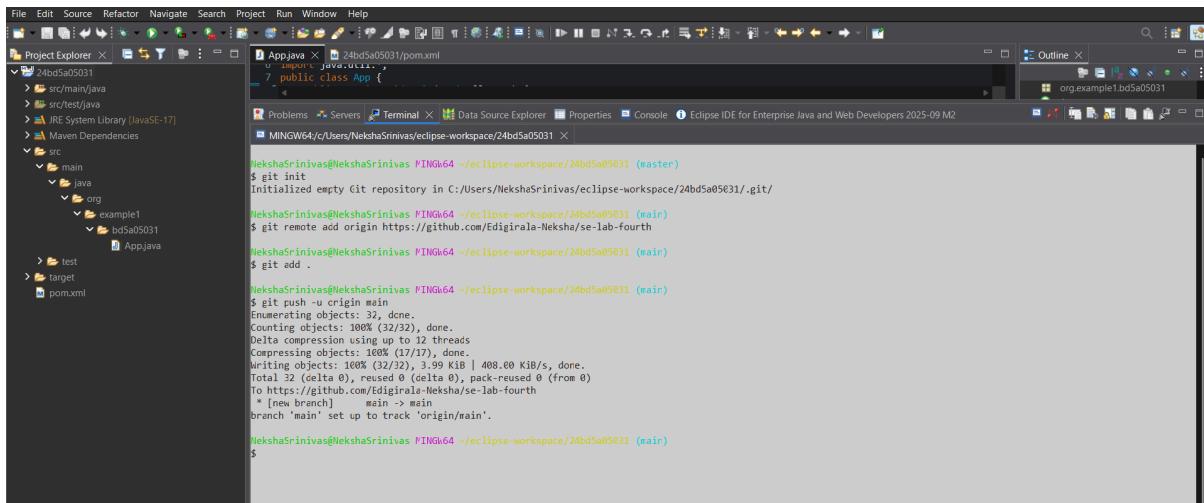
```
package org.example1.bd5a05031;
public class App {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double a, b, result = 0;
        char op;
        System.out.print("Enter first number: ");
        a = sc.nextDouble();
        System.out.print("Enter operator (+, -, *, /): ");
        op = sc.next().charAt(0);
        System.out.print("Enter second number: ");
        b = sc.nextDouble();
        switch (op) {
            case '+': result = a + b; break;
            case '-': result = a - b; break;
            case '*': result = a * b; break;
            case '/': result = b != 0 ? a / b : Double.NaN; break;
            default: System.out.println("Invalid operator");
        }
        System.out.println("Result: " + result);
        sc.close();
    }
}
```

```
terminated> App [Java Application] C:\Users\NekhaSrinivas\p2\pool\plugins\org.eclipse.jdt.core\bin\org.eclipse.jdt.core\openjdk\hotspot\jre\full\win32\x86_64_21.0.8\20250724-1412\jre\bin\javaw.exe (12-Aug-2025, 11:44:15 am - 11:44:28 am elapsed: 0:03)
Enter first number: 5
Enter operator (+, -, *, /): +
Enter second number: 15
Result: 20.0
```

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



Step-8: push to the git repo



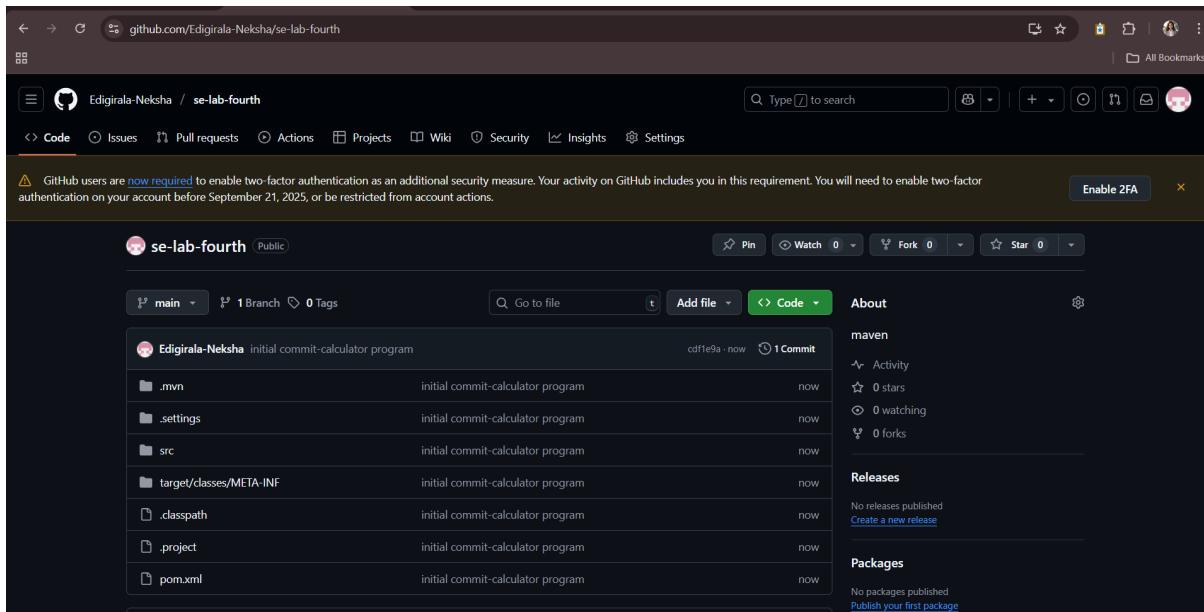
The screenshot shows the Eclipse IDE interface with the terminal window open. The terminal output is as follows:

```
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a05031 (master)
$ git init
Initialized empty Git repository in C:/Users/NekshaSrinivas/eclipse-workspace/24bd5a05031/.git/
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a05031 (nair)
$ git remote add origin https://github.com/Edigirala-Neksha/se-lab-fourth
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a05031 (nair)
$ git add .

NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a05031 (nair)
$ git push -u origin main
Enumerating objects: 32, done.
Counting objects: 32, done.
Delta compression using up to 12 threads
Compressing objects: 100% (32/32), done.
Writing objects: 100% (32/32), 3.99 KiB | 408.00 KiB/s, done.
Total 32 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Edigirala-Neksha/se-lab-fourth
 * [new branch]    main -> main
branch 'main' set up to track 'origin/main'.
```

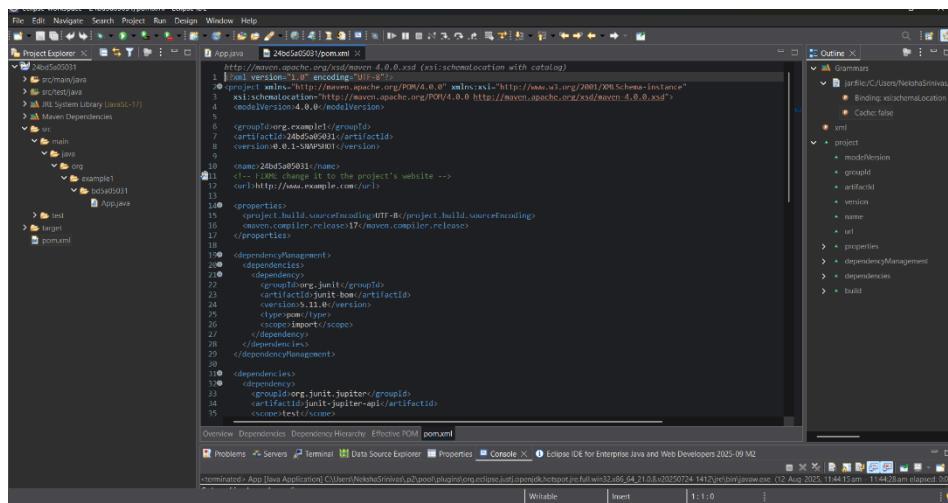
Git repo:

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



pom.xml file:

Shows the structure-

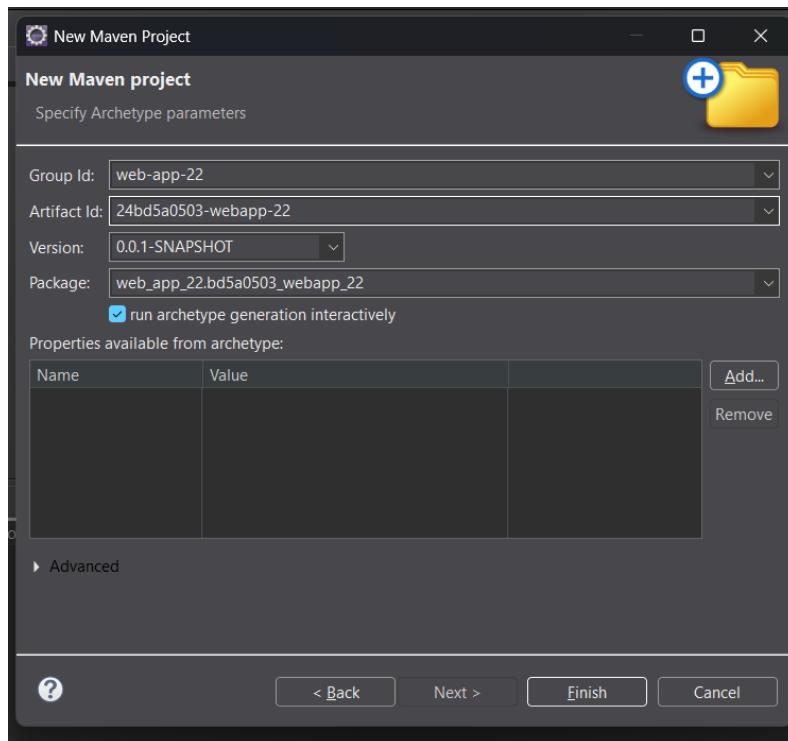


The screenshot shows the Eclipse IDE interface with the pom.xml file open in the center. The Outline view on the right side displays the structure of the XML document, including sections like project, properties, dependencyManagement, dependencies, and build. The code editor shows the XML content with various annotations and schema locations.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.example</groupId>
  <artifactId>myapp</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>MyApp</name>
  <url>http://maven.example.com/url</url>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.release>17</maven.compiler.release>
  </properties>
  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>org.junit</groupId>
        <artifactId>junit</artifactId>
        <version>11.0</version>
        <type>pom</type>
        <scope>import</scope>
      </dependency>
    </dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter-api</artifactId>
        <version>5.8.0</version>
      </dependency>
    </dependencies>
  </dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-engine</artifactId>
      <version>5.8.0</version>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-surefire-reporter</artifactId>
        <version>3.0.0-M5</version>
      </plugin>
    </plugins>
  </build>
</project>
```

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\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (02-Sept-2025, 7:19:56 pm) [pid: 13772]
Progress (1): 17/17 MB
Progress (1): 17 MB

Downloaded from central: https://repo.maven.apache.org/maven2/archetype-catalog.xml (17 MB at 9.1 MB/s)
[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] -----
```

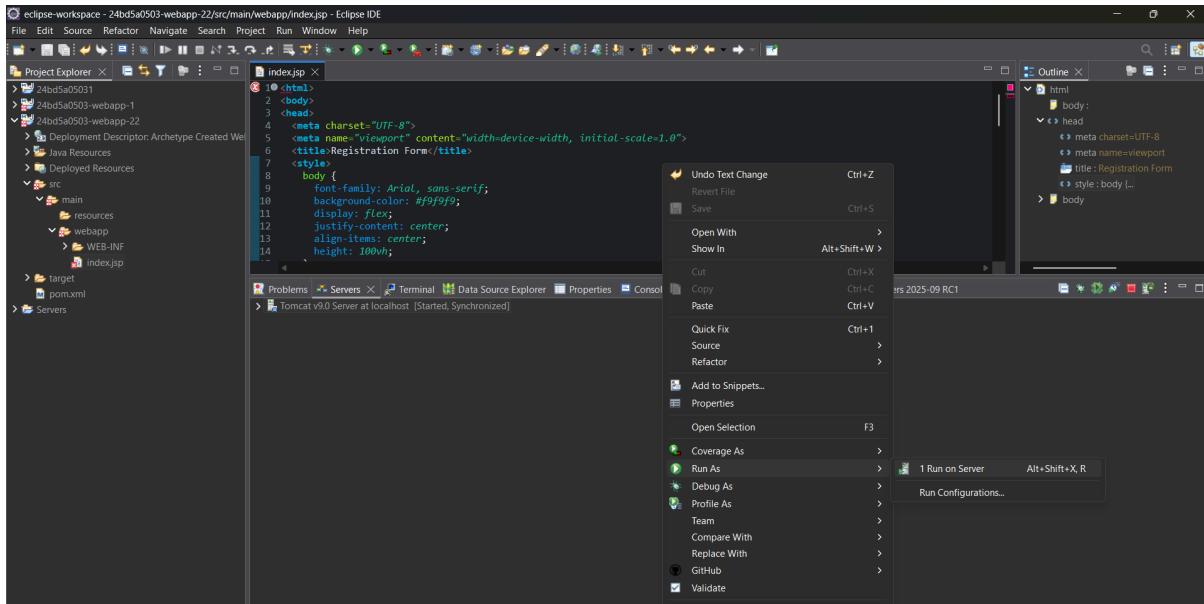
Step 4: write the html code for the web page:

```
index.jsp
1 <html>
2   <body>
3     <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Registration Form</title>
7     <style>
8       body {
9         font-family: Arial, sans-serif;
10        background-color: #f0f0f0;
11        display: flex;
12        justify-content: center;
13        align-items: center;
14        height: 100vh;
15      }
16      .form-container {
17        background: #fff;
18        padding: 20px 30px;
19        border-radius: 10px;
20        box-shadow: 0 4px 10px rgba(0,0,0,0.1);
21        width: 300px;
22      }
23      .form-container h2 {
24        text-align: center;
25        margin-bottom: 20px;
26      }
27      .form-container input {
28        width: 100%;
29        padding: 10px;
30        margin: 8px 0;
31        border: 1px solid #ccc;
32        border-radius: 5px;
33      }
34      .form-container button {
35        width: 100%;
36        padding: 10px;
37        background: #4CAF50;
38        color: white;
39        border: none;
40        cursor: pointer;
41      }
42    </style>
43  </head>
44  <body>
45    <div class="form-container">
46      <h2>Registration Form</h2>
47      <form>
48        <div>
49          <label>First Name <input type="text" name="first-name" /></label>
50        </div>
51        <div>
52          <label>Last Name <input type="text" name="last-name" /></label>
53        </div>
54        <div>
55          <label>Email <input type="email" name="email" /></label>
56        </div>
57        <div>
58          <label>Password <input type="password" name="password" /></label>
59        </div>
60        <div>
61          <button type="submit">Register</button>
62        </div>
63      </form>
64    </div>
65  </body>
66 </html>
```

Web-page:

```
index.jsp X
32     border-radius: 5px,
33 }
34 .form-container button {
35     width: 100%;
36     padding: 10px;
37     background: #4CAF50;
38     color: white;
39     border: none;
40     border-radius: 5px;
41     cursor: pointer;
42 }
43 .form-container button:hover {
44     background: #45a049;
45 }
46 </style>
47 </head>
48 <body>
49     <div class="form-container">
50         <h2>Registration Form</h2>
51         <form action="#" method="post">
52             <label for="fullname">Full Name</label>
53             <input type="text" id="fullname" name="fullname" placeholder="Enter your name" required>
54
55             <label for="email">Email</label>
56             <input type="email" id="email" name="email" placeholder="Enter your email" required>
57
58             <label for="password">Password</label>
59             <input type="password" id="password" name="password" placeholder="Enter password" required>
60
61             <label for="confirm">Confirm Password</label>
62             <input type="password" id="confirm" name="confirm" placeholder="Confirm password" required>
63
64             <button type="submit">Register</button>
65         </form>
66     </div>
67 </body>
68 </html>
69
```

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 main content area is a white card with a rounded border, titled "Registration Form". The form contains 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.

← → ⌂ localhost:8080/24bd5a0503-webapp-22/index.jsp

Registration Form

Full Name

Enter your name

Email

Enter your email

Password

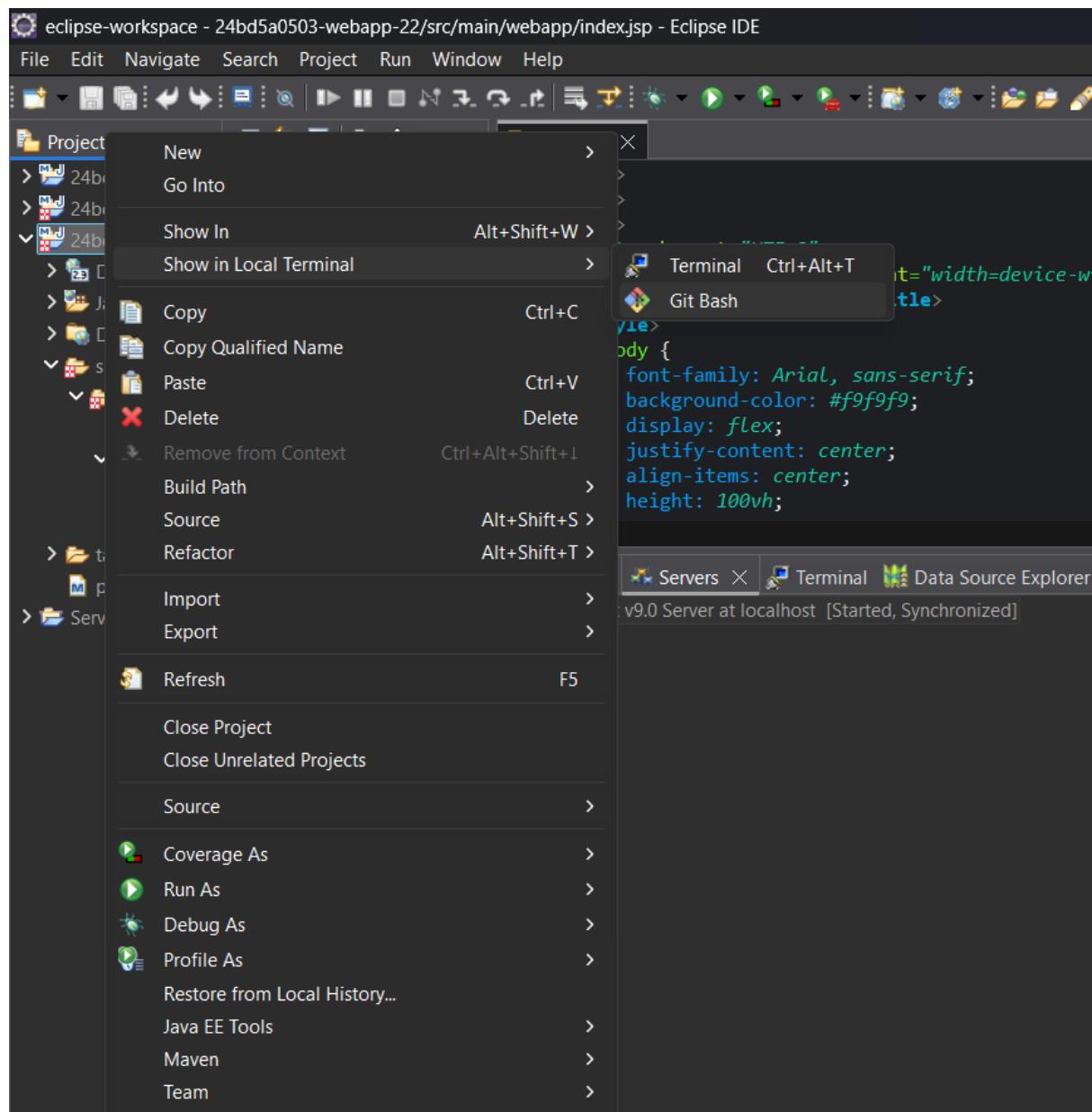
Enter password

Confirm Password

Confirm password

Register

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

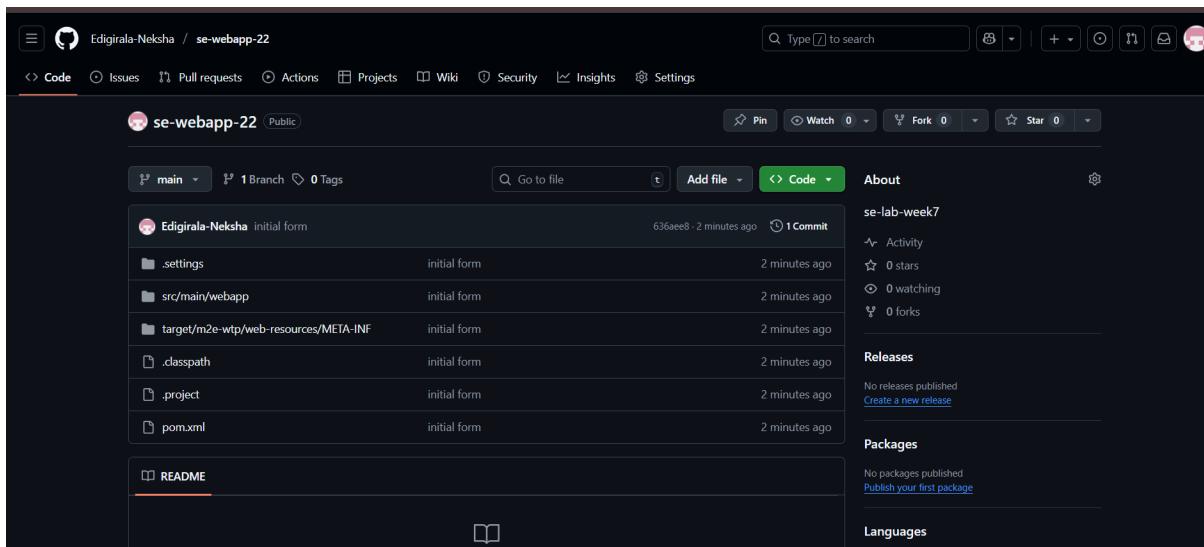


Step 8: use the command of git to push the maven web project

```
MINGW64:/c/Users/NekshaSrinivas/eclipse-workspace/24bd5a05... MINGW64:/c/Users/NekshaSrinivas/eclipse-workspace/24bd5a05... MINGW64:/c/Users/NekshaSrinivas/eclipse-workspace/24bd5a05...  
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0503-webapp-22 (master)  
$ git init  
Initialized empty Git repository in C:/Users/NekshaSrinivas/eclipse-workspace/24bd5a0503-webapp-22/.git/  
  
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0503-webapp-22 (main)  
$ git add .  
  
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0503-webapp-22 (main)  
$ git commit -m "initial form"  
[main (root-commit) 636aee8] initial form  
16 files changed, 254 insertions(+)  
create mode 100644 .classpath  
create mode 100644 .project  
create mode 100644 .settings/.jsdtscope  
create mode 100644 .settings/org.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/WEB-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 is public and contains one branch named 'main'. The 'About' section shows the repository was created by 'Edigirala-Neksha' and has 1 commit. The 'Releases' section indicates no releases have been published. The 'Languages' section shows the repository is written in Java.

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: sha256:7c06e91f61fa88c08cc74f7e1b7c69ae24910d745357e0df1d2c0322aaf2
0f9
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

Step 2: Running the Container

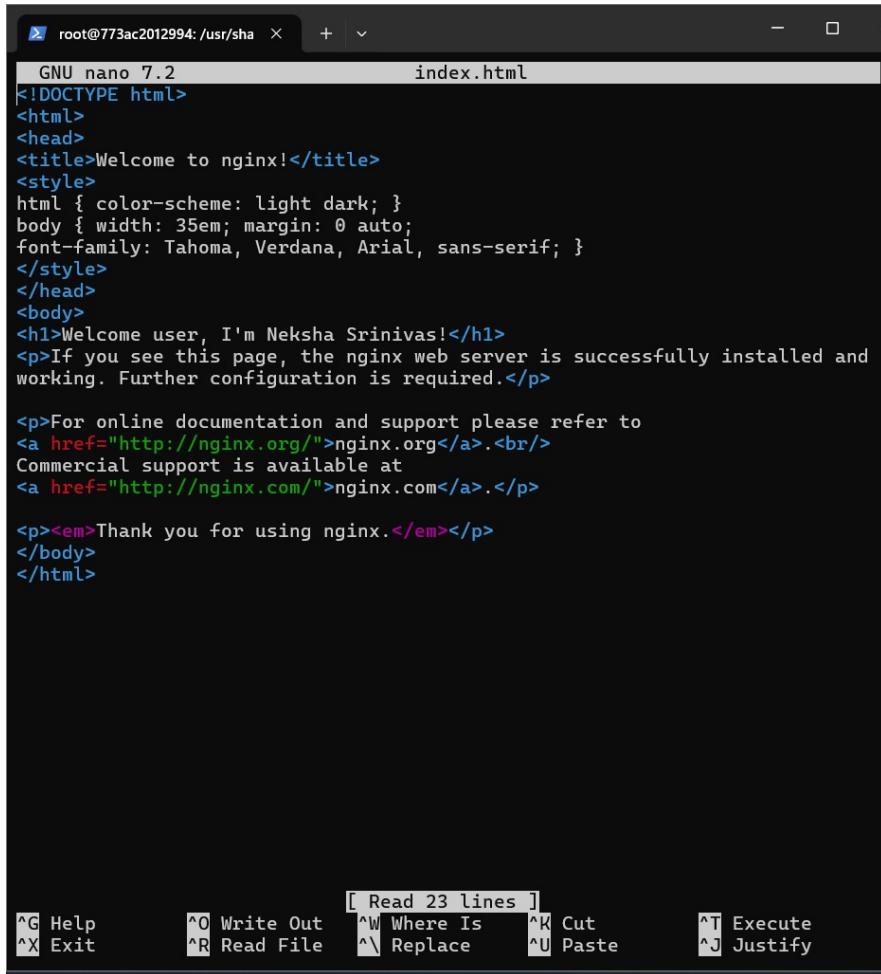
We created and started a new container named **myubuntu**, mapping port **3000** on the host to port **80** inside the container.

```
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



```
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>

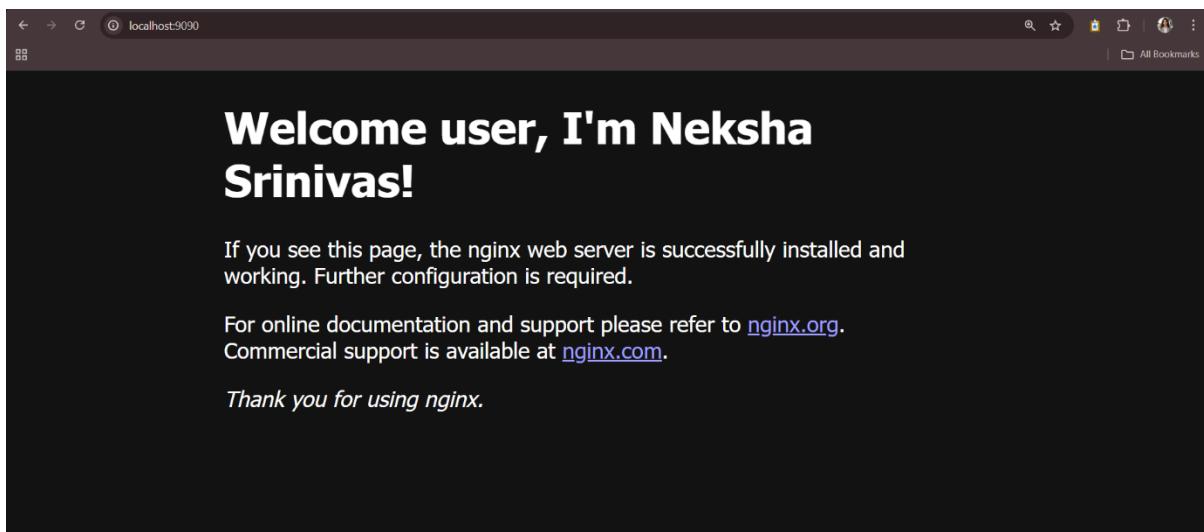
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

[Read 23 lines]

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^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: sha256:153126502820131f25f36cc59f7c4557275621bcd7a54b48c8ff
d4409685efd 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        29739008 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 44.0s (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 .dockerignore                            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                         24.4s
=> exporting to image                                     6.5s
-> => exporting layers                                    4.5s
=> => exporting manifest sha256:99d816a6b717e709d838937a995f24d0121e 0.0s
=> => exporting config sha256:f021a40f65d4b684b65cd403292af90ec68210 0.0s
=> => exporting attestation manifest sha256:04dc38eee96b84e155b083e5 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]
6a5ccfd4b031: Pushed
edd67216c21: Pushed
b71466b94f26: Mounted from nekshasrinivas/img-commmit-1
004a734bd8b1: 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> |
```

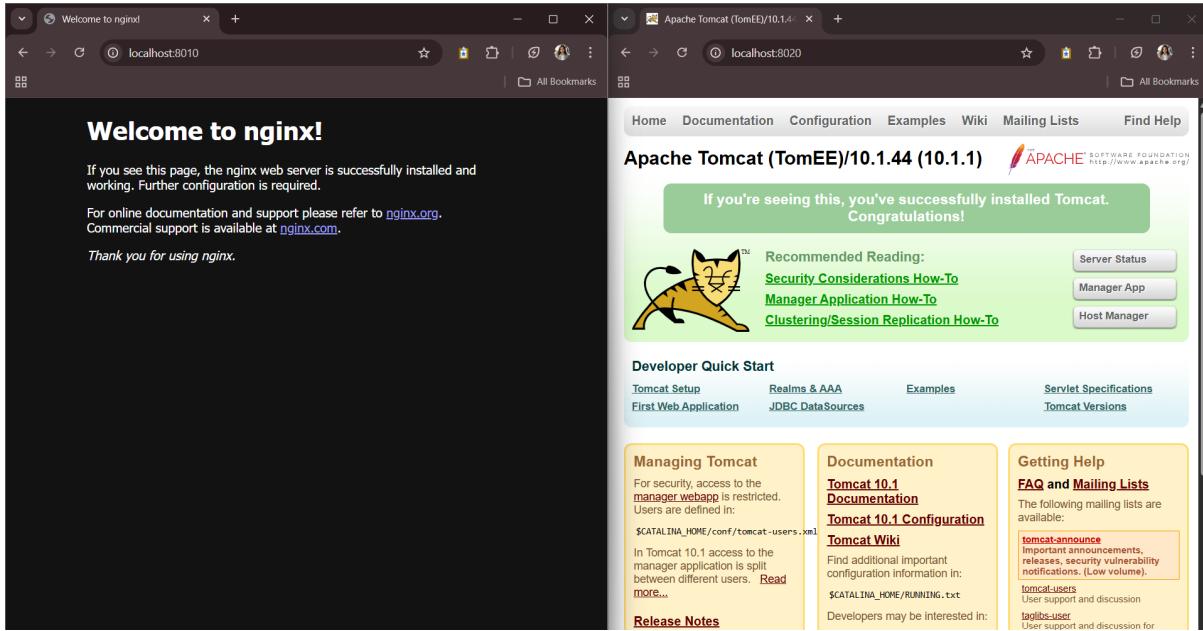
DOCKER COMPOSE FILE:

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



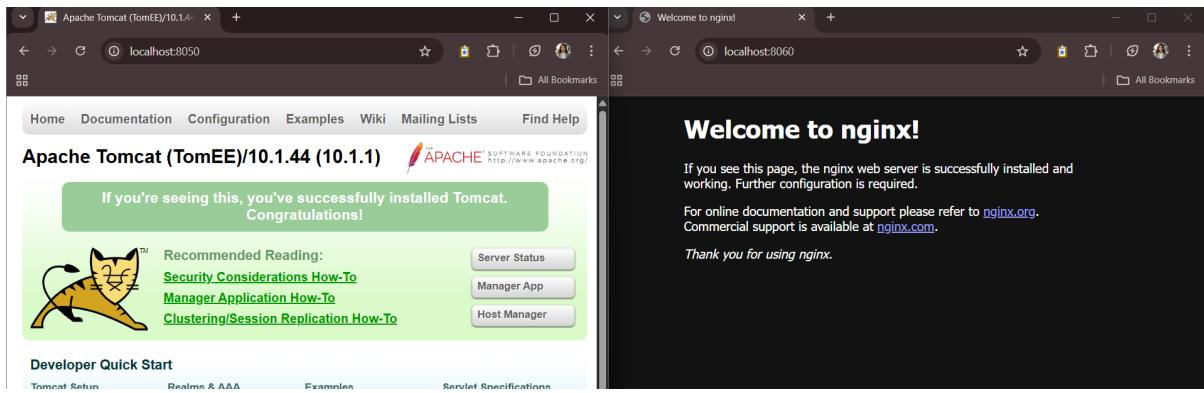
Step 3: Using docker file to run two servers parallelly

```
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...
  ✓ Container comp-1-server-db-1   St...
  ✓ Container comp-1-server-web-1  S...
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

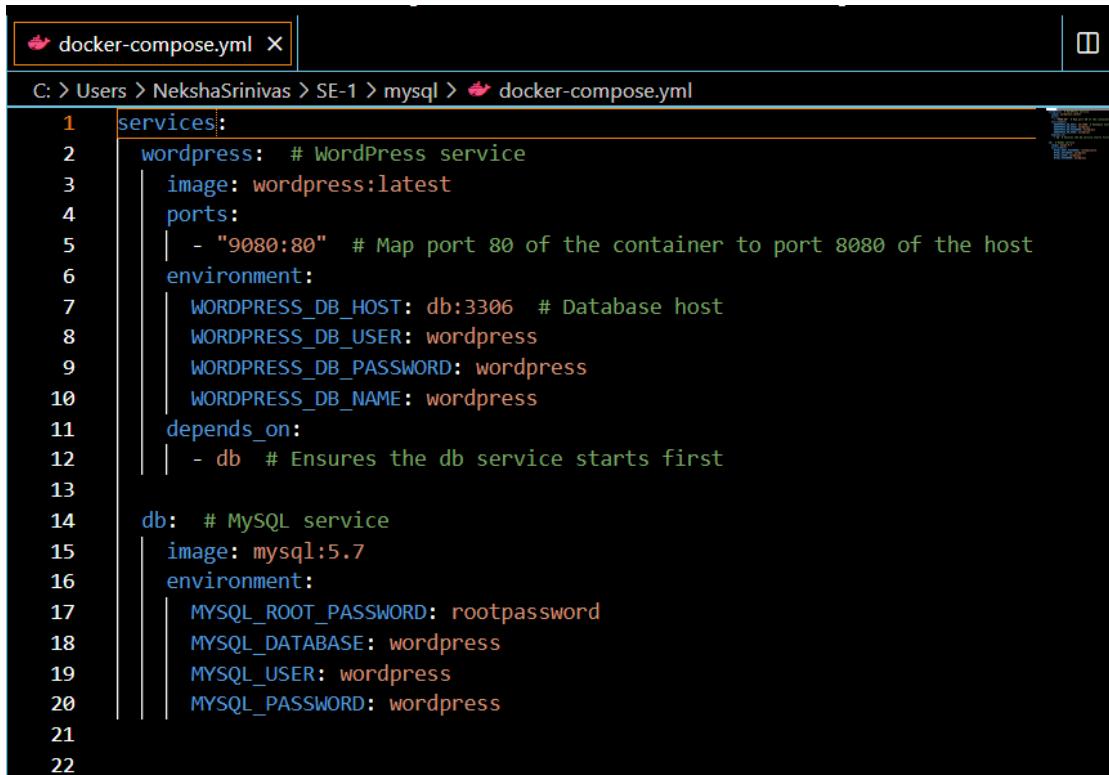
PS C:\Users\NekshaSrinivas\SE-1\mysql>
PS C:\Users\NekshaSrinivas\SE-1\mysql> ren docker-compose.txt docker-compose
.yml
PS C:\Users\NekshaSrinivas\SE-1\mysql> ls

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:



```

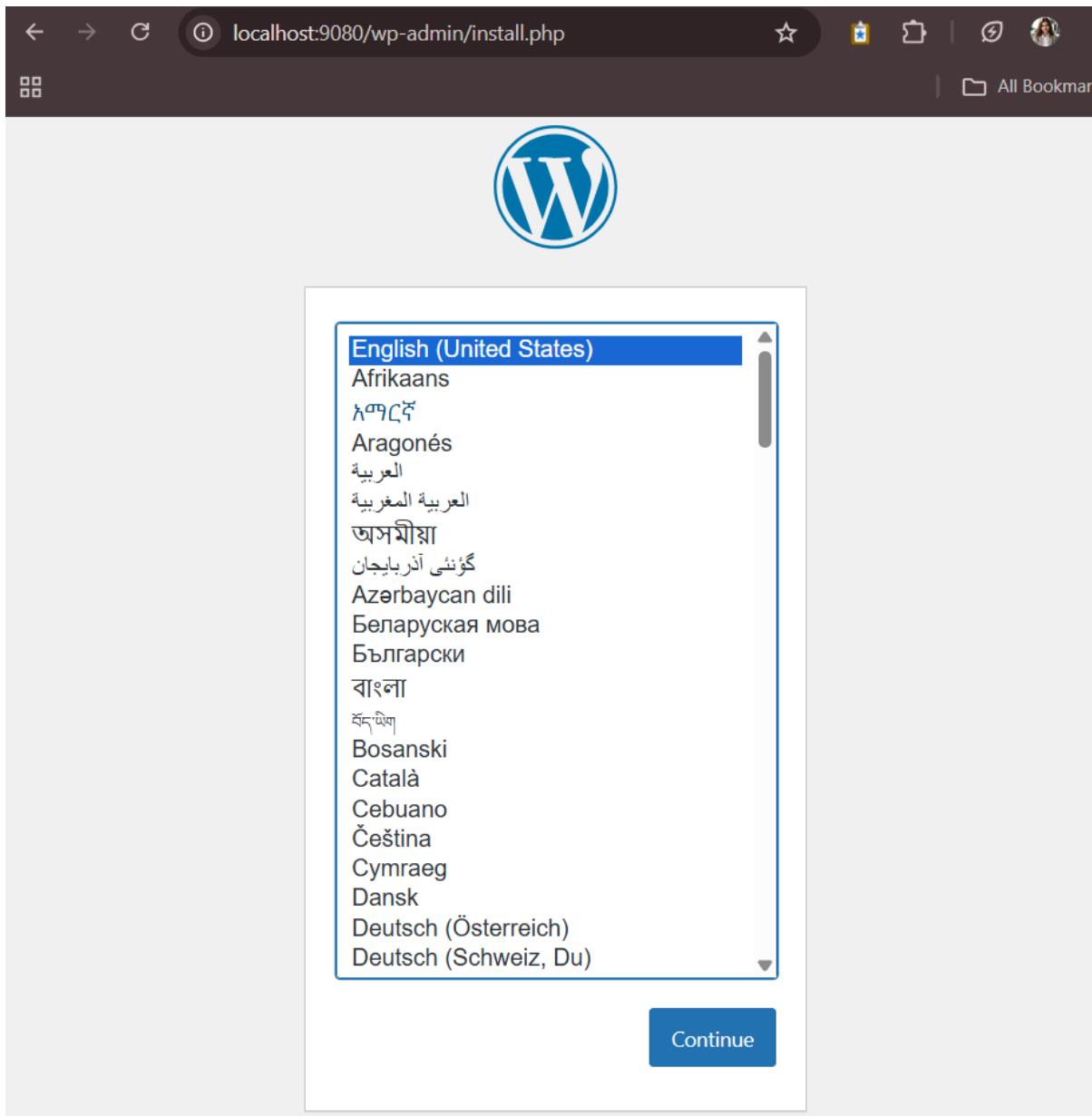
services:
  wordpress: # WordPress service
    image: wordpress:latest
    ports:
      - "9080:80" # Map port 80 of the container to port 8080 of the host
    environment:
      WORDPRESS_DB_HOST: db:3306 # Database host
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
      WORDPRESS_DB_NAME: wordpress
    depends_on:
      - db # Ensures the db service starts first
  db: # MySQL service
    image: mysql:5.7
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      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

edigralaneksha@gmail.com

Double-check your email address before continuing.

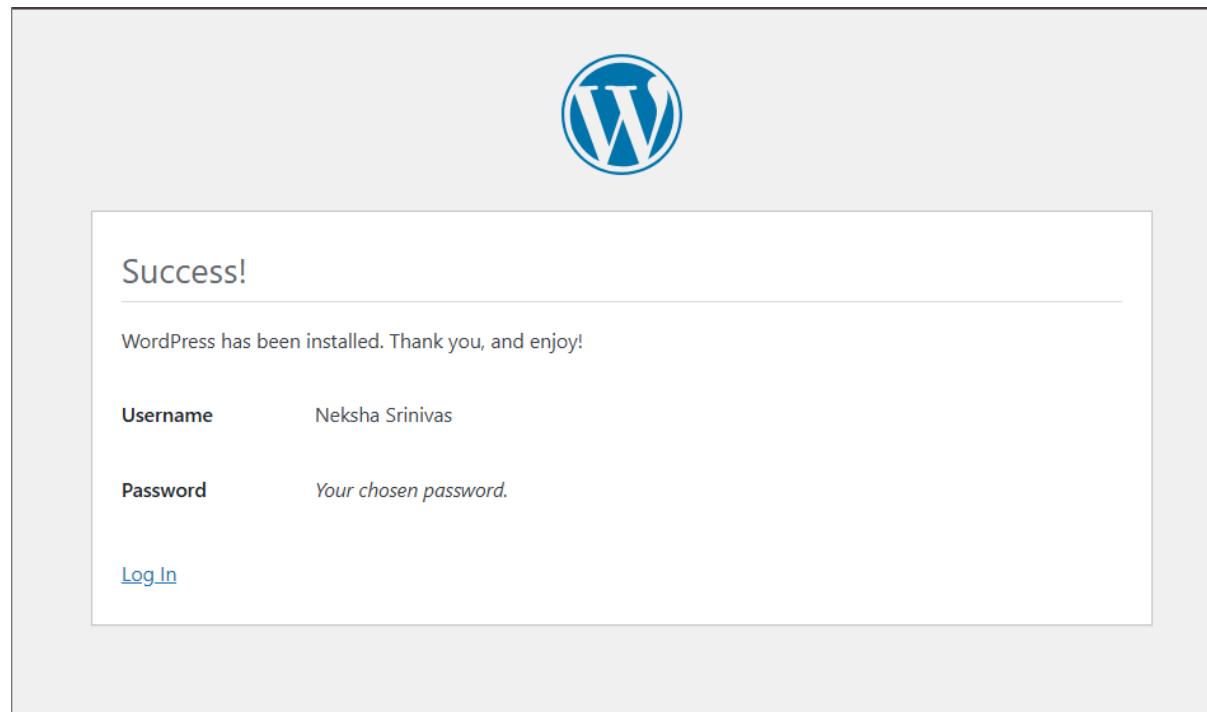
Search engine visibility

Discourage search engines from indexing this site

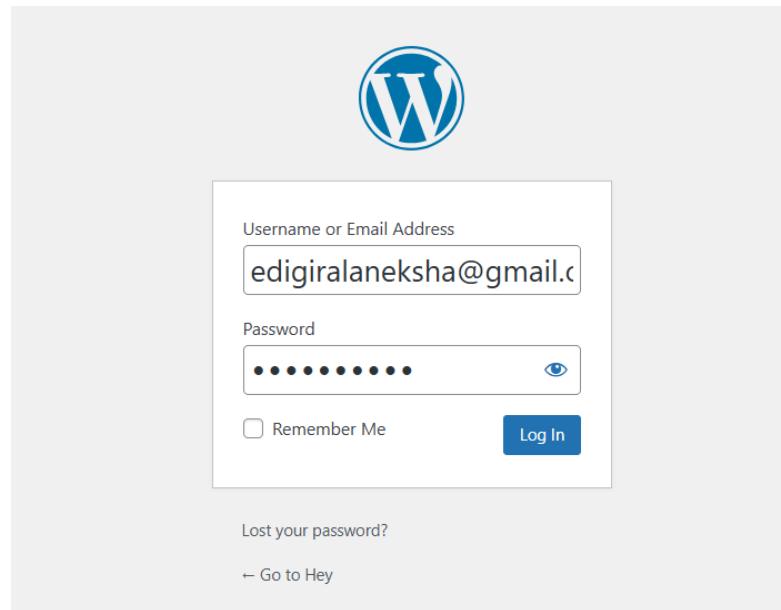
It is up to search engines to honor this request.

[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

The screenshot shows the WordPress dashboard at `localhost:9080/wp-admin/`. At the top right, there is a user profile for "Neksha Srinivas" with options to "Edit Profile" and "Log Out". The main header says "Welcome to WordPress!" and "Learn more about the 6.8.2 version.". Below this, there are three sections: "Author rich content with blocks and patterns", "Customize your entire site with block themes", and "Switch up your site's look & feel with Styles". Each section has a brief description and a link to "Open site editor" or "Edit styles". At the bottom left, there is a "Site Health Status" box showing "No information" and a "Quick Draft" box with a title input field.

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:

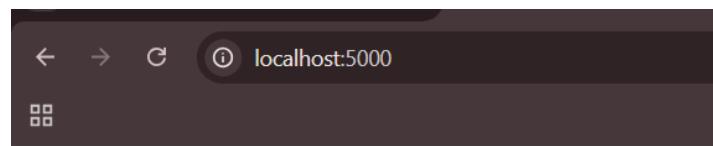
```
Mode           LastWriteTime          Length Name
----           -----              ----
-a---          28-08-2025    10:02           187 app.py
-a---          28-08-2025    10:03            82 docker-compose.yml
-a---          28-08-2025    10:02           105 Dockerfile

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="2025-08-28T10:24:45+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
    ✓ 04fa42a56901 Pull complete               75.7s
    ✓ 500d7b2546c4 Pull complete               1.8s
    ✓ ecc6cc933381 Pull complete               38.2s
    ✓ 5cd63fb67c17 Pull complete               38.5s
    ✓ 4d3eacc36b14 Pull complete               1.7s
    ✓ 9476b8faedba Pull complete               1.7s
    ✓ 789fa151603e Pull complete               3.5s
    ✓ 1756a372d796 Pull complete               1.8s
    ✓ bc0f5543b464 Pull complete               1.9s
    ✓ 131412d69359 Pull complete               67.6s
    ✓ 03ca01bc78d4 Pull complete               42.5s
    ✓ 03ca01bc78d4 Pull complete               1.8s
#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
```

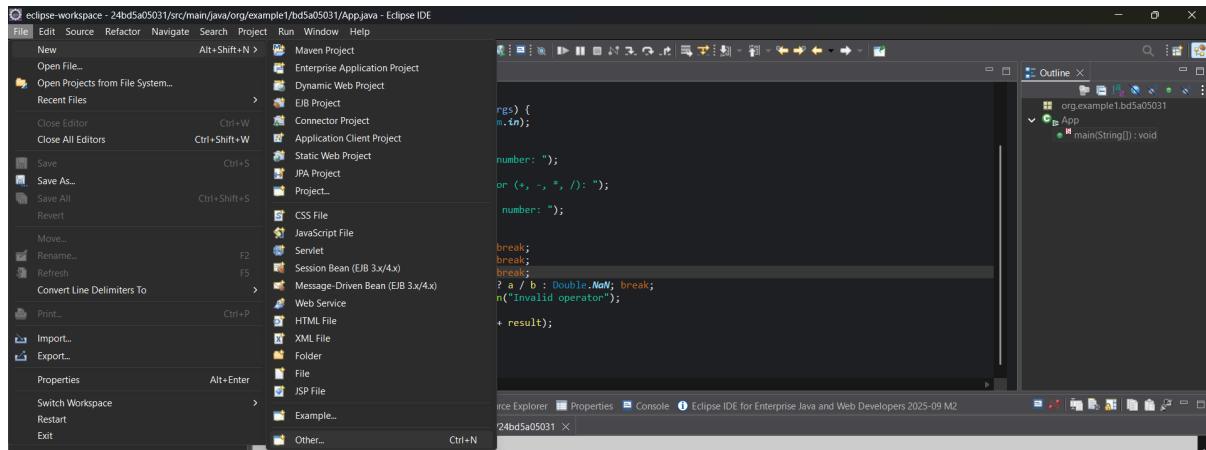
Step 4: Open the local host to view the custom page:



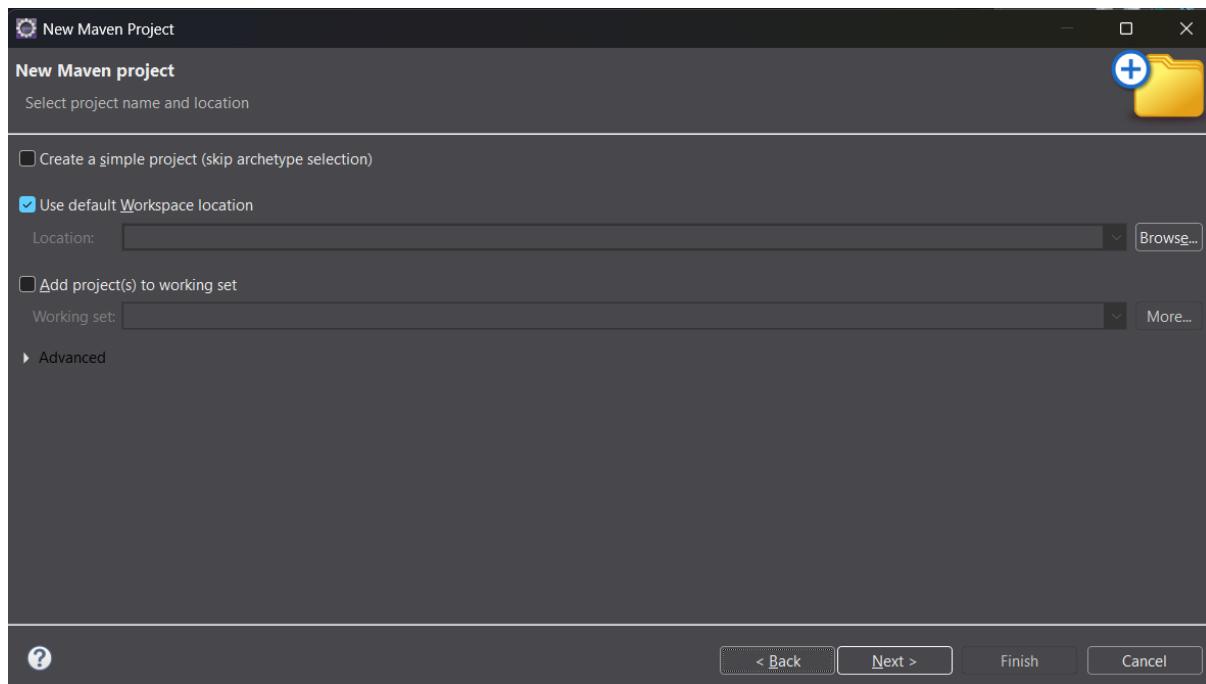
Hello from 24BD5A0503- NEKSHASRINIVAS

7. Creating a Multi-Module Maven Project

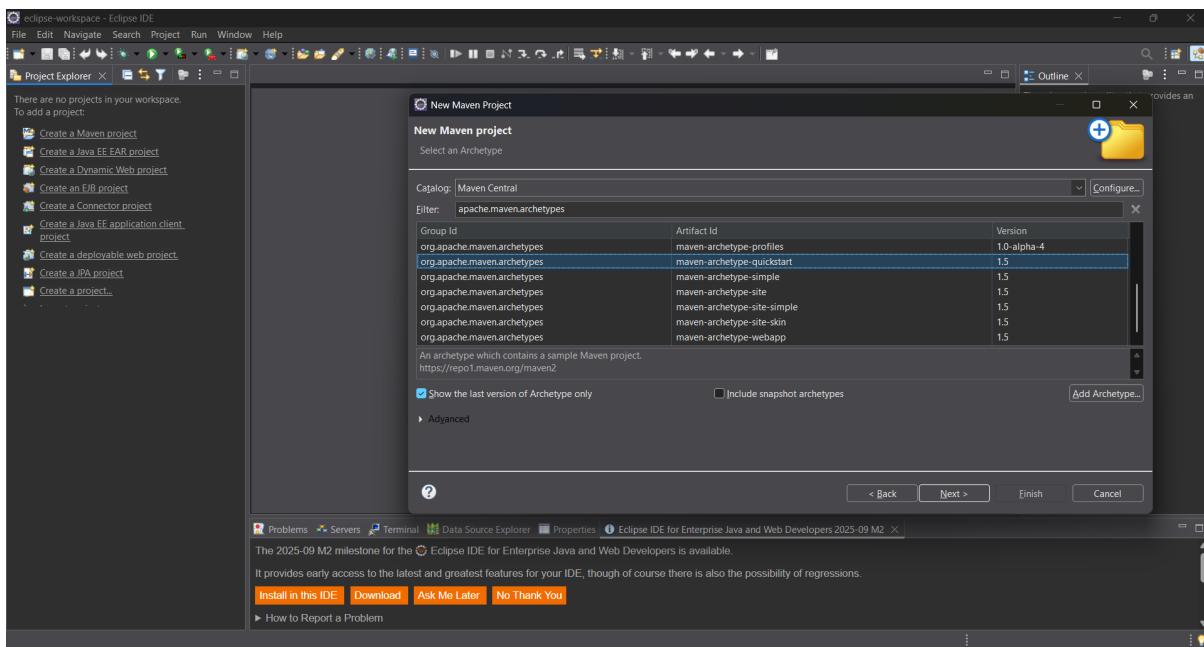
Step-1: Open the eclipse and click on file>new>Maven project



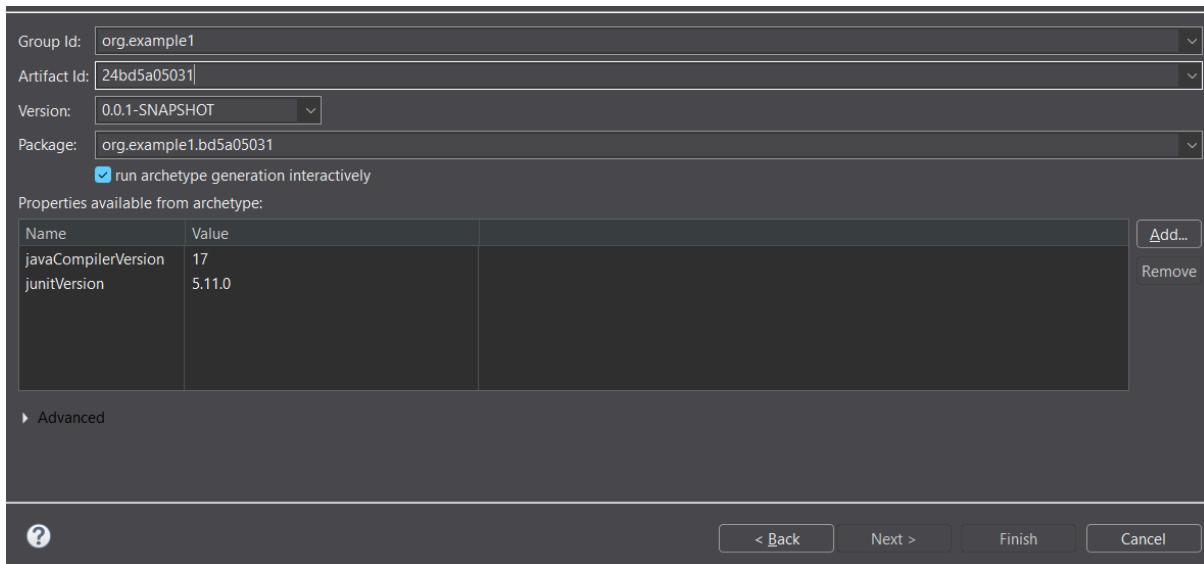
Step-2: select the default workspace and click on next



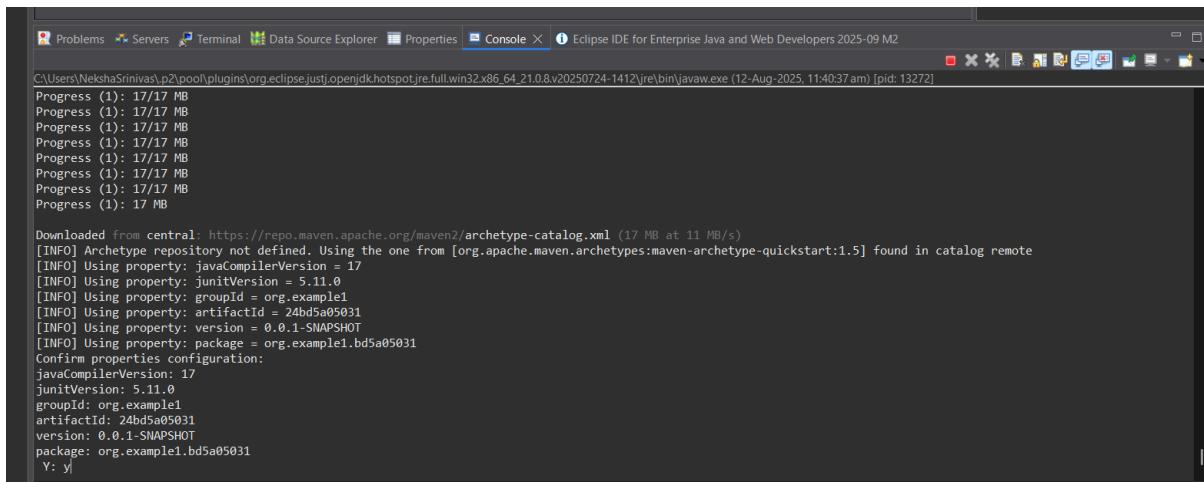
Step-3: in the filter option select the one maven-archetype-quickstart



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

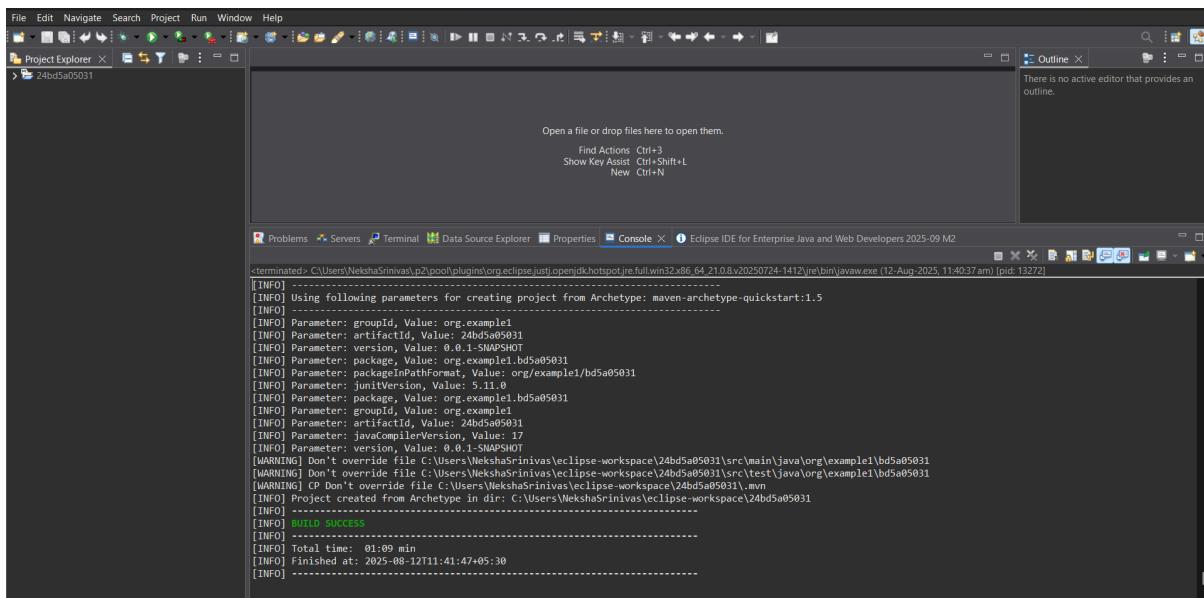


The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the following text:

```
C:\Users\NekshaSrinivas\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (12-Aug-2025, 11:40:37 am) [pid: 13272]
Progress (1): 17/17 MB
Progress (1): 17 MB

Downloaded from central: https://repo.maven.apache.org/maven2/archetype-catalog.xml (17 MB at 11 MB/s)
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archetypes:maven-archetype-quickstart:1.5] found in catalog remote
[INFO] Using property: javaCompilerVersion = 17
[INFO] Using property: junitVersion = 5.11.0
[INFO] Using property: groupId = org.example1
[INFO] Using property: artifactId = 24bd5a05031
[INFO] Using property: version = 0.0.1-SNAPSHOT
[INFO] Using property: package = org.example1.bd5a05031
Confirm properties configuration:
javaCompilerVersion: 17
junitVersion: 5.11.0
groupId: org.example1
artifactId: 24bd5a05031
version: 0.0.1-SNAPSHOT
package: org.example1.bd5a05031
Y: y|
```

Step-6: BUILD SUCCESS will be shown



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the following text:

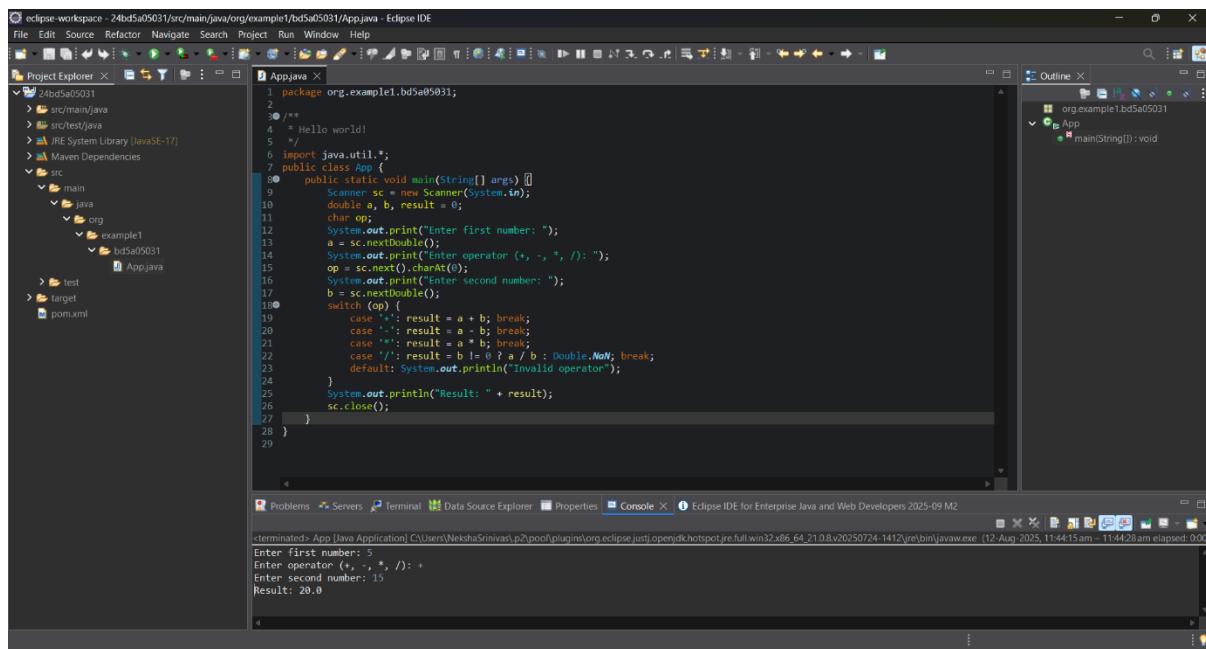
```
File Edit Navigate Search Project Run Window Help
Project Explorer X
> 24bd5a05031

Open a file or drop files here to open them.
Find Actions Ctrl+F
Show Key Assist Ctrl+Shift+N
New Ctrl+N

There is no active editor that provides an outline.

C:\Users\NekshaSrinivas\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (12-Aug-2025, 11:40:37 am) [pid: 13272]
[INFO] -----
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.5
[INFO] -----
[INFO] Parameter: groupId, Value: org.example1
[INFO] Parameter: artifactId, Value: 24bd5a05031
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] Parameter: package, Value: org.example1.bd5a05031
[INFO] Parameter: packageInPathFormat, Value: org.example1.bd5a05031
[INFO] Parameter: junitVersion, Value: 5.11.0
[INFO] Parameter: package, Value: org.example1.bd5a05031
[INFO] Parameter: groupId, Value: org.example1
[INFO] Parameter: artifactId, Value: 24bd5a05031
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] Parameter: package, Value: org.example1.bd5a05031
[WARNING] Don't override file C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a05031\src\main\java\org\example1\bd5a05031
[WARNING] Don't override file C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a05031\src\test\java\org\example1\bd5a05031
[WARNING] CP Don't override file C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a05031\.mvn
[INFO] Project created from Archetype in dir: C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a05031
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:09 min
[INFO] Finished at: 2025-08-12T11:41:47+05:30
[INFO] -----
```

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

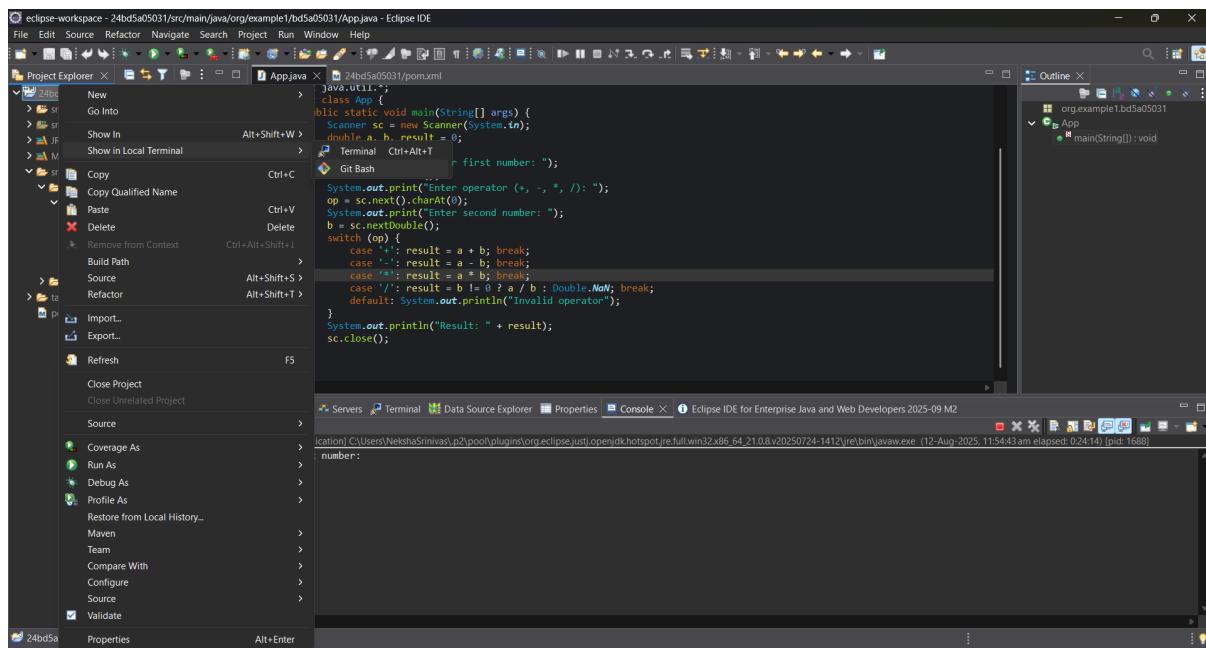


The screenshot shows the Eclipse IDE interface with the code for `App.java` in the center. The code is a simple Java application that takes two numbers from the user and performs arithmetic operations based on the operator entered. The code uses `Scanner` to read input and `System.out.println` to display results. The terminal window at the bottom shows the execution of the program and its output.

```
package org.example1.bd5a05031;
public class App {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double a, b, result = 0;
        char op;
        System.out.print("Enter first number: ");
        a = sc.nextDouble();
        System.out.print("Enter operator (+, -, *, /): ");
        op = sc.next().charAt(0);
        System.out.print("Enter second number: ");
        b = sc.nextDouble();
        switch (op) {
            case '+': result = a + b; break;
            case '-': result = a - b; break;
            case '*': result = a * b; break;
            case '/': result = b != 0 ? a / b : Double.NaN; break;
            default: System.out.println("Invalid operator");
        }
        System.out.println("Result: " + result);
        sc.close();
    }
}
```

```
terminated> App [Java Application] C:\Users\NekhaSrinivas\p2\pool\plugins\org.eclipse.jdt.core\bin\org.eclipse.jdt.core\openjdk\hotspot\jre\full\win32\x86_64_21.0.8\20250724-1412\jre\bin\javaw.exe (12-Aug-2025, 11:44:15 am - 11:44:28 am elapsed: 0:03)
Enter first number: 5
Enter operator (+, -, *, /): +
Enter second number: 15
Result: 20.0
```

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



Step-8: push to the git repo

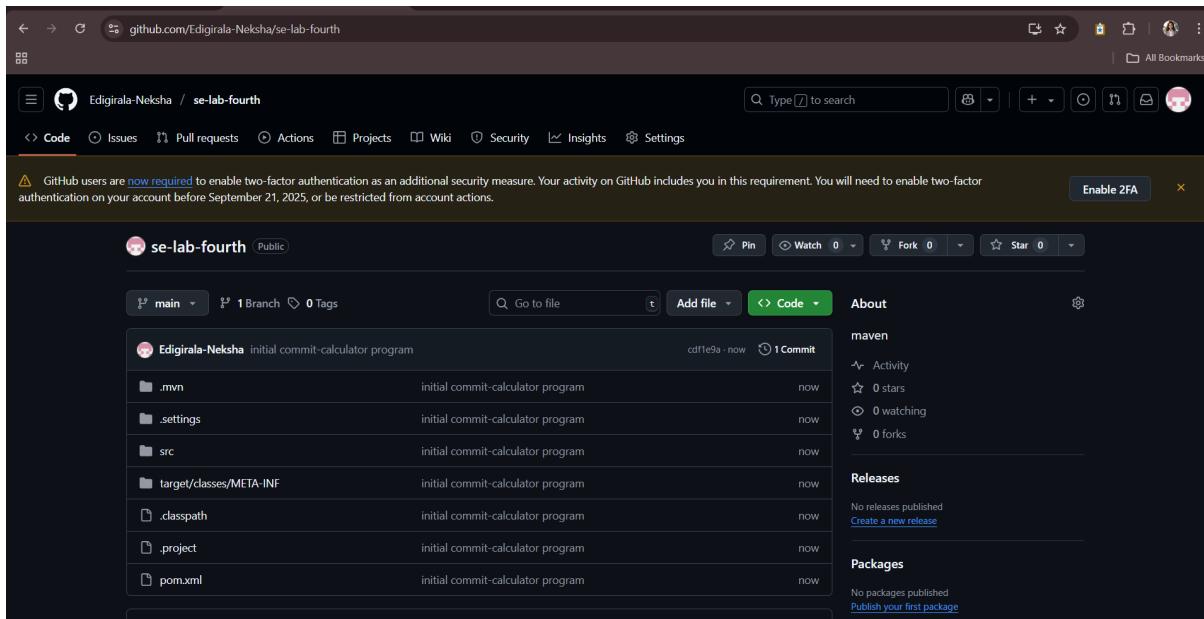
The screenshot shows the Eclipse IDE interface with the terminal window open. The terminal output shows the user pushing changes to a GitHub repository named 'se-lab-fourth'. The command \$ git push -u origin main was run, which initialized an empty Git repository, added the remote origin, and pushed the 'main' branch.

```
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a05031 (master)
$ git init
Initialized empty Git repository in C:/Users/NekshaSrinivas/eclipse-workspace/24bd5a05031/.git/
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a05031 (main)
$ git remote add origin https://github.com/Edigirala-Neksha/se-lab-fourth
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a05031 (main)
$ git add .

NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a05031 (main)
$ git push -u origin main
Enumerating objects: 32, done.
Counting objects: 32 (32/32), done.
Delta compression using up to 12 threads
Compressing objects: 100% (32/32), done.
Writing objects: 100% (32/32), 3.99 KiB | 408.00 KiB/s, done.
Total 32 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Edigirala-Neksha/se-lab-fourth
 * [new branch]    main -> main
branch 'main' set up to track 'origin/main'.
```

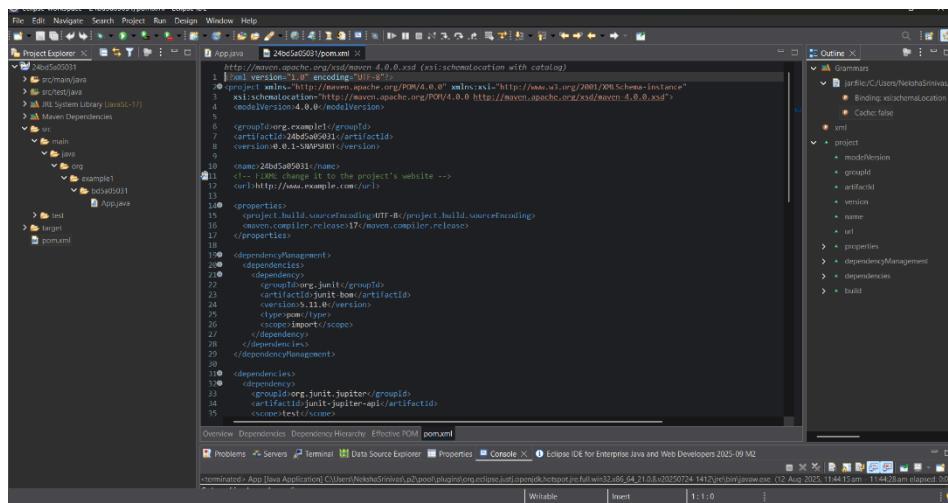
Git repo:

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



pom.xml file:

Shows the structure-

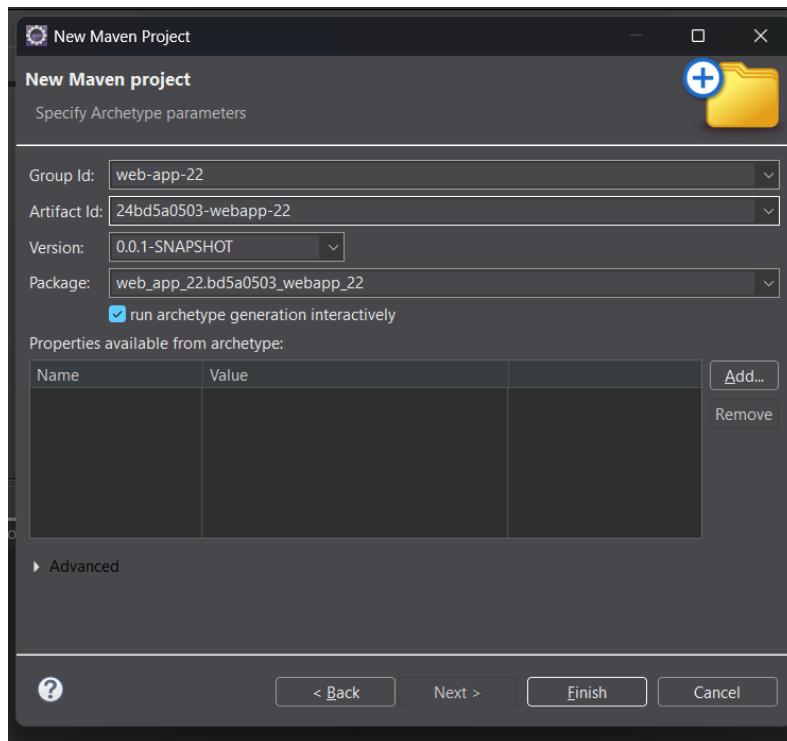


The screenshot shows the Eclipse IDE interface with the pom.xml file open in the central editor area. The Outline view on the right side displays the hierarchical structure of the XML document, including sections like project, properties, dependencyManagement, dependencies, and build.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.example</groupId>
  <artifactId>myapp</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>MyApp</name>
  <url>http://maven.example.com/url</url>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.release>17</maven.compiler.release>
  </properties>
  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>org.junit</groupId>
        <artifactId>junit</artifactId>
        <version>11.0</version>
        <type>pom</type>
        <scope>import</scope>
      </dependency>
    </dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter-api</artifactId>
        <version>5.8.0</version>
      </dependency>
    </dependencies>
  </dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-engine</artifactId>
      <version>5.8.0</version>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-surefire-reporter</artifactId>
        <version>3.0.0-M1</version>
      </plugin>
    </plugins>
  </build>
</project>
```

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\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (02-Sept-2025, 7:19:56 pm) [pid: 13772]
Progress (1): 17/17 MB
Progress (1): 17 MB

Downloaded from central: https://repo.maven.apache.org/maven2/archetype-catalog.xml (17 MB at 9.1 MB/s)
[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

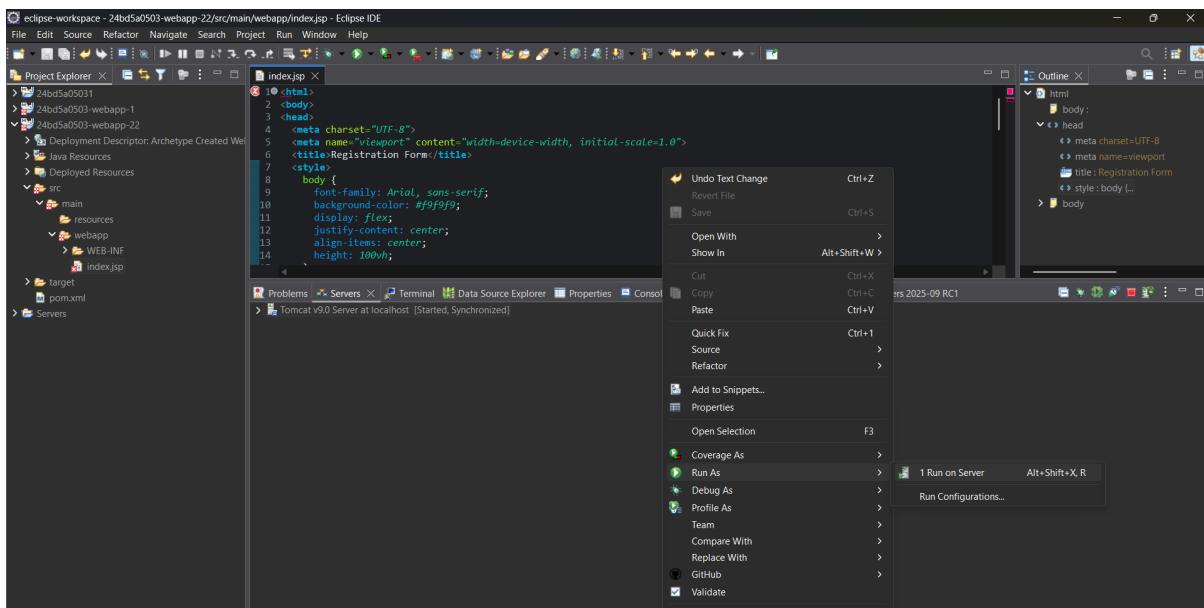
```
package: web_app_22.bd5a0503_webapp_22
Y: y
[INFO] -----
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-webapp:1.0
[INFO] -----
[INFO] Parameter: basedir, Value: C:\Users\NekshaSrinivas\eclipse-workspace
[INFO] Parameter: package, Value: web_app_22.bd5a0503_webapp_22
[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
1 <html>
2   <body>
3     <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Registration Form</title>
7     <style>
8       body {
9         font-family: Arial, sans-serif;
10        background-color: #f0f0f0;
11        display: flex;
12        justify-content: center;
13        align-items: center;
14        height: 100vh;
15      }
16      .form-container {
17        background: #fff;
18        padding: 20px 30px;
19        border-radius: 10px;
20        box-shadow: 0 4px 10px rgba(0,0,0,0.1);
21        width: 300px;
22      }
23      .form-container h2 {
24        text-align: center;
25        margin-bottom: 20px;
26      }
27      .form-container input {
28        width: 100%;
29        padding: 10px;
30        margin: 8px 0;
31        border: 1px solid #ccc;
32        border-radius: 5px;
33      }
34      .form-container button {
35        width: 100%;
36        padding: 10px;
37        background: #4CAF50;
38        color: white;
39        border: none;
40        cursor: pointer;
41      }
42    </style>
43  </head>
44  <body>
45    <div class="form-container">
46      <h2>Registration Form</h2>
47      <form>
48        <div>
49          <label>First Name <input type="text" name="firstName" /></label>
50        </div>
51        <div>
52          <label>Last Name <input type="text" name="lastName" /></label>
53        </div>
54        <div>
55          <label>Email <input type="email" name="email" /></label>
56        </div>
57        <div>
58          <label>Password <input type="password" name="password" /></label>
59        </div>
60        <div>
61          <label>Confirm Password <input type="password" name="confirmPassword" /></label>
62        </div>
63        <div>
64          <button type="submit">Register</button>
65        </div>
66      </form>
67    </div>
68  </body>
69 </html>
```

```
index.jsp X
32     border-radius: 5px;
33 }
34 .form-container button {
35     width: 100%;
36     padding: 10px;
37     background: #4CAF50;
38     color: white;
39     border: none;
40     border-radius: 5px;
41     cursor: pointer;
42 }
43 .form-container button:hover {
44     background: #45a049;
45 }
46 </style>
47 </head>
48 <body>
49     <div class="form-container">
50         <h2>Registration Form</h2>
51         <form action="#" method="post">
52             <label for="fullname">Full Name</label>
53             <input type="text" id="fullname" name="fullname" placeholder="Enter your name" required>
54
55             <label for="email">Email</label>
56             <input type="email" id="email" name="email" placeholder="Enter your email" required>
57
58             <label for="password">Password</label>
59             <input type="password" id="password" name="password" placeholder="Enter password" required>
60
61             <label for="confirm">Confirm Password</label>
62             <input type="password" id="confirm" name="confirm" placeholder="Confirm password" required>
63
64             <button type="submit">Register</button>
65         </form>
66     </div>
67 </body>
68 </html>
69
```

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 main content area is a white card with a rounded border, titled "Registration Form". The form contains 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.

← → ⌂ localhost:8080/24bd5a0503-webapp-22/index.jsp

Registration Form

Full Name

Enter your name

Email

Enter your email

Password

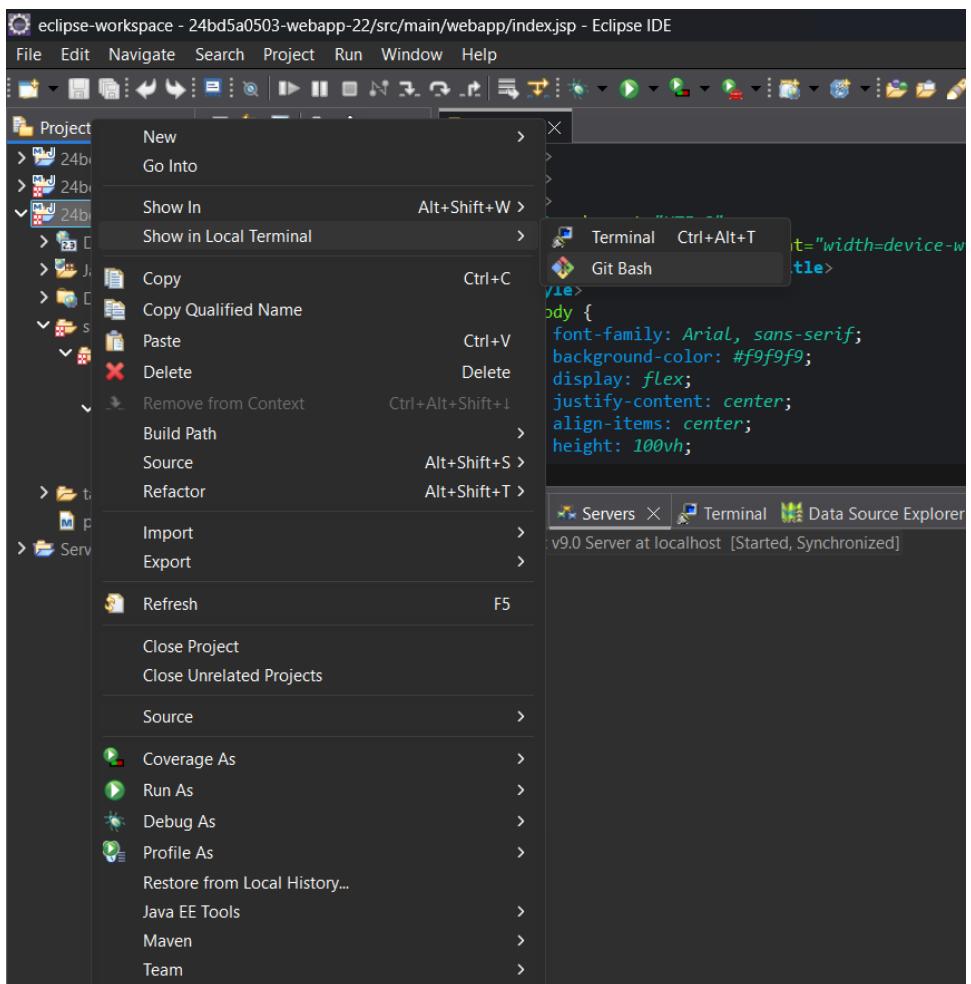
Enter password

Confirm Password

Confirm password

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/24bd5a0503-webapp-22$ git init
Initialized empty Git repository in C:/Users/NekshaSrinivas/eclipse-workspace/24bd5a0503-webapp-22/.git/
MINGW64:/c/Users/NekshaSrinivas/eclipse-workspace/24bd5a0503-webapp-22$ git add .
NekshaSrinivas@NekshaSrinivas MINGW64 ~/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/org.eclipse.jdt.core.prefs
 create mode 100644 .settings/crg.eclipse.wst.core.prefs
 create mode 100644 .settings/org.eclipse.wst.common.component
 create mode 100644 .settings/crg.eclipse.common.project.facet.core.xml
 create mode 100644 .settings/org.eclipse.wst.jsdt.ui.superType.container
 create mode 100644 .settings/crg.eclipse.wst.jsdt.ui.superType.name
 create mode 100644 .settings/org.eclipse.wst.validation.prefs
 create mode 100644 pom.xml
 create mode 100644 src/main/webapp/WEB-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

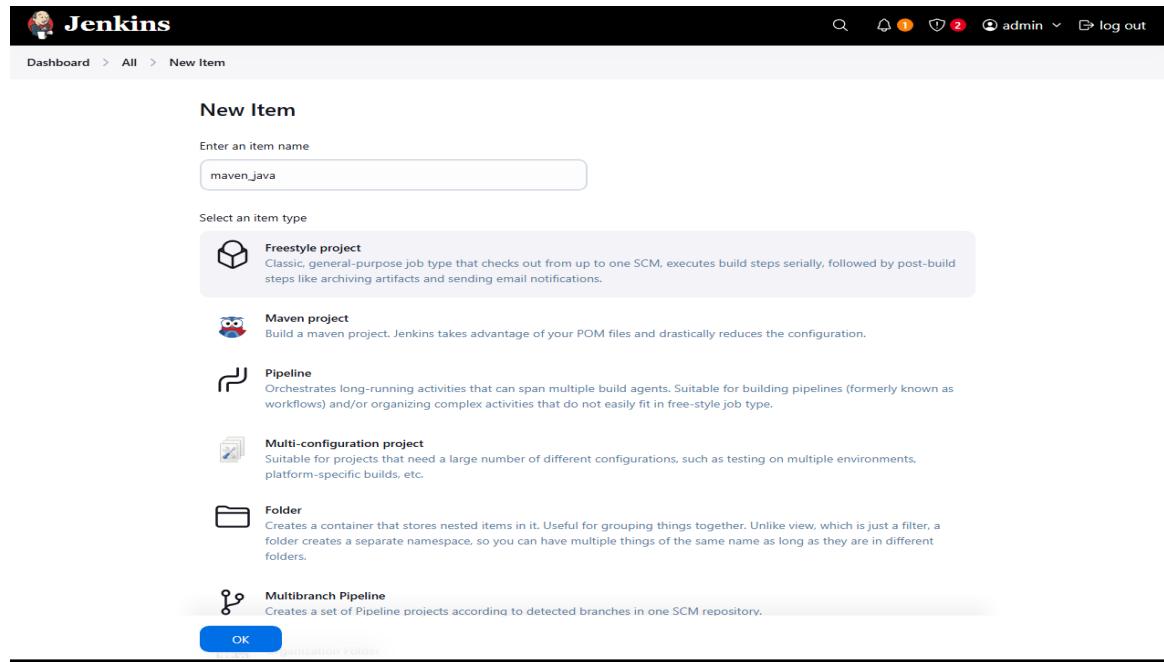
The screenshot shows a GitHub repository page for 'se-webapp-22'. The repository is public and has 1 branch and 0 tags. The 'About' section shows the repository was created by 'Edigirala-Neksha' and has 1 commit. The 'Activity' section shows 0 stars, 0 watching, and 0 forks. The 'Releases' section shows no releases published. The 'Languages' section shows no packages published.

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"



Step 2: Configuration of maven_java project

Give the description

The screenshot shows the Jenkins configuration page for a job named "Mavenjava". The "General" tab is selected. The "Description" field contains "Java Build demo". There are two checkboxes at the bottom: "Discard old builds" (unchecked) and "GitHub project" (unchecked). A blue "Enabled" switch is on the right.

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 Config [Jenkins]". The "Source Code Management" tab is selected. The "Git" option is selected. The "Repository URL" field contains "https://github.com/SarvikaSomishetty/eclipse-maven-projects.git". The "Credentials" dropdown is set to "- none -".

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 a job named "Mavenjava". The left sidebar lists various configuration sections: General, Source Code Management, Triggers, Environment, Build Steps (which is selected and highlighted in grey), and Post-build Actions. The main content area is titled "Configure" and contains several configuration options. At the top, there are checkboxes for "Add timestamps to the Console Output", "Inspect build log for published build scans", "Terminate a build if it's stuck", and "With Ant". Below this is a section titled "Build Steps" with the sub-instruction "Automate your build process with ordered tasks like code compilation, testing, and deployment." Two "Invoke top-level Maven targets" steps are listed. Both steps have "MAVEN_HOME" selected in the "Maven Version" dropdown and "clean" in the "Goals" input field. There is an "Advanced" button with a dropdown arrow next to the goals field. At the bottom of the build steps section 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 left sidebar lists configuration sections: General, Source Code Management, Triggers, Environment, Build Steps (which is selected), and Post-build Actions. The main area displays a 'Build Steps' section with a 'Goals' field containing 'clean' and an 'Advanced' dropdown. Below this is a 'Post-build Actions' section with a note about defining actions like notifications or artifact archiving. At the bottom are 'Save' and 'Apply' buttons.

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 configuration page for the 'Mavenjava' job. The 'Post-build Actions' section is expanded, displaying two actions:

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

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

If the build is success:

The screenshot shows a web browser window with the Jenkins interface. The title bar indicates the current tab is 'maven_web_build [Jenkins]'. The main content area shows the 'maven_web_build' project details:

- Status:** Green checkmark icon, indicating the build is successful.
- Changes:** Shows 'web build demo'.
- Workspace:** Shows a folder icon.
- Build Now:** A button to trigger a build.
- Configure:** A gear icon.
- Delete Project:** A trash bin icon.
- Rename:** A pencil icon.

Last Successful Artifacts: Shows a cube icon and a link to 'maven_web_test'.

Downstream Projects: Shows a green checkmark icon and a link to 'maven_web_test'.

Permalinks: A list of recent builds:

- Last build (#2), 27 min ago
- Last stable build (#2), 27 min ago
- Last successful build (#2), 27 min ago
- Last completed build (#2), 27 min ago

Builds: A table showing two builds:

Build #	Timestamp
#2	11:43 AM
#1	11:42 AM

Taskbar: Shows the Windows Start button, a search bar with 'Type here to search', and various pinned application icons (File Explorer, Edge, File History, Task View, Google Chrome, Firefox, FileZilla). On the right, it shows system status: battery level, signal strength, volume, language (ENG), date (07-10-2025), and time (12:11).

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

New Item

Enter an item name
maven_java_test

Select an item type

- Freestyle project**
Classic general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Maven project
Build a maven project. Jenkins takes advantage of your POM files 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 organizing complex activities that do not easily fit in 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 view, 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 4: Configuration of maven_java project

Give the description

MavenJava_Test Config [Jenkins]

localhost:8888/job/MavenJava_Test/configure

Verify it's you

Configure General

Enabled

General

Description
Test demo

Plain text Preview

Discard old builds ?

GitHub project

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', 'None' is selected. Under 'Environment', the checkbox 'Delete workspace before build starts' is checked. Other environment options like 'Use secret text(s) or file(s)', 'Provide Configuration files', 'Add timestamps to the Console Output', and 'Inspect build log for published build scans' are unchecked. At the bottom, there are 'Save' and 'Apply' buttons.

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', a 'Copy artifacts from another project' step is added. The 'Project name' is set to 'Mavenjava'. The 'Which build' dropdown is set to 'Latest successful build' with the 'Stable build only' checkbox checked. The 'Artifacts to copy' field contains '**/*'. The 'Target directory' field is empty. There are checkboxes for 'Flatten directories', 'Optional', 'Fingerprint Artifacts', and 'Include Build Number', with 'Fingerprint Artifacts' checked. At the bottom, there are 'Save' and 'Apply' buttons.

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

Click on apply and save

The screenshot shows the Jenkins configuration page for the 'MavenJava_Test' job. In the 'Post-build Actions' section, there is a step titled 'Archive the artifacts' with the pattern '**/*' entered. Below the configuration area, there are 'Save' and 'Apply' buttons.

In the dashboard you will find MavenJava and MavenJava_Test

The screenshot shows the Jenkins dashboard with several jobs listed in a table:

S	W	Name	Last Success	Last Failure	Last Duration
🔴	☁️	INTERNAL_JAVA	9 mo 3 days #34	40 sec #15454	0.67 sec
🟢	☀️	Mavenjava	13 days #2	N/A	11 sec
🟢	☀️	MavenJava_Test	13 days #3	N/A	3.4 sec
🔴	☁️	new	9 mo 3 days #3	13 days #4	31 sec
🟢	☀️	web_build	9 mo 9 days #8	N/A	8.2 sec
🔴	☁️	web_deploy	N/A	9 mo 9 days #15	0.31 sec
🟢	☀️	web_test	9 mo 9 days #12	N/A	3.4 sec

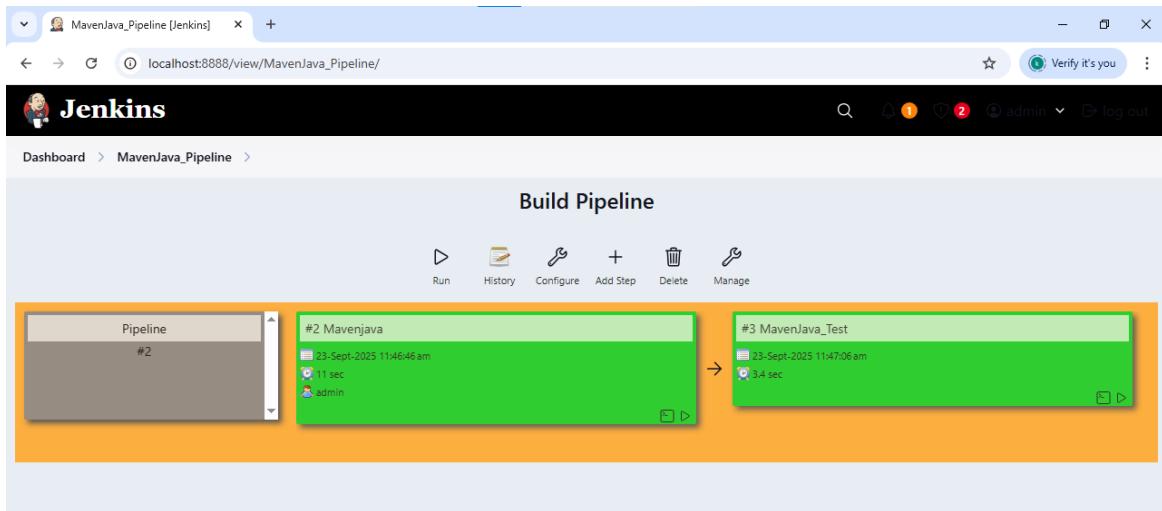
If you open the MavenJava file the following will be shown in case on no errors

The screenshot shows the Jenkins interface for the 'Mavenjava' job. The top navigation bar includes links for 'Dashboard', 'Mavenjava', 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', and 'Rename'. The main content area is titled 'Mavenjava' with a green checkmark icon. It displays the 'Java Build demo' configuration. On the left, there's a sidebar with 'Builds' (listing #2 from September 23, 2025, at 11:46 AM and #1 at 11:45 AM) and 'Permalinks' (link to last build). The right side shows 'Last Successful Artifacts' with a list of files and their sizes, such as .classpath (1.65 KiB), .project (1.06 KiB), and Dockerfile (131 B). Below this is a section for 'Downstream Projects' with a link to 'MavenJava_Test'. A 'Permalinks' section at the bottom lists the last build (#2) and last stable build (#2).

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' job. The top navigation bar includes links for 'Dashboard', 'MavenJava_Test', 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', and 'Rename'. The main content area is titled 'MavenJava_Test' with a green checkmark icon. It displays the 'Test demo' configuration. On the left, there's a sidebar with 'Builds' (listing #3 from September 23, 2025, at 11:47 AM, #2 at 11:46 AM, and #1 at 11:45 AM) and 'Permalinks' (link to 'Mavenjava'). The right side shows 'Last Successful Artifacts' with a list of files and their sizes, identical to the MavenJava job. Below this is a section for 'Upstream Projects' with a link to 'Mavenjava'. A 'Permalinks' section at the bottom lists the last build (#3), last stable build (#3), and last successful build (#3), all dated 13 days ago.

MavenJava_pipeline

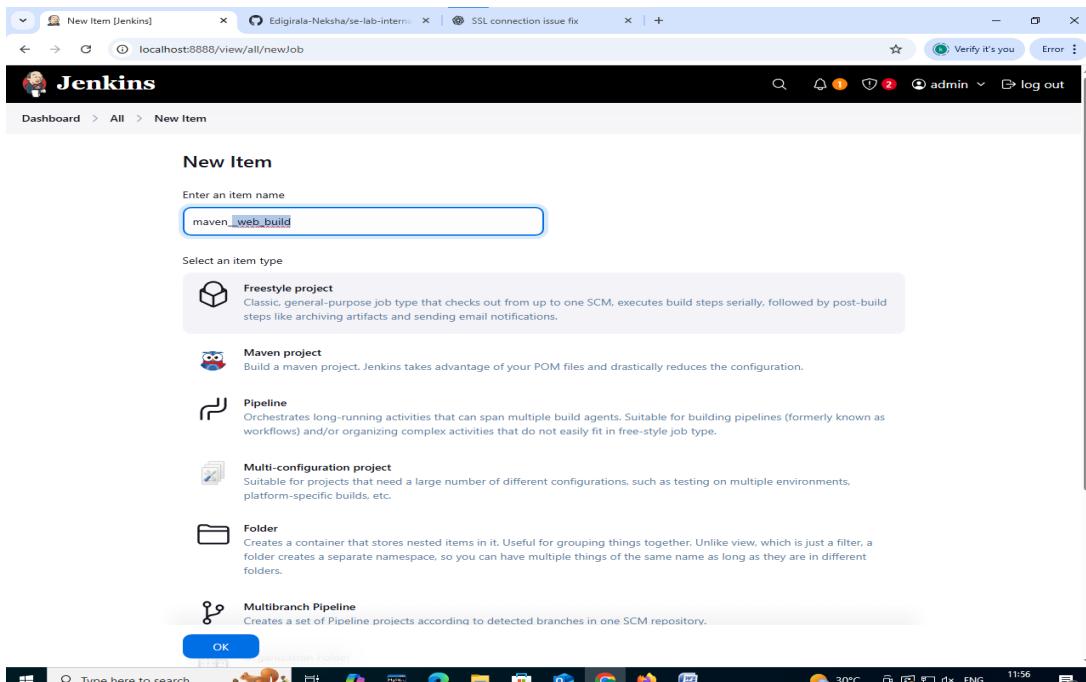


II. Maven Web Automation Steps:

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 'maven_web_build Config [Jenkins]', 'Edigirala-Neksha/se-lab-intern...', and 'SSL connection issue fix'. The main header displays the Jenkins logo, a search bar, and notification icons for 1 error and 2 warnings. The user is logged in as 'admin'. The page title is 'Configuration' under 'maven_web_build'.

Configure General

General tab selected. Description field contains 'web build demo'. Enabled toggle switch is checked. Other tabs include Source Code Management, Triggers, Environment, Build Steps, and Post-build Actions.

Source Code Management section:

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

Buttons: Save (blue), Apply, and Advanced dropdown.

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

The screenshot shows the Jenkins configuration interface for a job named "maven_web_build". The left sidebar lists several configuration sections: General, Source Code Management (selected), Triggers, Environment, Build Steps, and Post-build Actions. The main content area is titled "Source Code Management" and contains instructions to "Connect and manage your code repository to automatically pull the latest code for your builds." A radio button is selected for "Git", and the "Repositories" section is expanded. It shows a "Repository URL" input field containing "https://github.com/Edigirala-Neksha/se-lab-internal-1.git" and a "Credentials" dropdown set to "- none -". Below these are "Add Repository" and "Advanced" buttons. Another section, "Branches to build", is also expanded, showing a "Branch Specifier (blank for 'any')" input field containing "*/main" and an "Add Branch" button. At the bottom of the configuration area are "Save" and "Apply" buttons. The browser's address bar shows the URL "localhost:8888/job/maven_web_build/configure". The taskbar at the bottom of the screen includes icons for File Explorer, Edge, File, Mail, Google Chrome, and Firefox, along with system status indicators like battery level, temperature (30°C), and date/time (07-10-2025).

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

The screenshot shows the Jenkins configuration page for the 'maven_web_build' job. The left sidebar lists 'General', 'Source Code Management', 'Triggers', 'Environment', 'Build Steps' (which is selected), and 'Post-build Actions'. The main area is titled 'Build Steps' with the sub-instruction 'Automate your build process with ordered tasks like code compilation, testing, and deployment.' Below this, there are two 'Invoke top-level Maven targets' steps. The first step has 'MAVEN_HOME' selected in the 'Maven Version' dropdown and 'clean' in the 'Goals' input field. The second step also has 'MAVEN_HOME' selected in the 'Maven Version' dropdown and 'install' in the 'Goals' input field. At the bottom of the build steps section is a 'Save' button.

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 configuration page for the 'maven_web_build' job. The 'Post-build Actions' section is active. It contains two actions:

- Archive the artifacts**: Set to archive files matching the pattern '**/*'. An 'Advanced' dropdown is visible.
- Build other projects**: Set to build the project 'maven_web_test'. Trigger options include 'Trigger only if build is stable' (selected), 'Trigger even if the build is unstable', and 'Trigger even if the build fails'.

At the bottom, there are 'Save' and 'Apply' buttons. The status bar at the bottom right shows 'REST API' and '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"

New Item

Enter an item name
maven_web_test

Select an item type

- Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Maven project**
Build a maven project. Jenkins takes advantage of your POM files 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 organizing complex activities that do not easily fit in 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 view, 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_web_test project

Give the description

Configure

General

Enabled

Description
test demo

Plain text [Preview](#)

Discard old builds ?

GitHub project

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 'maven_web_test' job. The left sidebar lists 'General', 'Source Code Management' (selected), 'Triggers', 'Environment', 'Build Steps', and 'Post-build Actions'. The main content area has two sections: 'Source Code Management' and 'Environment'. In 'Source Code Management', 'None' is selected. In 'Environment', the checkbox 'Delete workspace before build starts' is checked. Other options like 'Advanced' and 'Temporary build if it's stuck' are also visible.

Configure

Source Code Management

Connect 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

Temporary build if it's stuck

Save Apply

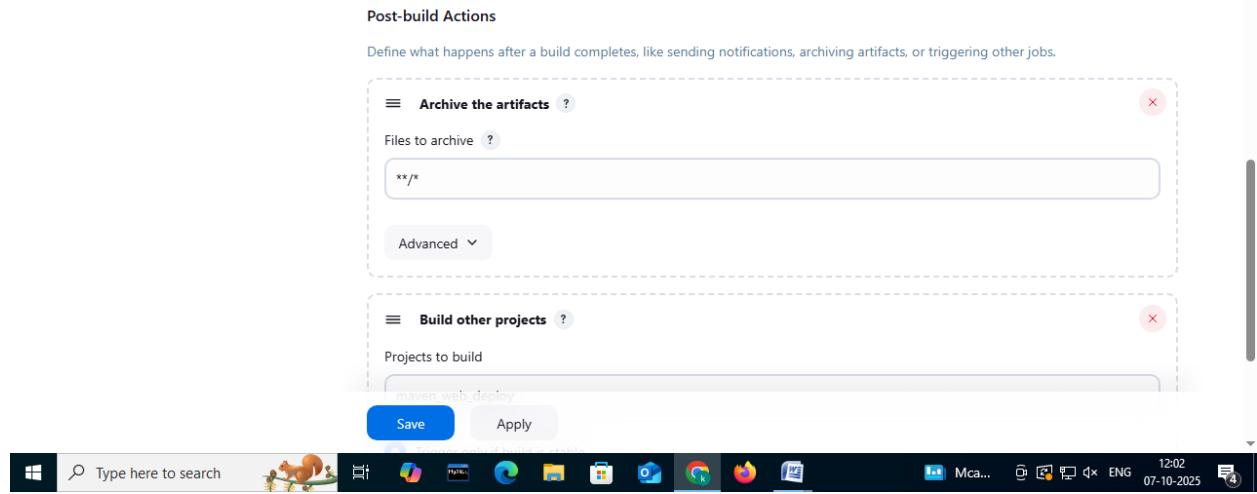
In the build steps click on add build step > select copy artifacts from another project > give project name as maven_web_build > give artifacts to copy as */*

The screenshot shows the Jenkins configuration interface for the 'maven_web_test' job. In the 'Build Steps' section, a 'Copy artifacts from another project' step is selected. The 'Project name' field contains 'maven_web_build'. The 'Which build' dropdown is set to 'Latest successful build' with the 'Stable build only' checkbox checked. The 'Artifacts to copy' field contains '**/*'. The 'Target directory' field is empty. There are also sections for 'Parameter filters' and checkboxes for 'Flatten directories', 'Optional', 'Fingerprint Artifacts', and 'Include Build Number'. At the bottom 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 test

The screenshot shows the Jenkins configuration interface for the 'maven_web_test' job. In the 'Post-build Actions' section, an 'Invoke top-level Maven targets' step is selected. The 'Maven Version' field contains 'MAVEN_HOME' and the 'Goals' field contains 'test'. An 'Advanced' dropdown is open. At the bottom are 'Add build step' and 'Post-build Actions' buttons.

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”



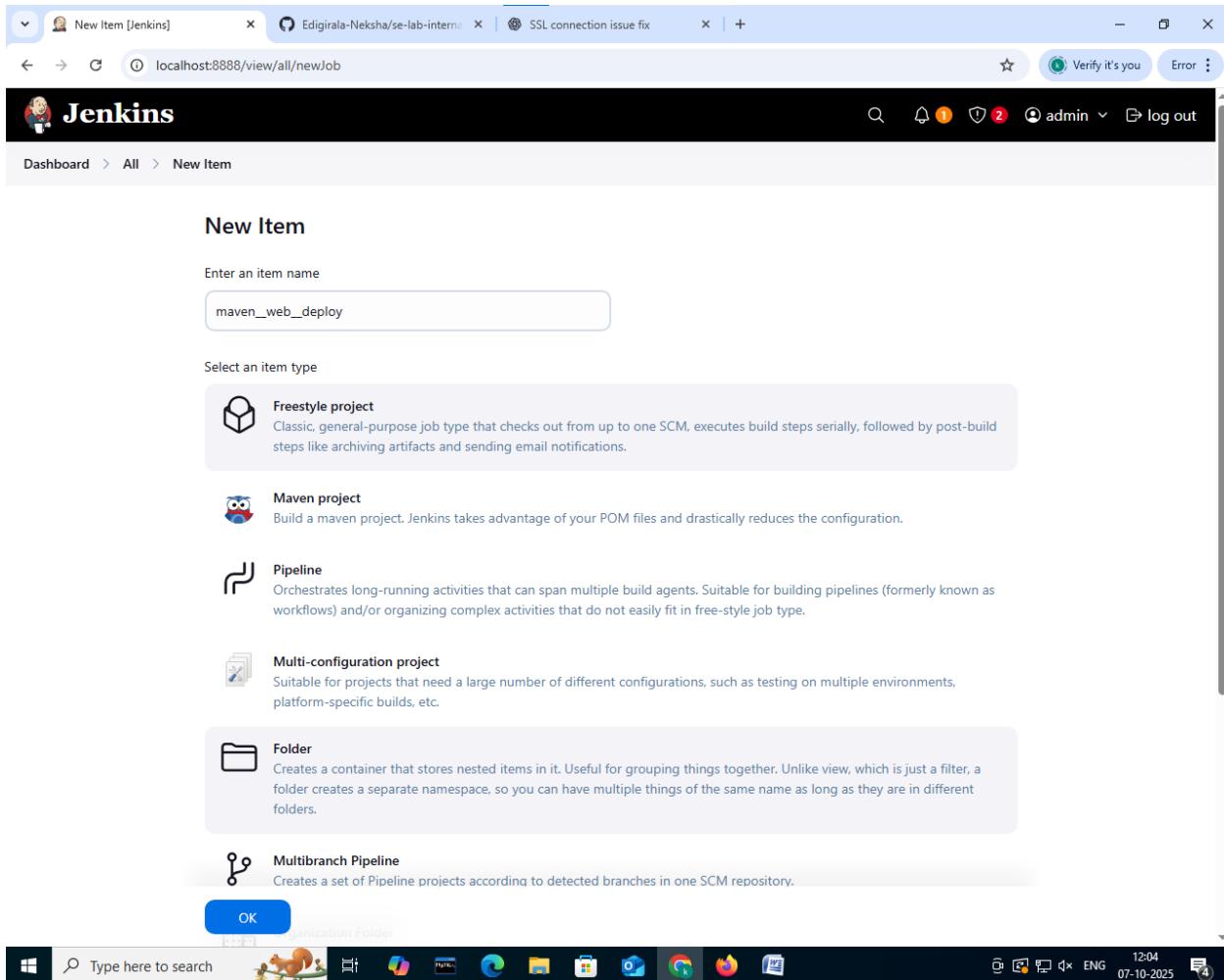
If the build is success:

The screenshot shows a Windows desktop environment. At the top, there is a taskbar with several pinned icons: Start, File Explorer, Task View, Microsoft Edge, File History, Google Chrome, Mozilla Firefox, and File Explorer again. To the right of the taskbar, it displays the date and time as 07-10-2025 12:37. In the center, a Microsoft Edge browser window is open, showing the Jenkins interface for a project named "maven_web_test". The browser's address bar shows the URL "localhost:8888/view/webpipeline/job/maven_web_test/". The Jenkins page displays various project details, including build history, upstream projects ("maven_web_build"), downstream projects ("maven_web_deploy"), and permalinks to recent builds.

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 page for a job named "maven_web_deploy". The "General" tab is selected. In the "Description" field, the text "deploy demo" is entered. The "Enabled" checkbox is checked. On the left sidebar, there are tabs for General, Source Code Management, Triggers, Environment, Build Steps, and Post-build Actions. Under "Source Code Management", there are several checkboxes: "Discard old builds", "GitHub project", "Permission to Copy Artifact", "This project is parameterized", "Throttle builds", and "Execute concurrent builds if necessary". A "Save" button is visible 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 left sidebar lists "General", "Source Code Management" (selected), "Triggers", "Environment", "Build Steps", and "Post-build Actions".

Source Code Management: Set to "None".

Triggers:

- 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

Terminate a build if it's stuck

Build Steps: (Not visible in the screenshot)

Bottom Bar: Includes a search bar, pinned icons for various tools (Windows, File Explorer, Edge, etc.), and system status indicators (NIFTY, ENG, 12:07, 07-10-2025).

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 the Jenkins configuration page for a job named "maven_web_deploy". The left sidebar lists several configuration sections: General, Source Code Management, Triggers, Environment, Build Steps (which is currently selected), and Post-build Actions. The main content area is titled "Build Steps" and contains a section for "Copy artifacts from another project". This section includes fields for "Project name" (set to "maven_web_test"), "Which build" (set to "Latest successful build" with a checkbox for "Stable build only"), "Artifacts to copy" (set to "**/*"), and "Artifacts not to copy" (empty). There are also fields for "Target directory" and "Parameter filters". At the bottom of this section are checkboxes for "Flatten directories", "Optional", "Fingerprint Artifacts" (which is checked), and "Include Build Number". Below this section are "Save" and "Apply" buttons. The status bar at the bottom of the browser window shows the date and time as 07-10-2025 12:07.

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 left sidebar lists various configuration sections: General, Source Code Management, Triggers, Environment, Build Steps, and Post-build Actions. The "Post-build Actions" section is currently selected and highlighted.

Post-build Actions

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

Deploy war/ear to a container

WAR/EAR files: `**/*.war`

Context path: `webpath`

Containers

Tomcat 9.x Remote

Credentials: `admin/*****`

+ Add

Tomcat URL: `https://localhost:8080/`

Advanced ▾

Save Apply

The Jenkins interface is running on a Windows 10 desktop, as indicated by the taskbar at the bottom which includes icons for File Explorer, Edge, and various system status indicators.

If the build is success:

The screenshot shows a Jenkins job page for 'maven_web_deploy'. The top navigation bar includes tabs for 'maven_web_deploy [jenkins]', 'Edigirala-Neksha/se-lab-intern...', 'Apache Tomcat/9.0.98', 'Jenkins support for Java 21', and a '+' button. The URL is 'localhost:8888/job/maven_web_deploy/'. The Jenkins logo is at the top left, and the user 'admin' is logged in.

The main content area displays the job name 'maven_web_deploy' with a green checkmark icon. Below it is the description 'deploy demo'. A sidebar on the left contains links for 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', and 'Rename'. The 'Builds' section lists the following builds:

Build	Timestamp
#13	12:36 PM
#12	12:35 PM
#11	12:31PM
#10	12:28PM
#9	12:20PM
#8	12:19PM
#7	12:12PM
#6	11:50AM
#5	11:50AM
#4	11:48 AM

Upstream Projects listed are 'maven_web_test' (green checkmark). Permalinks for each build are provided. The system status bar at the bottom shows a Windows taskbar with icons for File Explorer, Edge, and other applications, along with system information like temperature (30°C), battery level, and date/time (07-10-2025).

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 indicates the URL is `localhost:8888/newView`. The main area is titled "New view". On the left, there's a sidebar with links like "New Item", "Build History", "Project Relationship", "Check File Fingerprint", "Manage Jenkins", and "My Views". A dropdown menu for "Build Queue" shows "No builds in the queue". Another dropdown for "Build Executor Status" shows "0/2". The main form has a "Name" field containing "maven_web_pipeline" and a "Type" section where "Build Pipeline View" is selected (indicated by a blue circle). Below it, "List View" and "My View" are shown as options. At the bottom right of the form is a "Create" button. The footer of the page includes links for "REST API" and "Jenkins 2.489".

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 main area is titled 'Edit View' and contains several sections:

- Pipeline Flow**: Set to 'Based on upstream/downstream relationship'. A note explains this mode derives the pipeline structure based on trigger relationships.
- Upstream / downstream config**: A dropdown menu is open, showing 'Select Initial Job' with 'maven_web_build' selected.
- Trigger Options**: Buttons for 'Save' and 'Apply' are visible.

A sidebar on the left includes links for 'New Item', 'Build History', 'Edit View' (which is currently selected), 'Delete View', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. Below the sidebar are sections for 'Build Queue' (empty) and 'Build Executor Status' (0/2).

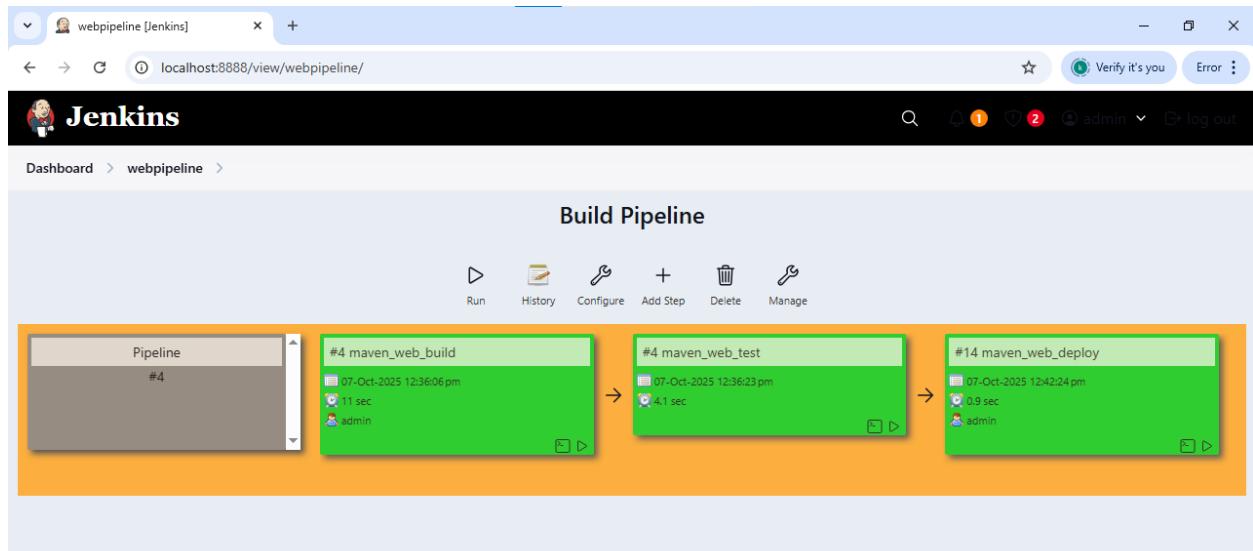
Click on apply and save

The screenshot shows the Jenkins 'Edit View' configuration page for a view named 'maven_web_pipeline'. The main area is titled 'Edit View' and contains several sections:

- Widgets**: A section for customizing widgets. It includes checkboxes for 'Filter build queue' (unchecked) and 'Filter build executors' (unchecked). A note states that checked boxes will filter the view.
- Buttons**: Buttons for 'Save' and 'Apply'.

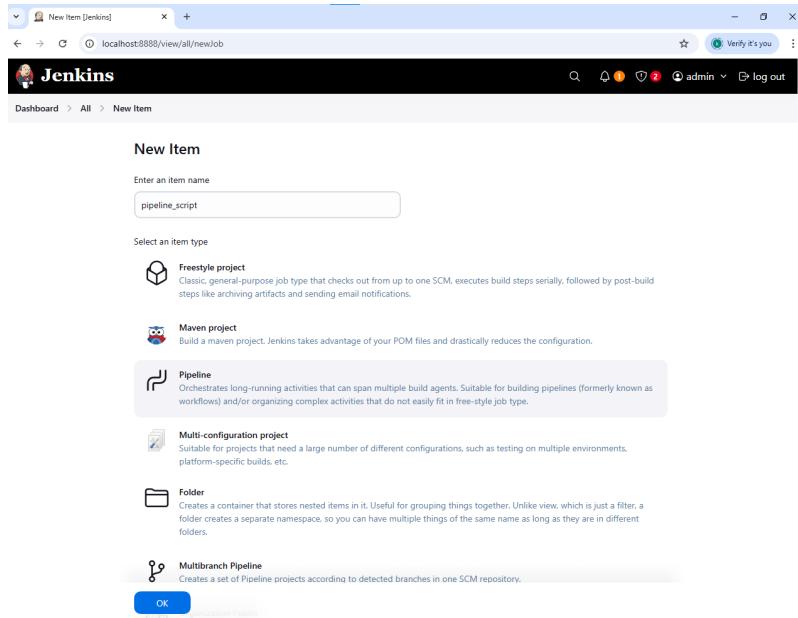
A sidebar on the left includes links for 'New Item', 'Build History', 'Edit View' (selected), 'Delete View', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. Below the sidebar are sections for 'Build Queue' (empty) and 'Build Executor Status' (0/2).

In the stage view it we be shown as:

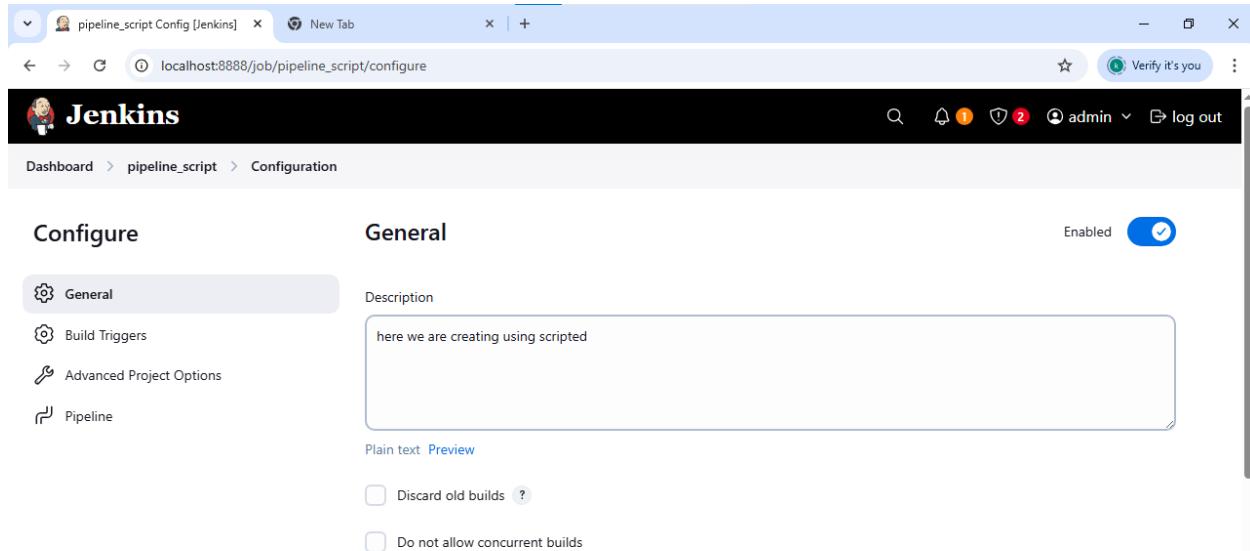


9.Pipeline Creation using script

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 "Build Triggers", several options like "Build after other projects are built", "Build periodically", and "GitHub hook trigger for GITScm polling" are listed. The "Advanced Project Options" section has an "Advanced" dropdown. The "Pipeline" section shows the "Definition" set to "Pipeline script". A code editor displays a Groovy script:

```
1 > pipeline {
2   agent any
3   tools
```

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

Step 4: click on apply and then save

The screenshot shows the Jenkins 'Configure' screen for a project named 'pipeline_script'. The 'Advanced Project Options' tab is selected. The 'Pipeline' section is expanded, showing a 'Definition' dropdown set to 'Pipeline script'. Below it is a code editor containing a Groovy pipeline script:

```
1 > pipeline {
2   >   agent any
3   >   tools{
4     >     maven 'MAVEN-HOME'
5   >
6   >   stages {
7     >     stage('git repo & clean') {
8       >       steps {
9         >         //bat "rmdir /s /q mavenjava"
10        >         bat "git clone https://github.com/SarvikaSomishetty/eclipse-maven-projects.git"
11        >         bat "mvn clean -f eclipse-maven-projects"
12      >
13    >   }
14   >   stage('install') {
15     >     steps {
16       >       bat "mvn install -f eclipse-maven-projects"
17     >
18   > }
```

Below the script, there is a checked checkbox for 'Use Groovy Sandbox'. At the bottom of the page, there are 'Save' and 'Apply' buttons. The top of the browser window shows the URL as 'localhost:8888/job/pipeline_script/configure'.

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 a user icon, the job name 'pipeline_script [Jenkins]', and a 'Verify it's you' button. The main content area has a dark header with the Jenkins logo and the job name. Below the header, a breadcrumb trail shows 'Dashboard > pipeline_script >'. The left sidebar contains links for Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Stages, Rename, and Pipeline Syntax. The right side features a 'Stage View' section with a table showing execution times for five stages: Declarative: Tool Install (296ms), git repo & clean (5s), install (9s), test (3s), and package (4s). A summary at the top of the table states 'Average stage times: (full run time: ~26s)'. Below the table, a box indicates 'Oct 07 11:02' and 'No Changes'. The bottom section displays a 'Builds' table with one entry: '#2 11:02 AM'. The 'Permalinks' section lists four recent builds.

Declarative: Tool Install	git repo & clean	install	test	package
296ms	5s	9s	3s	4s
Oct 07 11:02	No Changes			
296ms	5s	9s	3s	4s

Permalinks

- Last build (#2), 4 min 29 sec ago
- Last stable build (#2), 4 min 29 sec ago
- Last successful build (#2), 4 min 29 sec ago
- Last completed build (#2), 4 min 29 sec ago

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.

```
  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:
  Type  Reason  Age   From          Message
  ----  -----  --   --   -----
  Normal Scheduled  68s  default-scheduler  Successfully assigned default/mynginx-79bb8756c7-wpslj to minikube
  Normal Pulled   67s  kubelet        Pulling image "nginx"
  Normal Pulled   65s  kubelet        Successfully pulled image "nginx" in 2.416s (2.416s including waiting). Image size: 159974475 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 [:1]: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:
```

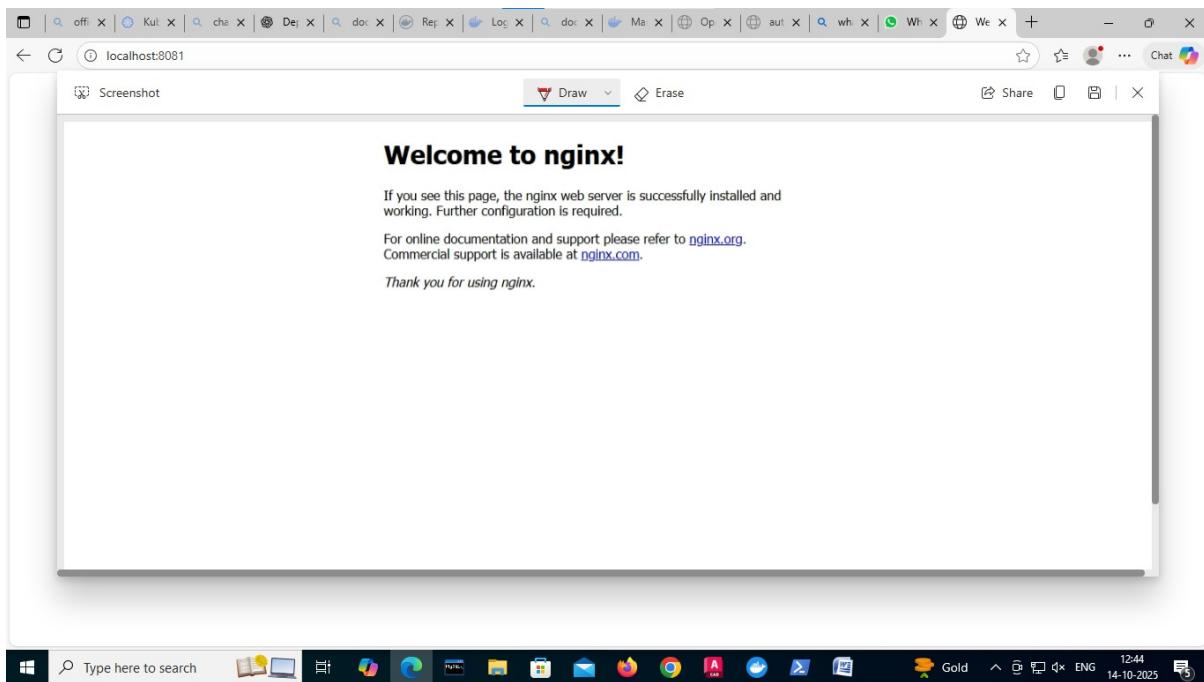
Step-6:Check the NGINX Service Details

- 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:91ee: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 new WAF actions → https://ngrok.com/r/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      rt1      rt5      p50      p90
                        2       0       0.00    0.00    30.28    30.47

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', the 'Git' option is selected. The 'Repository URL' field contains 'https://github.com/Edigirala-Neksha/se-lab-internal-1.git'. The 'Branches to build' field contains '/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 after other projects are built' are available but not selected. Buttons for 'Save' and 'Apply' are 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 for 'se-lab-internal-1'. The 'General' tab is selected. In the left sidebar, 'Webhooks' is highlighted. The main area shows the repository name 'se-lab-internal-1' and a section for 'Require contributors to sign off on web-based commits'. Below that is the 'Default branch' section, which currently has 'main' selected. The 'Releases' section is also visible.

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>

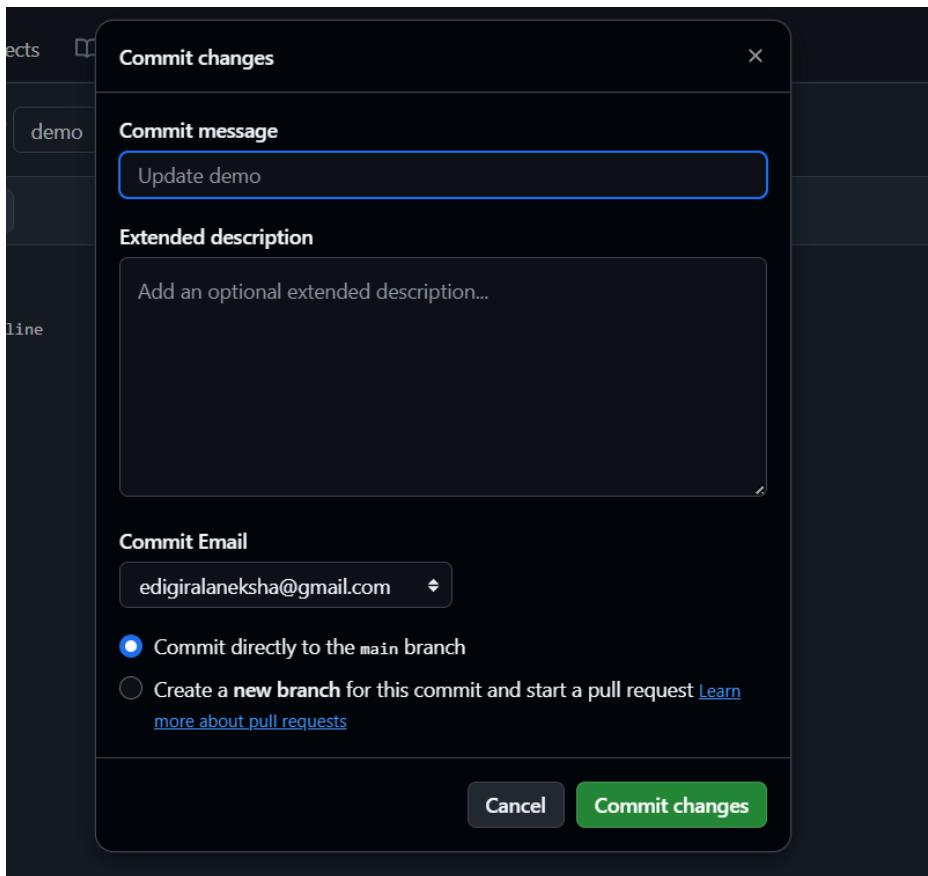
Payload url: <https://corkier-darla-handsome.ngrok-free.dev/github-webhook/>

The screenshot shows the 'Webhooks / Add webhook' form. The 'Payload URL' field contains 'https://corkier-darla-handsome.ngrok-free.dev/github-webhook/'. The 'Content type' dropdown is set to 'application/x-www-form-urlencoded'. The 'SSL verification' section has 'Enable SSL verification' selected. The 'Which events would you like to trigger this webhook?' section has 'Just the push event.' selected.

Step 8: make changes in the files in github

A screenshot of a GitHub repository named "se-lab-internal-1". The "Code" tab is selected. In the center, there's a file editor for a file named "demo" in the "main" branch. The code contains three lines: "demooooooo", "webhook", and "xxxxxx-new line!". Above the code editor, there are buttons for "Cancel changes" and "Commit changes...". On the left, there's a sidebar with a tree view of the repository's structure, including ".settings", "src", "target", ".classpath", ".project", "Dockerfile", "demo", and "pom.xml". A status bar at the bottom indicates keyboard navigation instructions.

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 the Jenkins logo and the project name 'job_webhook_java'. A sidebar on the left lists various project management options: Status (highlighted), Changes, Workspace, Build Now, Configure, Delete Project, GitHub Hook Log, Rename, and Credentials. The main content area is titled 'Permalinks' and lists four recent builds. Below this is a 'Builds' section with a 'Pending' row for build #3 and a 'Today' row for build #2 (11:36 am). A 'Filter' input field is also present.

- 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
- Last completed build (#2), 3 min 46 sec ago

Build	Timestamp
#3	In the quiet period. Expires in 2.9 sec
#2	11:36 am

The screenshot shows the Jenkins interface for the job `job_webhook_java`. On the left, a sidebar lists various project management options: Status, 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 displays three recent builds: #3 (11:41 am), #2 (11:36 am), and #1 (11:35 am), each indicated by a green checkmark.

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 a back link to Previous Build. The main content area shows the build was started by GitHub push by Edigirala-Neksha at 28-Oct-2025, 11:41:02 am. It includes a summary of the run spent and a git log entry for revision bc52c46b2c311be243984889d4970f7839687de from the repository https://github.com/Edigirala-Neksha/se-lab-internal-1.git. The git log also lists changes such as 'Update demo'.

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 'Plugins' page. A search bar at the top contains the text 'email'. Below it, a table lists several plugins:

- Email Extension**: Version 1925.v1598902b_58dd. Status: 100, Enabled, with a red 'X' icon.
- Email Extension Template Plugin**: Version 233.v1eb_88fc160b_5. Status: 100, Enabled, with a red 'X' icon.
- Mailer Plugin**: Version 522.va_995fa_cfb_8b_d. Status: 100, Enabled, with a red 'X' icon.

A message box in the center says: "Failed to load: Pipeline (workflow-aggregator 608.v67378e9d3db_1) - Failed to load: Pipeline: Basic Steps (workflow-basic-steps 1098.v808b_fd7f8cf4)". A toggle switch next to the message is turned off.

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

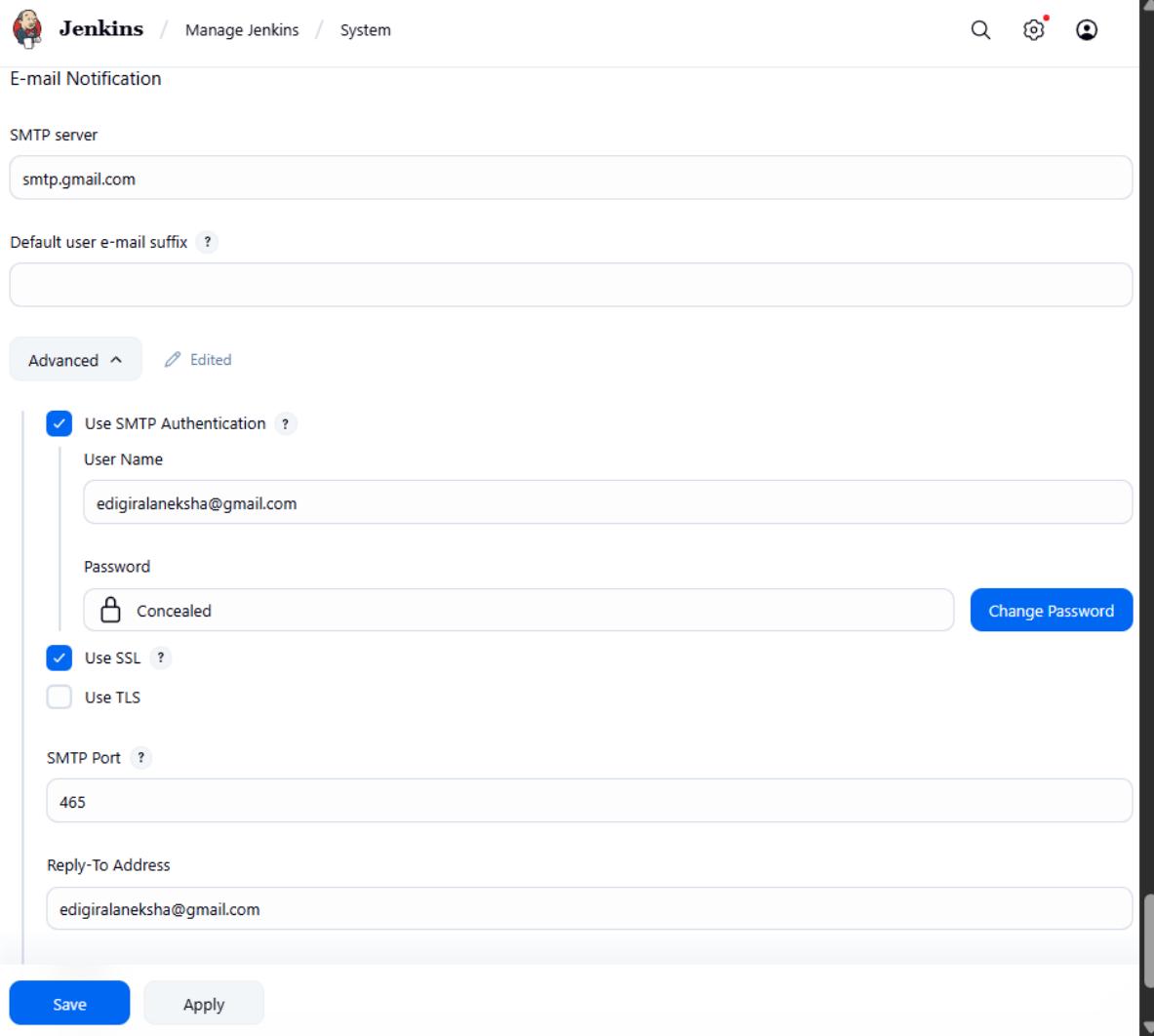
Use SSL ?

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
Default Recipients	Leave empty or provide default emails
Triggers	Select as per needs (e.g., Failure)

Extended E-mail Notification

SMTP server
smtp.gmail.com

SMTP Port
465

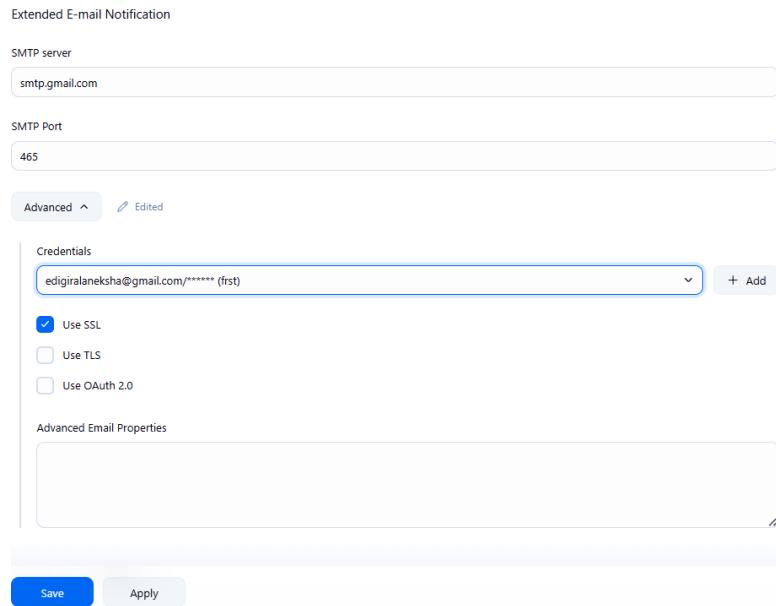
Advanced ^ Edited

Credentials
edigiralaneksha@gmail.com/***** (frst) + Add

Use SSL
 Use TLS
 Use OAuth 2.0

Advanced Email Properties

Save Apply

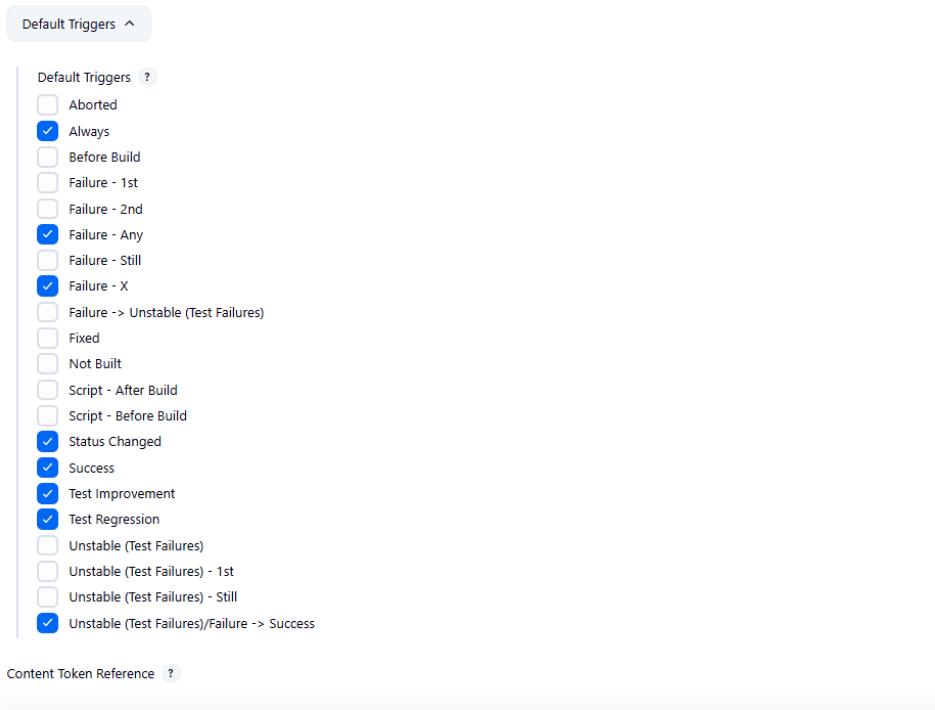


Default Triggers ^

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".

General Section:

- Description:** java webhook
- Enabled:**
- Discard old builds:**
- GitHub project:**
- Notify when Job configuration changes:**
- This project is parameterized:**
- Throttle builds:**
- Execute concurrent builds if necessary:**
- Advanced:**

Source Code Management:

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

- None:**
- Git:**

Buttons: Save Apply

ii. In the Post-build Actions section:

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

A. Fill in the fields:

Field	Value
Project Recipient List	Add recipient email addresses (comma-separated)
Content Type	Default (text/plain) or text/html
Triggers	Select events (e.g., Failure, Success, etc.)
Attachments	(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** ? X

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.

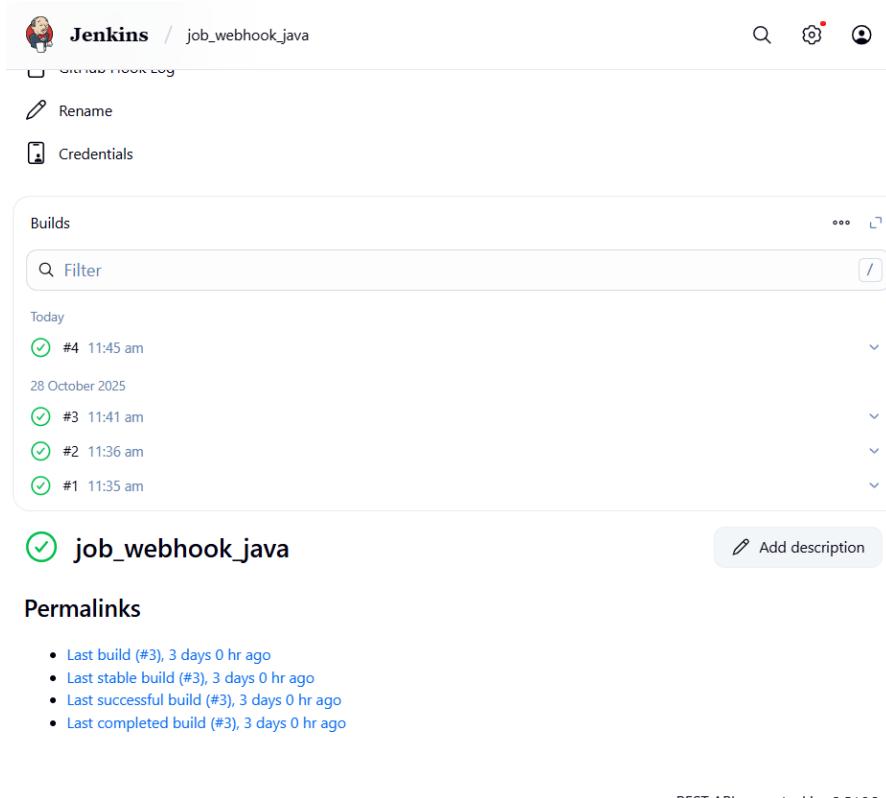
edigiralaneksha@gmail.com,nekshasri99@gmail.com

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**



The screenshot shows the Jenkins interface for a job named "job_webhook_java".

Job Information:

- Icon: Jenkins logo
- Name: Jenkins / job_webhook_java
- Build Log: Shows "jenkins.log" and "jenkins.log.1" with a "View Log" link.
- Rename: Option to rename the job.
- Credentials: Option to manage credentials.

Build History:

- Builds: A table showing build history from today to October 28, 2025.
- Today:
 - #4 11:45 am (green circle)
 - #3 11:41 am (green circle)
 - #2 11:36 am (green circle)
 - #1 11:35 am (green circle)
- 28 October 2025:
 - #4 11:45 am (green circle)
 - #3 11:41 am (green circle)
 - #2 11:36 am (green circle)
 - #1 11:35 am (green circle)

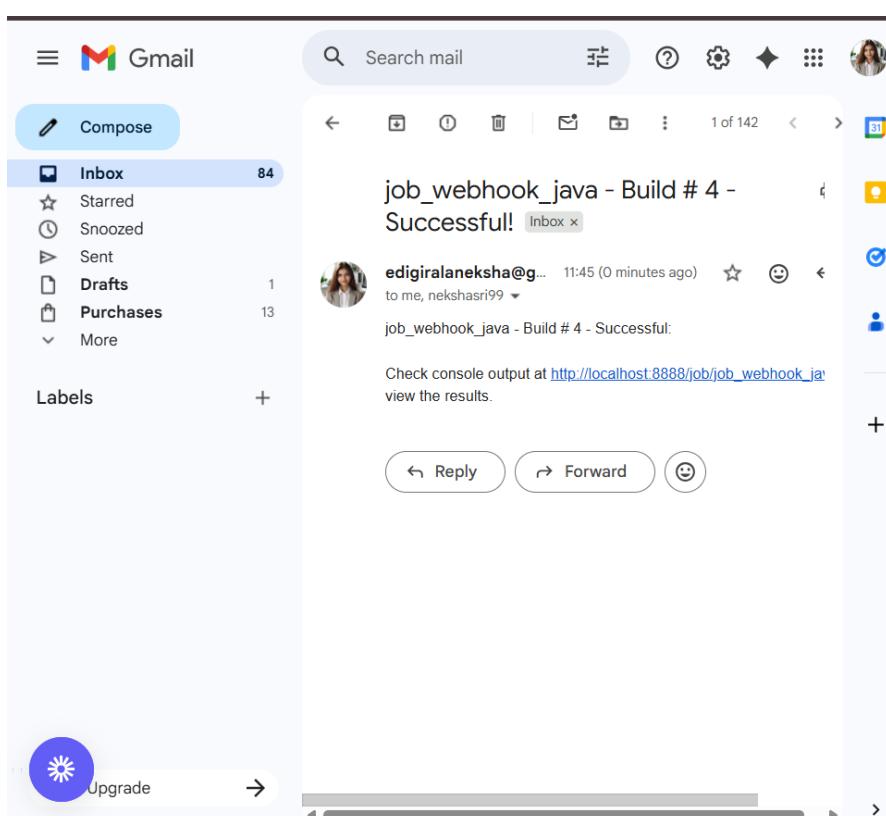
Job Details:

- Job Name: job_webhook_java
- Add description: Option to add a 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



The screenshot shows a Gmail inbox with one unread email from "edigiralaneksha@gmail.com" regarding a Jenkins build.

Inbox Summary:

- Compose button
- Inbox (84 messages)
- Starred (0)
- Snoozed (0)
- Sent (0)
- Drafts (1 message)
- Purchases (13 messages)
- More

Email Preview:

job_webhook_java - Build # 4 - Successful!

edigiralaneksha@gmail.com 11:45 (0 minutes ago) to me, nekshasri99

Check console output at http://localhost:8888/job/job_webhook_java view the results.

Bottom Navigation:

- Reply button
- Forward button
- Smiley face button

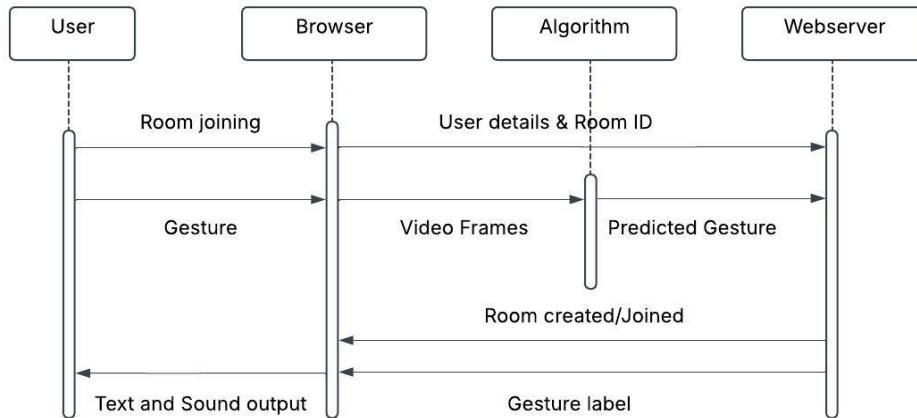
TUNEORA – A Music Web

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.

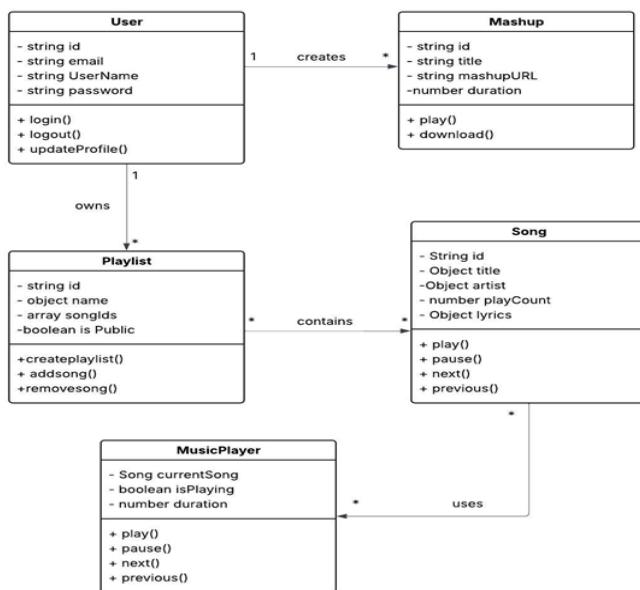
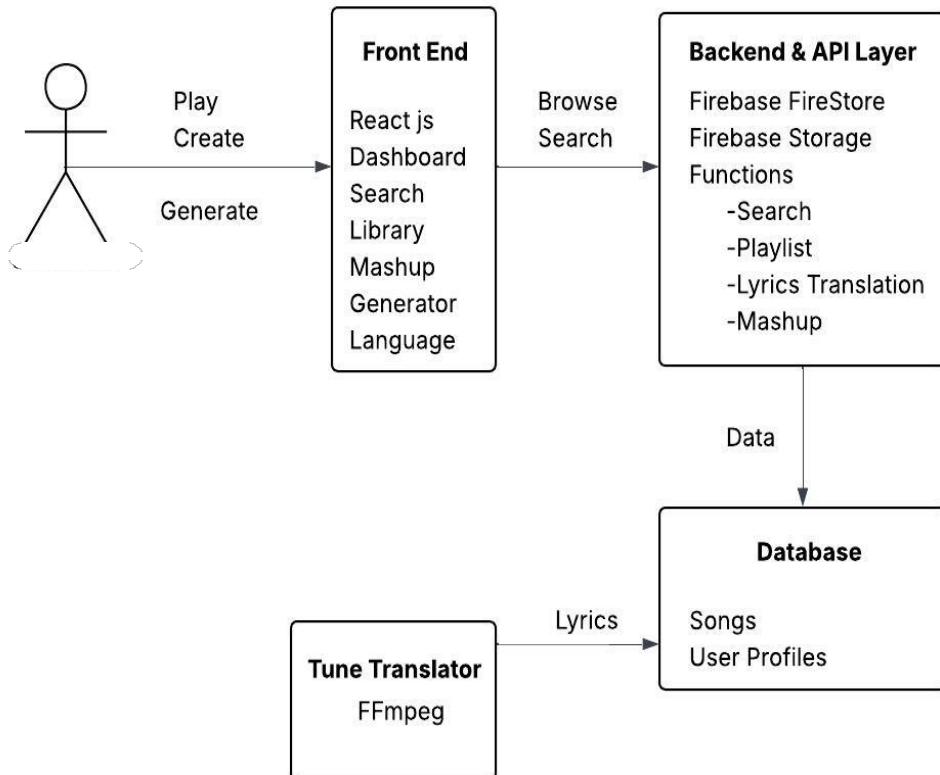


Fig 5: Class Diagram for TuneOra

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, Inbox, History, and Help. The main area has a breadcrumb navigation: ALLv2EN-US... > Modules > AWS Academ... > Launch AWS Academy Learner Lab. At the top, it says "Used \$0 of \$50" and has buttons for Start Lab, End Lab, AWS Details, Readme, and Reset. A dropdown menu shows "EN-US". The central part is titled "Learner Lab" and contains a "Environment Overview" section with links to 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. Below this is a large blue V-shaped graphic with orange arrows at the top.

Step 2: Scroll down and select Launch AWS Academy Lab

The screenshot shows the AWS Academy module page. The sidebar includes Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main content area has a "Home" tab selected. Under "AWS Academy Learner Lab", there's a link labeled "Launch AWS Academy Learner Lab". Other sections include "AWS Academy Learner Lab Compliance and Security" (with a "Complete All Items" button) and "Module Knowledge Check" (worth 100 pts). The bottom of the page shows a URL: <https://awsacademy.instructure.com/courses/141967/modules/items/13720316>.

Step 3: click on start lab

The screenshot shows the AWS Academy Learner Lab interface. On the left is a sidebar with icons for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main area has a top navigation bar with 'AWS' (red dot), 'Used \$0 of \$50', '00:00', and buttons for 'Start Lab', 'End Lab', 'AWS Details', 'Readme', 'Reset', and a close button. Below the navigation is a terminal window showing a command prompt: 'eee_W_5353255@runweb195092:~\$'. To the right is a 'Learner Lab' panel with a dropdown menu set to 'EN-US'. The 'Learner Lab' section contains 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.

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

This screenshot is identical to the one above, showing the AWS Academy Learner Lab interface. The 'AWS' service is selected in the top navigation bar. The terminal window shows a command prompt: 'eee_W_3940257@runweb155453:~\$'. The 'Learner Lab' panel on the right is visible with its list of links.

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

The screenshot shows the AWS EC2 Dashboard in the N. Virginia region. The left sidebar includes sections for Instances, Images, and Elastic Block Store. The main area features a 'Resources' summary and a 'Launch instance' section. The 'Launch instance' section contains a 'Launch instance' button and a note stating 'Note: Your instances will launch in the US East (N. Virginia) Region'. To the right, there's a 'Service health' section showing 'AWS Health Dashboard' and a status message 'This service is operating normally.', and an 'Account attributes' section with various settings like 'Default VPC' and 'Settings'.

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

The screenshot shows the 'Launch an instance' wizard in the N. Virginia region. The top navigation bar includes tabs for 'Launch AWS Academy Learner', 'Launch an instance | EC2 | us-east-1', and 'Two-factor authentication - Git'. The main form has a 'Name and tags' section where 'Name' is set to 'week-122'. Below it is an 'Application and OS Images (Amazon Machine Image)' section with a search bar and a 'Quick Start' tab selected, showing options like Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. On the right, a 'Summary' section shows 'Number of instances' set to 1, and a large orange 'Launch instance' button at the bottom right.

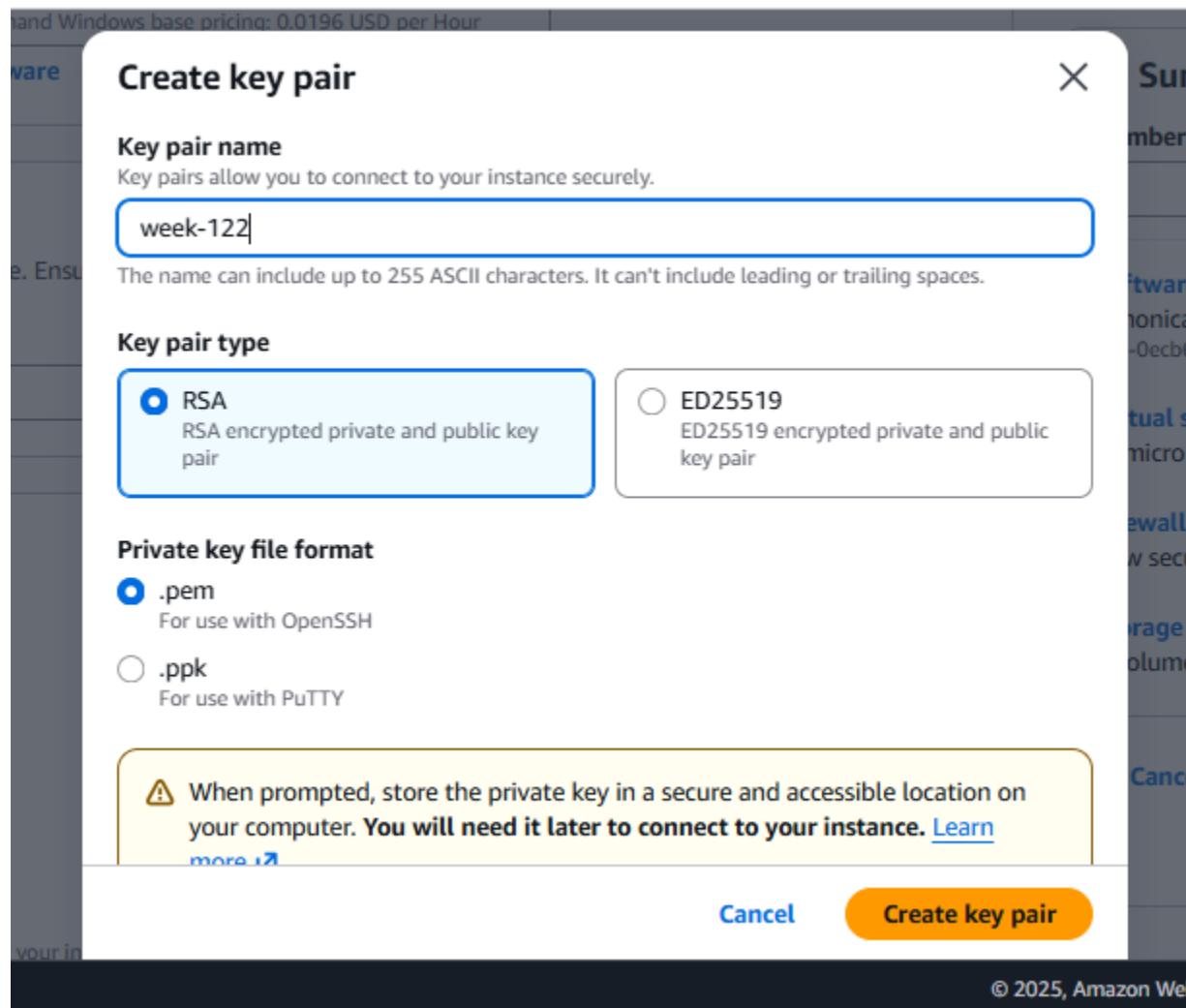
Step 6: Select the ubuntu server

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the left sidebar, under 'Quick Start', the 'Ubuntu' option is selected. Below it, the 'Amazon Machine Image (AMI)' section displays the 'Ubuntu Server 24.04 LTS (HVM), SSD Volume Type' AMI (ami-0ecb62995f68bb549). The 'Description' panel states: 'Ubuntu Server 24.04 LTS (HVM). EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).'. The 'Architecture' dropdown is set to '64-bit (x86)'. On the right, the 'Summary' panel shows the instance configuration: 1 instance, AMI (Canonical, Ubuntu, 24.04, amd64 noble image), Virtual server type (t3.micro), Firewall (New security group), Storage (1 volume(s) - 8 GiB), and a 'Launch instance' button.

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

The screenshot shows the AWS EC2 'Launch an instance' wizard. The 'Architecture' dropdown is now set to '64-bit (x86)'. The 'Instance type' section shows the 't3.micro' option selected, with details: Family: t3, 2 vCPU, 1 GiB Memory, Current generation: true, On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour, On-Demand SUSE base pricing: 0.0104 USD per Hour, On-Demand Linux base pricing: 0.0104 USD per Hour, On-Demand RHEL base pricing: 0.0392 USD per Hour, On-Demand Windows base pricing: 0.0196 USD per Hour. The 'Additional costs apply for AMIs with pre-installed software' note is visible. The 'Key pair (login)' section indicates that a key pair is required for secure connection. The 'Summary' panel remains the same as in Step 6, showing 1 instance of the selected AMI, t3.micro instance type, and other configurations.

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) ▾ [Create new VPC](#)

Subnet | [Info](#)

No preference ▾ [Create new subnet](#)

Availability Zone | [Info](#)

No preference ▾ [Enable additional zones](#)

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

ssh TCP 22

Remove

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

Anywhere Add CIDR, prefix list or security group e.g. SSH for admin desktop

0.0.0.0/0 X

⚠ 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. [X](#)

Add security group rule

▼ Summary

Number of instances | [Info](#)

1

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd64... [read more](#)

ami-0ebc62995f69bb549

Virtual server type (instance type)

t3.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

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

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

The screenshot shows the AWS EC2 'Launch an instance' configuration interface. In the 'Configure storage' section, a root volume of 8 GiB (gp3) is selected. The 'Number of instances' dropdown is set to 2. The 'Launch instance' button is visible at the bottom right.

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

The screenshot shows a progress bar indicating the instance is launching. The status message reads: 'Please wait while we launch your instance. Do not close your browser while this is loading.' A progress bar shows 33% completion.

The screenshot shows the AWS EC2 Instances Launch an instance page. At the top, there is a green success message: "Successfully initiated launch of Instances (i-0f868f6d463f89656, i-0a5aa6fe5d0039e34)". Below this, there is a "Launch log" link. The main area is titled "Next Steps" with a sub-section "Q. What would you like to do next with these instances, for example "create alarm" or "create backup"?" followed by a numbered list of six options:

- Create billing usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing usage thresholds. [Create billing alerts](#)
- Connect to your instance**: Once your instance is running, log into it from your local computer. [Learn more](#)
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. [Connect an RDS database](#), [Create a new RDS database](#), [Learn more](#)
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. [Create EBS snapshot policy](#)
- Manage detailed monitoring**: Enable or disable detailed monitoring for the instance. [Learn more](#)
- Create Load Balancer**: Create an application, network gateway or classic Elastic Load Balancer. [Create Load Balancer](#)
- Create AWS budget**: AWS Budgets allows you to create budgets, forecast. [Create AWS budget](#)
- Manage CloudWatch alarms**: Create or update Amazon CloudWatch alarms for the instance. [Create CloudWatch alarm](#)

At the bottom of the page, there are links for "CloudShell", "Feedback", and copyright information: "© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

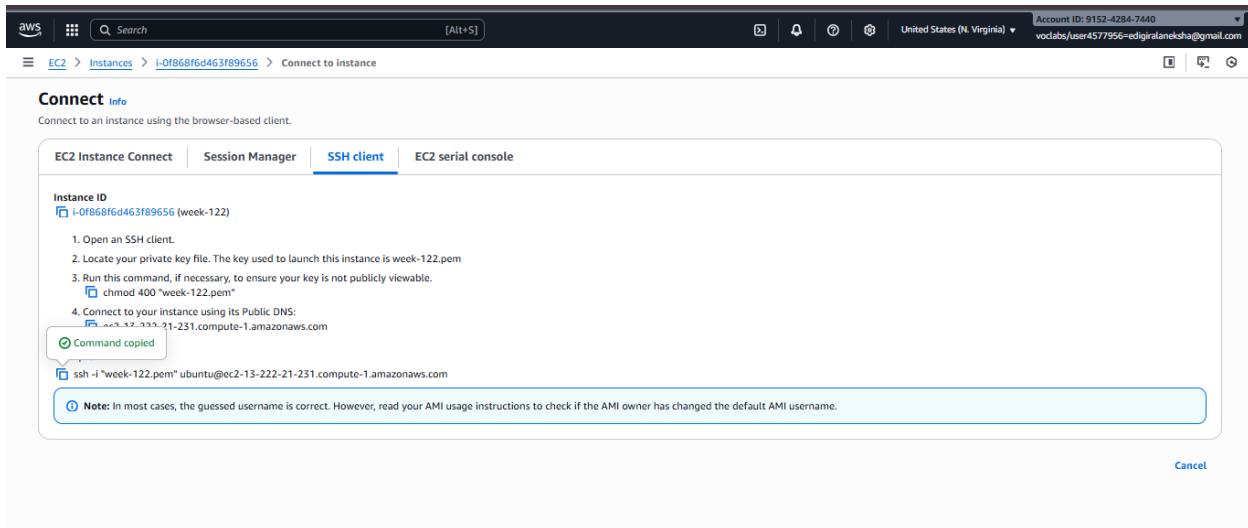
Step 13: In the instances the created ones will be shown, you can also rename the instance , changed week-1222 to “webapp”

The screenshot shows the AWS EC2 Instances page. On the left, there is a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, Images, AMIs, AMI Catalog, and Elastic Block Store. The main area shows a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
week-12	i-0d69dc97b6767aaf3	Terminated	t3.micro	-	View alarms	us-east-1a	-	-
week-122	i-0a5aa6fe5d0039e34	Running	t3.micro	Initializing	View alarms	us-east-1c	ec2-13-220-246-239.co... 13.220	
webapp	i-0f868f6d463f89656	Running	t3.micro	Initializing	View alarms	us-east-1c	ec2-13-222-21-231.co... 13.222	

Below the table, there is a detailed view for the instance "i-0f868f6d463f89656 (webapp)". It includes tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under the Details tab, there is an "Instance summary" section with fields for Instance ID (i-0f868f6d463f89656), Public IPv4 address (13.222.21.231), Private IPv4 addresses (172.31.9.83), Public DNS (ec2-13-222-21-231.compute-1.amazonaws.com), and Instance state (Running).

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



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/1AVLSqfmfnMphv5Ad4A1Mwo8qECo.
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%
  Swap usage:   0%            Users logged in:    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 B]
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 B]
Get:53 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Get:54 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [384 B]
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 debootstrap docker-buildx docker-compose-v2 docker-doc rinse
  zfs-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, 8 newly installed, 0 to remove and 10 not upgraded.
Need to get 76.0 MB of archives.
After this operation, 288 MB 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 noble/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 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 dns-root-data all 2024071801~ubuntu0.24.04.1 [5918 B]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 dnsmasq-base amd64 2.90-2ubuntu0.1 [376 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 28.2.2-0ubuntu1~24.04.1 [28.3 MB]
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 MB in 1s (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 ...
Unpacking runc (1.3.3-0ubuntu1~24.04.2)
```

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: Pull complete
```

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
c5fd91cf9a9b4f0625d4d2c0d896406e8da76929ed75a3f9ccc1699fb08535
root@ip-172-31-9-83:/home/ubuntu/ar/maven-practice#
```

Step 21: Check the images and the containers

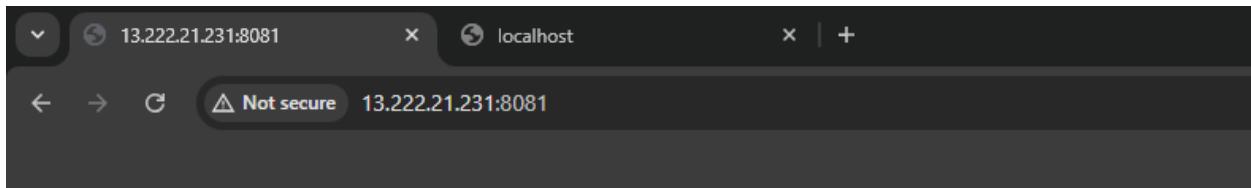
```
root@ip-172-31-9-83:/home/ubuntu/ar/maven-practice# sudo docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
img1           latest    a67a112ce8ac  2 minutes ago  413MB
dep1           latest    28efbe56e633  29 minutes ago  413MB
tomcat          9.0      2a4887a16e43  12 hours 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
84e7f9ce5ec2  dep1      "catalina.sh run"  9 minutes ago    Up 9 minutes  0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp
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