



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

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



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
((Artificial Intelligence and Machine Learning))**

LAB RECORD

SOFTWARE ENGINEERING LAB

B.Tech. III YEAR I SEM (RKR21)

ACADEMIC YEAR 2024-25



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad



NBA
NATIONAL BOARD FOR
TECHNICAL EDUCATION

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 INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad



Daily Laboratory Assessment Sheet

Name of the Lab:

Name of the Student:

Class:

HT. No:

S.No.	Name of the Experiment	Date	Observation Marks (3M)	Record Marks (4M)	Viva Voice Marks (3M)	Total Marks (10M)	Signature of Faculty
	TOTAL						

INDEX

S.NO	CONTENTS	PAGE NO
I	Vision/Mission /PEOs/POs/PSOs	
II	Syllabus	
III	Course outcomes, C0-PO Mapping	
Exp No:	List of Experiments	
1	INSTALLATION OF STARUML, GIT BASH AND GITHUB ACCOUNT CREATION	
2	IEEE – SRS (SOFTWARE REQUIREMENTS SPECIFICATION) WITH UML DIAGRAMS	
3	INSTALLATION OF ECLIPSE, MAVEN, JDK, TOMCAT, CONFIGURING TOMCAT TO ECLIPSE	
4	A. BASIC GIT COMMANDS - VERSION, CONFIG, INIT, STATUS, ADD, COMMIT, DIFF, HELP B. GIT COMMANDS: WORKING WITH LOCAL AND REMOTE REPOSITORIES - BRANCHES, CHECKOUT, MERGE, REVERT, LOG C. GIT COMMANDS: WORKING WITH REMOTE REPOSITORIES - REMOTE, CLONE, PULL, PUSH, FORK	
5	A. CREATING MAVEN JAVA PROJECT USING ECLIPSE AND PUSH INTO TO GITHUB. B. CREATING MAVEN WEB PROJECT USING ECLIPSE AND PUSH INTO TO GITHUB. C. INSTALLATION OF JENKINS.	
6	A. BUILDING THE CI/CD FREESTYLE PIPELINE USING JENKINS FOR MAVEN JAVA PROJECT B. BUILDING THE CI/CD FREESTYLE PIPELINE USING JENKINS FOR MAVEN WEB PROJECT WITH POLL SCM C. BUILDING THE CI/CD SCRIPTED PIPELINE USING JENKINS FOR MAVEN JAVA PROJECT WITH POLL SCM	
7	A. INSTALLATION OF DOCKER, MINIKUBE, ACCOUNT IN DOCKERHUB B. DOCKER CLI COMMANDS C. MODIFY AND PUSH DOCKER IMAGE D. CREATE AND PUSH DOCKERFILE IMAGE E. RUNNING MULTIPLE CONTAINERS USING DOCKER COMPOSE F. DEPLOYING AND SCALING APPLICATIONS USING MINIKUBE G. DEPLOYING AND MANAGING MONITORING SYSTEMS USING NAGIOS IN DOCKER	

Exp No:	List of Experiments	PAGE NO
8	<p>A. AWS ACADEMY LEARNING ACCOUNT CREATION</p> <p>B. PROJECT DEPLOYMENT IN THE AWS CLOUD USING EC2 INSTANCE</p> <p>C. MAVEN WEB PROJECT DEPLOYMENT IN THE AWS CLOUD USING EC2 INSTANCE</p>	



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

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



Department of Computer Science & Engineering(AI & ML)

Vision of the Institution:

To be the fountain head of latest technologies, producing highly skilled, globally competent engineers.

Mission of the Institution:

- To provide a learning environment that inculcates problem solving skills, professional, ethical responsibilities, lifelong learning through multi modal platforms and prepare students to become successful professionals.
- To establish Industry Institute Interaction to make students ready for the industry.
- To provide exposure to students on latest hardware and software tools.
- To promote research-based projects/activities in the emerging areas of technology convergence.
- To encourage and enable students to not merely seek jobs from the industry but also to create new enterprises
- To induce a spirit of nationalism which will enable the student to develop, understand India's challenges and to encourage them to develop effective solutions.
- To support the faculty to accelerate their learning curve to deliver excellent service to students



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

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



Department of Computer Science & Engineering (AI & ML)

Vision of the Department:

To be among the region's premier teaching and research Computer Science and Engineering departments producing globally competent and socially responsible graduates in the most conducive academic environment

Mission of the Department:

- To provide faculty with state-of-the-art facilities for continuous professional development and research, both in foundational aspects and of relevance to emerging computing trends.
- To impart skills that transform students to develop technical solutions for societal needs and inculcate entrepreneurial talents.
- To inculcate an ability in students to pursue the advancement of knowledge in various specializations of Computer Science and Engineering and make them industry-ready.
- To engage in collaborative research with academia and industry and generate adequate resources for research activities for seamless transfer of knowledge resulting in sponsored projects and consultancy.
- To cultivate responsibility through sharing of knowledge and innovative computing solutions that benefit the society-at-large.
- To collaborate with academia, industry and community to set high standards in academic excellence and in fulfilling societal responsibilities



Department of Computer Science & Engineering (AI & ML)

PROGRAM OUTCOMES (POs)

PO1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct Investigations of Complex Problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and Team Work: Function effectively as an individual, and as a member or

leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY (AN AUTONOMOUS INSTITUTE)

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



Department of Computer Science & Engineering(AI & ML)

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: An ability to analyze the common business functions to design and develop appropriate Computer Science solutions for social upliftment.

PSO2: Shall have expertise on the evolving technologies like Python, Machine Learning, Deep Learning, Internet of Things (IOT), Data Science, Full stack development, Social Networks, Cyber Security, Big Data, Mobile Apps, CRM, ERP etc.



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

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



NBA
ACCREDITED

Department of Computer Science & Engineering(AI & ML)

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will endeavor to excel in their chosen careers as professionals, researchers and entrepreneurs on a global platform.

PEO2: Graduates will demonstrate the ability to solve challenges in the fields of Engineering and Technology simultaneously catering to societal needs.

PEO3: Graduates will strive to improve their learning curve by practicing Continuing Professional Development (CPD).

PEO4: Graduates will, at all times, adopt a professional demeanor by communicating effectively, working collaboratively, and maintaining the ethics & core values as befitting their education in interdisciplinary and emerging fields.



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad



B. Tech. in COMPUTER SCIENCE AND ENGINEERING

III Year I Semester Syllabus (RKR21)

SOFTWARE ENGINEERING LAB (21CC505PC)

Common to CSE, IT, CSE (AI&ML) and CSE (DS)

L	T	P	C
0	0	3	1.5

Pre-requisites/ Co-requisites:

1. 21CC502PC – Software Engineering Course
2. 21CS401PC- Java Programming Course

Course Objectives: The course will help to

1. Formulate problem statements and Software Requirement Specifications by comprehensively grasping project requirements.
2. Demonstrate proficiency in designing, developing, and testing diverse project modules.
3. Utilize Git Framework and GitHub while implementing Continuous Integration/Continuous Deployment (CI/CD) pipelines through Jenkins.
4. Implement project deployment using Docker and Kubernetes.
5. Acquire knowledge in AWS cloud infrastructure.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Transform end-user needs into system and software requirements through a structured process.
2. Depict the system's high-level design using CASE tools based on the software requirements.
3. Employ Jenkins CI/CD for project building purposes.
4. Implement project deployment utilizing Docker and Kubernetes.
5. Create a project within the AWS Cloud environment.

Software to be used: The students must use JDK 11/17/21 Version, STAR UML, GIT Bash, Jenkins, Dockers Desktop, Mini KUBE, Eclipse, Tomcat, and Visual Studio Editor.

List of Experiments:

Do the following exercises for any one project given in the list of sample projects or any other projects?

1. Development of problem statement.
2. Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
3. Study and usage of any Design phase CASE tool
4. Creating the project and committing using Git and GitHub
5. Creating Maven Java and Maven Web project using Eclipse and Push them to GitHub.
6. Building the CI/CD pipeline using Jenkins for the project in the previous experiment.
7. Local Deployment of project using Docker, Kubernetes and Monitoring using Nagios tool.
8. Cloud Deployment of a project in the AWS Cloud using EC2 instance.

Sample Projects:

1. Book Bank
2. Online course reservation system
3. E-ticketing
4. Recruitment system
5. Hospital Management system
6. Online Banking System

TEXT BOOKS:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, Mc Graw Hill International Edition, 2015.
2. Software Engineering- Sommerville, 7th edition, Pearson Education, 2017.
3. The unified modeling language user guide Grady Brooch, James Rumbaugh, Ivar Jacobson, Pearson Education, 2016.
4. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, 2015.

REFERENCE BOOKS:

1. <https://kubernetes.io/docs/tutorials/hello-minikube/>
2. <https://minikube.sigs.k8s.io/docs/start/>
3. <https://www.jenkins.io/doc/>
4. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>
5. Introducing Maven by, Balaji Varanasi and Sudha Belida, APRESS publications.



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

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



NBA
NAAC

Department of Computer Science & Engineering (AI & ML)

Course Outcomes and CO-PO-PSO Mapping

Course Outcomes:

After learning the contents of this course, the student is able to

CO1	Transform end-user needs into system and software requirements through a structured process.
CO2	Depict the system's high-level design using CASE tools based on the software requirements.
CO3	Employ Jenkins CI/CD for project building purposes.
CO4	Implement project deployment utilizing Docker and Kubernetes.
CO5	Create a project within the AWS Cloud environment.

CO-PO-PSO MAPPING:

	CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO-1	PSO-2
Software Engineering Lab	CO1	3	3	3	2						2			2	1
	CO2	2	3	3	3	3					2			2	1
	CO3					3				2	2	2		1	2
	CO4					3	3	2	2		2	2	2	2	3
	CO5	3				3	2	2				3	3	3	3

Software and Hardware Requirements

1. Software Tools

- a. Star UML**
- b. Java 11/17/21 Version**
- c. Apache Tomcat 9**
- d. Eclipse IDE**
- e. Visual Studio**
- f. Jenkins**
- g. Git Bash for Windows**
- h. Windows Docker Desktop**
- i. AWS Cloud account (Basic Plan)**

3. Hardware Requirements

- a. Windows 10 Pro**
- b. 8 GB RAM (64-bit Processor)**
- c. 512 GB HDD**
- d. i5 Core Processor**
- e. BIOS-level hardware virtualization support is enabled in the BIOS settings.**

Experiment-1: Installation of StarUML, Git Bash and GitHub Account Creation

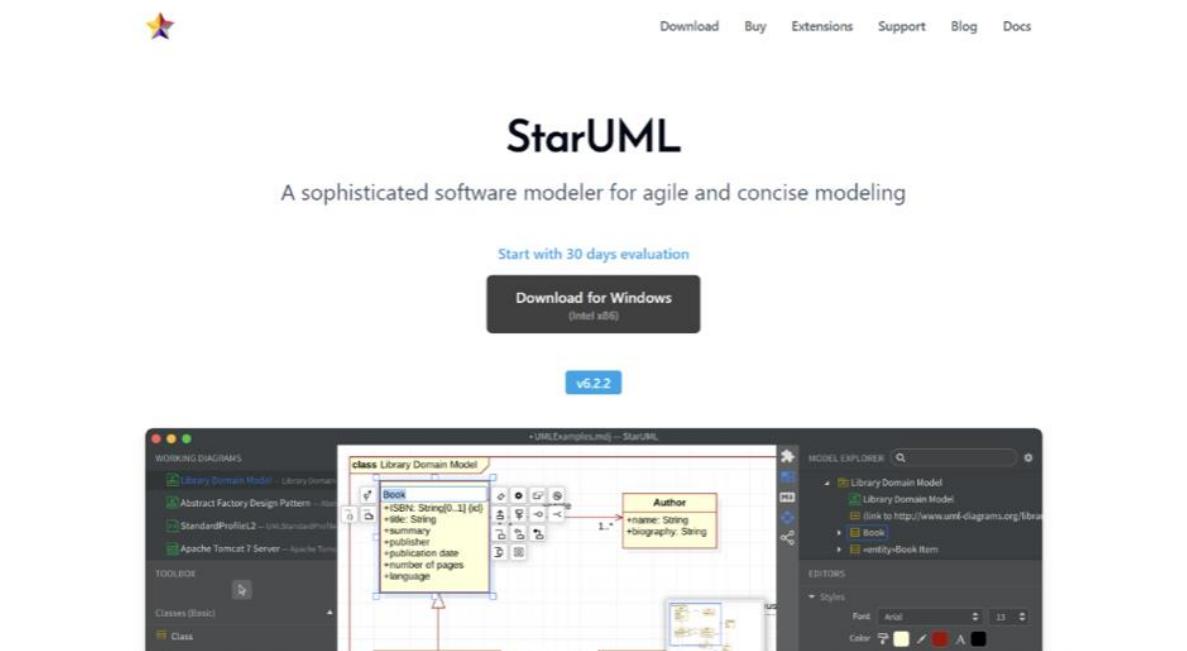
Aim: To install StarUML, Git Bash, and create a GitHub account.

Introduction:

1. **StarUML:** StarUML is a sophisticated software modeling tool that supports UML (Unified Modeling Language) for designing and describing software architecture and systems.
2. **UML Diagrams:** UML diagrams are graphical representations used to visualize, specify, construct, and document the structure and behavior of a software system.
3. **Git Bash and GitHub:**
 - o Git: Git is a popular version control system created by Linus Torvalds in 2005, designed for managing multiple versions of a code base across multiple developers or teams. Functions of Git include:
 - Managing projects with repositories
 - Cloning projects for local work
 - Controlling and tracking changes with staging and committing
 - Branching and merging for different parts and versions of a project
 - Pulling the latest project version locally
 - Pushing local updates to the main project
 - o GitHub: An online platform for hosting and collaborating on Git repositories.

Procedure:

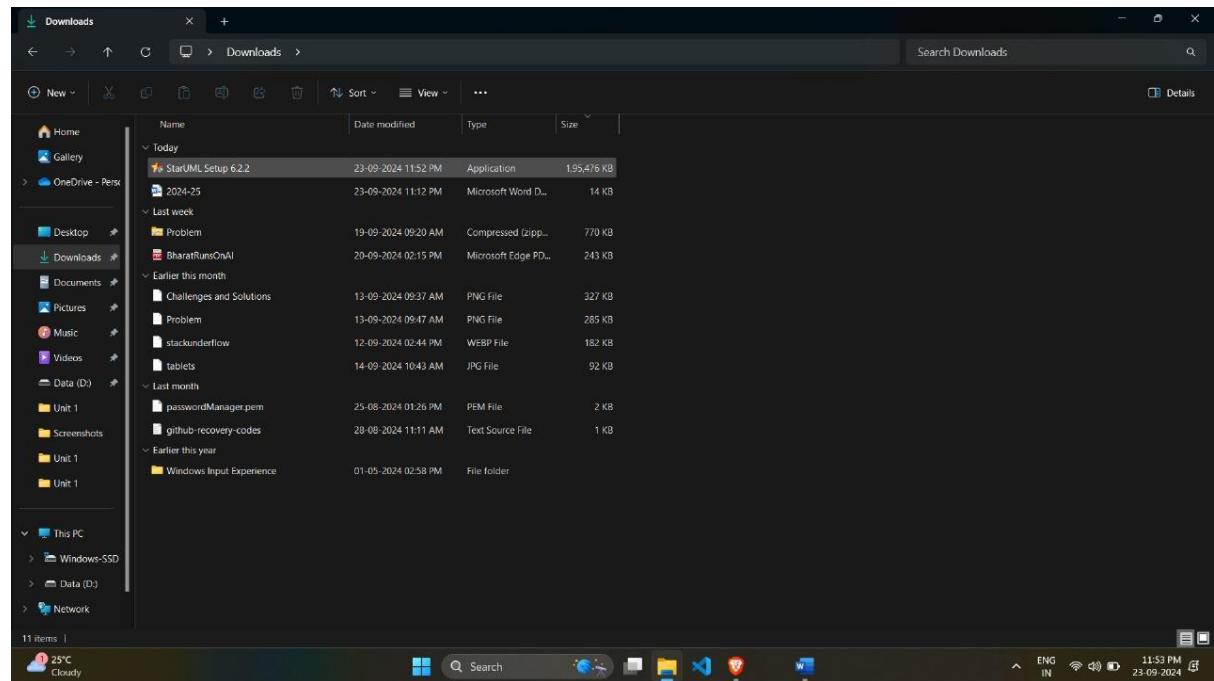
1. Download and Install StarUML from staruml.io website.



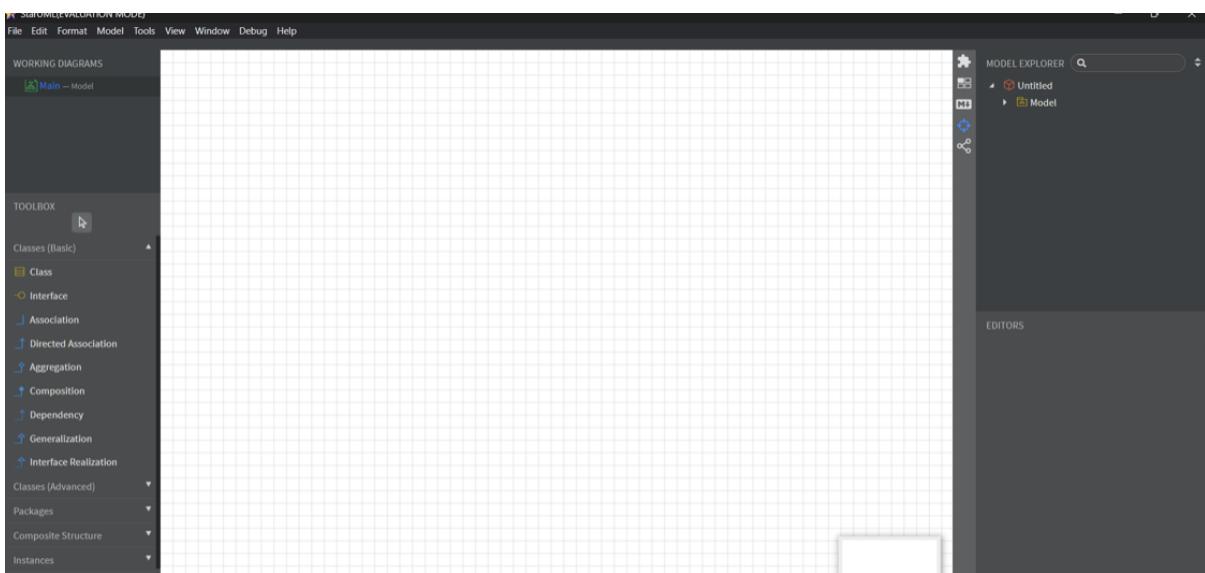
2. Click on download for windows button and select based on your system architecture.

The screenshot shows the StarUML download page. At the top, there's a navigation bar with links for Download, Buy, Extensions, Support, Blog, and Docs. Below the navigation is a large "Download" heading and a sub-headline "Start with 30 days evaluation". A blue button labeled "v6.2.2" is centered. Below this, there are three main download sections: macOS (with icons for Intel x86 and Apple arm64), Windows (with an icon for x86-64bit), and Linux (with icons for .deb and .rpm). The Windows section is highlighted with a black background. Below these sections, a note says "If you want to download for previous versions, you can get a link for previous versions by [finding your license key](#)". At the bottom left, there's a link to "download/releases/v6/StarUML_Setup.6.2.2.exe", and at the bottom right, a "Release Notes" link.

3. Install the StarUML software which will be present in the downloads section .



4. Explore available features



Installation of Git

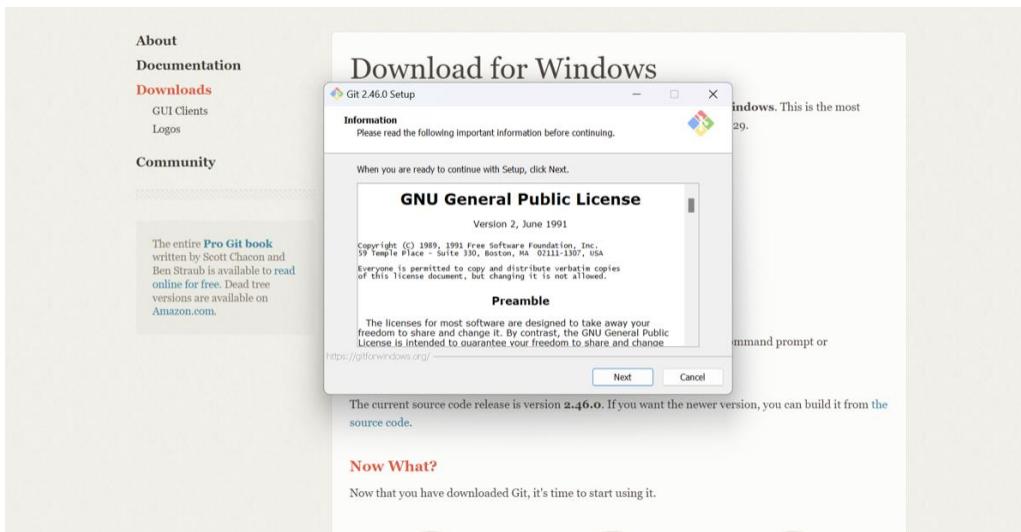
1. Go to the Download section of <https://git-scm.com/>

The screenshot shows the official Git website. On the left, there's a sidebar with links for About, Documentation, Downloads (GUI Clients, Logos), and Community. A sidebar box also mentions the 'Pro Git book'. The main content area is titled 'Download for Windows'. It features a prominent download link for the 'latest (2.46.0) 64-bit version of Git for Windows'. Below this, there are sections for 'Other Git for Windows downloads' (Standalone Installer, 32-bit Git for Windows Setup, Portable, 32-bit Git for Windows Portable, 64-bit Git for Windows Portable), 'Using winget tool' (with a command example), and a note about the current source code release being version 2.46.0. A 'Now What?' section at the bottom encourages users to start using their newly downloaded Git.

2.Select the latest version of Git software for the required OS and download it.

This screenshot is similar to the one above, showing the 'Download for Windows' section of the Git website. However, a dark overlay window titled 'Recent download history' is displayed in the top right corner. This window lists three recent downloads: 'Git-2.46.0-64-bit.exe' (65.1 MB, Done), 'StarUML Setup 6.2.2 (1).exe' (Cancelled), and 'StarUML Setup 6.2.2.exe' (191 MB, 2 hours ago). A 'Full download history' link is also visible at the bottom of the overlay.

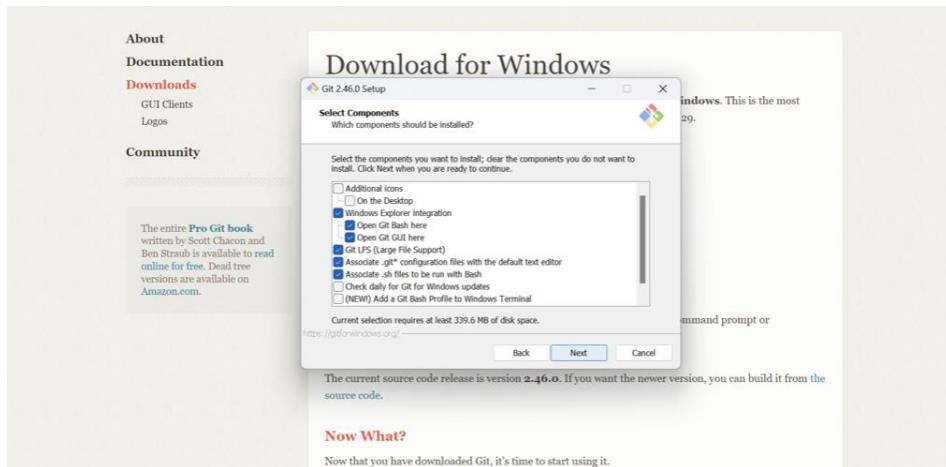
3.Install the downloaded Git software. Click next



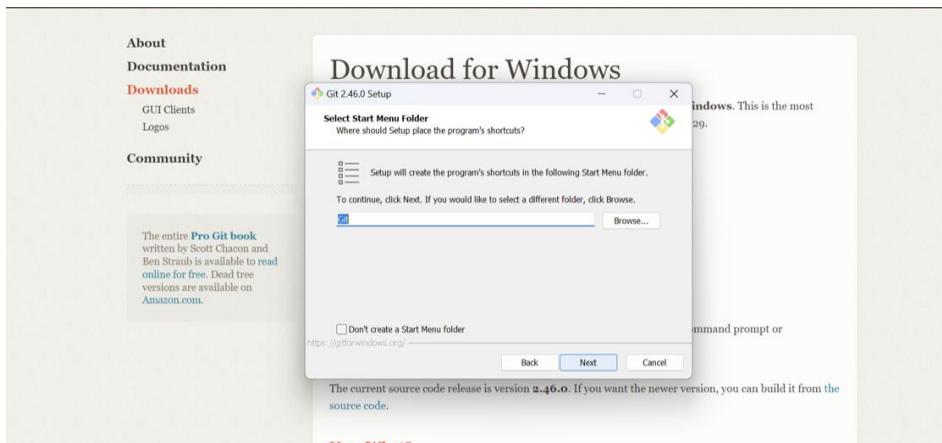
4.Select the destination location for the installation.



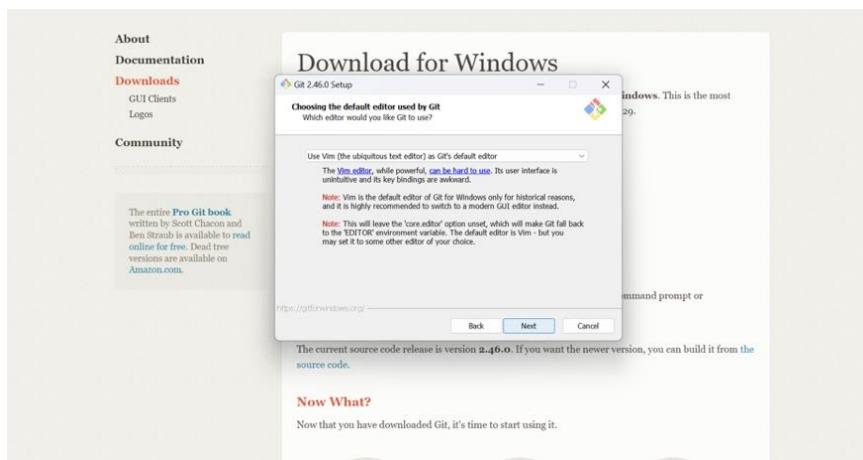
5.Select the required components and click next.



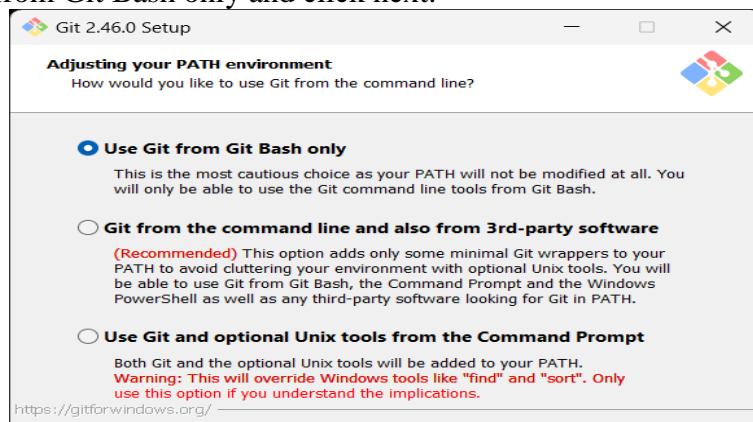
6.Select the start menu folder and click next.



7.Choose the default editor and click next.



8.Select use Git from Git Bash only and click next.



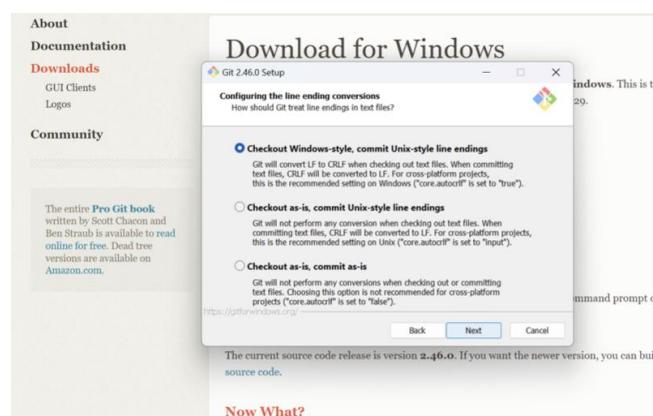
9. Choose the SSH executable and click next.



9. Choose the HTTPS transport library as OpenSSL library. Click next.



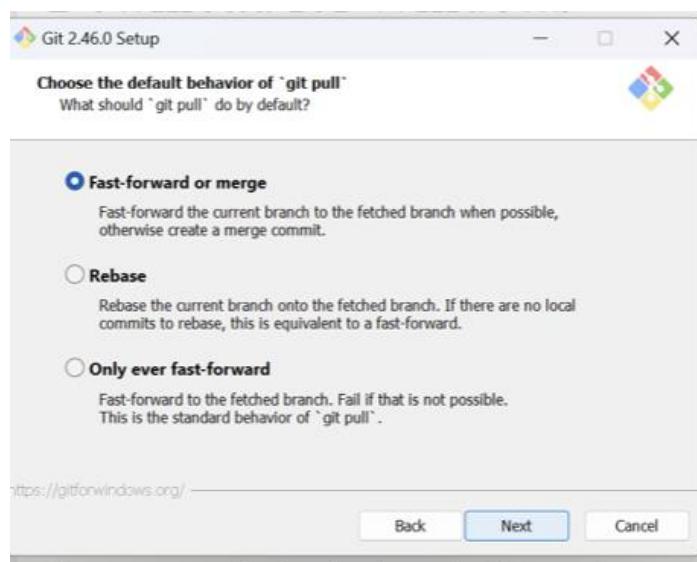
10. Configure the line ending conversions and Click next to proceed.



11.Select the terminal emulator to use with Git Bash and Click next.

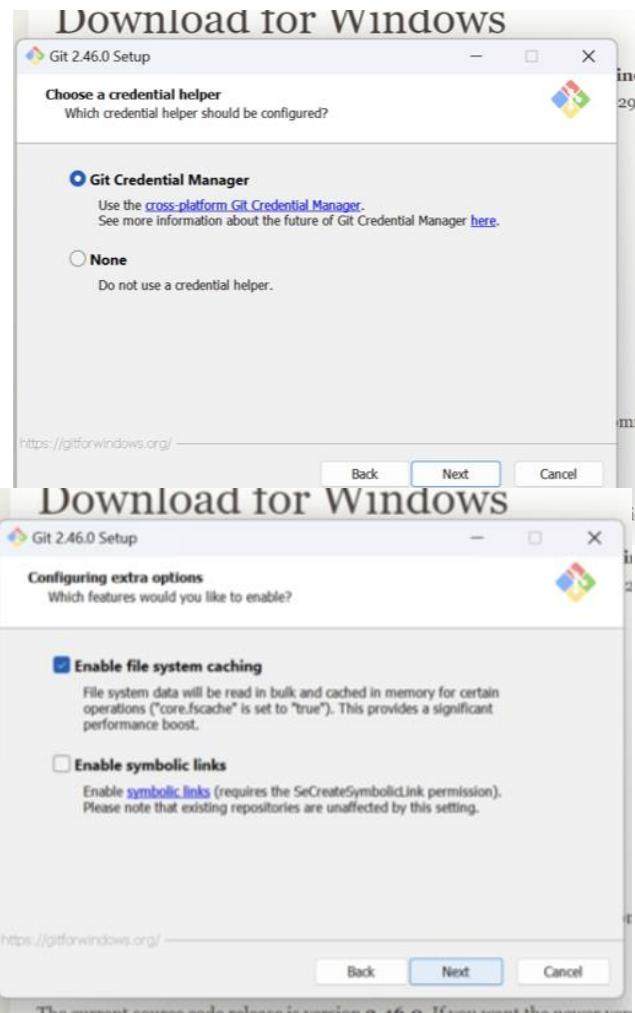


12.Choose the default actions of git pull. Click next.



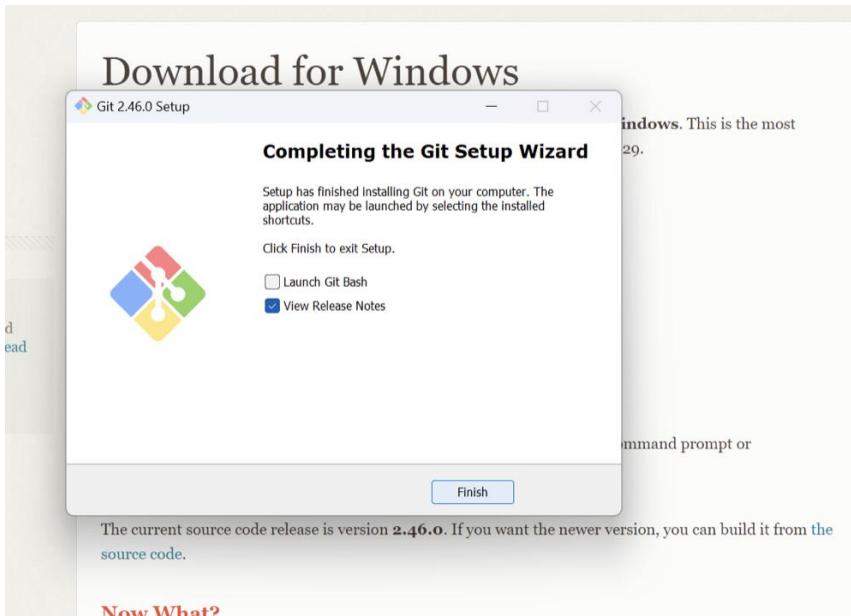
13.Select the Git credential manager. Click next.

14.Enable
click next.



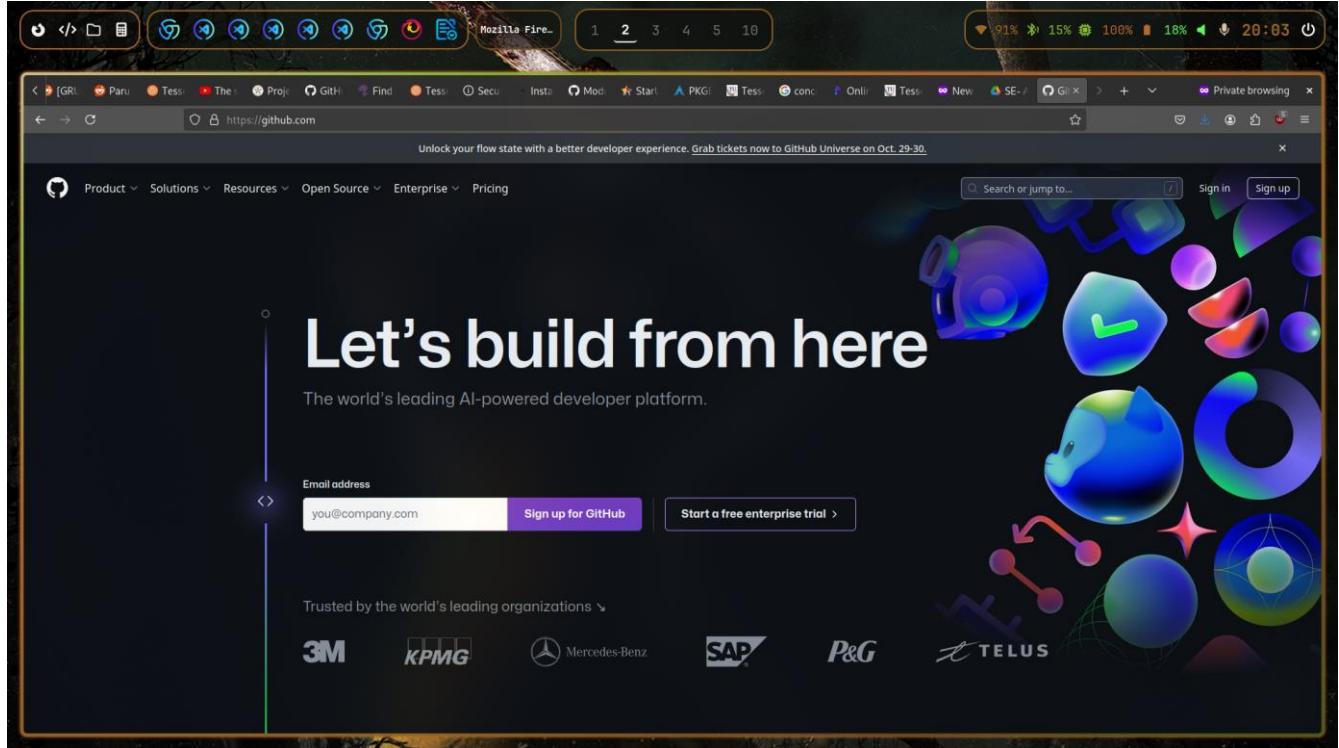
file system caching and
click next.

15.Click on install. And Click on finish to complete the installation process.

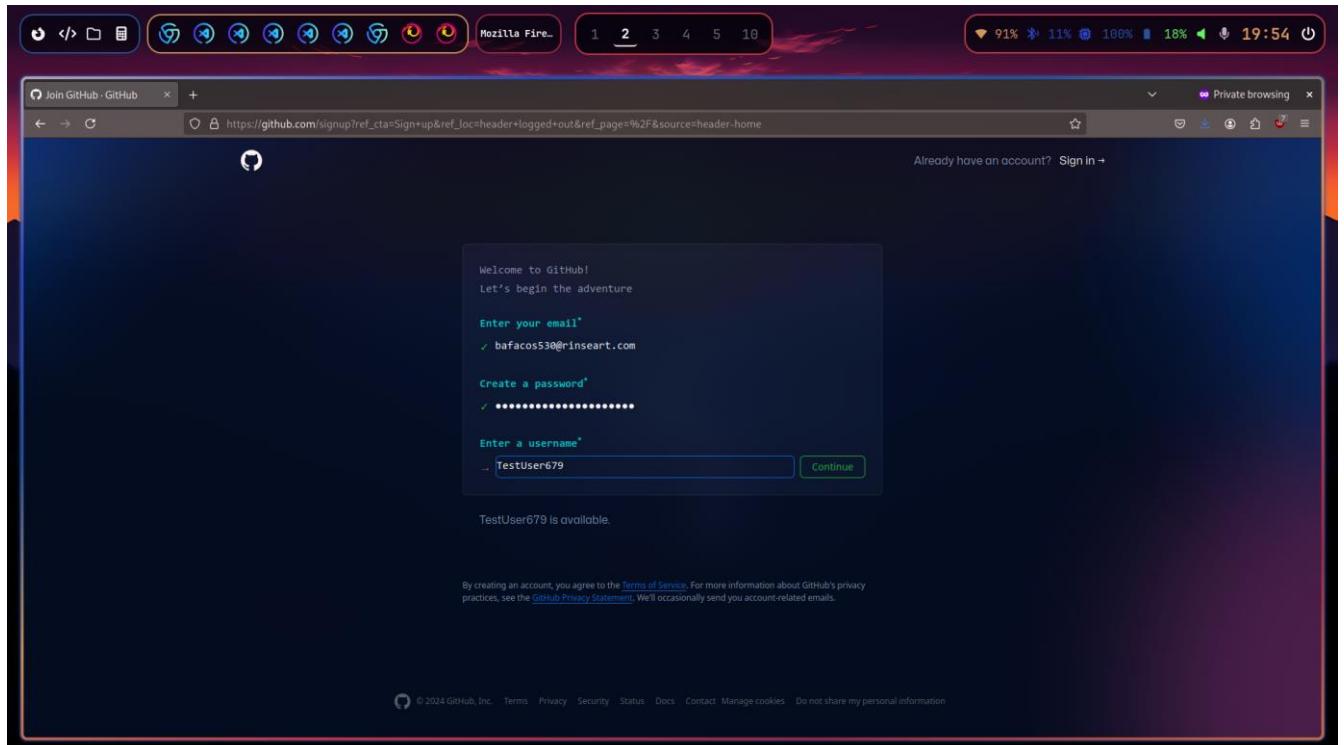


Creating GitHub account

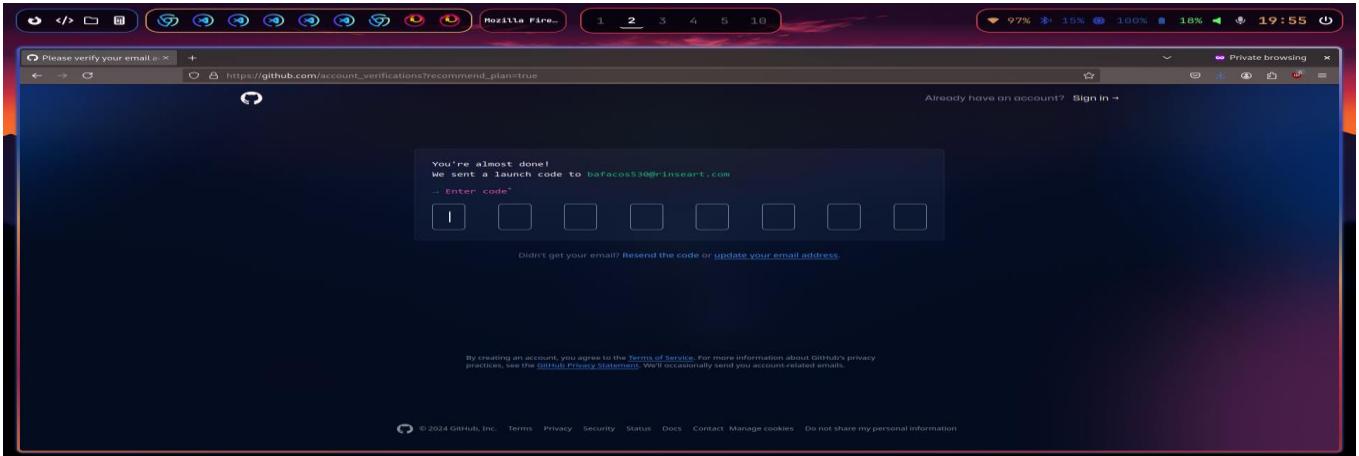
1. Go to <https://github.com> and click on Sign Up



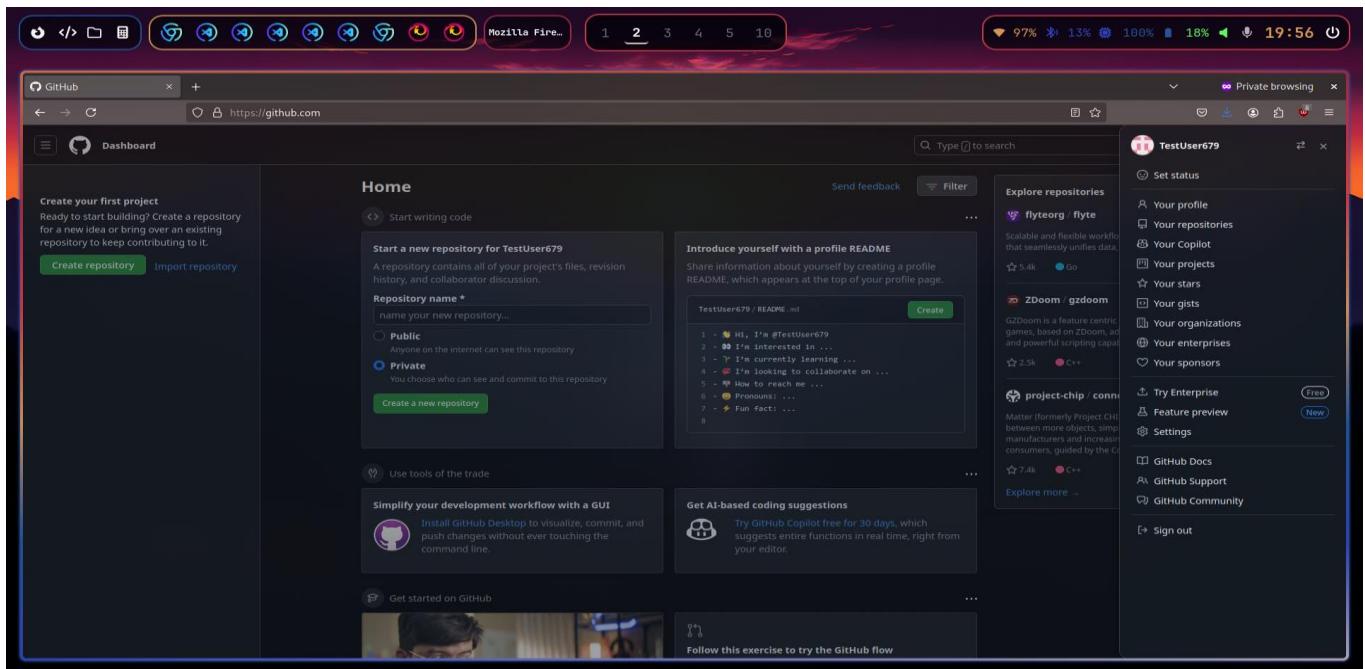
2. Enter your Email address and create a password



3. Click continue, then enter the verification code sent to your email



4. You are now logged in to GitHub



Result: Installed Star UML and explored it's functionality, installed Git SCM and created a GitHub account.

Experiment-2: Preparation of Software Requirements Specification Document

Aim: To study and understand the components and importance of a Software Requirements Specification (SRS).

Introduction:

1. **SRS:** A Software Requirements Specification is a document that comprehensively describes the functional and non-functional requirements of a software system. It serves as a formal agreement between stakeholders and developers, outlining what the software must achieve.
2. **Purpose of SRS:**
 - Ensures a clear understanding of project goals and requirements.
 - Acts as a reference throughout the development lifecycle.
 - Facilitates communication among stakeholders, developers, and testers.
3. **Key Components of SRS:**
 - **Functional Requirements:** Define specific behaviors or functions of the software.
 - **Non-Functional Requirements:** Include performance, security, and usability considerations.
 - **System Constraints:** Highlight limitations or dependencies of the project.



Software Requirements Specification For E-Ticketing

Version 3.0 approved

Prepared by:

- | | |
|-------------|--------------|
| 1. Aditya | - 22BD1A6602 |
| 2. Harsha | - 22BD1A660J |
| 3. Armaan | - 22BD1A6611 |
| 4. Anand | - 22BD1A6615 |
| 5. Varshith | - 22BD1A6618 |

Keshav Memorial Institute of Technology

11-10-2024.

Table of Contents

1. Introduction.....	
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions.....	2
1.4 Product Scope.....	2
1.5 References.....	2
2. Overall description.....	
2.1 Product Perspective.....	3
2.2 Product Functions.....	3
2.3 Operating Environment.....	4
2.4 User Characteristics.....	4
2.5 Design and Implementation Constraints.....	4
2.6 User documentation.....	5
2.7 Assumptions and Dependencies.....	5
3. Specific Requirements.....	
3.1 User Interfaces.....	7
3.2 Hardware Interfaces	7
3.3 Software Interfaces.....	7
3.4 Communications Interfaces.....	8
4. System Features.....	
4.1 Global database.....	8
4.2 Login interface.....	8
4.3 Payment Processing.....	9
4.4 Admin Management.....	10
4.5 Notifications & Alerts.....	10
5. Nonfunctional Requirements.....	
5.1 Performance Requirements	11
5.2 Safety Requirements	11
5.3 Security Requirements	11
5.4 Software Quality Attributes	11
5.5 Business Rules	12
6. Other Requirements.....	
6.1 Regulatory Compliance.....	13
6.2 Data Backup & Recovery.....	13
6.3 User Accessibility.....	13
6.4 Localization & Internationalization.....	14
6.5 Scalability.....	14
6.6 Session Management	14
Appendix A: Glossary.....	14
Appendix B: Analysis Models.....	15
Appendix C: To Be Determined List.....	24

Revision History

Name	Date	Reason for changes	Version
Week-1	11-10-24	SRS Template	1.0
Week-2	17-10-24	SRS (DIAGRAMS)	2.0
Week-3	25-10-2024	SRS Final document	3.0

1. Introduction

The Software Requirement Specification is designed to document and describe the agreement between the customer and developer regarding the specification of the software product requested.

This documentation is done to provide a clear idea of customer requirements. This document can be used as reference in further development of the software system.

1.1 Purpose:

This is the Software Requirements Specification (SRS) for our **E-TICKETING** platform. The project focuses on simplifying ticket booking for events, travel, and other activities through an online system. It will allow users to purchase tickets conveniently without the need to visit physical counters. We will also provide event organizers and service providers with the ability to upload and manage their events or transport schedules for sale. The platform aims to streamline the entire process between the ticket buyer and the provider, making the transaction quick and easy.

1.2 Document Convention:

- **Heading:**
 - Font-Size:16
 - Font-Style: Bold
 - Font: Times New Roman
- **Subheading:**
 - Font-Size:14
 - Font-Style: Bold
 - Font: Times New Roman
- **Content:**
 - Font-Size:12
 - Font: Times New Roman.

1.3 Intended Audience and Reading Suggestions:

This document serves as a prototype for the **E-TICKETING** platform. It is designed for various audiences with different purposes in mind. Developers can use this document to design and implement the project; managers can use it for planning, managing costs, and tracking timelines. Event organizers and service providers can use it to understand the system's features, while users can assess if the platform meets their needs and suggest any improvements. Testers can refer to this document for testing functionalities. The functional and non-functional requirements outlined here are also beneficial for developers during the development process.

1.4 Product Scope:

Our project is similar to existing e-ticketing systems but introduces several new features and improvements. The platform will be available 24/7, with periodic maintenance scheduled once a month. The primary goal is to enhance user convenience by allowing them to book tickets from anywhere, saving significant time compared to traditional ticketing methods. Service providers won't need to maintain physical stock and can manage bookings digitally. Our system will provide users with ticket recommendations based on their preferences and needs. Users will also have the ability to rate event organizers or service providers based on the quality of the service, enabling better customer experience and feedback management. User data will be secured against breaches and unauthorized access. Regular surveys will be conducted to gather user feedback and continuously improve the platform. A robust database will be maintained for both users and service providers.

User details will be kept safe and secure from data theft. We conduct surveys on how we can make our site better and what changes can be made. We also maintain a database for different users.

1.5 References:

References were taken from various ticketing systems and platforms, including industry standards and best practices from ticketing websites like Ticketmaster, Eventbrite, IRCTC, BookMyShow, and other similar platforms.

2. Overall Description

2.1 Product Perspective

Our project is an evolution of existing **e-ticketing** systems. It will include key functionalities found in ticketing platforms, such as allowing event organizers or service providers to set up digital listings, customers to browse through available events or services, and administrators to manage event categories, accept or reject service providers, and maintain platform policies. The system will be **mobile-friendly** with highly customizable themes. Additionally, it incorporates several new advancements such as an **advanced search engine** for easy ticket search, **high data security** to protect user information from breaches, and a **user-friendly interface** that ensures a smooth experience for users of all types.

2.2 Product functions

2.2.1 Administrator

- Administrators will have the ability to insert, modify, and delete event listings or service offerings.
- Can accept or reject event organizers or service providers based on platform policies and compliance with payment methods.
- Can add, edit, and arrange events or services into categories for easier navigation.
- Will be able to notify users about special discounts, promotional offers, and relevant updates.
- Administrators can record canceled or refunded tickets based on user actions.

2.2.2 Customers/Users

- Users will receive up-to-date information about new events or services, as well as ongoing discounts or special offers.
- They will be able to manage and modify their account details, including payment information and preferences.
- Users can search for tickets by event, category, location, or any relevant search criteria to find exactly what they're looking for.

- They will have the ability to cancel or modify their bookings according to the platform's terms and conditions.
- Users can suggest events or services to be added to the platform, and provide reviews or feedback on the event or service providers.

2.3 Operating Environment

- This e-ticketing platform will function seamlessly across all major web browsers, including but not limited to Firefox, Chrome, Safari, and Edge. For a model reference, it is compatible with modern versions of these browsers with JavaScript and HTML5 capabilities.
- The platform will be accessible on Windows (7, 8, 10, 11), macOS, and Linux operating systems.
- Minimum system requirements include a Pentium 4 processor or higher.
- The platform requires at least 1GB of RAM or higher for optimal performance.

2.4 User Characteristics

The users of this e-ticketing platform include customers, event organizers, and administrators who maintain the system.

- **Customers** are individuals booking tickets for events, transport, or other services. They are assumed to have a basic understanding of computers and internet browsing.
- **Event organizers** or service providers are responsible for uploading event details and managing ticket sales on the platform. They should have the necessary skills to manage their listings and handle bookings.
- **Administrators** are responsible for maintaining the overall system, ensuring smooth operations, and resolving any issues that arise. They should have a deeper understanding of the internal modules of the platform and be capable of troubleshooting problems as needed.

2.5 Design and Implementation Constraints

- **Stable Internet Connection:** Users must have a device with a stable internet connection and internet browsing capabilities to access the e-ticketing platform.

- **Database Access:** The information about users, ticket availability, and event or transport listings will be stored in a database, which the platform can access in real-time.
- **Availability:** The platform will be available **24/7** to ensure that users can book tickets at any time.
- **Platform Compatibility:** The software is designed to be cross-platform and can run on any operating system, including mobile devices.
- **User Authentication:** Users must log in with the correct username and password to access their accounts and perform actions such as booking or managing tickets.

2.6 User Documentation

The user documentation for the e-ticketing system will provide comprehensive guides for various users, ensuring they can effectively navigate and utilize the platform. The documentation will include:

- **User Manuals for Customers:** Detailed step-by-step guides explaining how to create an account, search for available tickets, filter based on destination or price, book tickets, make payments, and manage bookings (e.g., cancellations or refunds).
- **Administrator Manuals:** Guides for system administrators on how to manage user accounts, view and edit ticket listings, update destinations, oversee payment systems, and generate detailed reports on user activity and transactions.
- **Training Guides for Support Staff:** These will help customer support teams assist users with ticket booking issues, payment queries, and troubleshooting common problems.
- **Online Help and FAQs:** A well-organized help section available on the platform itself, providing quick access to common issues, FAQs, and troubleshooting steps for users to solve problems without additional assistance.

Additionally, video tutorials and interactive demos will be developed for key features of the system, such as booking tickets and using advanced search filters.

2.7 Assumptions and Dependencies

The website requires following third party products.

- **MongoDB:** For storing user data, ticket availability, and booking details in a scalable NoSQL database.

- **Express.js**: To handle the server-side application logic and API requests.
- **React.js**: For building an interactive and dynamic user interface that offers a seamless user experience.
- **Node.js**: To power the backend server, manage API routes, and handle requests efficiently.

3.External Interface Requirements

3.1 User Interfaces

The e-ticketing system will feature a user-friendly, responsive interface designed for seamless navigation. Users will have access to an intuitive dashboard, allowing them to search for tickets by location, date, price, and category. The interface will support filters for narrowing down ticket options and real-time updates on ticket availability. The ticket booking process will be clear, with options to review and confirm bookings, view payment history, and manage user profiles.

3.2 Hardware Interfaces

The system will be compatible with multiple devices, including Windows, Mac, and Linux systems, as well as mobile platforms (iOS and Android). It will require a minimum hardware specification of a 1.7 GHz processor and 2 GB of RAM to ensure smooth operation. It will also support standard peripherals like monitors, keyboards, and touch-screen devices.

3.3 Software Interfaces

The system will be developed using **JavaScript**, **Node.js**, and **React** for frontend, with **MySQL** for database management. It will also integrate with third-party payment systems such as **PayPal**, **Stripe**, and **Google Pay**. The backend will be deployed on cloud platforms like **AWS** or **Heroku** and will support RESTful APIs for seamless communication between client and server.

3.4 Communication Interfaces

The system will use **HTTP/HTTPS** for secure data communication, ensuring encryption through SSL certificates. Web services will allow users to interact with the system, including receiving email or SMS notifications for ticket confirmations or cancellations. The platform will also support integration with external ticketing services through APIs.

4. System Features

4.1 User Registration and Login

4.1.1 Description and Priority

This feature allows users to create accounts and log in using email and password or via social login (Google, Facebook). High priority as it is essential for user interaction with the system.

4.1.2 Stimulus/Response Sequences

- **Stimulus:** A user submits registration information.
- **Response:** The system creates a new user account and logs them in, displaying a confirmation message.

4.1.3 Functional Requirements

- **REQ-1:** The system must validate unique emails during registration.
- **REQ-2:** Users should be able to log in using social media credentials.
- **REQ-3:** Users must receive a confirmation email after registration.

4.2 Ticket Booking and Management

4.2.1 Description and Priority

This feature allows users to search for tickets, view availability, and book tickets for events or travel. High priority, as this is the core function of the system.

4.2.2 Stimulus/Response Sequences

- **Stimulus:** A user selects ticket preferences (e.g., location, date).
- **Response:** The system displays available tickets, updates based on filters, and confirms the booking once payment is processed.

4.2.3 Functional Requirements

- **REQ-4:** The system must display real-time availability for tickets.
- **REQ-5:** Users should be able to add multiple tickets to a cart for bulk purchase.
- **REQ-6:** The system must provide a booking confirmation after payment.

4.3 Payment Processing

4.3.1 Description and Priority

This feature handles payment processing through various gateways, ensuring secure transactions. High priority for ensuring successful ticket purchases.

4.3.2 Stimulus/Response Sequences

- **Stimulus:** A user proceeds to payment after selecting tickets.
- **Response:** The system processes the payment, confirms it, and updates the user's account with the transaction.

4.3.3 Functional Requirements

- **REQ-7:** The system must support multiple payment methods (credit card, PayPal, Google Pay).
- **REQ-8:** Payment information must be encrypted for security.
- **REQ-9:** The system should notify the user of payment success or failure.

4.4 Admin Management

4.4.1 Description and Priority

Admins will manage events, users, payments, and ticket availability. This feature is critical to maintaining system integrity and ensuring correct ticketing information.

4.4.2 Stimulus/Response Sequences

- **Stimulus:** An admin logs in and updates an event's ticket availability.
- **Response:** The system reflects the changes in real-time for all users.

4.4.3 Functional Requirements

- **REQ-10:** Admins must be able to add, edit, or delete events and ticket listings.
- **REQ-11:** The system must generate reports on ticket sales and user activity.
- **REQ-12:** Admins should be able to view and manage user accounts.

4.5 Notifications and Alerts

4.5.1 Description and Priority

This feature sends real-time notifications about booking status, cancellations, or payment issues.
High priority for keeping users informed.

4.5.2 Stimulus/Response Sequences

- **Stimulus:** A user books a ticket or an event is cancelled.
- **Response:** The system sends a notification via email or SMS.

4.5.3 Functional Requirements

- **REQ-13:** The system must send booking confirmations to users.
- **REQ-14:** The system should notify users of cancellations or changes in ticket availability.
- **REQ-15:** The system must ensure notifications are sent in real time.

5. Nonfunctional Requirements

5.1 Performance Requirements

The e-ticketing system must be optimized for high performance, capable of handling large volumes of ticket searches, transactions, and user queries efficiently. Data retrieval for ticket availability, payment history, or booking details must occur within 3 seconds under normal operating conditions. The system should support up to 5000 concurrent users without performance degradation, and search results for available tickets should be generated in less than 5 seconds. Additionally, real-time ticket availability updates should reflect changes (e.g., sold-out events) within 2 seconds.

5.2 Safety Requirements

Safety is critical to prevent loss of transaction data or incorrect bookings. The system must include a reliable backup mechanism, with automated backups scheduled hourly. A rollback feature must be available to allow administrators to undo erroneous actions, such as mistaken cancellations or payments. The system should ensure consistency in ticket inventory to avoid overselling or booking errors during high-demand periods.

5.3 Security Requirements

The e-ticketing system must prioritize security by implementing encryption for both data at rest and data in transit. Role-based access control will be enforced to limit access to sensitive data (e.g., payment information) to authorized users only. Payment transactions will be secured via SSL and tokenization to prevent unauthorized access or fraud. An audit trail will record all access and modifications to user accounts, tickets, and payment details for accountability.

5.4 Software Quality Attributes

- **Reliability:** The system must maintain 99.9% uptime, ensuring users can access ticketing services without interruptions.
- **Scalability:** The system must be designed to scale horizontally, accommodating increased traffic during peak booking periods.

- **Maintainability:** The system must have a modular architecture, allowing for easy updates, bug fixes, and the integration of new features.
- **Usability:** The interface must be intuitive, with minimal training required for both users and administrators.
- **Compatibility:** The system will integrate smoothly with third-party payment processors and ticket distribution services, using industry-standard protocols (e.g., REST APIs).

5.5 Business Rules

The system must be flexible enough to handle various ticketing models (prepaid, postpaid) and discount schemes (bulk tickets, early bird discounts). It must comply with local accounting standards, providing accurate reports for tax purposes and ensuring ticket availability and pricing are properly updated in real time. Refund and cancellation policies must be automated, issuing refunds based on predefined rules.

6. Other Requirements

6.1 Regulatory Compliance

The system must comply with data privacy regulations, such as the **General Data Protection Regulation (GDPR)** and **California Consumer Privacy Act (CCPA)**, ensuring the secure handling of user data. It must also adhere to **Payment Card Industry Data Security Standard (PCI DSS)** for processing payment information securely.

6.2 Data Backup and Recovery

Automated daily backups must be performed to prevent data loss. A **disaster recovery** plan must ensure that the system can be restored with minimal downtime in case of failure, including data replication across multiple locations to guarantee ticketing services remain available even during emergencies.

6.3 User Accessibility

The system must comply with **Web Content Accessibility Guidelines (WCAG) 2.1** to ensure accessibility for users with disabilities. This includes support for screen readers, keyboard navigation, and alternative text for images, making the system usable for all.

6.4 Localization and Internationalization

The system will support **multiple languages and currencies**, enabling a global audience to use the platform. Localization features will include the adaptation of date formats, currency symbols, and local ticketing standards.

6.5 Scalability

The system must be designed with scalability in mind, ensuring that it can efficiently handle increased traffic during peak times, such as high-demand ticket releases. This includes scalable cloud infrastructure to manage server loads dynamically.

6.6 Session Management

The system must implement **secure session management**, including automatic session expiration after a period of inactivity and **multi-factor authentication (MFA)** for additional security. The system will ensure secure logout protocols, especially on shared or public devices.

Appendix A: Glossary

- **REST API:** A set of rules allowing systems to communicate and exchange data securely over the web.
- **SSL Encryption:** Secure Sockets Layer, a standard security protocol for establishing encrypted links between web servers and browsers.
- **Tokenization:** The process of substituting sensitive data with unique identification symbols that retain essential information without compromising security.

Appendix B:

Analysis Models (Included as appropriate, such as use case diagrams, data flow diagrams, etc.)

Use case template

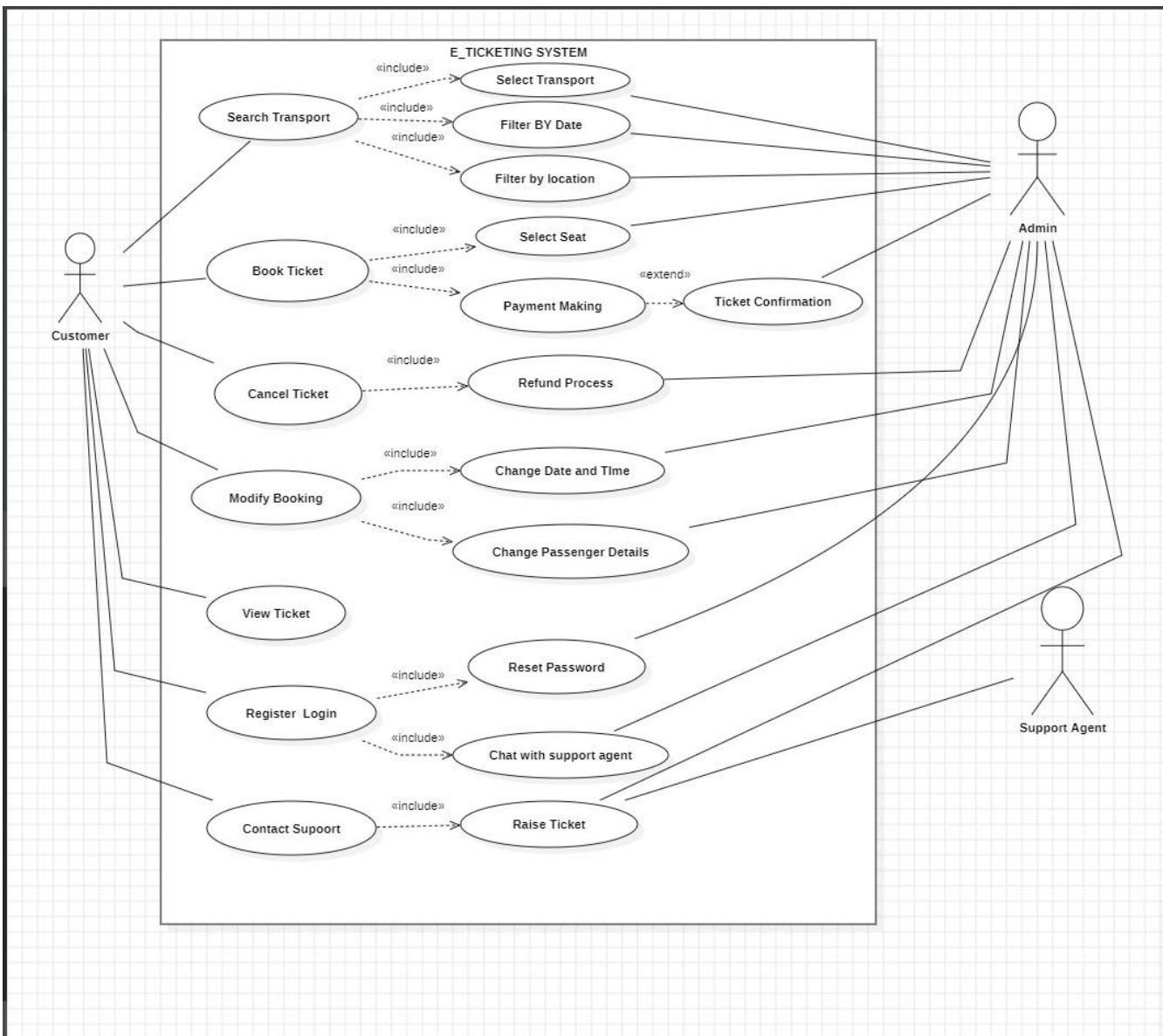
Use Case ID:	0123456789		
Use Case Name:	E-Ticketing		
End Objective:	Automate and facilitate the whole process of shopping		
Created by:	1. Aditya 2. Harsha 3. Armaan 4. Anand 5. Varshith	On(date):	October 16, 2024
User/Actor:	Customer, Seller and Administrator		
Trigger:	Customer buying ticket by logging in the site		
Basic/Normal Flows			
User Actions		System Actions	

The user tries to log in but doesn't have an account.	The page requests the user to register an account on the registration page before proceeding with the login.
The user tries to log in by entering details.	If the details entered are incorrect, the system displays a message: "Please check the username or password entered," prompting the user to re-enter the correct details.
The user tries to book a ticket.	If the ticket is not available, the system displays a "Not available" message for that specific booking.
The user completes ticket booking and proceeds to payment.	The system confirms the availability of the ticket and redirects the user to the payment page.

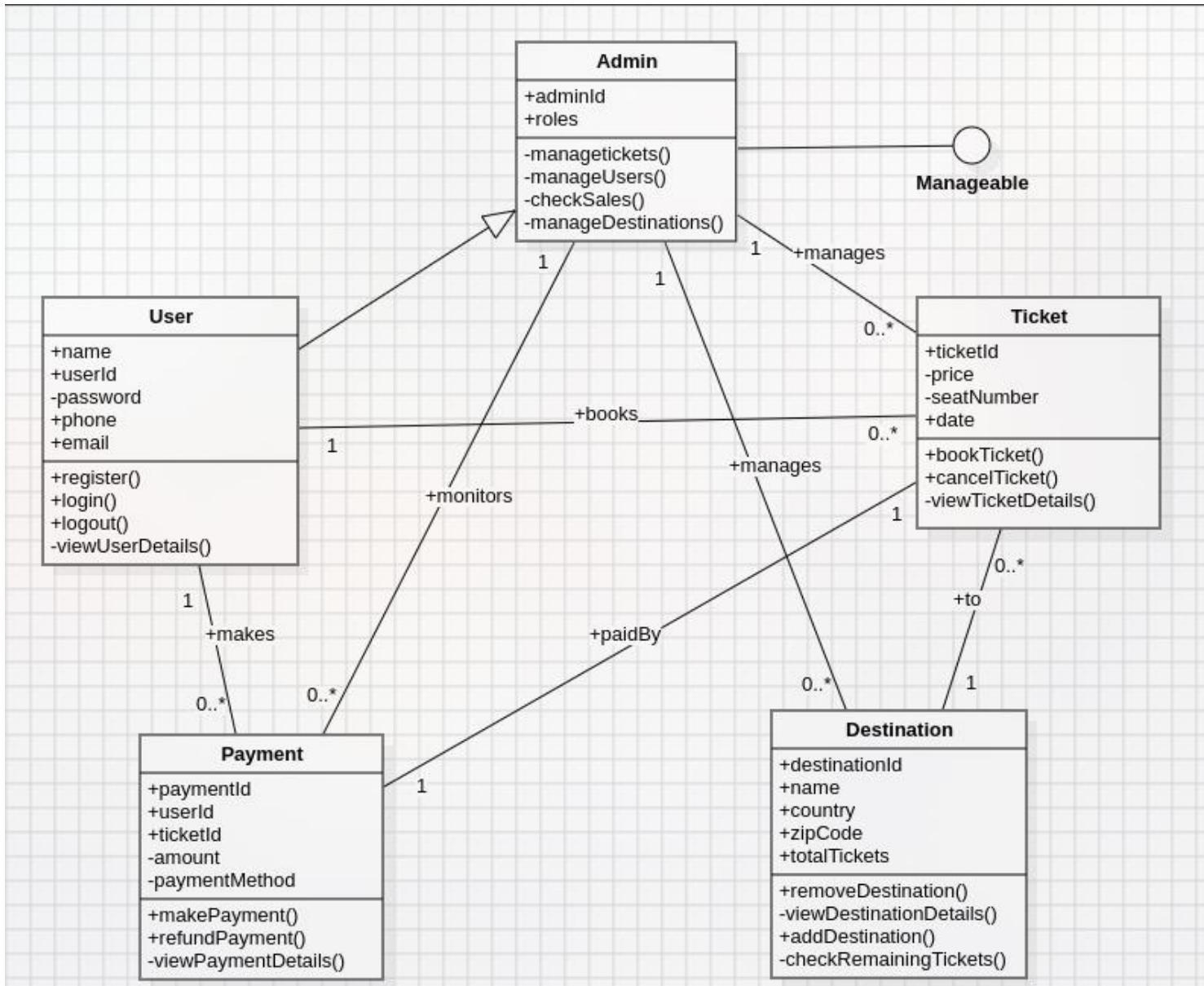
Exception Flows

User Actions	System Actions
The user tries to log in but doesn't have an account.	The system prompts the user to register an account on the registration page before proceeding with the login.
The user tries to log in by entering details.	If the login details entered are incorrect, the system displays a message: "Please check the username or password entered," prompting the user to re-enter the correct details.
The user tries to book a ticket.	If the ticket is not available, the system displays a "Not available" message for that specific booking.

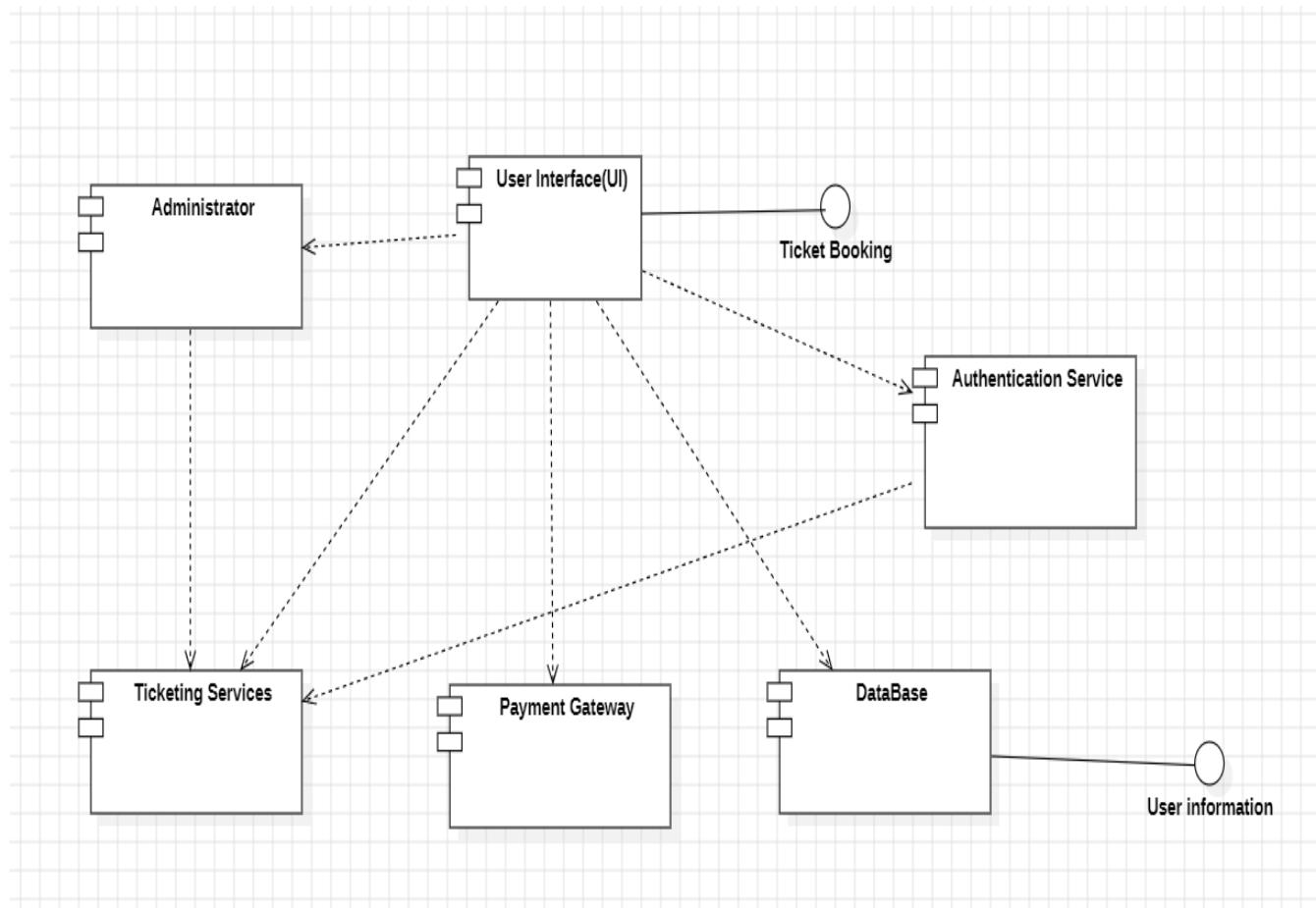
Use-Case Diagram



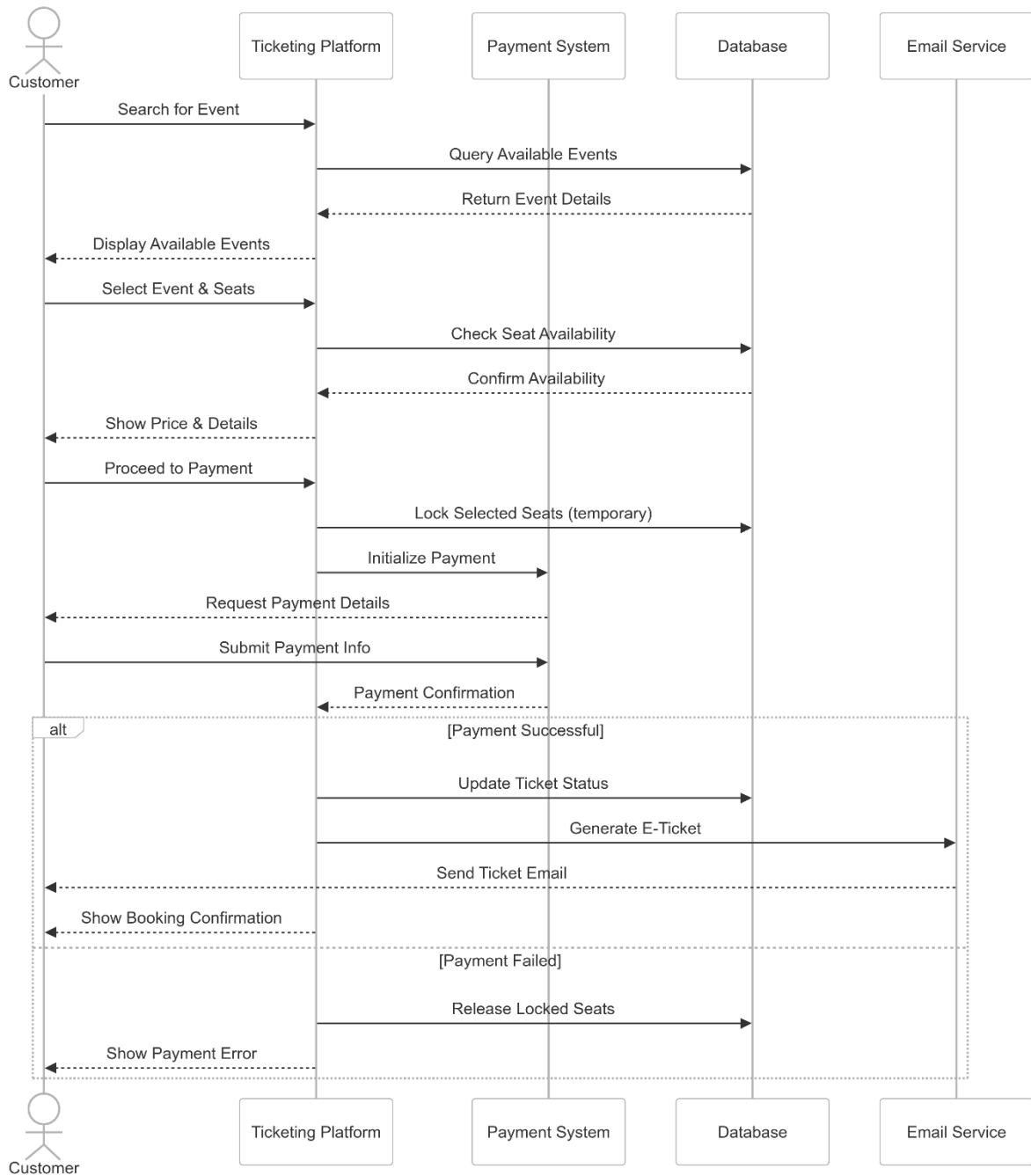
Class Diagram



Component Diagram



Sequence Diagram



Appendix C: To Be Determined (TBD) List

This appendix lists items that need further clarification or finalization during the project's development. These items will be addressed as the project evolves, and final decisions will be made during subsequent stages.

1. Third-Party Payment Systems

- The exact payment gateways (e.g., PayPal, Stripe, Google Pay) to be integrated for processing payments securely and efficiently are still under evaluation.

2. Finalized User Roles and Permissions

- User roles and permission levels, especially for the **Admin** class, need further definition. This will include detailed permissions (e.g., adding events, managing users, generating reports) and access control for sensitive actions.

3. Ticket Types and Categories

- Finalization of the specific ticket types (e.g., VIP, standard, student discounts) and categories (e.g., transportation, event types) the system will support.

Result: The Software Requirements Specification (SRS) document was made, understood and its key components identified, ensuring clarity in project requirements.

Experiment-3: Installation of Maven, Eclipse and Tomcat

Aim: To install Maven, Eclipse, and Tomcat.

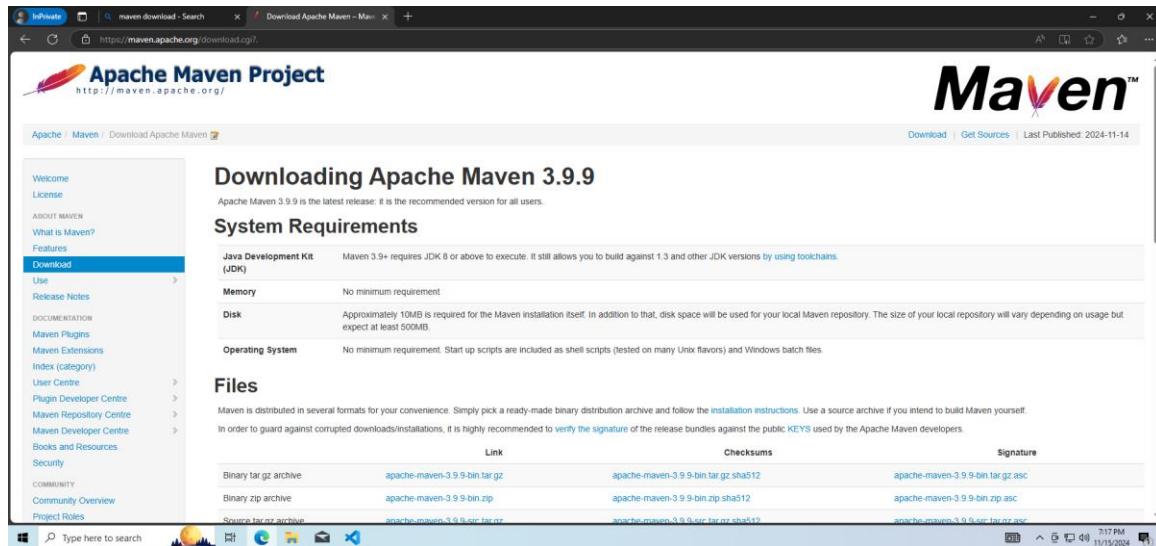
Introduction:

1. **Maven:** Maven is a build automation tool primarily used for Java projects. It helps in managing project dependencies, compiling code, packaging applications, and deploying them. It simplifies the build process by providing a consistent environment for building, testing, and deploying Java applications.
2. **Eclipse:** Eclipse is a popular Integrated Development Environment (IDE) used mainly for Java development. It provides tools for writing, debugging, and managing Java code. It also supports plugins for other languages and frameworks, making it versatile for a wide range of software development tasks.
3. **Tomcat:** Tomcat is an open-source web server and servlet container developed by Apache. It is used for running Java-based web applications. Tomcat executes Java servlets and renders JavaServer Pages (JSP), enabling dynamic content in web applications.

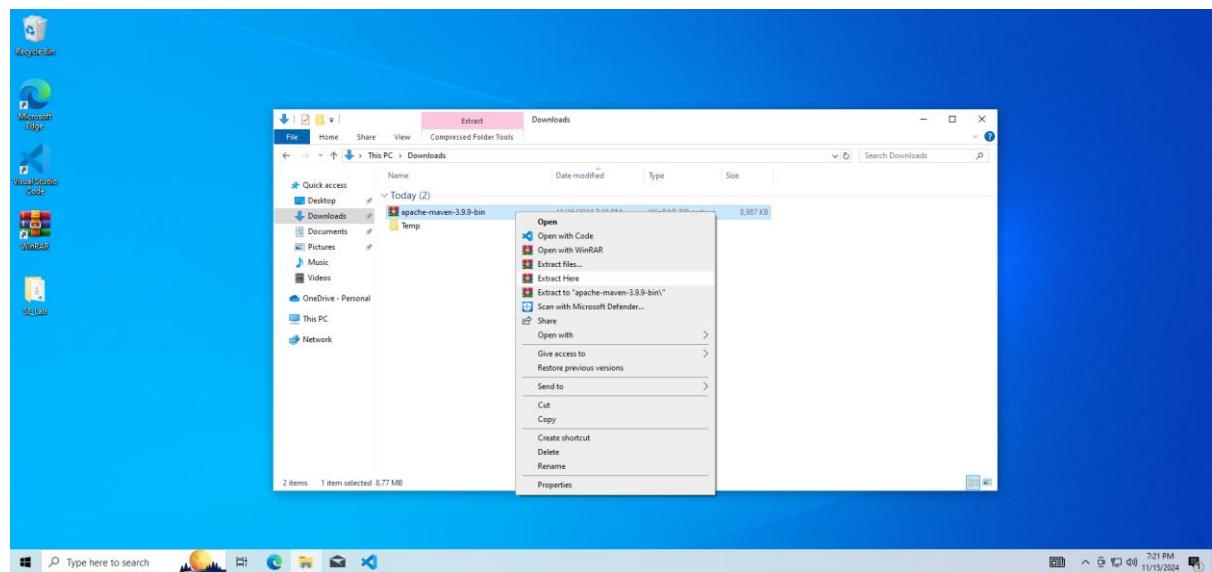
Procedure:

1.Maven Installation

Step 1: Go to <https://maven.apache.org/download.cgi>, and install the Binary zip archive for Maven.

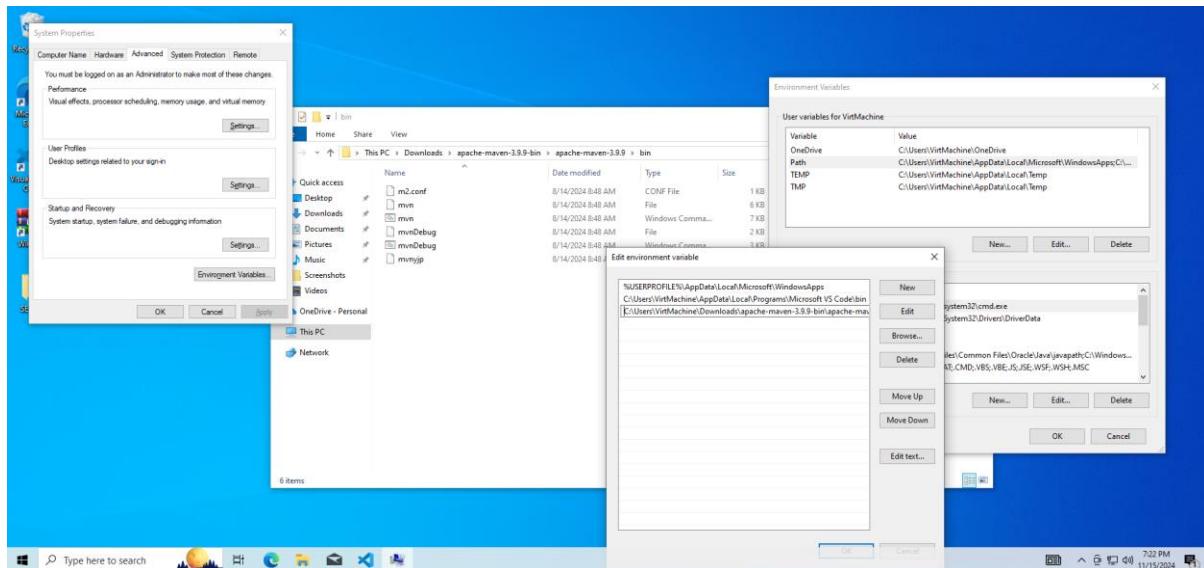


Step 2: Unzip the compressed folder present in the Downloads folder.

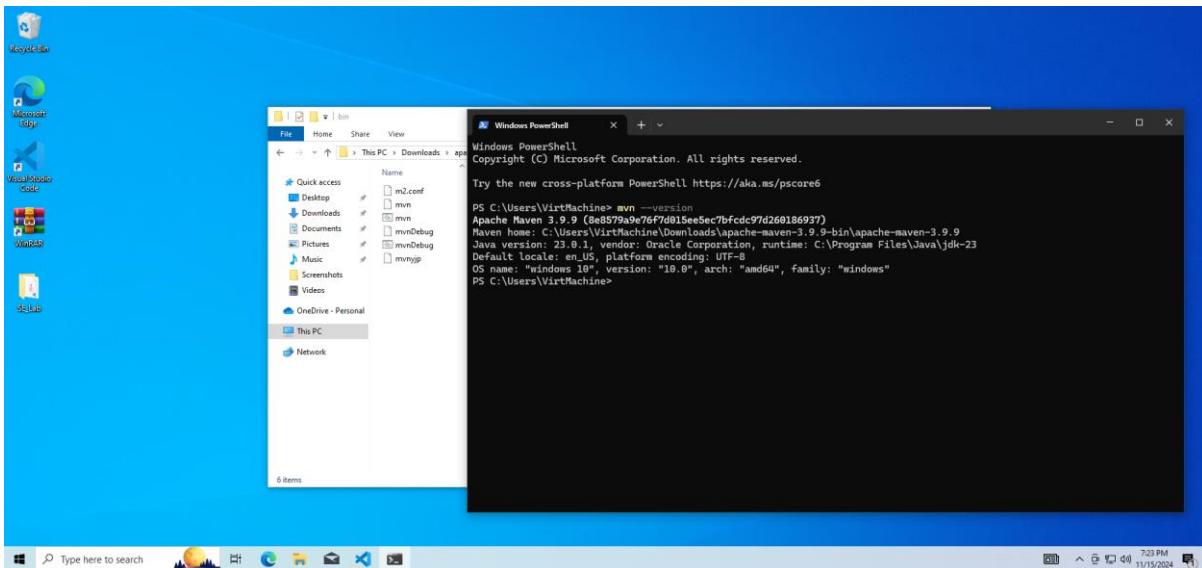


Step 3: Select the location for the extracted folder.

Step 4: Add classpath for the bin folder inside the Maven folder in the environment variables.

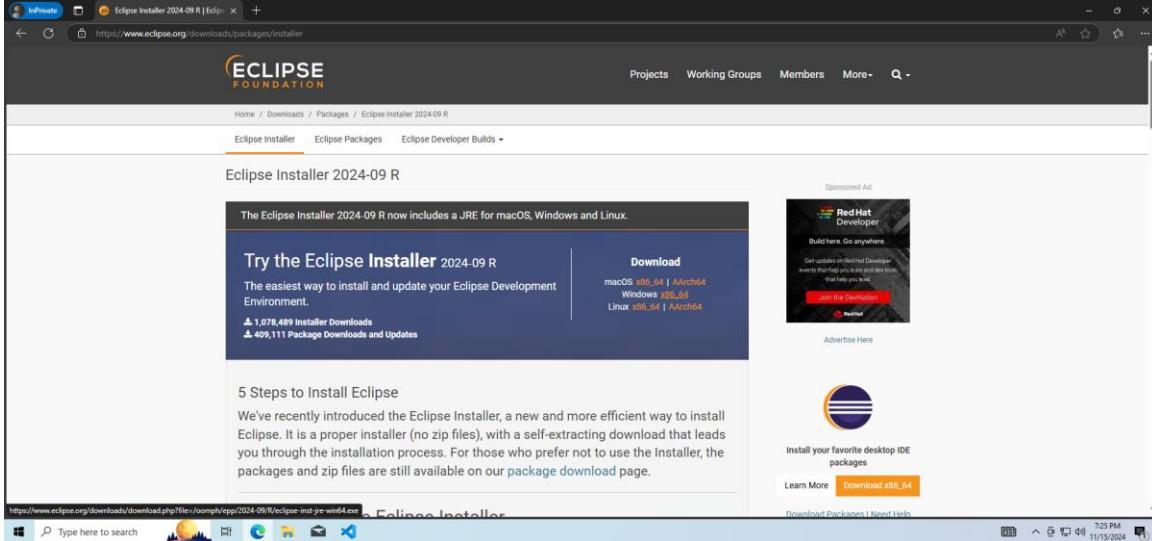


Step 5: Check maven installation

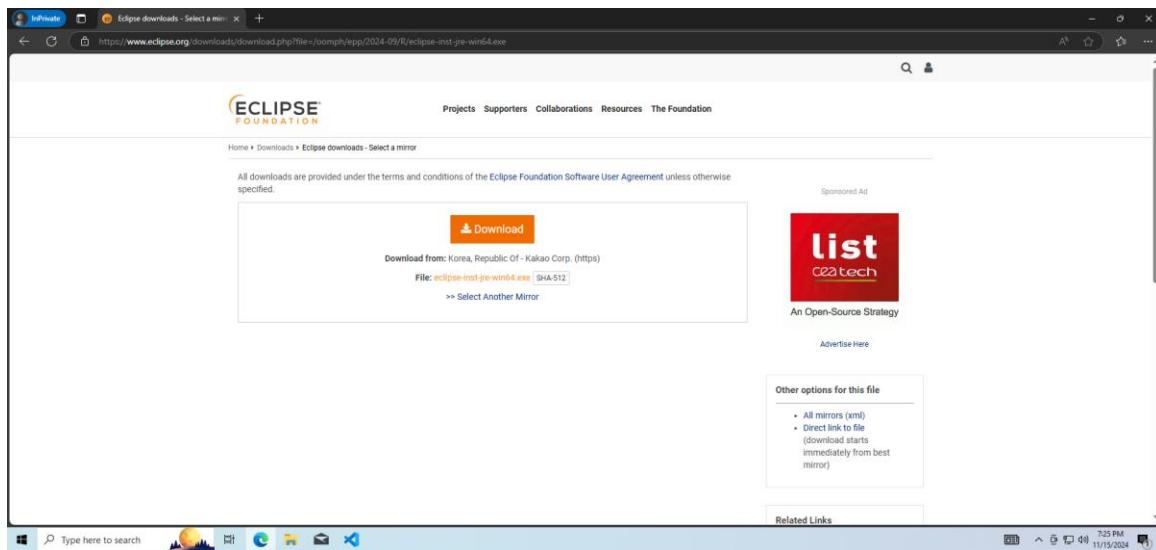


2. Eclipse Installation

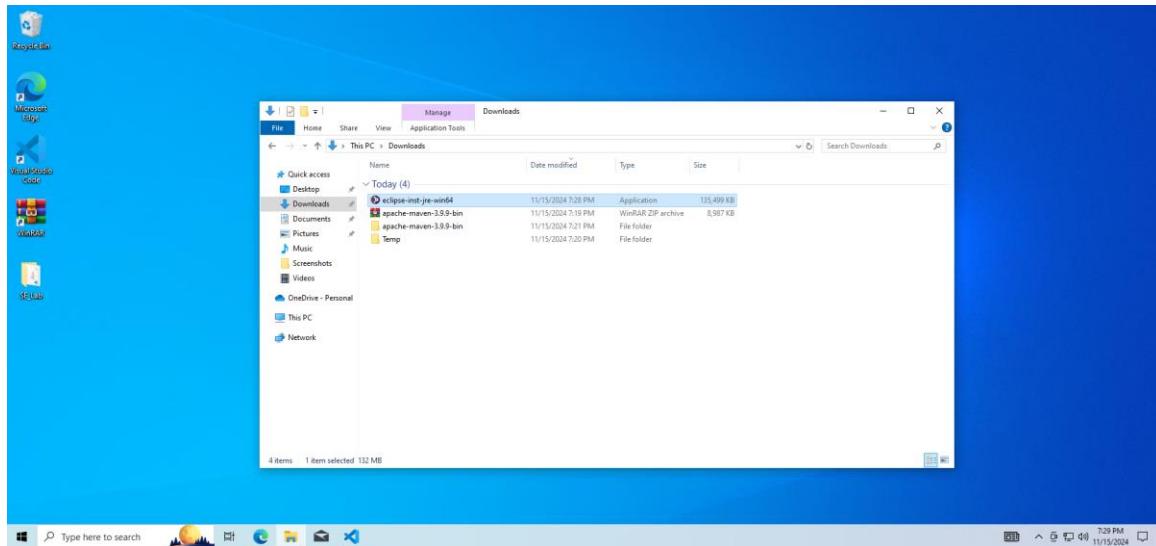
Step 1: Go to <https://eclipse.org/downloads>, and click on Download x86_64 to begin installation.



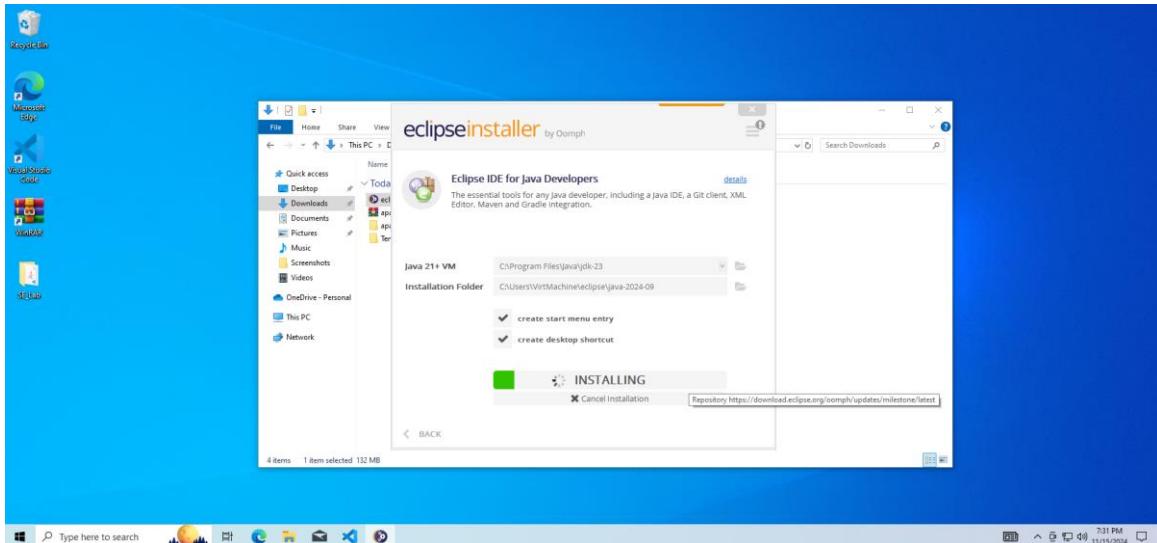
Step 2: Click download again on the next page.



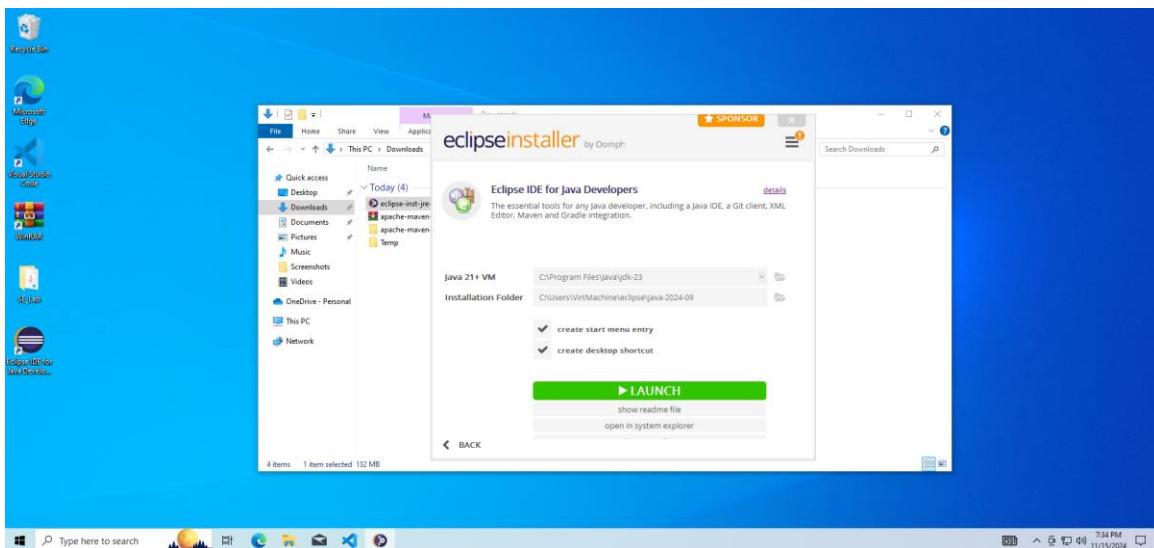
Step 3: Run the installer from the Downloads folder and click on “Eclipse IDE for Java Developers”.



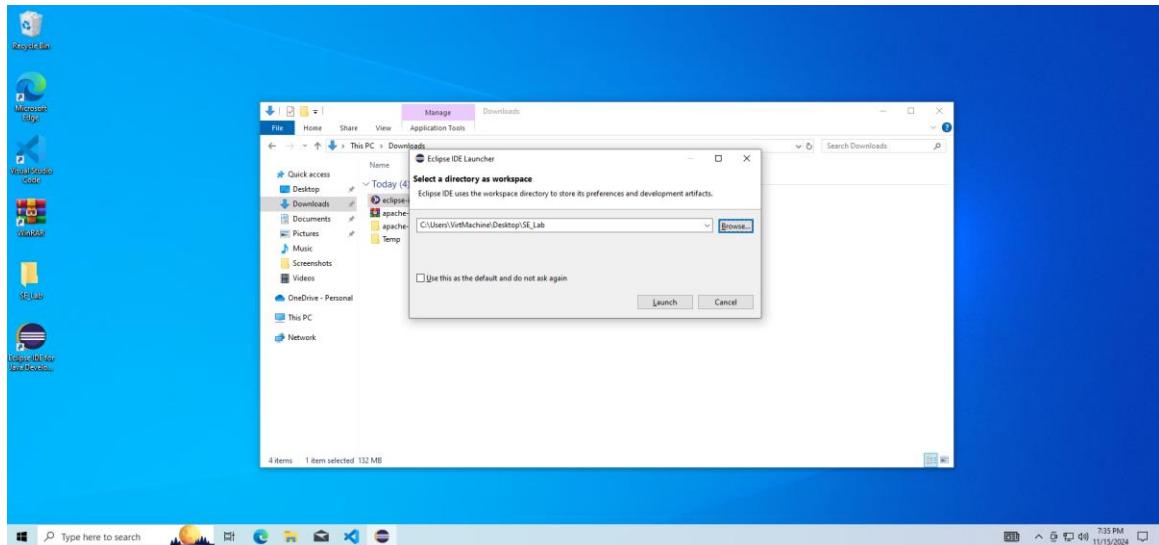
Step 4: Select the installation folder in the next page, and click on Install.



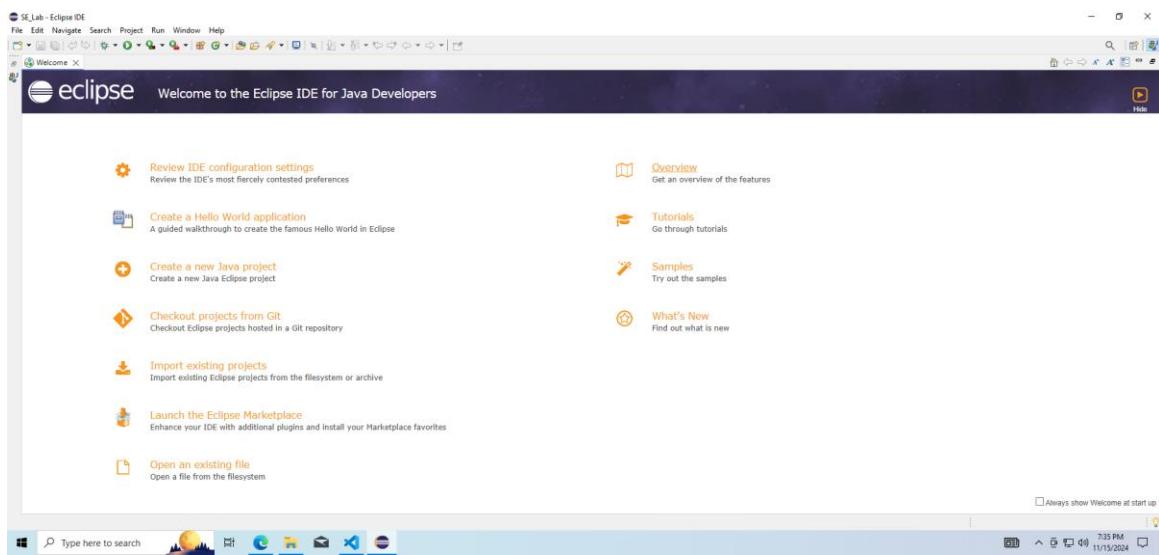
Step 5: Click on “Launch” after the installation is done.



Step 6: Select a directory as workspace for Eclipse and click on “Launch”.

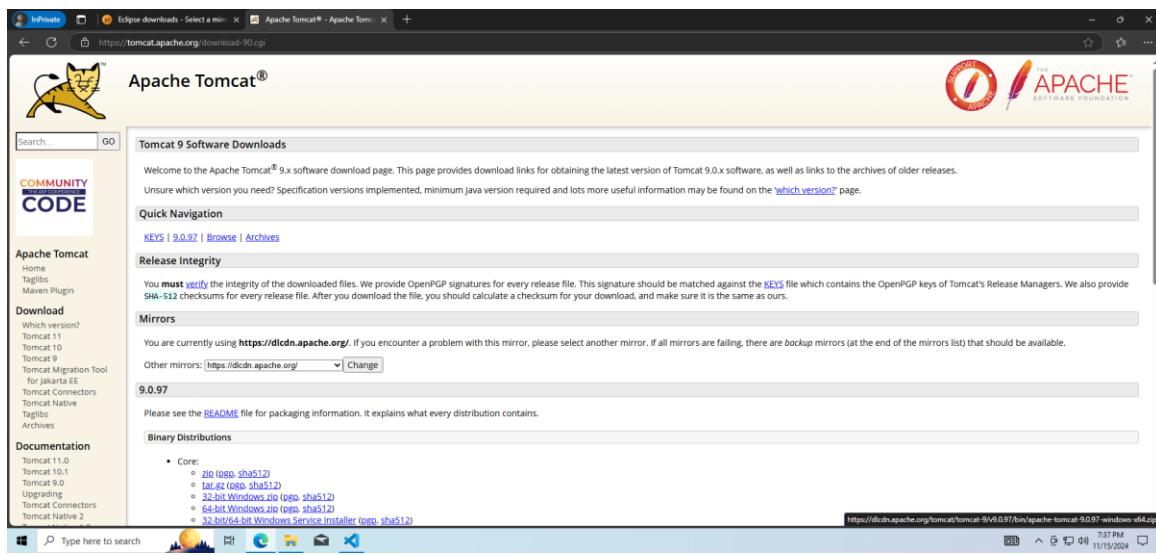


Step 7: You will see the welcome page in the Eclipse IDE

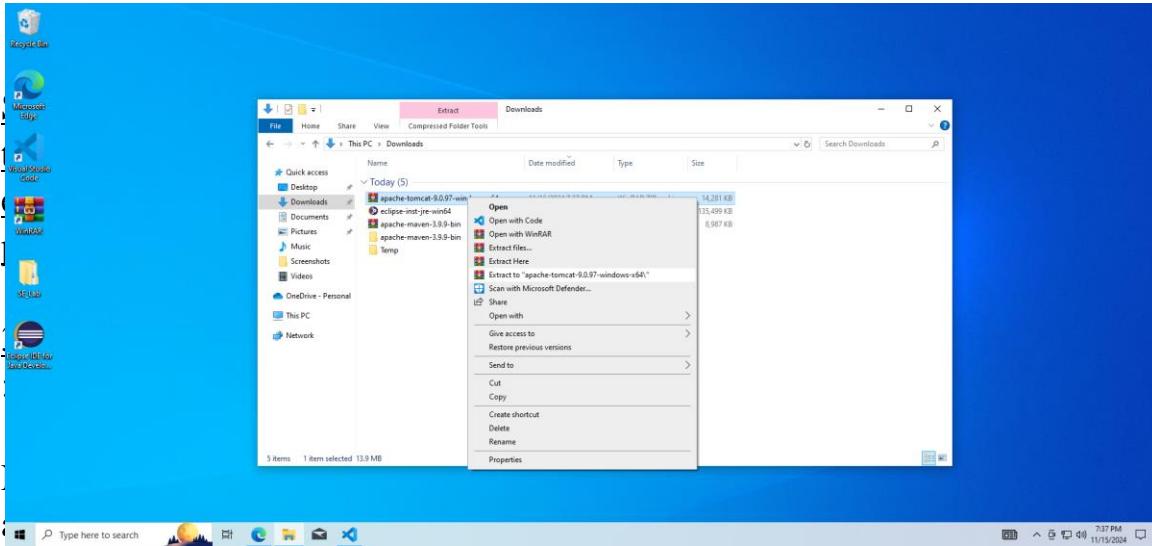


3. Tomcat Installation

Step 1: Go to <https://tomcat.apache.org/download-90.cgi>, and download the “64-bit Windows zip” folder for Tomcat.



Step 2: Unzip the apache-tomcat-9.0.97-windows-x64 folder by right clicking it and click on extract all and selecting the destination folder.



Navigate to the bin folder inside the extracted tomcat folder and run startup as shown.

```

PS C:\Users\VirtMachine\Downloads\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\bin> .\startup.bat
Using CATALINA_HOME: "C:\Users\VirtMachine\Downloads\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97"
Using CATALINA_TMPDIR: "C:\Users\VirtMachine\Downloads\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\temp"
Using ORE_HOME:
Using CATALINA_JAR: "C:\Program Files\Java\jdk-23"
Using CLASSPATH: "C:\Users\VirtMachine\Downloads\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\bin\bootstrap.jar;C:\Users\VirtMachine\Downloads\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\bin\tomcat-juli.jar"
Using CATALINA_OPTS: ""
PS C:\Users\VirtMachine\Downloads\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\bin>

```

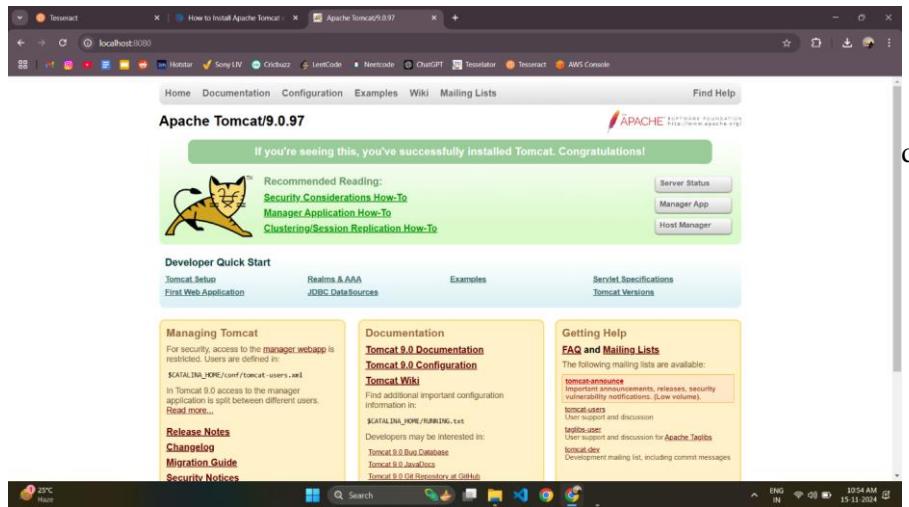
```

15-Nov-2024 19:43:27.067 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\Users\VirtMachine\Downloads\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\webapps\docs]
15-Nov-2024 19:43:27.528 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application in directory [C:\Users\VirtMachine\Downloads\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\webapps\docs] has finished in [453] ms
15-Nov-2024 19:43:27.528 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\Users\VirtMachine\Downloads\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\webapps\host-manager]
15-Nov-2024 19:43:28.008 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application in directory [C:\Users\VirtMachine\Downloads\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\webapps\host-manager] has finished in [678] ms
15-Nov-2024 19:43:28.009 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\Users\VirtMachine\Downloads\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\webapps\host-manager]
15-Nov-2024 19:43:28.160 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application in directory [C:\Users\VirtMachine\Downloads\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\webapps\host-manager] has finished in [602] ms
15-Nov-2024 19:43:28.160 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\Users\VirtMachine\Downloads\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\webapps\host-manager]
15-Nov-2024 19:43:28.223 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\Users\VirtMachine\Downloads\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\webapps\ROOT]
15-Nov-2024 19:43:28.254 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application in directory [C:\Users\VirtMachine\Downloads\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\webapps\ROOT] has finished in [31] ms
15-Nov-2024 19:43:28.270 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8080"]
15-Nov-2024 19:43:28.301 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in [1236] milliseconds

```

Step 4: The following window appears showing logs for tomcat. The tomcat server is running now!

Step 5: Navigate to <http://localhost:8080>. If you see the following, your installation was successful.

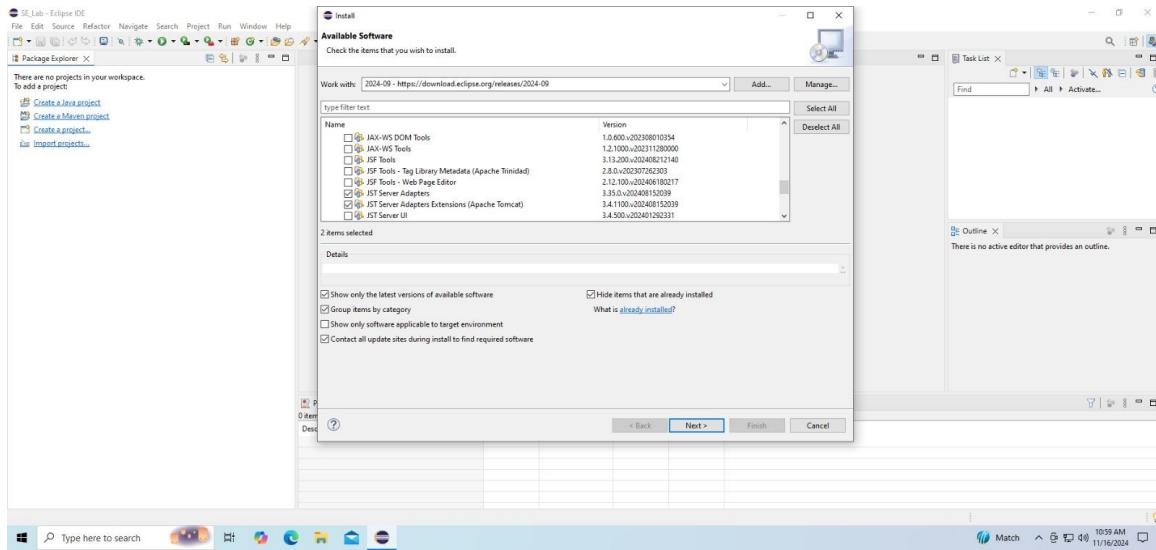


ddg

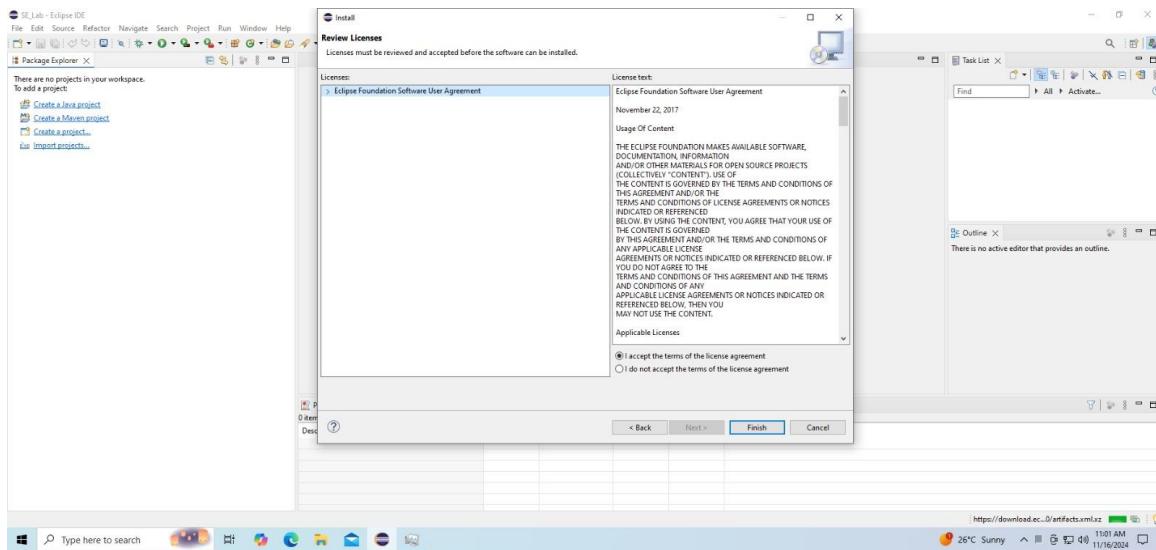
Step 6: Click on CTRL+C in the tomcat window, to stop the tomcat server.

4. Maven and Tomcat configuration

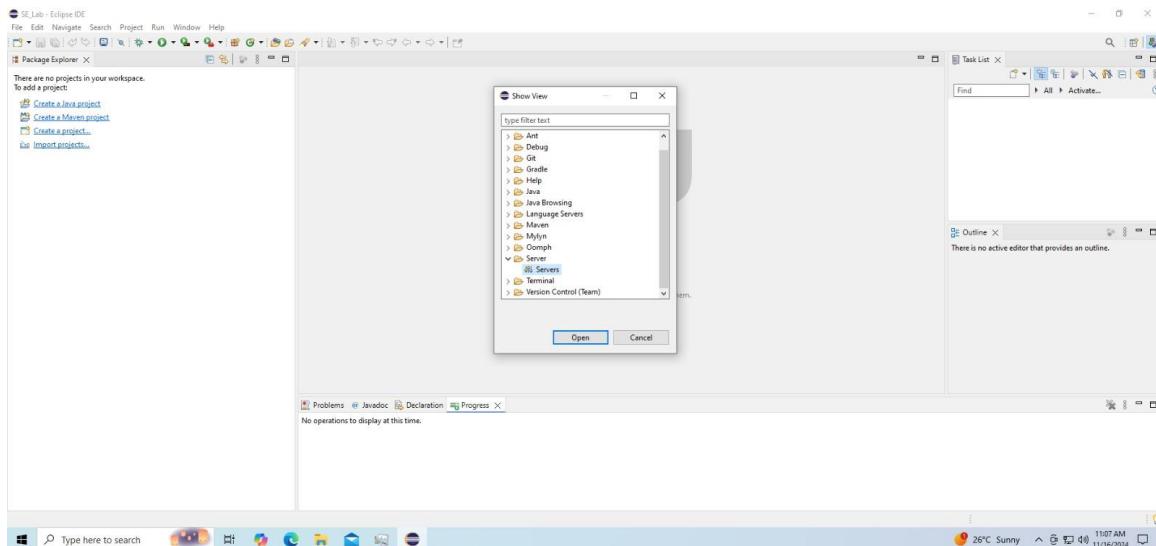
Step 1: Install Java Serverlet and Tomcat Server updates.



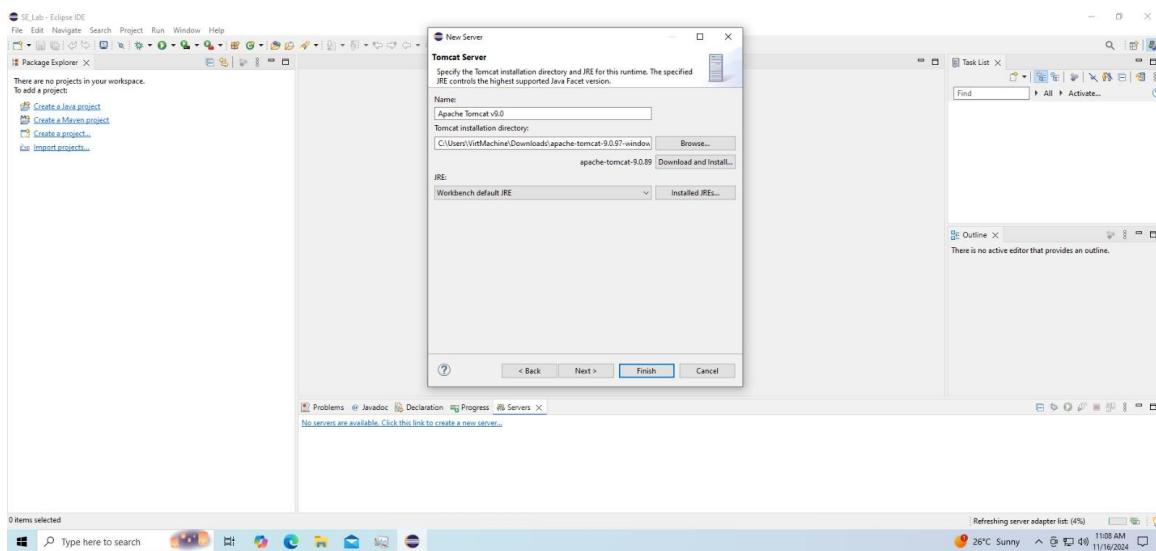
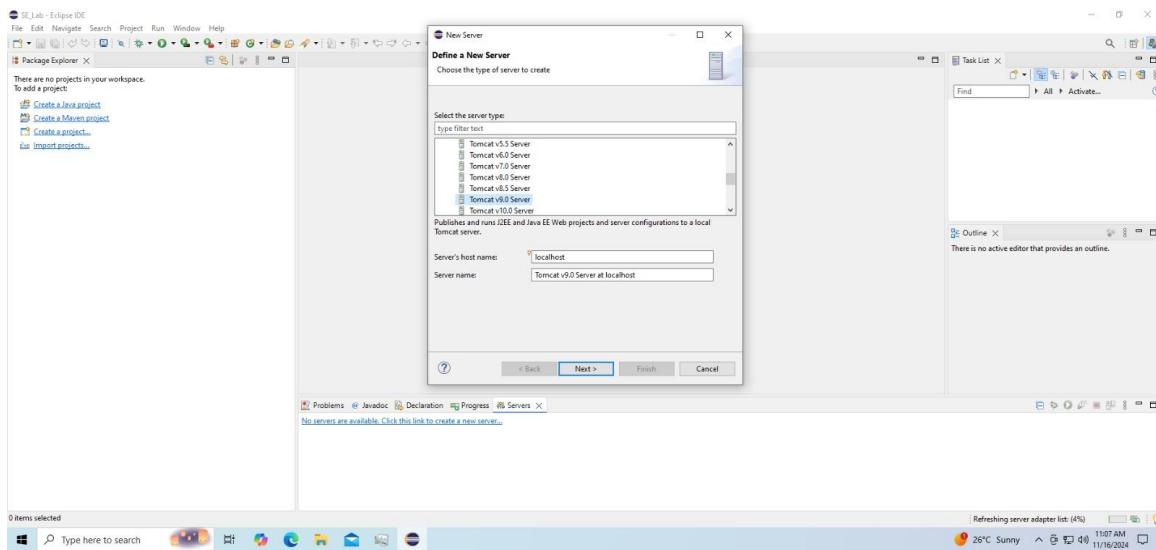
Step 2: Click on finish and restart Eclipse.



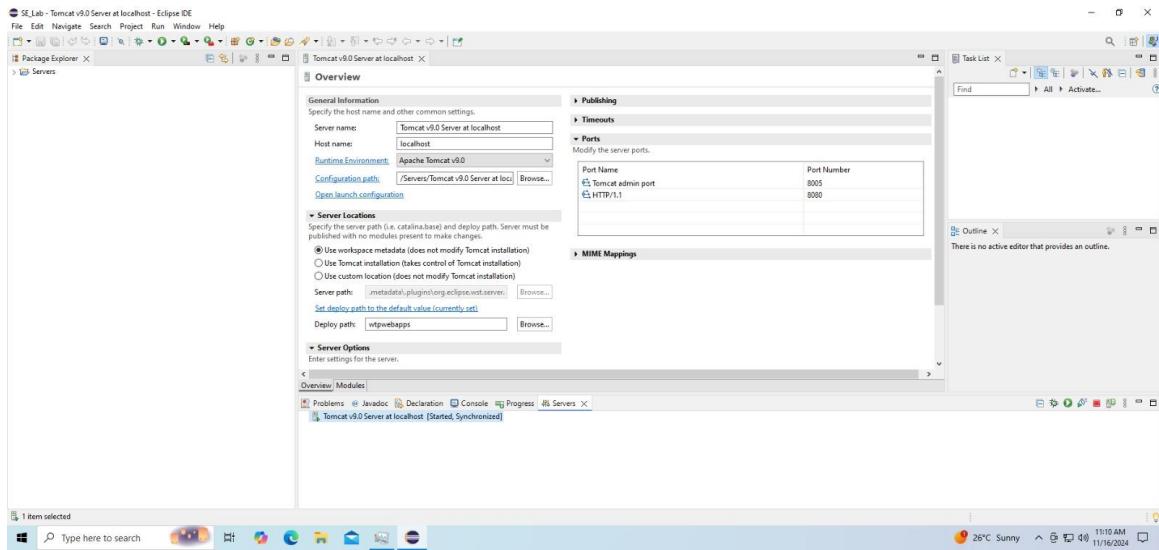
Step 3: Click on Servers in the Other tab in the Window tab.



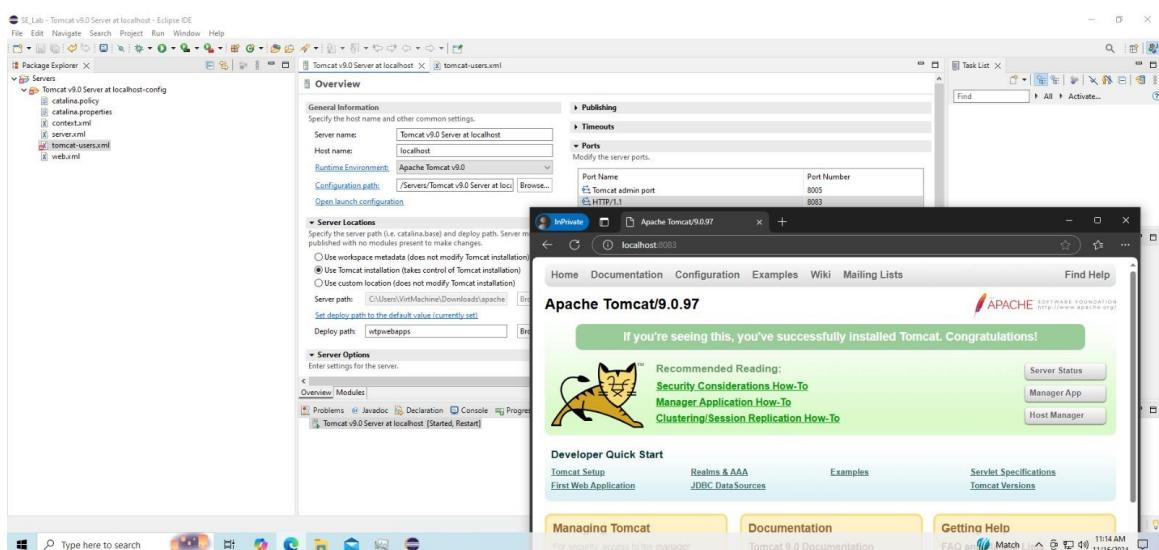
Step 5: Select your Tomcat version



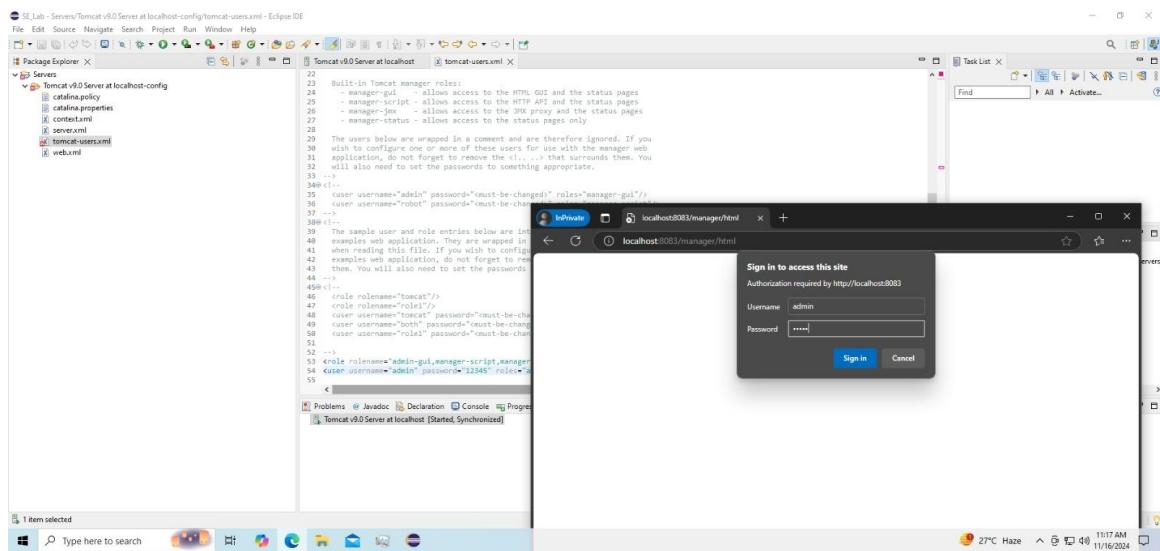
Step 6: Click on start server for Tomcat



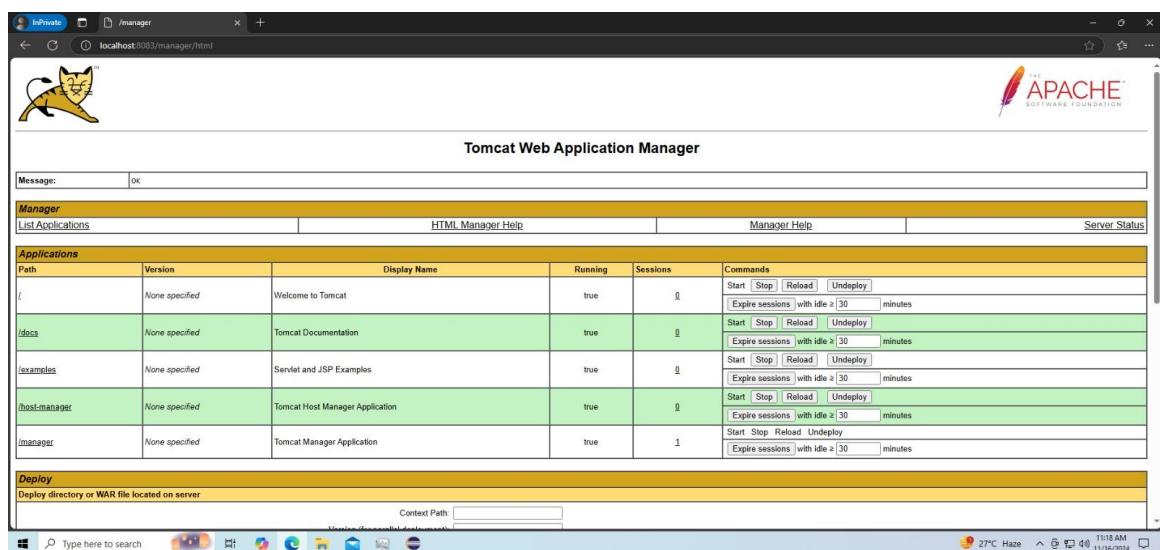
Step 7: The Tomcat Server is started on port 8083



Step 8: Modify tomcat-users.xml and add new user



Step 8: In the web browser, select manage apps, then login with your credentials.



Result: Eclipse, Maven and Tomcat were installed and Tomcat was successfully configured with Eclipse.

Experiment-4: Creating the project, and committing it using Git and GitHub

Aim: To explore Git Commands, create a repository on GitHub and committing a project using Git and GitHub.

Introduction:

1. **Git:** A distributed version control system that facilitates efficient management of code repositories, tracking changes, and collaboration across teams.
2. **GitHub:** An online platform for hosting Git repositories, enabling developers to collaborate on projects, track changes, and manage contributions seamlessly.
3. **Workflow Overview:**
 - o Initialize a local repository using Git.
 - o Create or modify files in the project.
 - o Stage and commit changes locally.
 - o Connect the local repository to GitHub using a remote URL.
 - o Push the project to GitHub for sharing and collaboration.

1. `git init`

```
MINGW64:/c/Users/Anand
$ Anand@DESKTOP-OHNEFJA MINGW64 ~ (se)
$ git --version
git version 2.46.1.windows.1
$ git config --global user.name "Anand"
$ git config --global user.email "anand@gmail.com"
$ git init
Reinitialized existing Git repository in C:/Users/Anand/.git/
$
```

2. git add and git status

```
MINGW64:/e/dlvs
$ Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
$ git add complete-bert.ipynb
$ Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
$
```

3. git commit

```
MINGW64:/e/dlvs
$ Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
$ git add complete-bert.ipynb
$ git commit
hint: Waiting for your editor to close the file...          0 [sig] bash 286! sigp
cket::process: Suppressing signal 18 to win32 process (pid 4704)
Aborting commit due to empty commit message.
$ Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
$
```

4. git diff

```
MINGW64:/e/dlvs
```

```
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
```

```
$ git diff
```

```
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
```

```
$ git status
```

```
On branch master
```

```
Untracked files:
```

```
(use "git add <file>,..." to include in what will be committed)
```

```
    SRS Team-13 (22R01A6611).docx
```

```
    Unit I - All Topics Combined.pdf
```

```
    complete-bertrand.ipynb
```

```
    complete.ipynb
```

```
    dlvs_Lab2.ipynb
```

```
    dlvs_Lab3.ipynb
```

```
    dlvs_Lab4.ipynb
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

```
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
```

```
$ git diff
```

```
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
```

5. git diff and git commit

```
MINGW64:/e/dlvs
```

```
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
```

```
$ git add complete-bertrand.ipynb
```

```
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
```

```
$ git commit
```

```
hint: Waiting for your editor to close the file...          0 [sig] bash 286! sigpa
```

```
cket::process: Suppressing signal 18 to win32 process (pid 4704)
```

```
Aborting commit due to empty commit message.
```

```
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
```

```
$
```

6. git branch -a

```
MINGW64:/e/dlvs
```

```
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
```

```
$ git branch
```

```
* master
```

```
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
```

```
$ git branch se
```

```
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
```

```
$ git branch
```

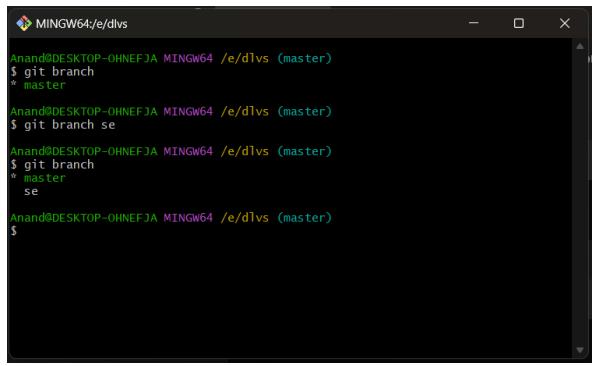
```
* master
```

```
  se
```

```
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
```

```
$
```

7. git branch



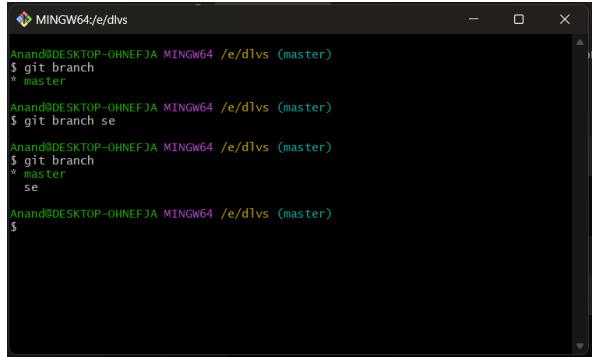
```
MINGW64:/e/dlvs
$ git branch
* master

Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
$ git branch se
* master
  se

Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
$ git branch -d se
* master

Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
$
```

8. git branch -d



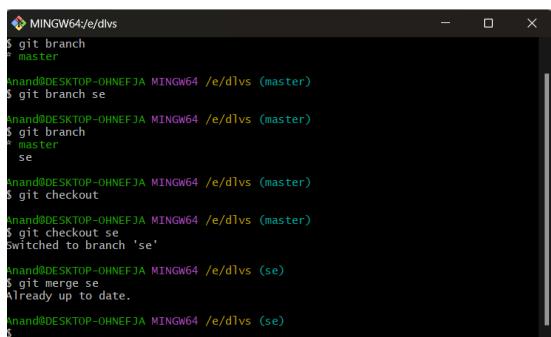
```
MINGW64:/e/dlvs
$ git branch
* master

Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
$ git branch se
* master
  se

Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
$ git branch -d se
* master

Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
$
```

11. git merge



```
MINGW64:/e/dlvs
$ git branch
* master

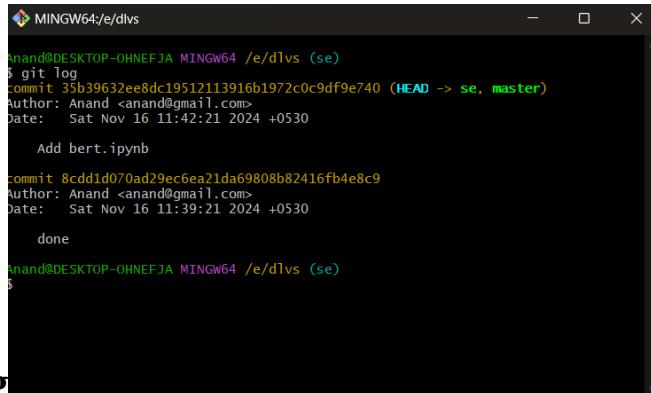
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
$ git branch se
* master
  se

Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (master)
$ git checkout se
* master
  se

Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (se)
$ git merge master
Auto-merging dlvs.c
CONFLICT (content): Merge conflict in dlvs.c
Automatic merge failed; fix conflicts and then commit the result.
* master
  se

Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (se)
$ git merge se
Already up to date.

Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (se)
$
```



```
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (se)
$ git log
commit 35b39632ee8dc19512113916b1972c0c9df9e740 (HEAD -> se, master)
Author: Anand <anand@gmail.com>
Date:   Sat Nov 16 11:42:21 2024 +0530

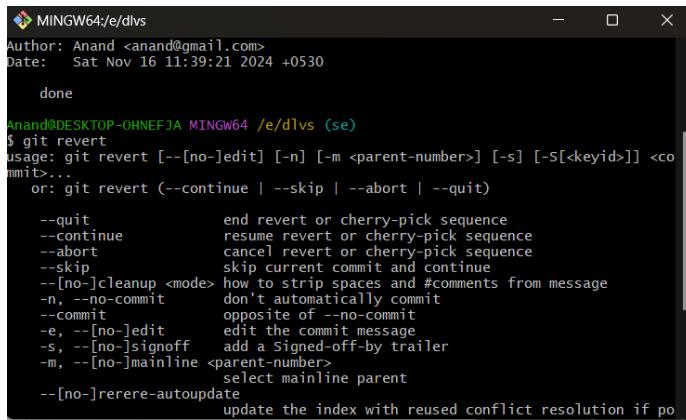
    Add bert.ipynb

commit 8cdd1d070ad29ec6ea21da69808b82416fb4e8c9
Author: Anand <anand@gmail.com>
Date:   Sat Nov 16 11:39:21 2024 +0530

    done
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (se)
$
```

12. git log

13. git revert

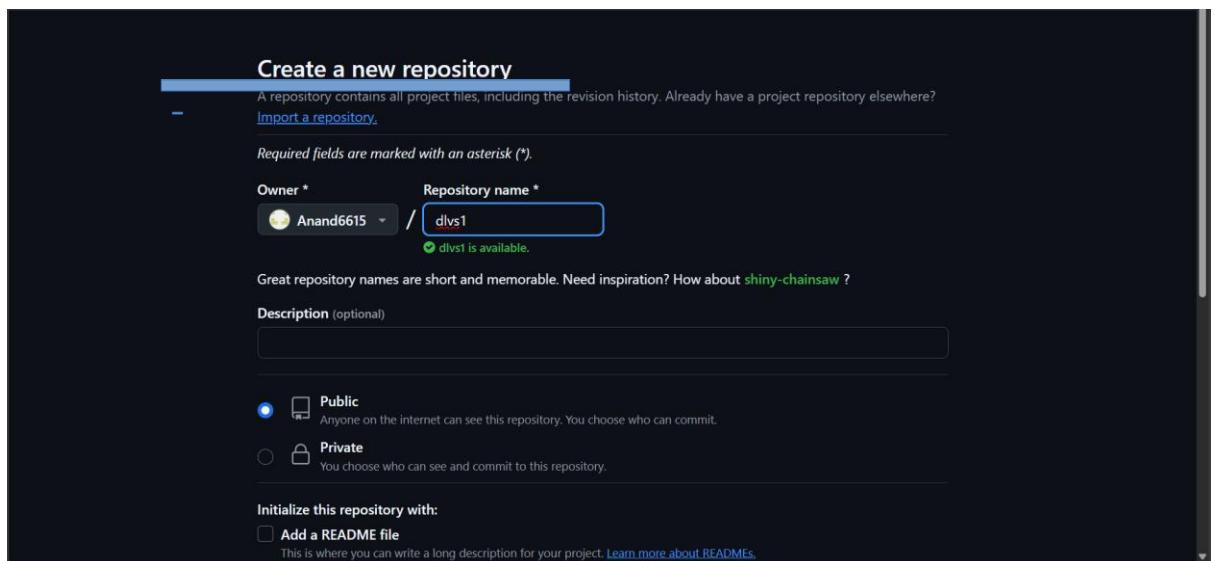


```
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (se)
Author: Anand <anand@gmail.com>
Date:   Sat Nov 16 11:39:21 2024 +0530

done
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (se)
$ git revert
usage: git revert [--no-edit] [-n] [-m <parent-number>] [-s] [-S<keyid>] <co
mmits>...
or: git revert (--continue | --skip | --abort | --quit)

--quit           end revert or cherry-pick sequence
--continue       resume revert or cherry-pick sequence
--abort          cancel revert or cherry-pick sequence
--skip           skip current commit and continue
--commit         opposite of --no-commit
-e, --no-edit    edit the commit message
-s, --no-signoff add a Signed-off-by trailer
-m, --no-mainline <parent-number>
                select mainline parent
--[no-]rerere-autoupdate
                update the index with reused conflict resolution if po
```

14. Create new GitHub repository



The screenshot shows the GitHub interface for creating a new repository. The title bar says "Create a new repository". Below it, a note says "A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)". A note below that says "Required fields are marked with an asterisk (*)".

The "Owner" field is set to "Anand6615". The "Repository name" field is set to "dlvs1", which is highlighted with a blue border and has a green checkmark next to it indicating it's available.

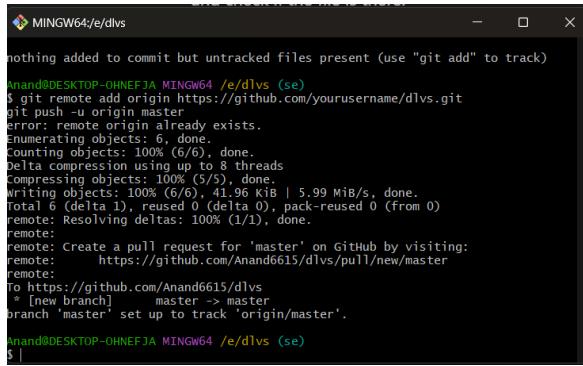
A note below the repository name says "Great repository names are short and memorable. Need inspiration? How about [shiny-chainsaw](#) ?".

The "Description (optional)" field is empty.

The "Visibility" section shows "Public" is selected (indicated by a blue circle), and "Anyone on the internet can see this repository. You choose who can commit." is displayed. The "Private" option is also shown with the note "You choose who can see and commit to this repository.".

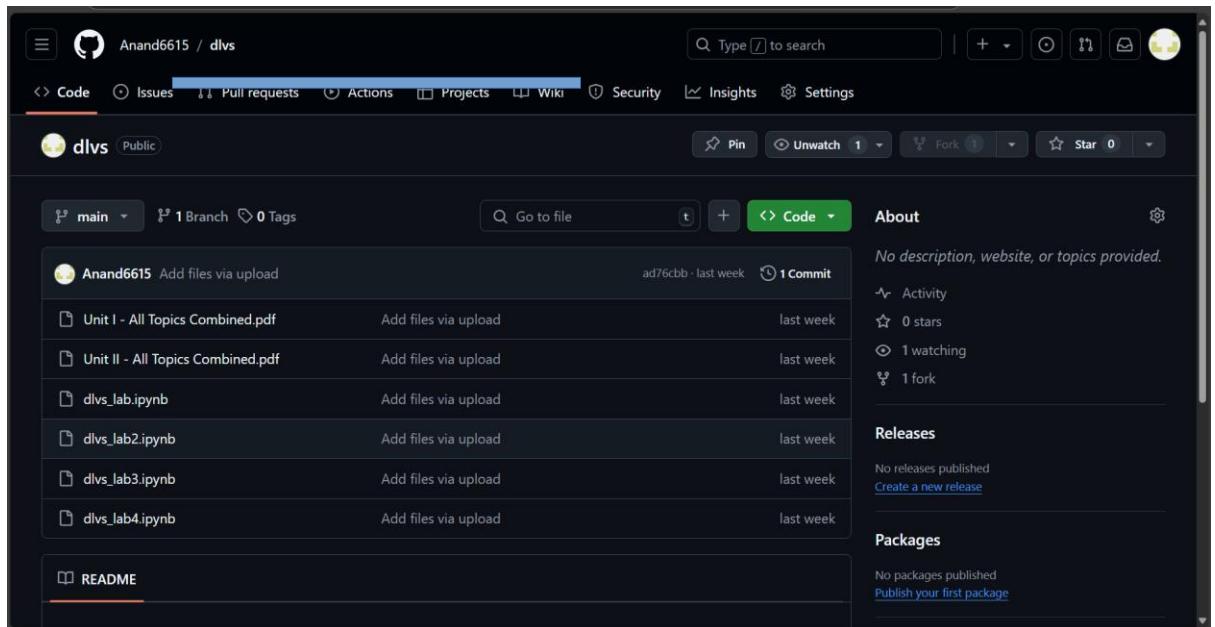
The "Initialize this repository with:" section has a checkbox for "Add a README file" which is unchecked. Below it, a note says "This is where you can write a long description for your project. [Learn more about READMEs.](#)"

15. git push



```
MINGW64:/e/dlvs
nothing added to commit but untracked files present (use "git add" to track)
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (se)
$ git remote add origin https://github.com/yourusername/dlvs.git
git push -u origin master
error: remote origin already exists.
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 41.96 KiB | 5.99 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
remote: Create a pull request for 'master' on GitHub by visiting:
remote: https://github.com/Anand6615/dlvs/pull/new/master
remote:
To https://github.com/Anand6615/dlvs
 * [new branch]    master -> master
branch 'master' set up to track 'origin/master'.
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (se)
$ |
```

16. After git push in GitHub



17. git clone

```
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (se)
$ git remote add origin https://github.com/Anand6615/dlvs

Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (se)
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Anand/.ssh/id_rsa):
/c/Users/Anand/.ssh/id_rsa already exists.
Overwrite (y/n)? y

Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (se)
$ cd "E:/sel"

Anand@DESKTOP-OHNEFJA MINGW64 /e/se1
$ git clone https://github.com/Anand6615/dlvs.git
Cloning into 'dlvs'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (8/8), 11.05 MiB | 964.00 KiB/s, done.

Anand@DESKTOP-OHNEFJA MINGW64 /e/se1
$
```

18. After `git clone`

19. `git pull`

```
raufu@LAPTOP-G21RPBHK MINGW64 ~/Desktop/test/Sample/Software-Engineering (main)
$ git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 956 bytes | 43.00 KiB/s, done.
From https://github.com/Armaan016/Software-Engineering
 * branch      main       -> FETCH_HEAD
   ce8a2ba..e3fe626 main       -> origin/main
Updating ce8a2ba..e3fe626
Fast-forward
 file.txt | 2 ++
 1 file changed, 1 insertion(+), 1 deletion(-)

raufu@LAPTOP-G21RPBHK MINGW64 ~/Desktop/test/Sample/Software-Engineering (main)
$
```

20. `git remote -v`

```

MINGW64/e/dlvs
  --[no-]cleanup <mode> how to strip spaces and #comments from message
  -n, --no-commit    don't automatically commit
  -c, --[no-]commit  opposite of --no-commit
  -e, --[no-]edit    edit the commit message
  -s, --[no-]signoff add a Signed-off-by trailer
  -m, --[no-]mainline <parent-number>
                     select mainline parent
  --[no-]rerere-autoupdate
                     update the index with reused conflict resolution if po
ssible
  --[no-]strategy <strategy>
                     merge strategy
  -X, --[no-]strategy-option <option>
                     option for merge strategy
  -S, --[no-]gpg-sign[=<key-id>]
                     GPG sign commit
  --[no-]reference   use the 'reference' format to refer to commits

Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (se)
$ git remote add origin https://github.com/Anand6615/dlvs
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (se)
$ |

```

21. Adding SSH Keys

```

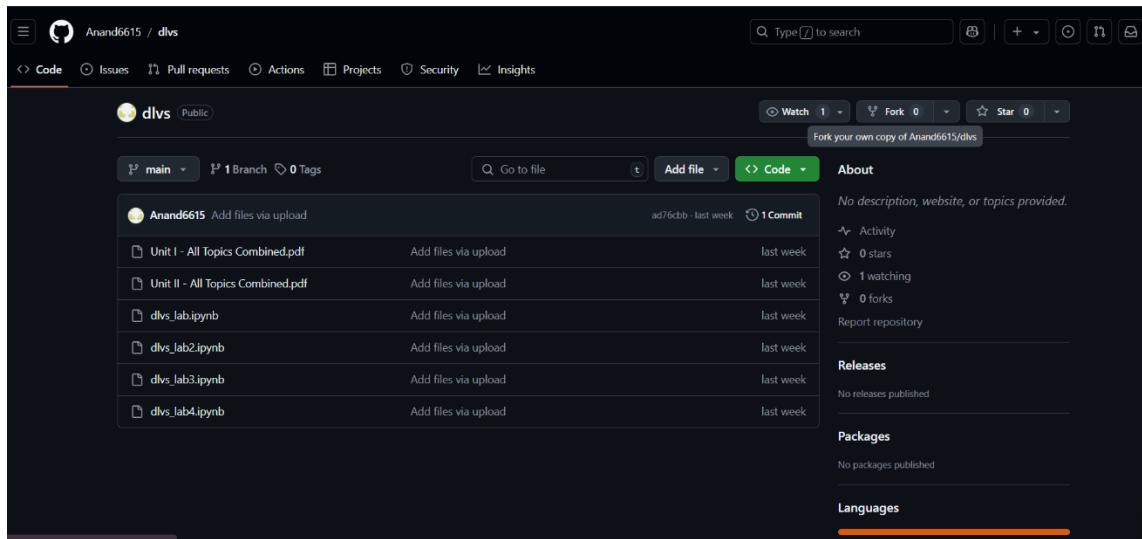
MINGW64/e/dlvs
  --[no-]rerere-autoupdate
                     update the index with reused conflict resolution if po
ssible
  --[no-]strategy <strategy>
                     merge strategy
  -X, --[no-]strategy-option <option>
                     option for merge strategy
  -S, --[no-]gpg-sign[=<key-id>]
                     GPG sign commit
  --[no-]reference   use the 'reference' format to refer to commits

Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (se)
$ git remote add origin https://github.com/Anand6615/dlvs
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (se)
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Anand/.ssh/id_rsa):
/c/Users/Anand/.ssh/id_rsa already exists.
Overwrite (y/n)?
Anand@DESKTOP-OHNEFJA MINGW64 /e/dlvs (se)
$ |

```

The screenshot shows the GitHub Settings interface for the user 'Anand (Anand6615)'. The left sidebar has sections like Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, Password and authentication, Sessions, and SSH and GPG keys (which is currently selected). The main area displays two sections: 'SSH keys' and 'GPG keys'. The 'SSH keys' section lists a single key named 'bert', which was added on Nov 6, 2024, and is last used within the last 2 weeks with Read/write permissions. The 'GPG keys' section indicates there are no GPG keys associated with the account.

22. Git Fork



23. git help

```
GIT-INIT(1)                               Git Manual                               GIT-INIT(1)

NAME
    git-init - Create an empty Git repository or reinitialize an existing one

SYNOPSIS
    git init [-q | --quiet] [--bare] [--template=<template-directory>]
              [--separate-git-dir=<git-dir>] [--object-format=<format>]
              [--ref-format=<format>]
              [-b <branch-name> | --initial-branches=<branch-name>]
              [--shared[=<permissions>]] [<directory>]

DESCRIPTION
    This command creates an empty Git repository - basically a .git directory with subdirectories for objects, refs/heads, refs/tags, and template files. An initial branch without any commits will be created (see the --initial-branch option below for its name).

    If the $GIT_DIR environment variable is set then it specifies a path to use instead of ./git for the base of the repository.

    If the object storage directory is specified via the $GIT_OBJECT_DIRECTORY environment variable then the sha1 directories are created underneath; otherwise, the default $GIT_DIR/objects directory is used.

    Running git init in an existing repository is safe. It will not overwrite things that are already there. The primary reason for rerunning git init is to pick up newly added templates (or to move the repository to another place if --separate-git-dir is given).

OPTIONS
    -q, --quiet
        Only print error and warning messages; all other output will be suppressed.

    --bare
        Create a bare repository. If GIT_DIR environment is not set, it is set to the current working directory.

    --object-format=<format>
        Specify the given object <format> (hash algorithm) for the repository. The valid values are sha1 and (if enabled) sha256. sha1 is the default.

        Note: At present, there is no interoperability between SHA-256 repositories and SHA-1 repositories.

    Historically, we warned that SHA-256 repositories may later need backward incompatible changes when we introduce such interoperability features. Today, we only expect compatible changes. Furthermore, if such changes prove to be necessary, it can be expected that SHA-256 repositories created with today's Git will be usable by future versions of Git without data loss.

    --ref-format=<format>
        Specify the given ref storage <format> for the repository. The valid values are:

        * files for loose files with packed-refs. This is the default.
```

Result: Git commands were explored, a project was successfully created using Git, and the repository was pushed to GitHub for collaborative development.

Experiment-5: Create Maven Java and Web Projects and Push them to GitHub

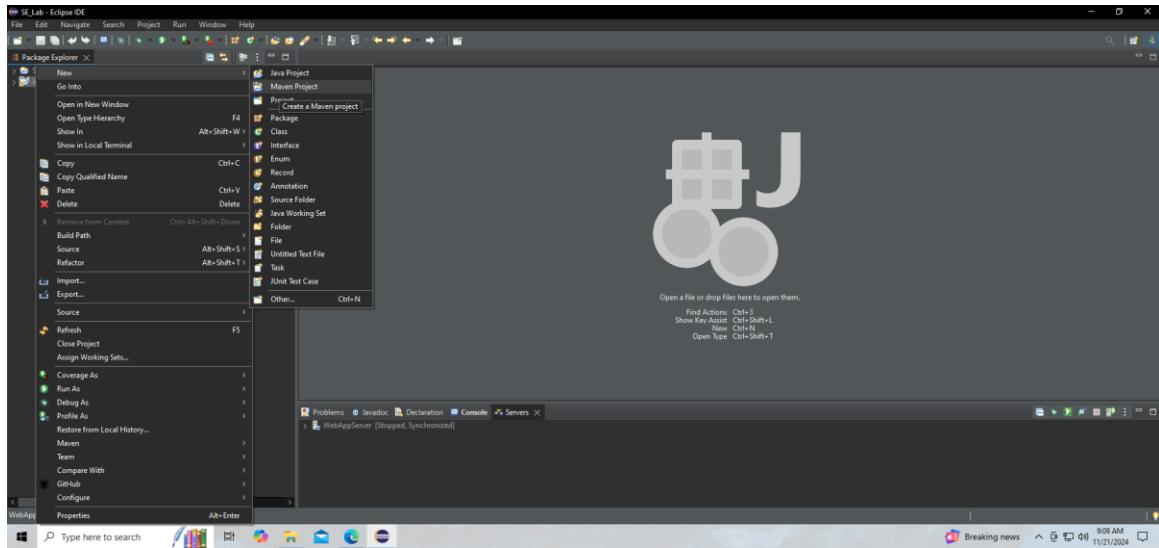
Aim: To install Maven, Tomcat, and Eclipse, configure Tomcat with Eclipse, create a Java and web project using Maven in Eclipse, and push the project to GitHub.

Introduction:

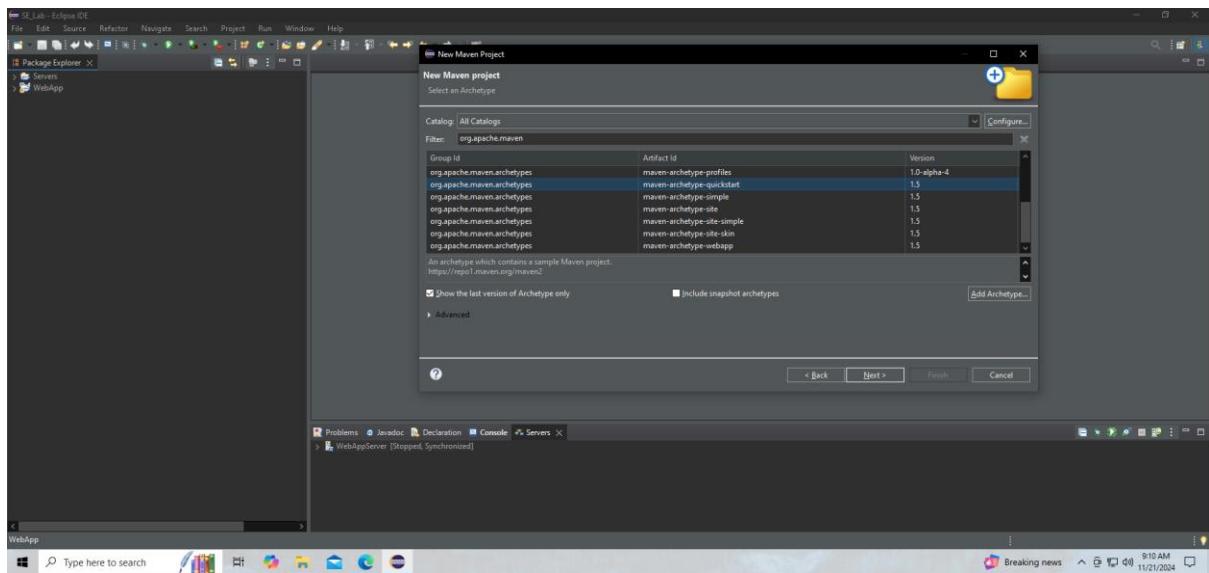
1. **Maven:** A powerful project management tool that simplifies the build process by automating dependency management, project structure setup, and build configurations. It follows a declarative approach through a pom.xml file.
2. **Tomcat:** An open-source web server and servlet container that supports the deployment of Java-based web applications. Configuring Tomcat with Eclipse enables seamless integration for development and testing.
3. **Eclipse:** A widely used Integrated Development Environment (IDE) for Java and other programming languages. It provides robust tools for creating, managing, and debugging applications.
4. **Workflow Overview:**
 - Install Maven, Tomcat, and Eclipse on the system.
 - Configure Tomcat with Eclipse for efficient web application development.
 - Use Maven to create a Java and web project in Eclipse, benefiting from dependency and build management.
 - Push the completed project to GitHub for version control and collaboration.

Creating Maven Java Project and pushing into GitHub

1. Creating new Maven Project

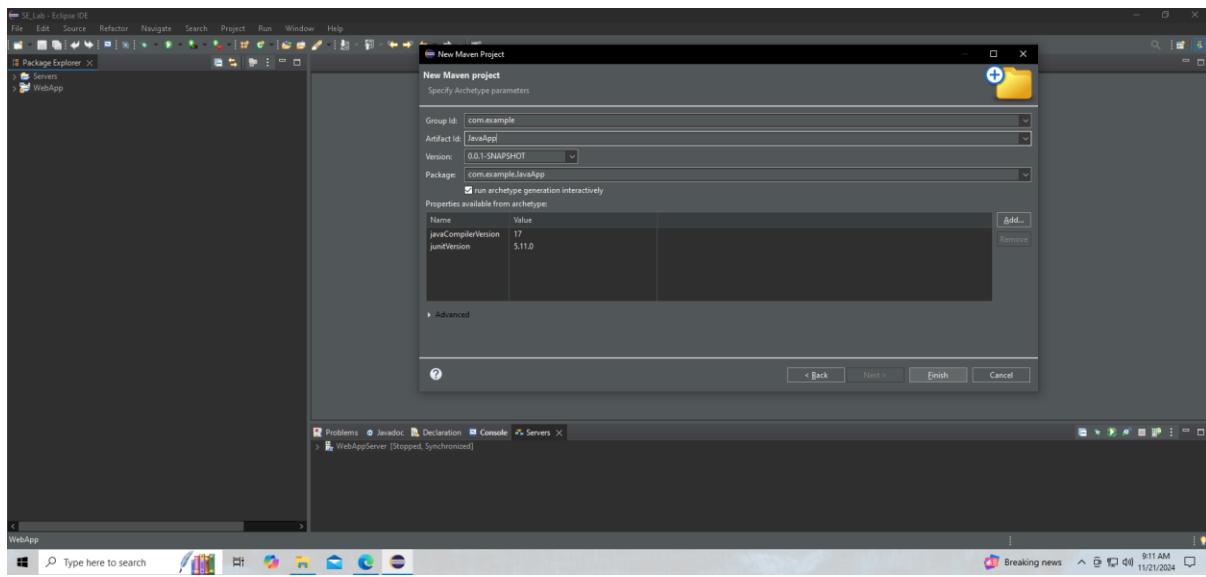


2. Selecting maven-archetype-quickstart

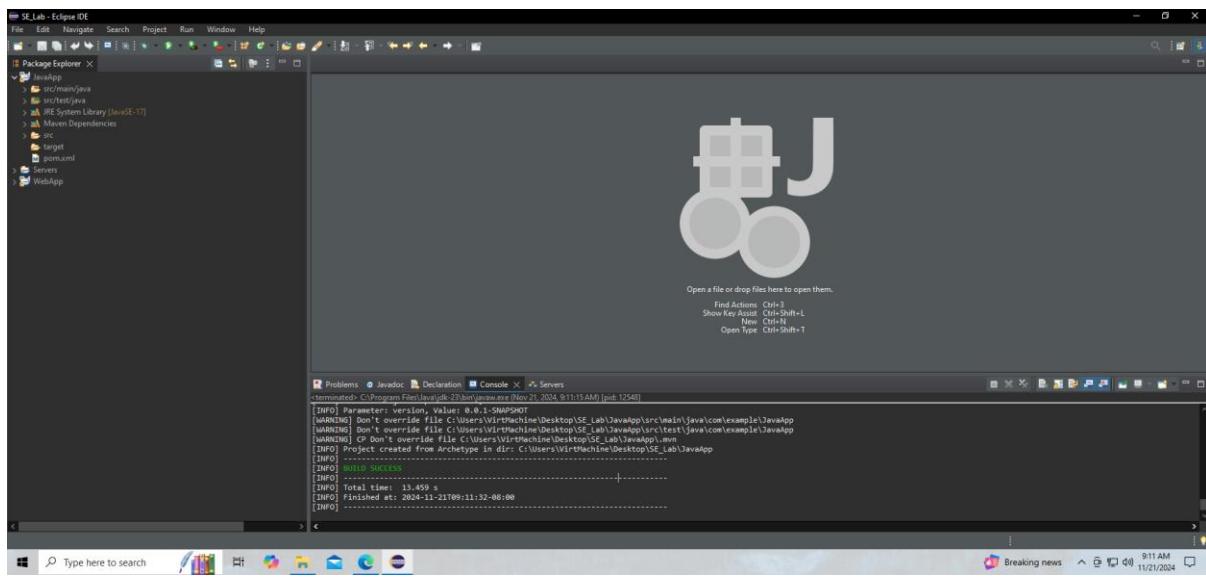


Group Id	Artifact Id	Version
org.apache.maven.archetypes	maven-archetype-profiles	1.0-alpha-4
org.apache.maven.archetypes	maven-archetype-quickstart	1.5
org.apache.maven.archetypes	maven-archetype-simple	1.5
org.apache.maven.archetypes	maven-archetype-site	1.5
org.apache.maven.archetypes	maven-archetype-site-simple	1.5
org.apache.maven.archetypes	maven-archetype-site-skin	1.5
org.apache.maven.archetypes	maven-archetype-webapp	1.5

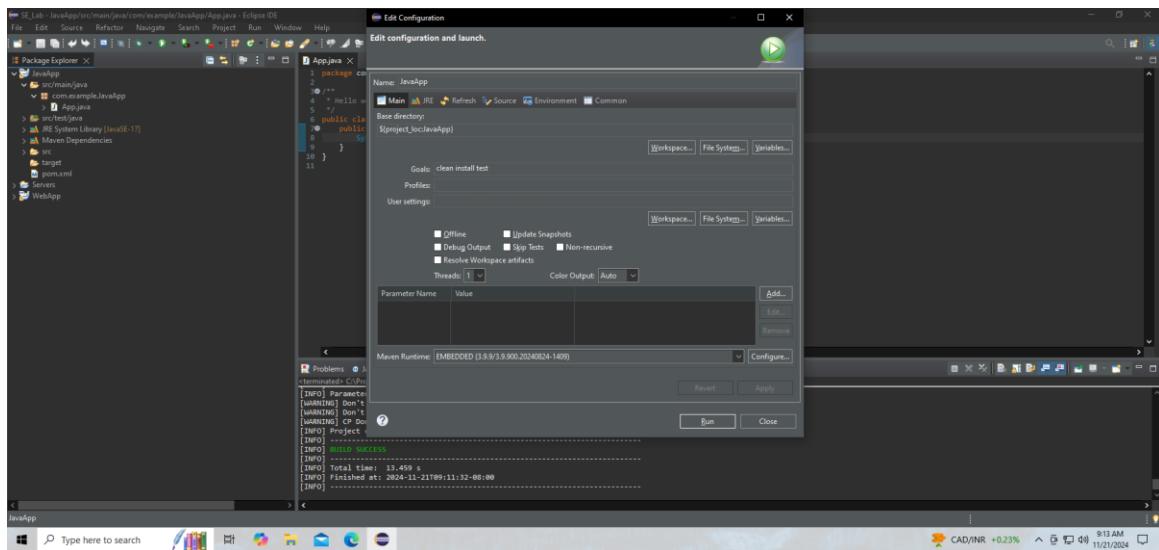
3. Creating new Group Id and Artifact Id



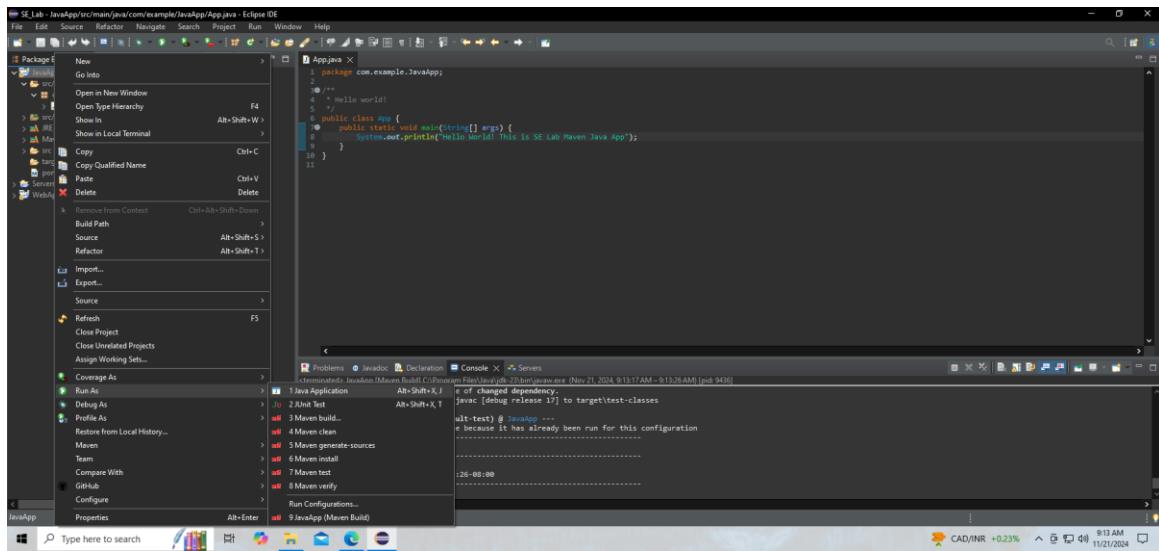
4. Project created



5. Creating Maven Build configuration



6. Run as Java Application



7. Output after running program

SE_Lab - JavaApp/src/main/java/com/example/javaApp/App.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X

JavaApp

src/main/java

com.example.javaApp

App.java

src/test/java

IREE System Library [Java8-17]

Maven Dependencies

pom.xml

Servers

WebApp

App.java

```
1 package com.example.javaApp;
2
3 /**
4 * Hello world!
5 */
6 public class App {
7     public static void main(String[] args) {
8         System.out.println("Hello World! This is SE Lab Maven Java App");
9     }
10 }
```

Problems Javadoc Declaration Console Servers

<terminated> App[Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 21, 2024, 9:10:49 AM - 9:10:49 AM) [pid: 8532]

Hello World! This is SE Lab Maven Java App

JavaApp Type here to search CAD/INR +0.23% 8:12 AM 11/21/2024

9. Pushing to GitHub

SE_Lab - JavaApp/src/main/java/com/example/javaApp/App.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X

JavaApp

src/main/java

com.example.javaApp

App.java

src/test/java

IREE System Library [Java8-17]

Maven Dependencies

pom.xml

Servers

WebApp

App.java

```
1 package com.example.javaApp;
2
3 /**
4 * Hello world!
5 */
6 public class App {
7     public static void main(String[] args) {
8         System.out.println("Hello World! This is SE Lab Maven Java App");
9     }
10 }
```

Problems Javadoc Declaration Console Terminal Servers

VirtMachine@DESKTOP-AV9VJUG MINGW64 /c/Users/VirtMachine/Desktop/SE_Lab/JavaApp

```
$ git init
Initialized empty Git repository in C:/Users/VirtMachine/Desktop/SE_Lab/JavaApp/.git/
$ git add .
$ git branch -m main
warning: In the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
$ git commit -m "Initial Commit"
[master (root-commit) 0000000] Initial Commit
 1 files changed, 1 insertions(+), 0 deletions(-)
$ git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>" to unstage)
  new file: .gitignore
  new file: .mvn/jme.conf
  new file: pom.xml
  new file: settings/org.eclipse.core.resources.prefs
  new file: settings/org.eclipse.jdt.core.prefs
  new file: settings/org.eclipse.m2e.core.prefs
  new file: pom.xml
```

Connected - Encoding: UTF-8 CAD/INR +0.23% 8:12 AM 11/21/2024

```
SE_Lab - JavaApp/src/main/java/com/example/JavaApp - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer X
JavaApp
  src/main/java
    com.example.JavaApp
      App.java
  src/test/java
  IRE System Library [JavaSE-17]
  Maven Dependencies
  .classpath
  .project
  pom.xml
  Servers
  WebApp

App.java X
1 package com.example.JavaApp;
2
3 /**
4 * Hello world!
5 */
6 public class App {
7     public static void main(String[] args) {
8         System.out.println("Hello World! This is SE Lab Maven Java App");
9     }
10 }

Problems Javadoc Declaration Console Terminal X Servers
MINGW64/c/Users/VirtMachine/Desktop/SE_Lab/JavaApp X
$ git commit -m "First commit"
[main (root-commit) f4dd222] First commit
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 .gitignore
 create mode 100644 .mvn/jmrc.config
 create mode 100644 .settings/org.eclipse.core.resources.prefs
 create mode 100644 .settings/org.eclipse.jst.core.prefs
 create mode 100644 .settings/org.eclipse.m2e.core.prefs
 create mode 100644 pom.xml
 create mode 100644 src/main/java/com/example/JavaApp/AppTest.java
 create mode 100644 test/java/com/example/JavaApp/AppTest.java
$ git remote add origin https://github.com/Thehejestic01/maven_java.git
$ git push -u
origin https://github.com/Thehejestic01/maven_java.git (fetch)
origin https://github.com/Thehejestic01/maven_java.git (push)
VirtMachine@DESKTOP-AV1JLQ MINGW64 ->/Desktop/SE_Lab/JavaApp (main)
$
```

Connected - Encoding: UTF-8

11. Successfully pushed into GitHub

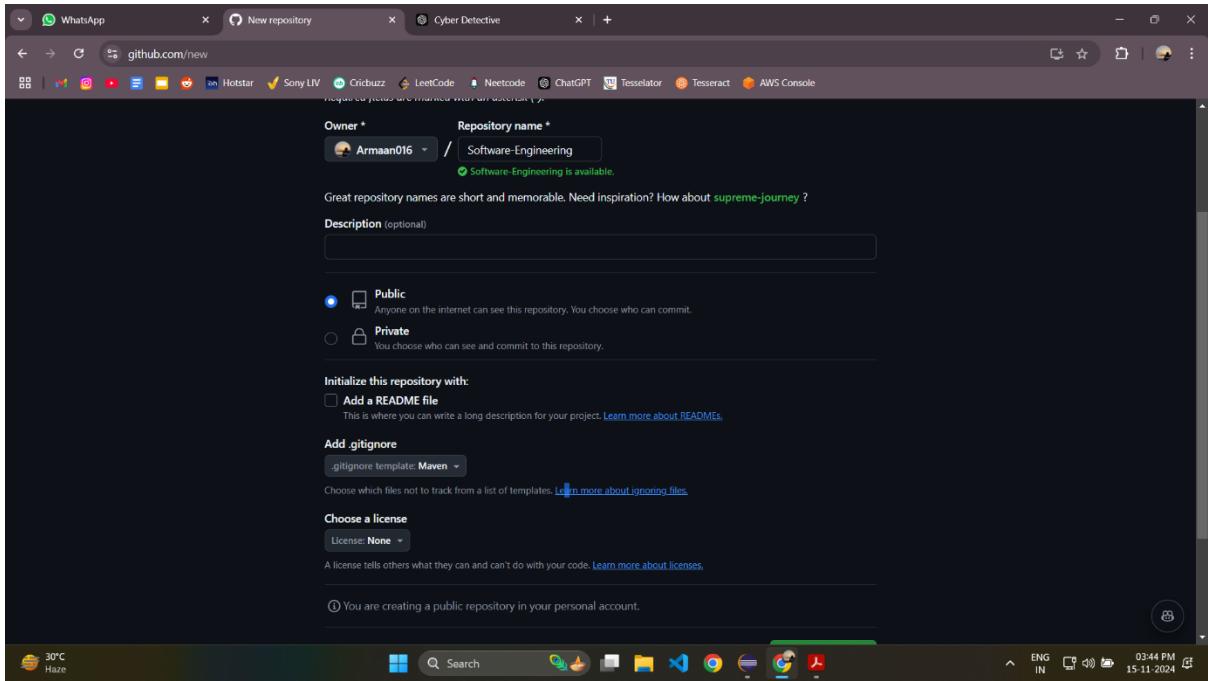
```
SE_Lab - JavaApp/src/main/java/com/example/JavaApp - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer X
JavaApp
  src/main/java
    com.example.JavaApp
      App.java
  src/test/java
  IRE System Library [JavaSE-17]
  Maven Dependencies
  .classpath
  .project
  pom.xml
  Servers
  WebApp

App.java X
1 package com.example.JavaApp;
2
3 /**
4 * Hello world!
5 */
6 public class App {
7     public static void main(String[] args) {
8         System.out.println("Hello World! This is SE Lab Maven Java App");
9     }
10 }

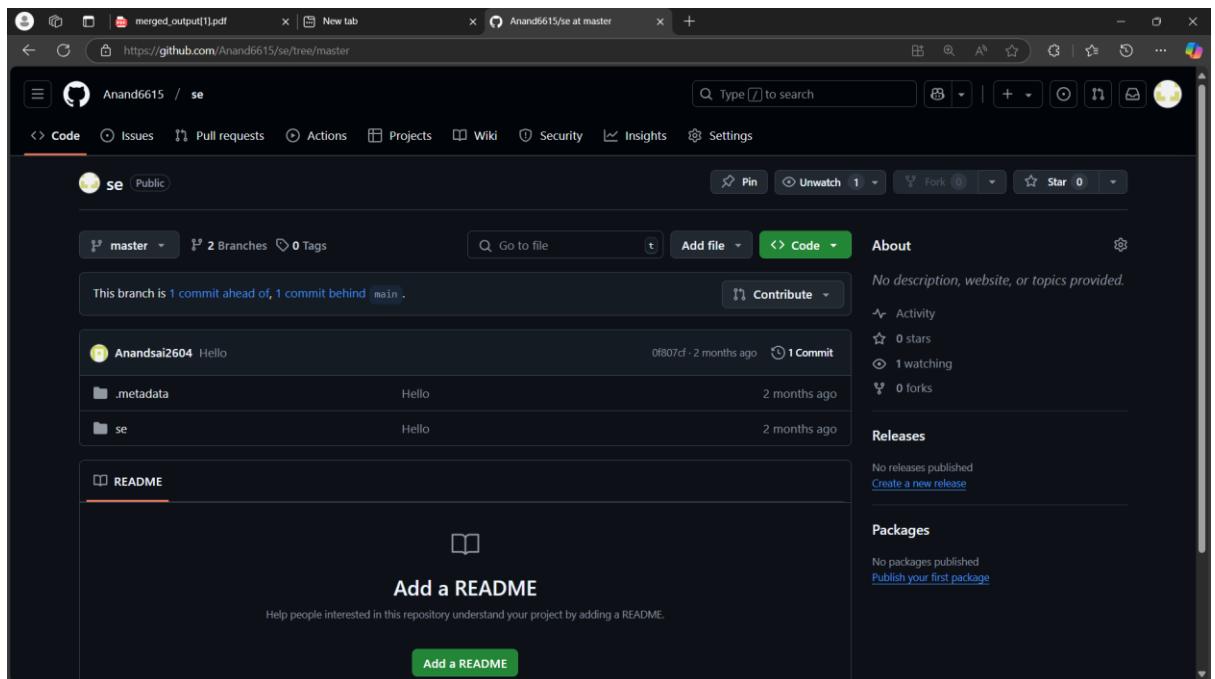
Problems Javadoc Declaration Console Terminal X Servers
MINGW64/c/Users/VirtMachine/Desktop/SE_Lab/JavaApp X
$ git push -u
Enumerating objects: 23, done.
Counting objects: 23, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (23/23), done.
Writing objects: 100% (23/23), done.
Total 23 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Thehejestic01/maven_java.git
 ! [new branch] main -> main (forced update)
branch 'main' set up to track 'origin/main'.
VirtMachine@DESKTOP-AV1JLQ MINGW64 ->/Desktop/SE_Lab/JavaApp (main)
$
```

Connected - Encoding: UTF-8

12. Creating new GitHub repository

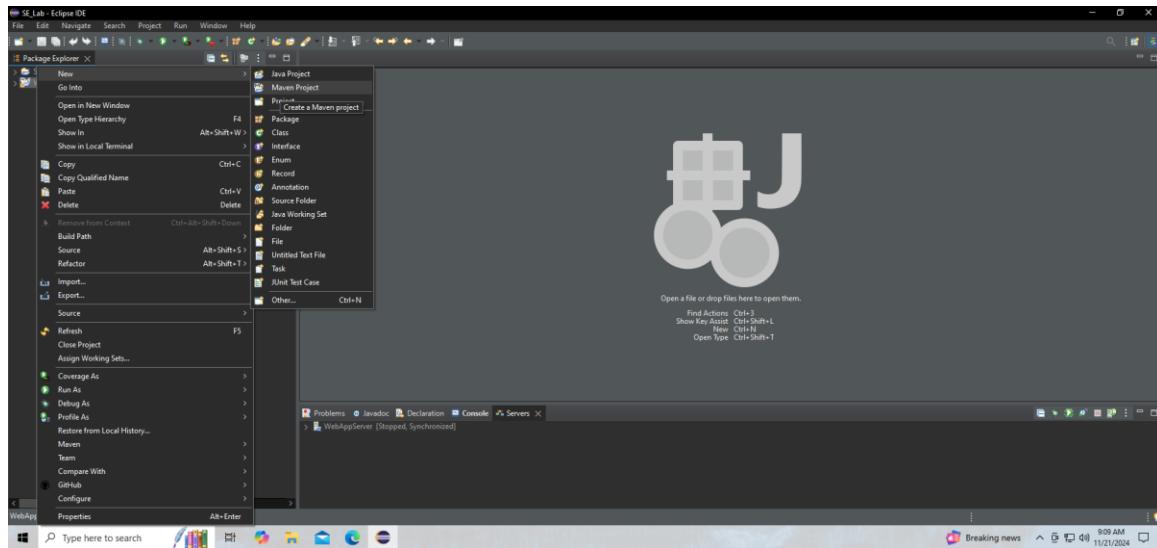


13. After pushing

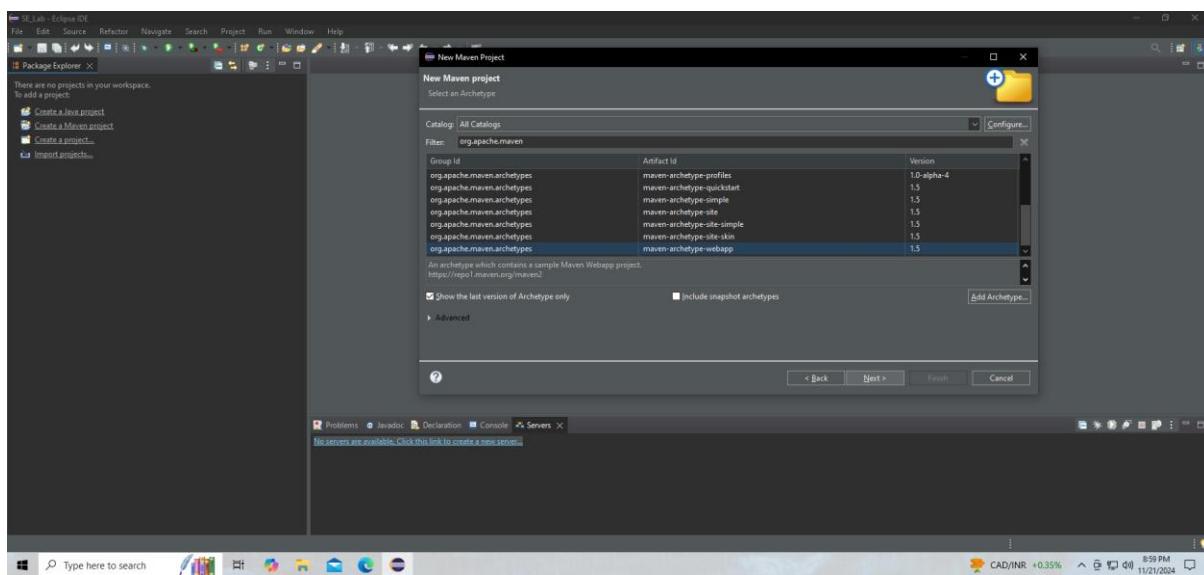


Creating Maven Web App and pushing into GitHub

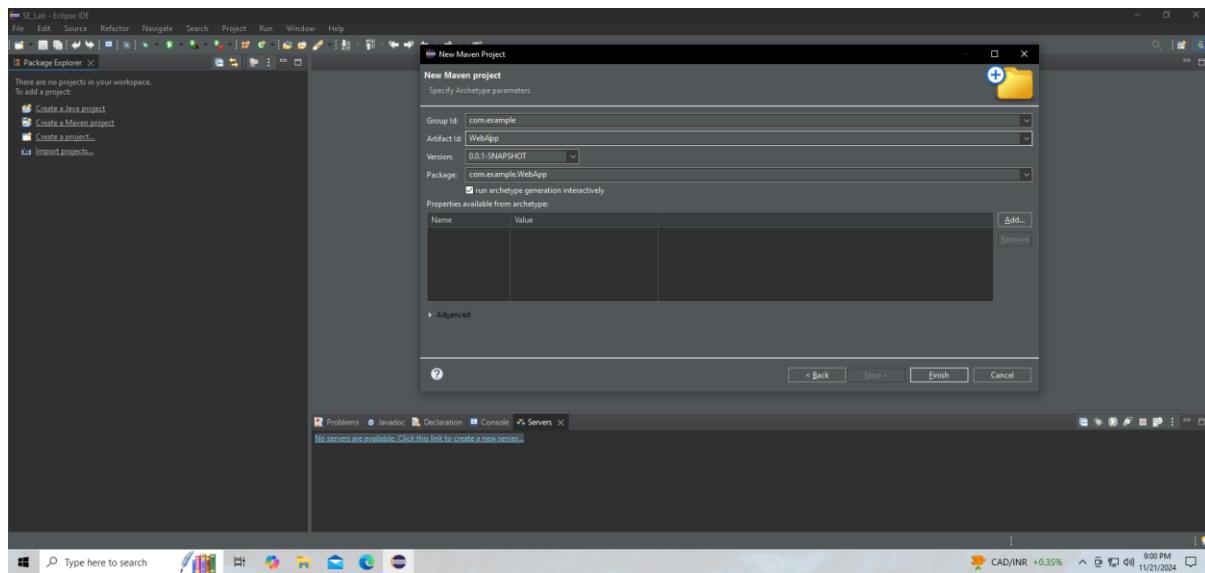
1. Creating new Maven Project



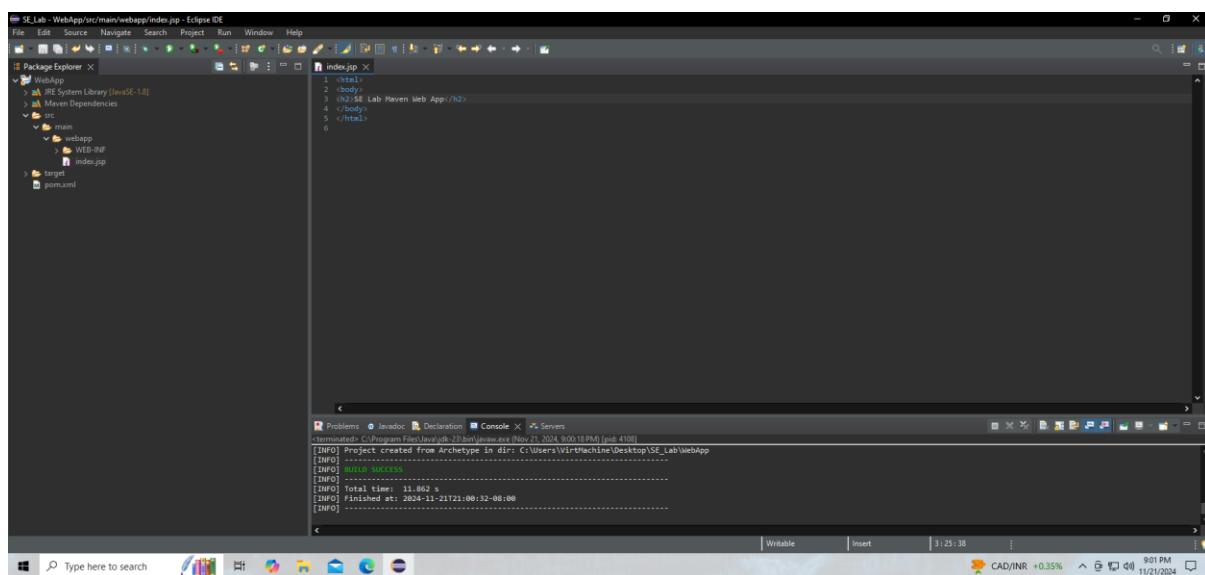
2. Selecting maven-archetype-webapp



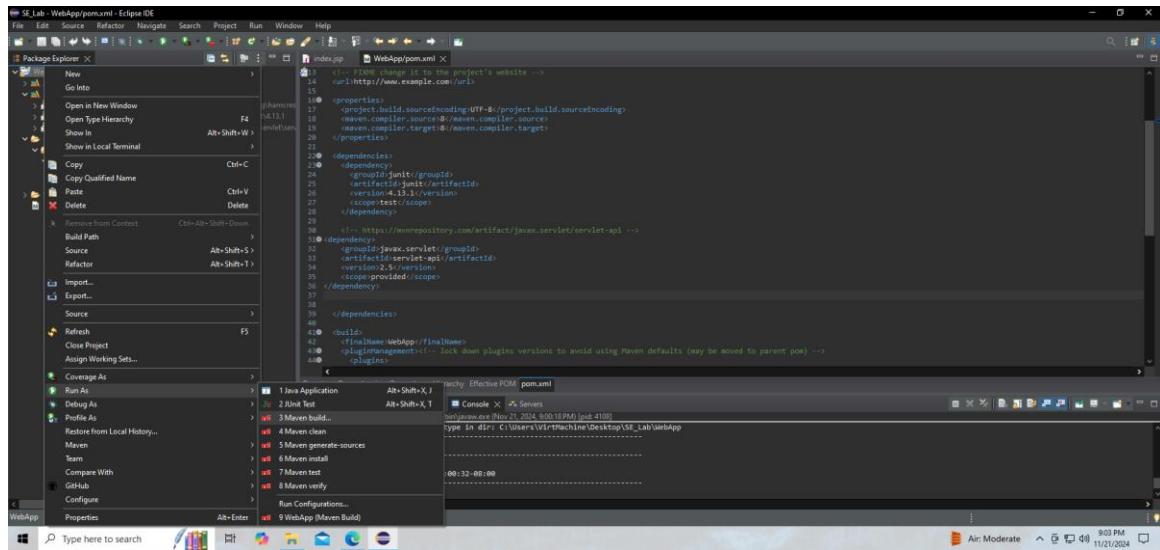
3. Creating new Group Id and Artifact Id



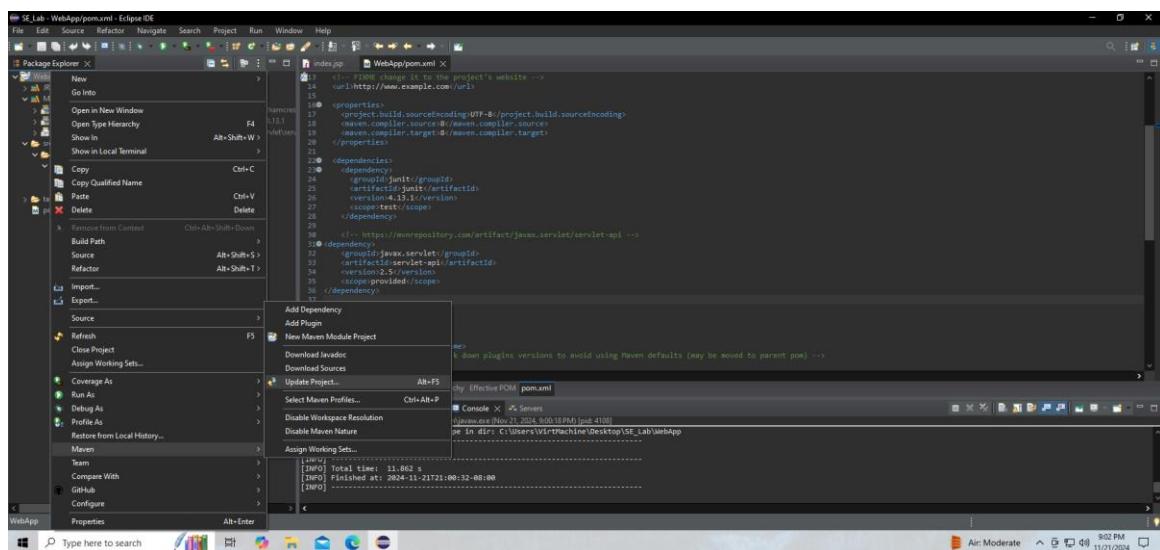
4. Project created



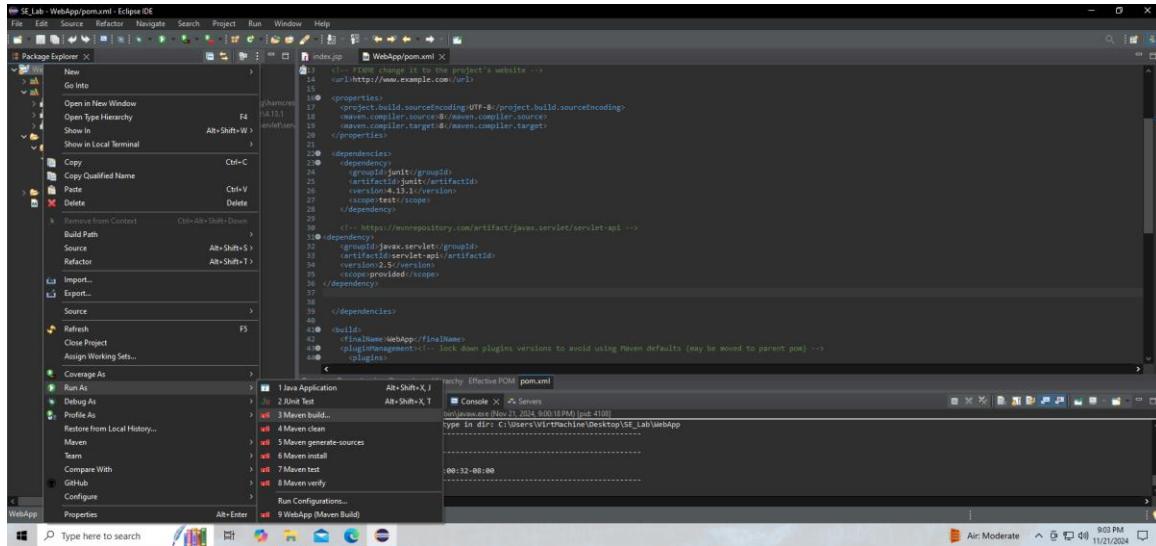
5. Add servlet-api dependency to pom.xml



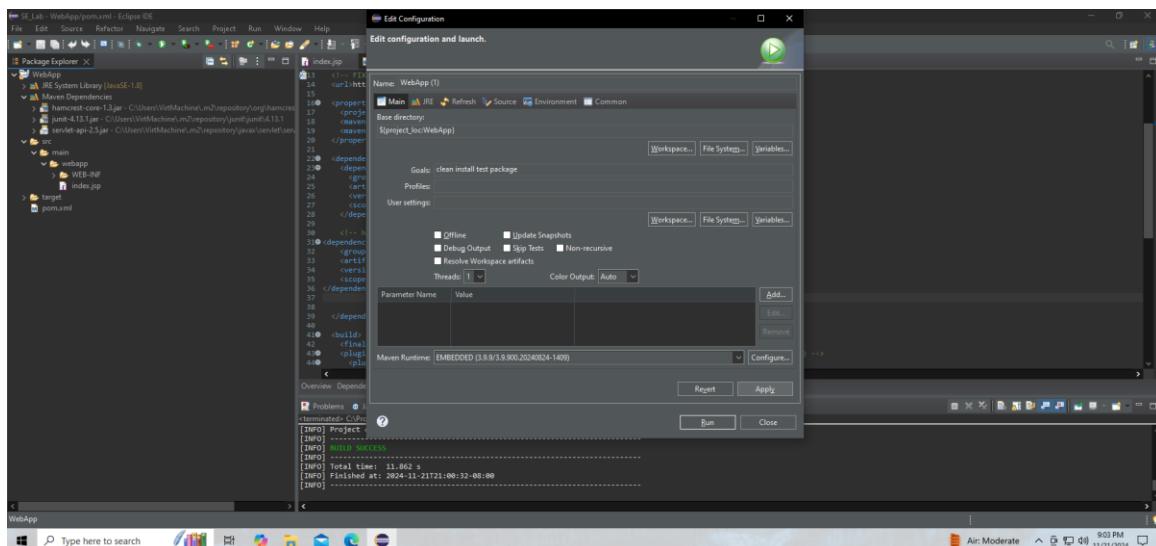
6. Click on Update Project



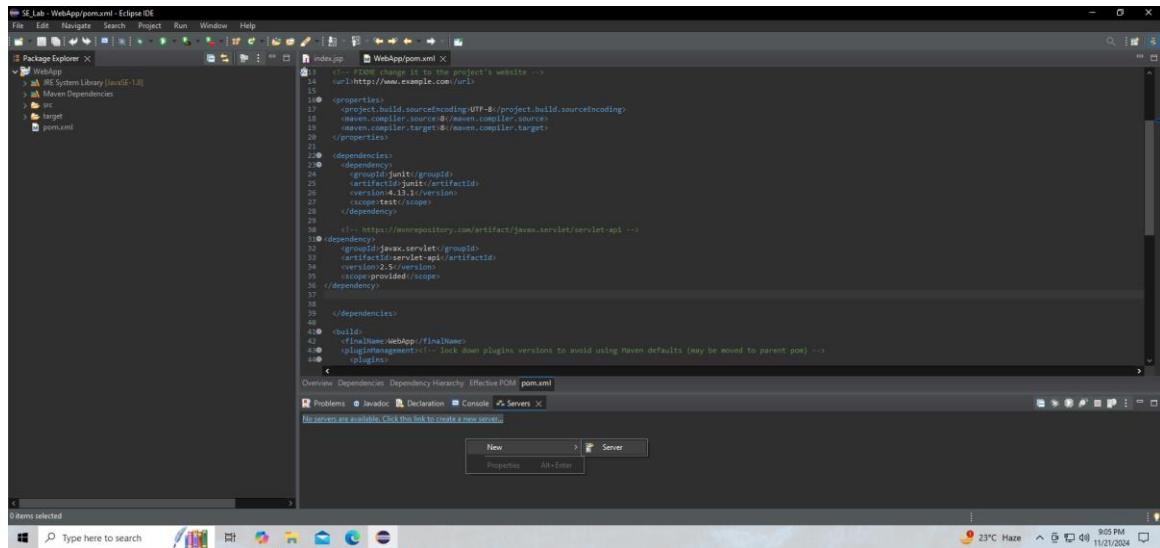
7. Click on Maven build



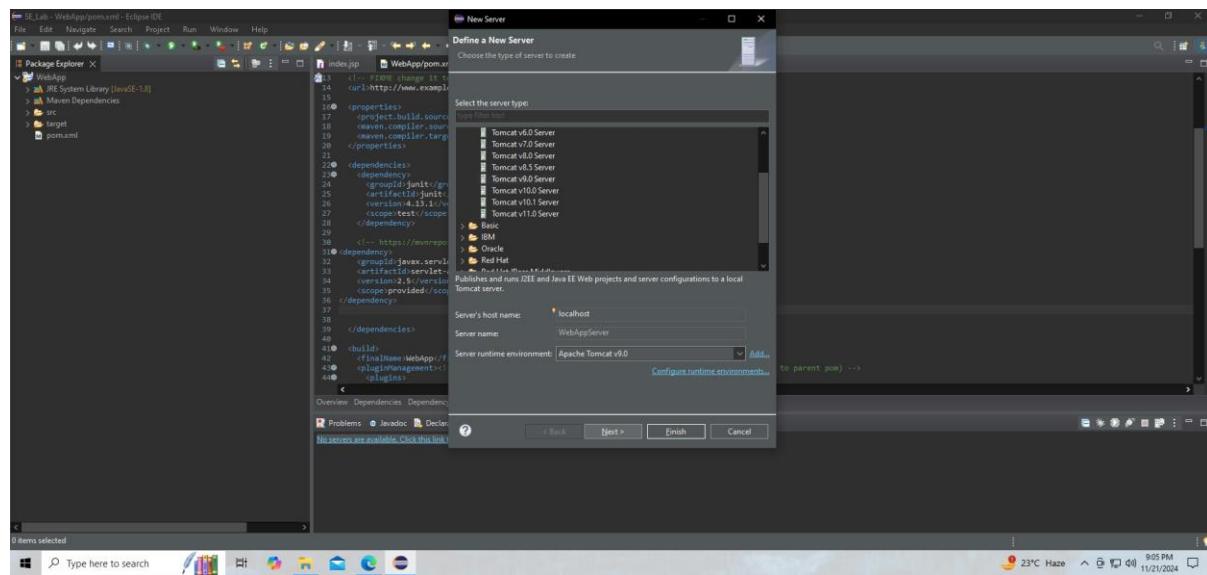
9. Add clean install test package and then run it



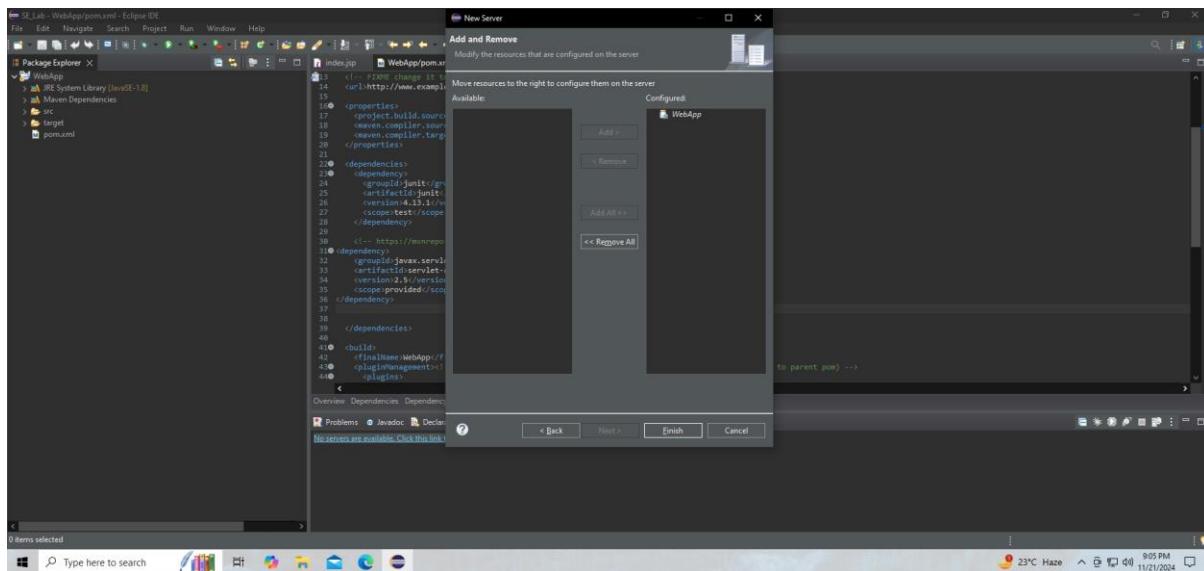
11. Click on Servers tab and create new server



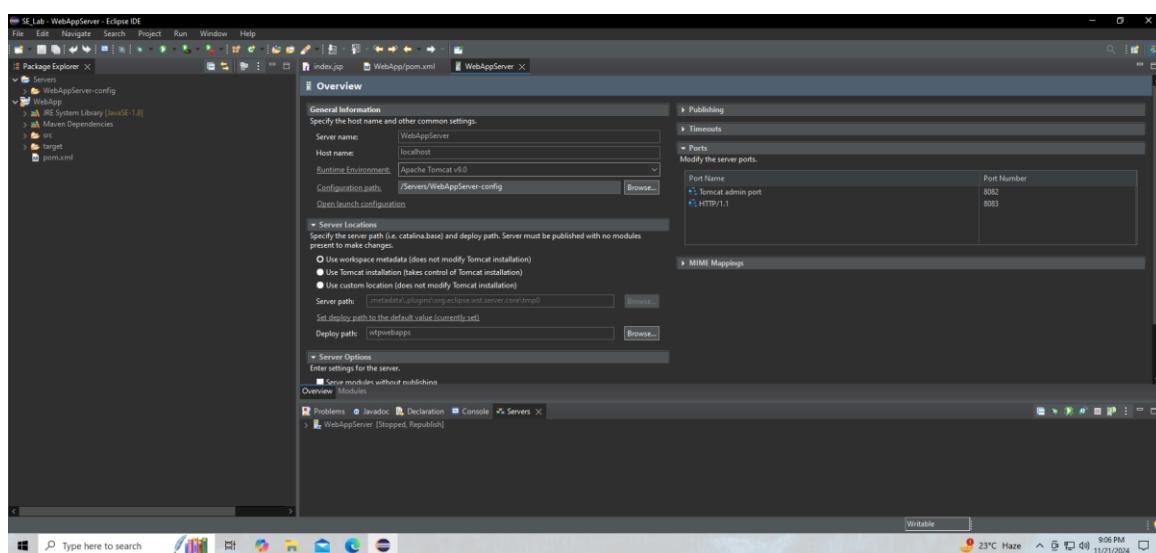
12. Select Tomcat v9.0 Server and click Next



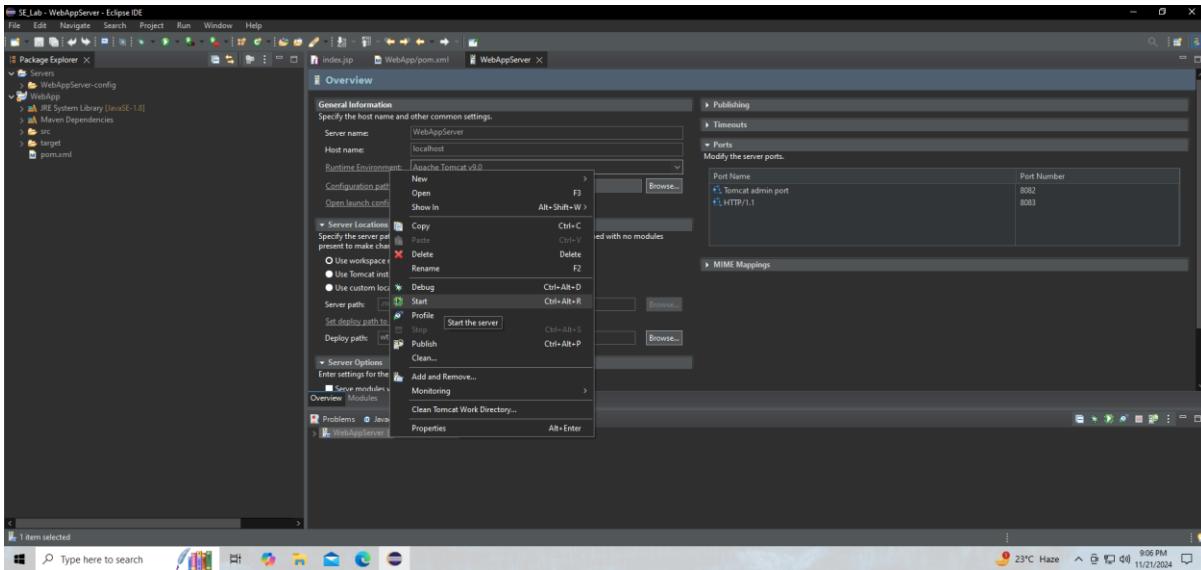
13. Click on your project and click Add, then Finish



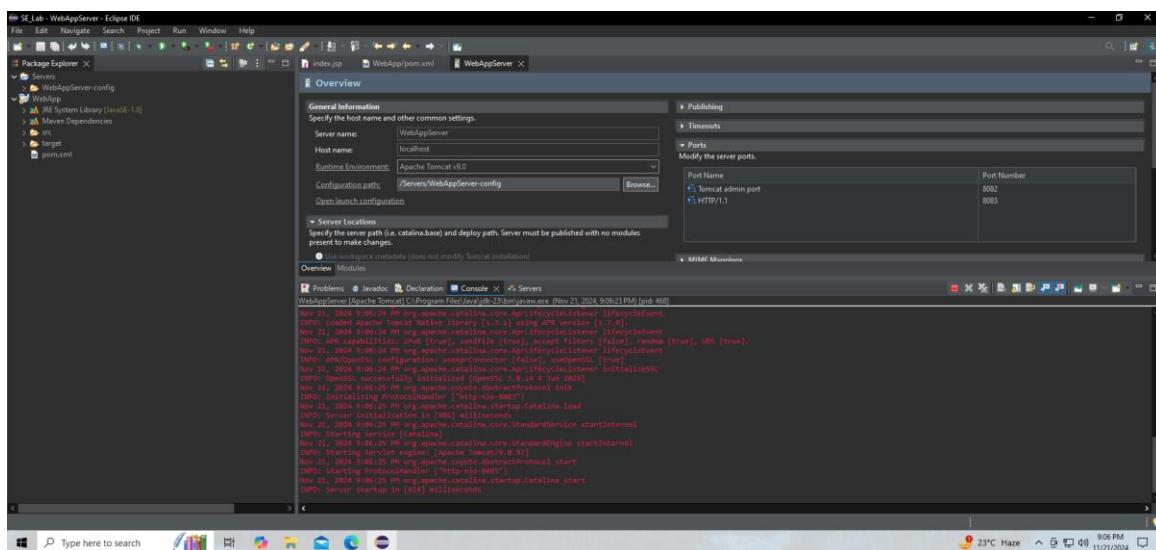
13. Tomcat Server Added



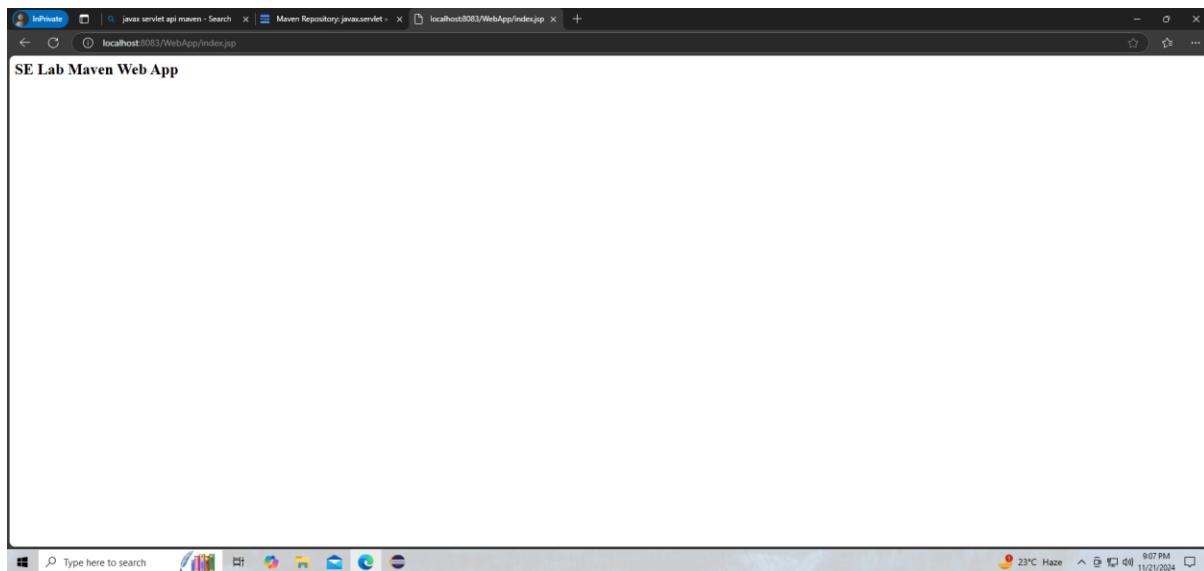
13. Click on Start



13. Tomcat Server started



13. Visit localhost:8083/WebApp/index.jsp



13. Pushing into GitHub

```
VirtMachine@DESKTOP-4AVU1GE ~\Desktop\SE_Lab\WebApp
$ git init
Initialized empty Git repository in C:/Users/VirtMachine/Desktop/SE_Lab/WebApp/
$ git add .
warning: LF will be replaced by CRLF in index.jsp.
warning: LF will be replaced by CRLF in pom.xml.
warning: LF will be replaced by CRLF in WebAppServer-config.xml.
warning: LF will be replaced by CRLF in WebAppServer.xml.
warning: LF will be replaced by CRLF in WebApp.pom.
warning: LF will be replaced by CRLF in WebAppServer.pom.
$ git status
On branch main

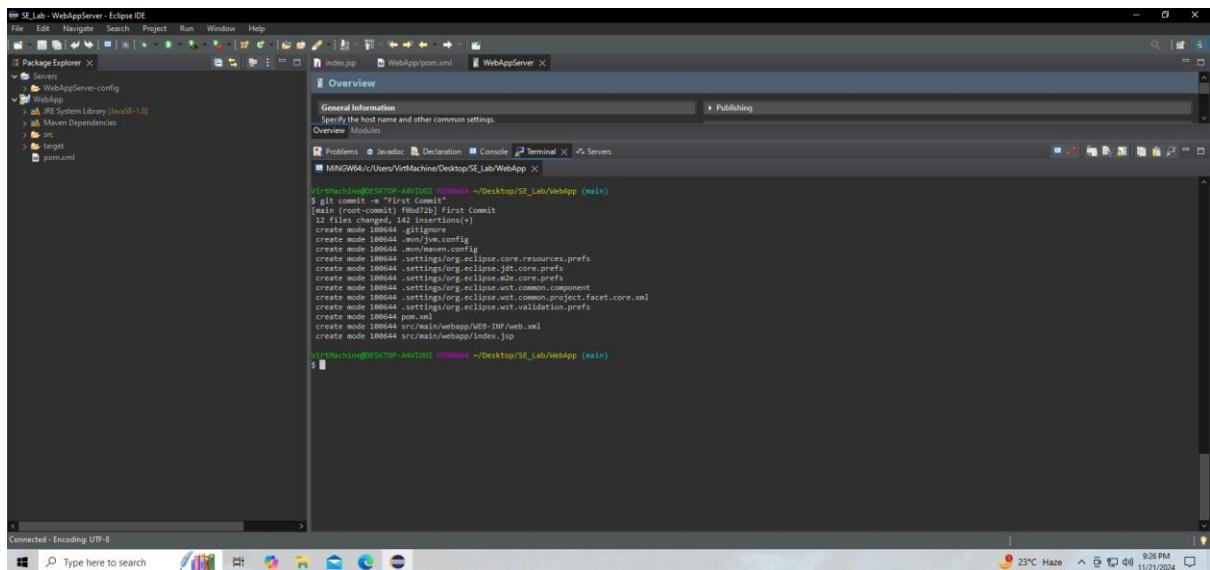
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        gitignore
        pom.xml
        src
        target

nothing added to commit but untracked files present (use "git add" to track)

VirtMachine@DESKTOP-4AVU1GE ~\Desktop\SE_Lab\WebApp [main]
$ git add .
warning: LF will be replaced by CRLF in index.jsp.
warning: LF will be replaced by CRLF in pom.xml.
warning: LF will be replaced by CRLF in WebAppServer-config.xml.
warning: LF will be replaced by CRLF in WebAppServer.xml.
warning: LF will be replaced by CRLF in WebApp.pom.
warning: LF will be replaced by CRLF in WebAppServer.pom.
$ git commit -m "Initial commit"
[master (root-commit) 0000000] Initial commit
  1 file changed, 1 insertion(+)
  create mode 100644 index.jsp
$
```



SE_Lab - WebAppServer - Eclipse IDE

File Edit Navigate Search Project Run Window Help

Package Explorer Servers WebAppServer-config WebApp JRE System Library [JavaSE-1.8] Maven Dependencies src target pom.xml

Overview General Information Specify the host name and other common settings. Overview Modules

Problems JavaDoc Declaration Console Terminal Servers MINGW64/c/Users/VinMachine/Desktop/SE_Lab/WebApp

```
$ git fetch https://github.com/thehejeticolif/maven_web_app.git
$ git remote add origin https://github.com/thehejeticolif/maven_web_app.git
$ git push -u --force origin main
origin https://github.com/thehejeticolif/maven_web_app.git (fetch)
origin https://github.com/thehejeticolif/maven_web_app.git (push)
Branch 'main' set up to track 'origin/main'.
$
```

Connected - Encoding: UTF-8

SE_Lab - WebAppServer - Eclipse IDE

File Edit Navigate Search Project Run Window Help

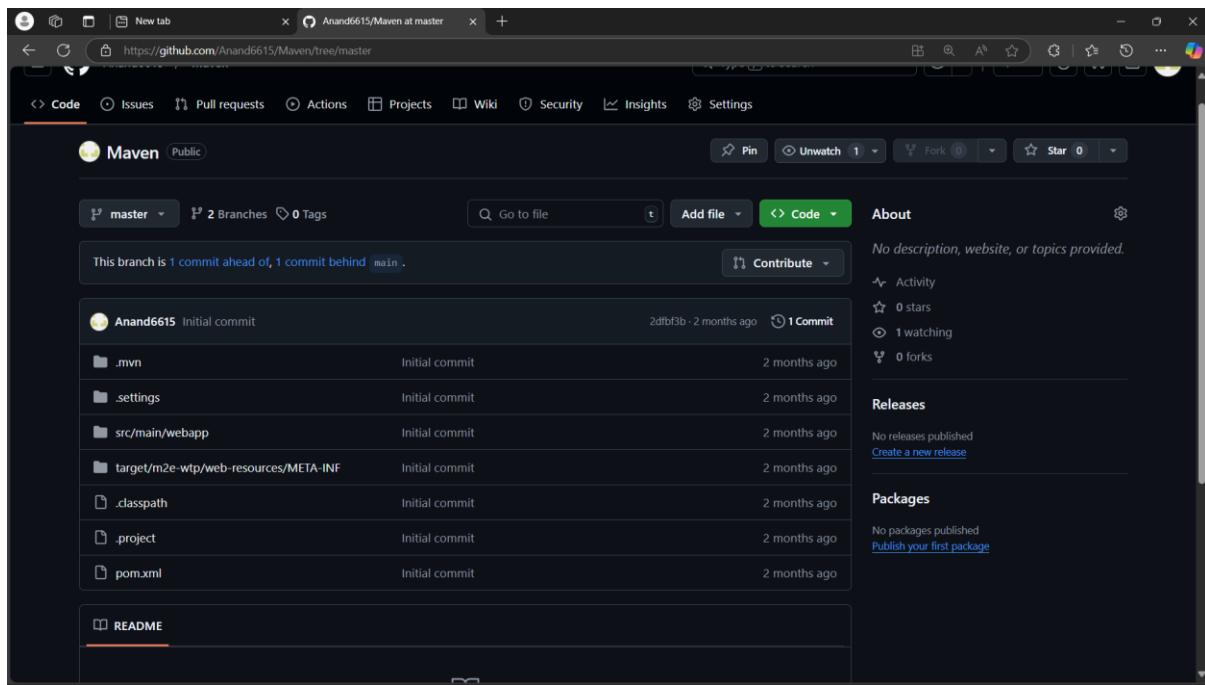
Package Explorer Servers WebAppServer-config WebApp JRE System Library [JavaSE-1.8] Maven Dependencies src target pom.xml

Overview General Information Specify the host name and other common settings. Overview Modules

Problems JavaDoc Declaration Console Terminal Servers MINGW64/c/Users/VinMachine/Desktop/SE_Lab/WebApp

```
$ git push -u --force origin main
Enumerating objects: 19, done.
Compressing objects: 100% (19/19), done.
Writing objects: 100% (19/19), 2.84 KiB | 2.84 MiB/s, done.
Total 19 (delta 8), pack-reused 0 (from 0)
To https://github.com/thehejeticolif/maven_web_app.git
 + aa14000...f0d072b main -> main (forced update)
branch 'main' set up to track 'origin/main'.
$
```

Connected - Encoding: UTF-8



Result: Maven Java and web projects were created and pushed to GitHub.

Experiment-6: Building a CI/CD pipeline using Jenkins for projects created in Experiment – 5

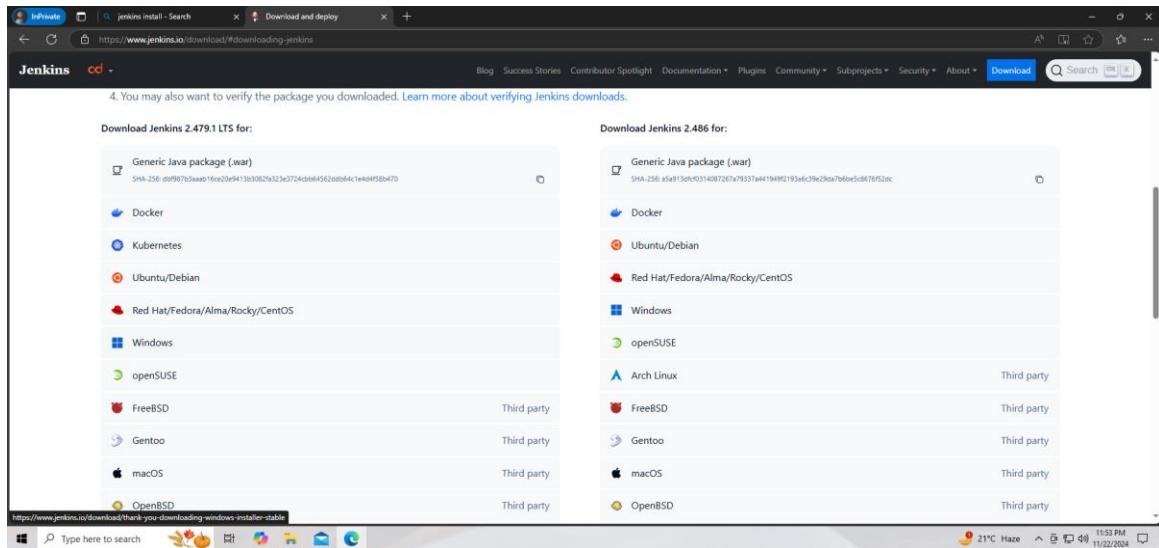
Aim: To install Jenkins and use it for Continuous Integration and Continuous Deployment (CI/CD) of the Maven Java and web projects created in the previous experiment.

Introduction:

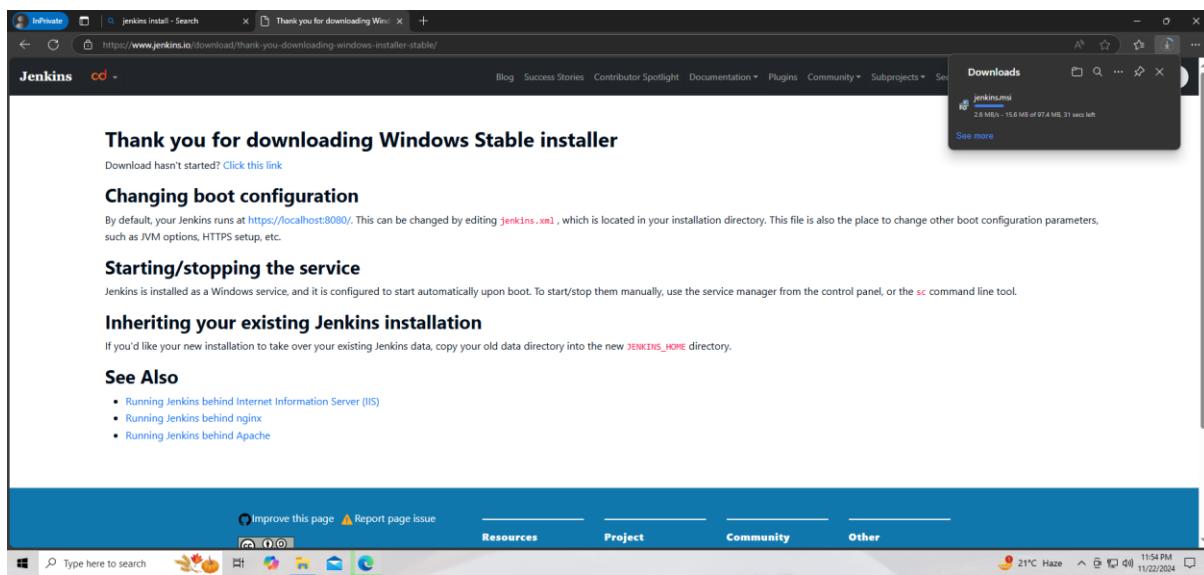
1. **Jenkins:** Jenkins is an open-source automation server widely used for implementing Continuous Integration (CI) and Continuous Deployment (CD). It automates building, testing, and deploying applications, thereby enhancing software delivery efficiency and reliability.
2. **Continuous Integration (CI):**
 - Automatically integrates code changes from multiple contributors.
 - Triggers builds and tests upon code updates to ensure stability and detect issues early.
3. **Continuous Deployment (CD):**
 - Extends CI by automating the release of stable builds to production environments.
 - Reduces manual intervention and accelerates deployment cycles.

Installation of Jenkins

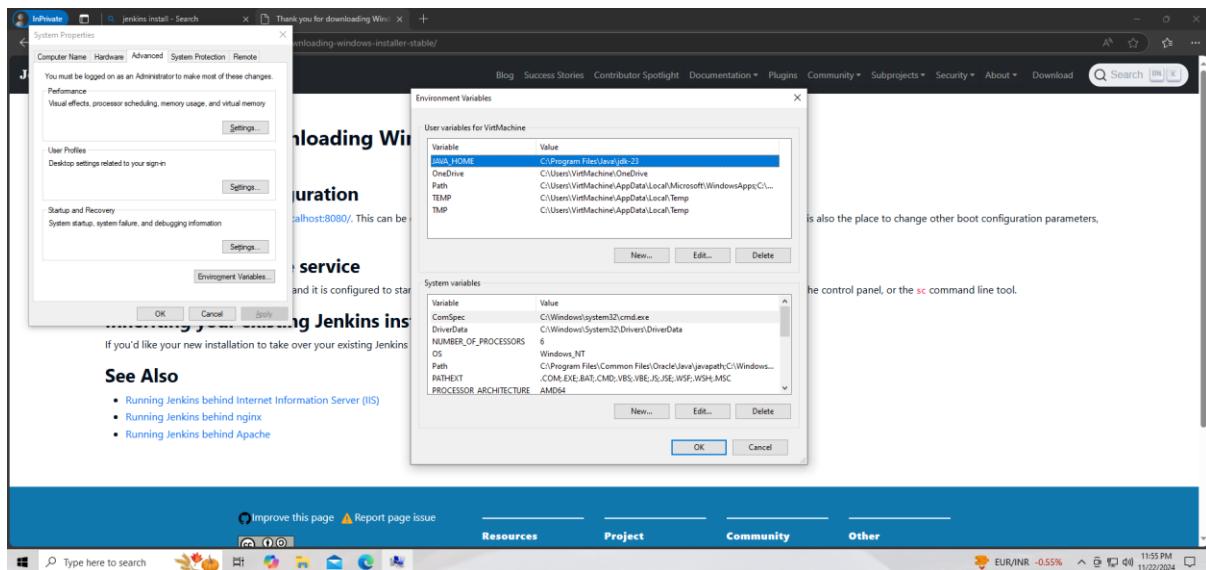
1. Download the Jenkins installer from the official website



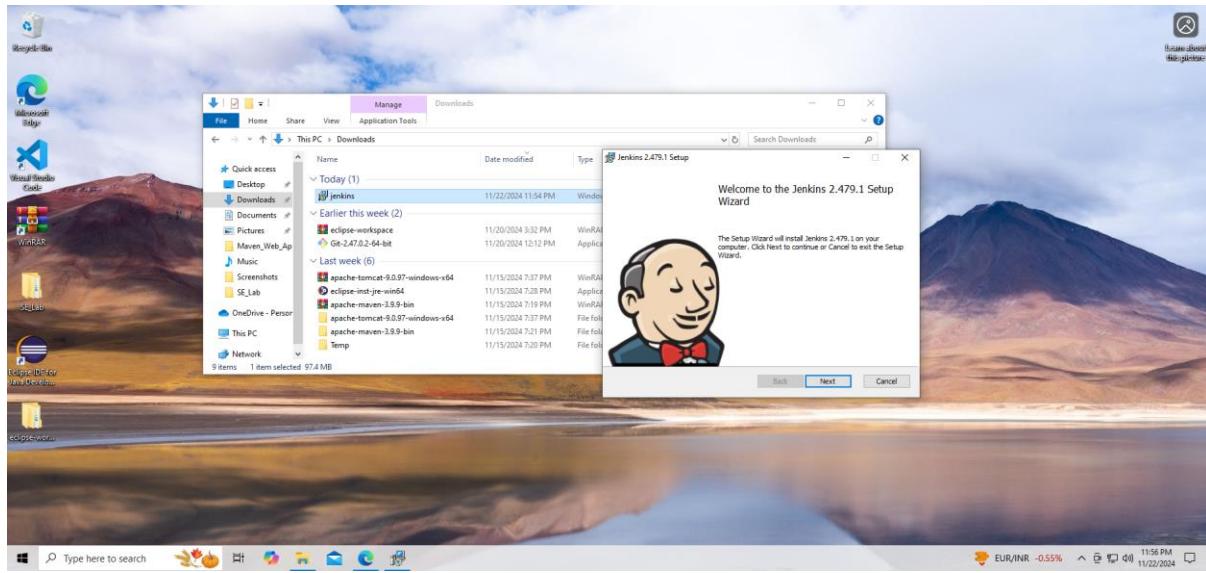
2. Wait for the download to complete



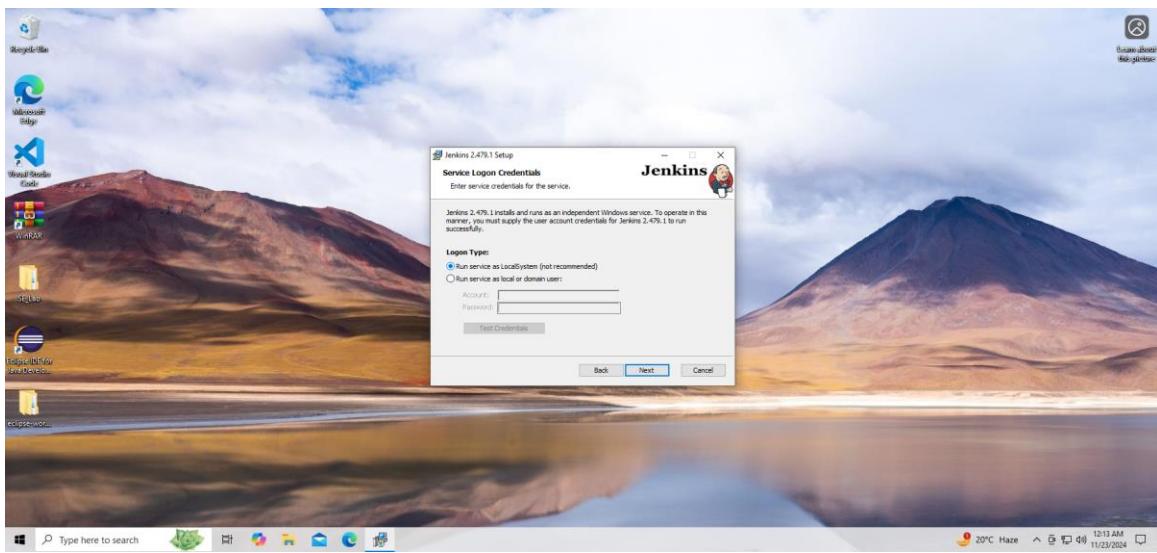
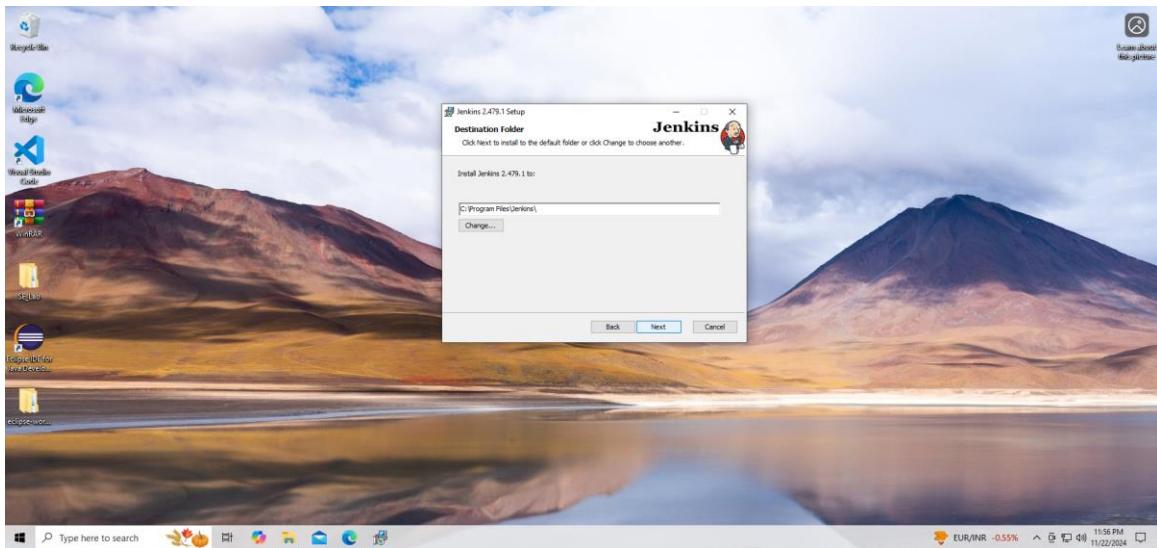
3. Add JAVA_HOME environment variable if not added

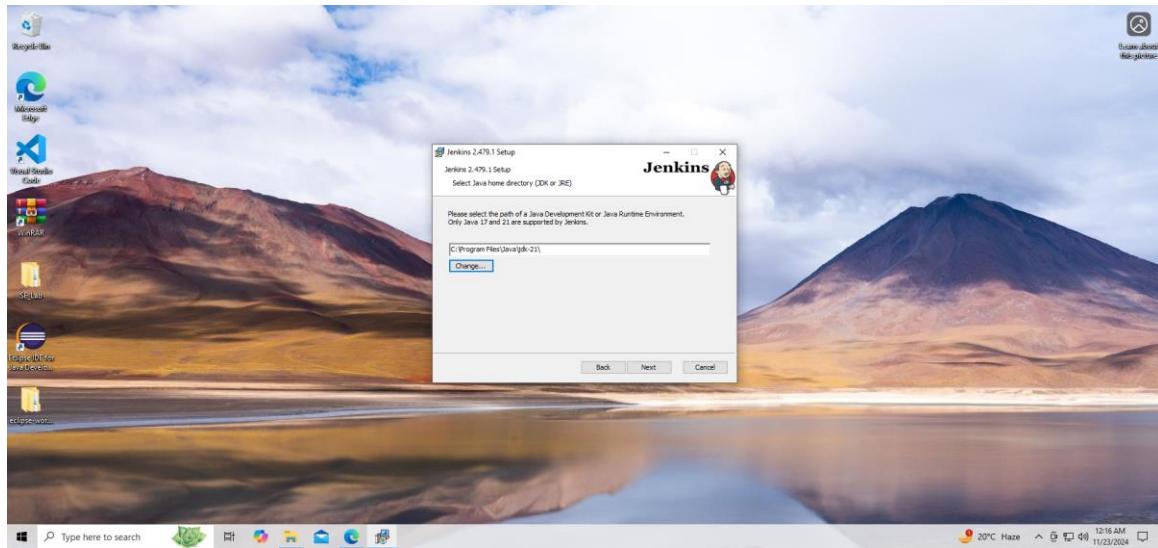
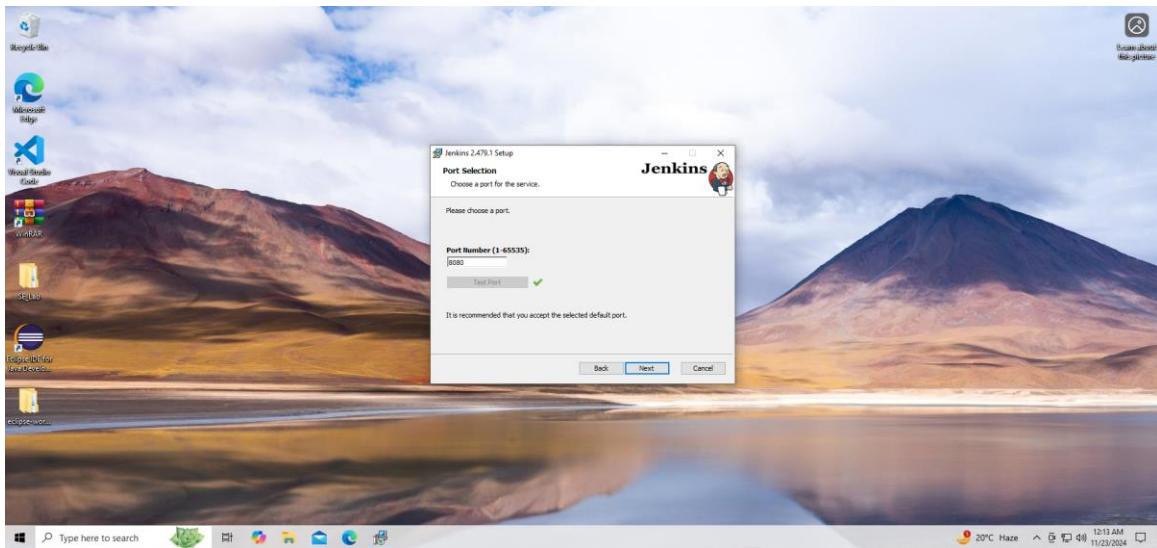


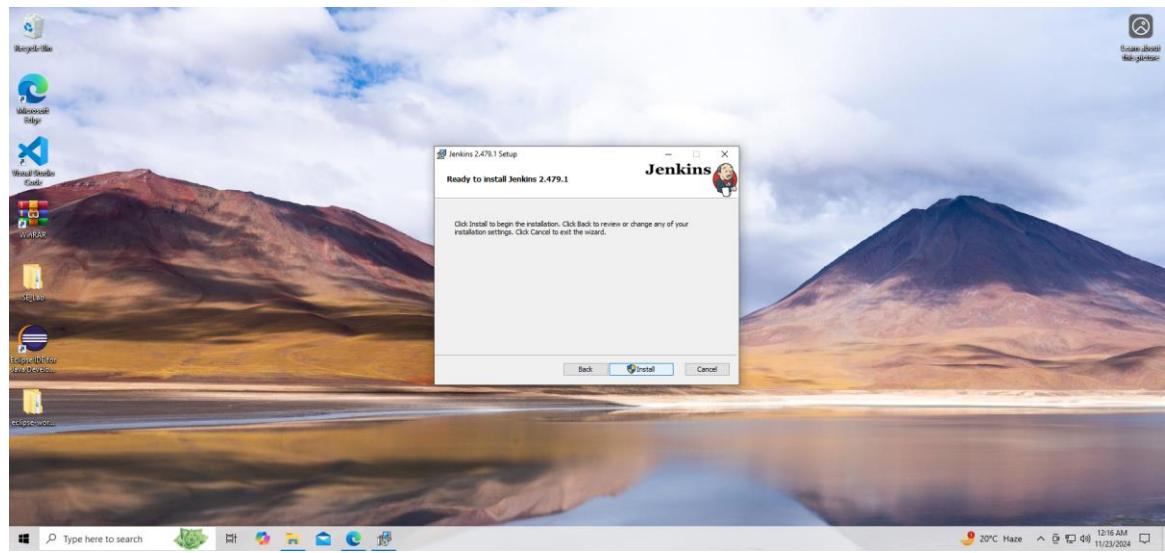
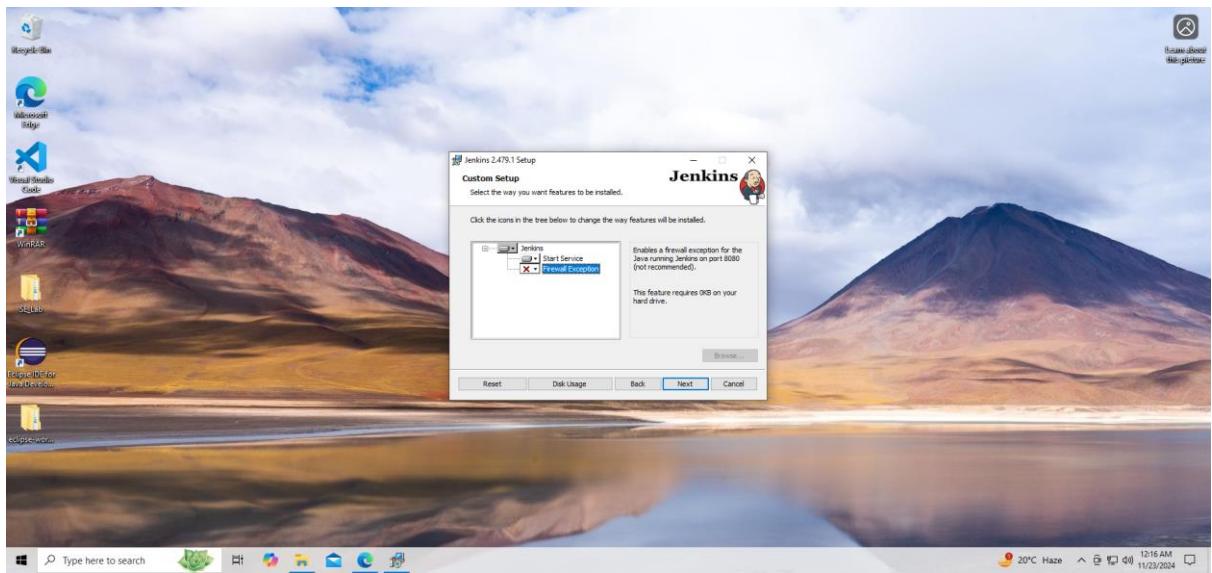
4. Run the Jenkins installer

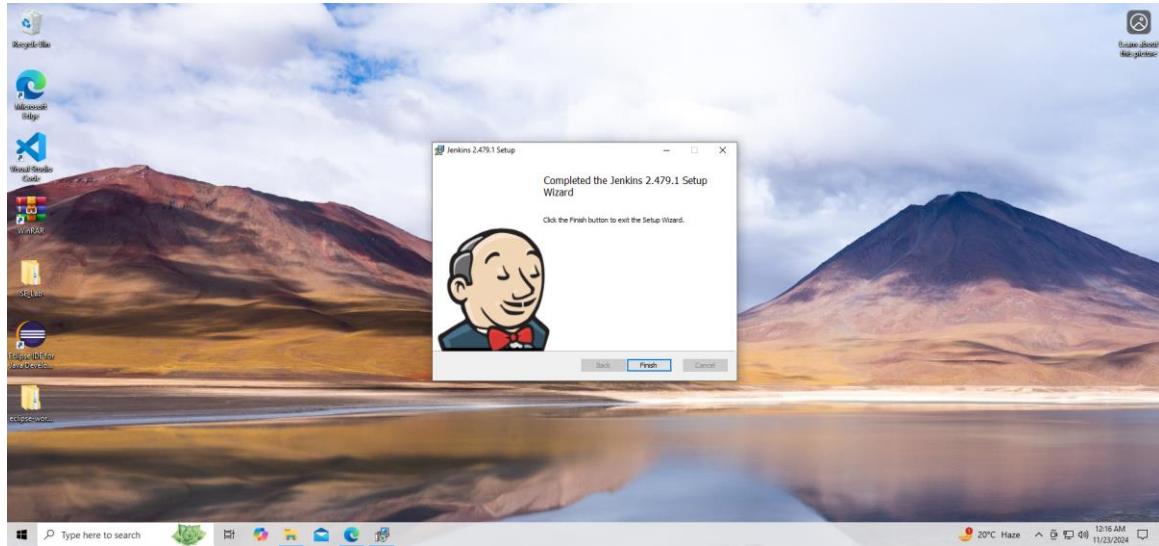


5. Follow on screen instructions



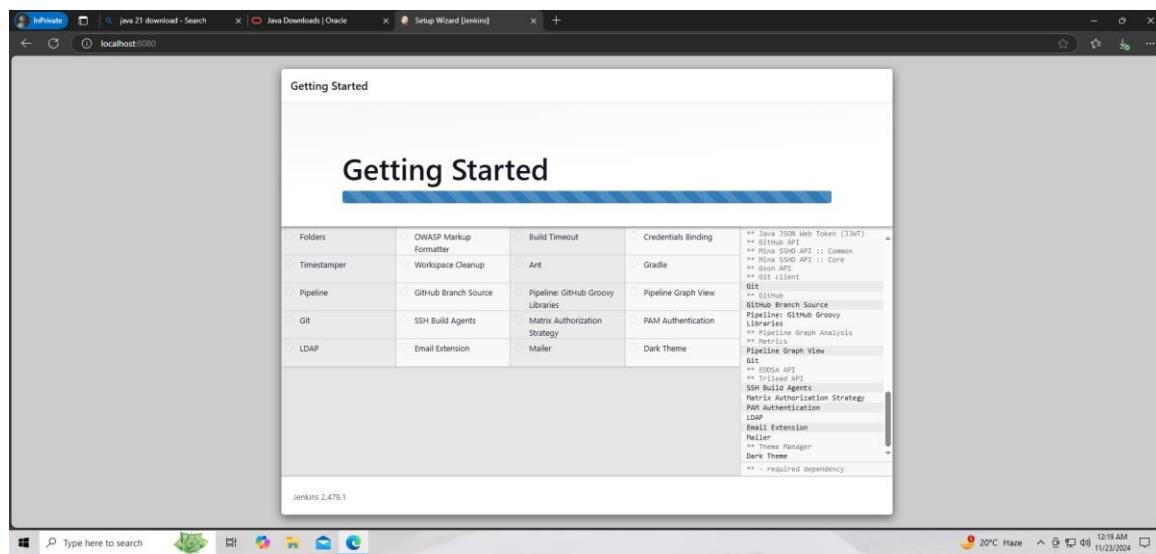
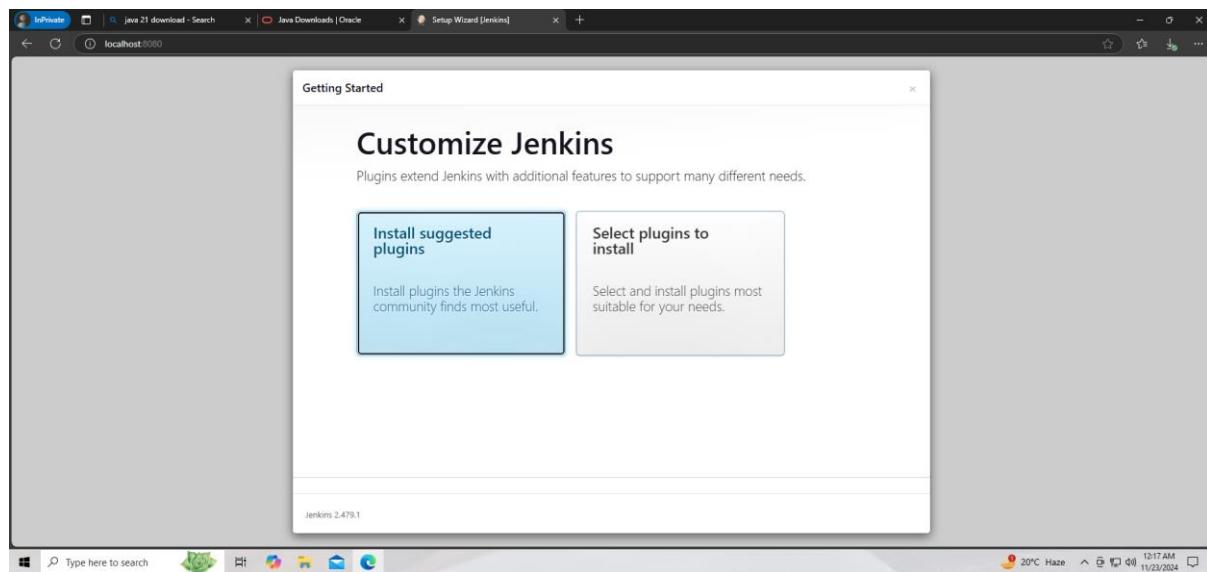


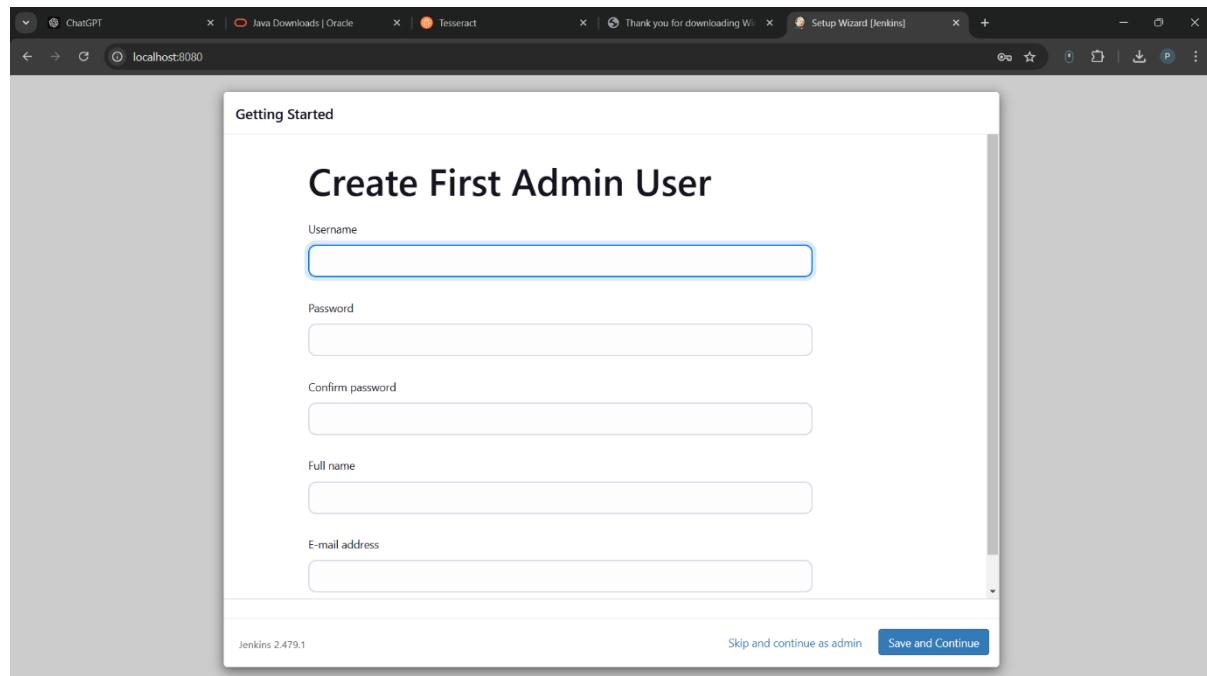
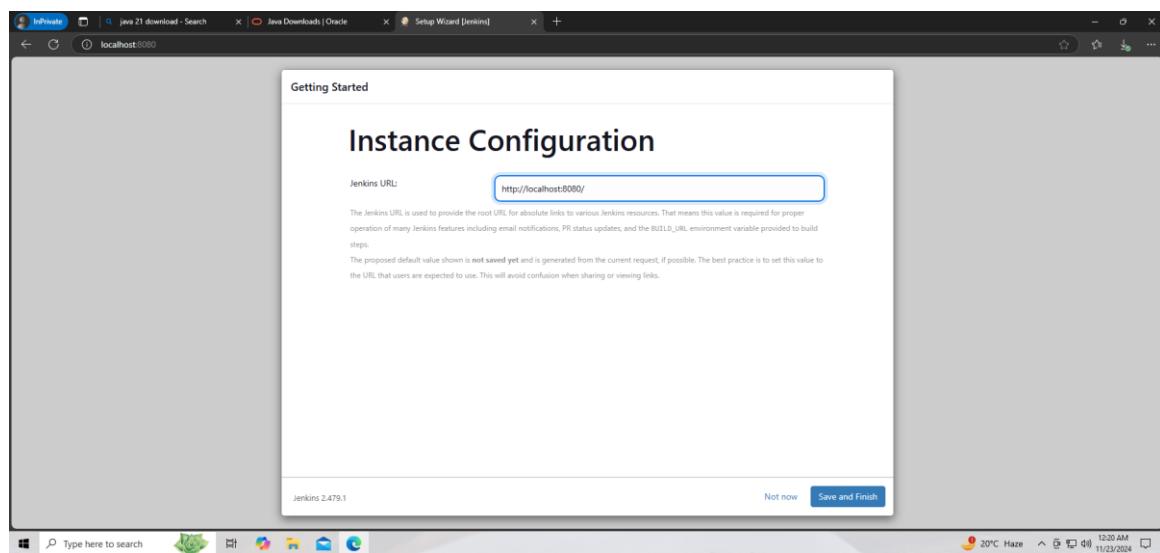


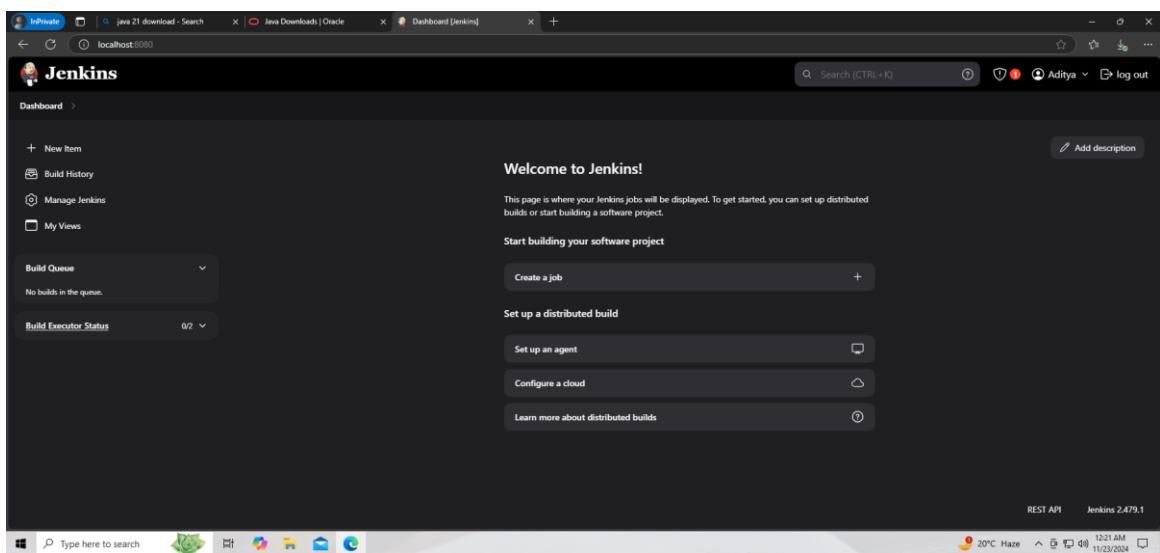
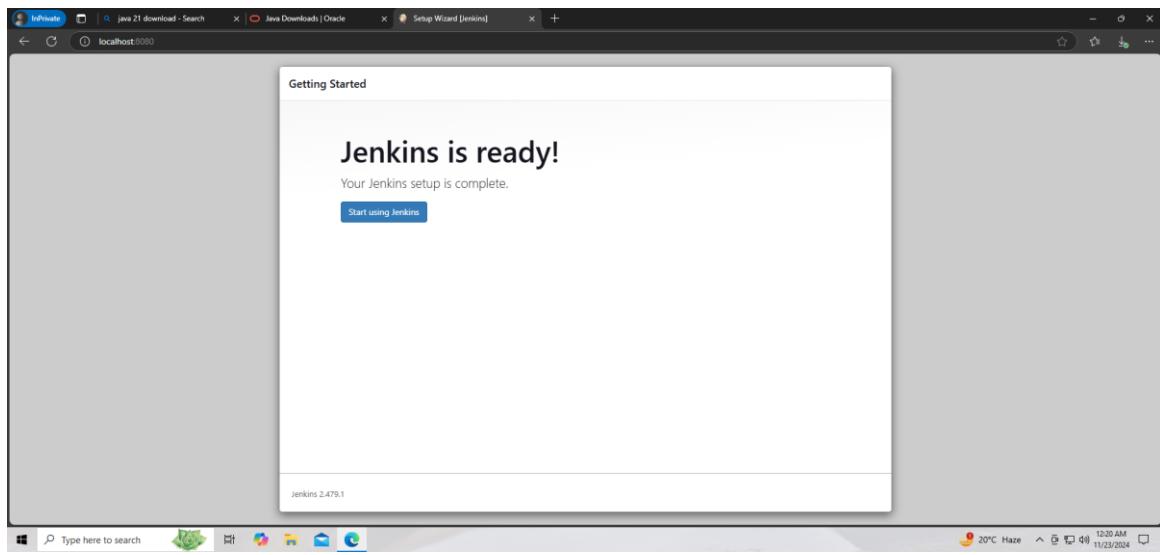


6. After installation, open localhost:8080 in browser

A screenshot of a web browser window. The address bar shows the URL "localhost:8080/login?from=%2F". The main content is a "Getting Started" dialog titled "Unlock Jenkins". It instructs the user that a password has been written to the log and provides the path "C:\ProgramData\Jenkins\jenkins\secrets\initialAdminPassword". It asks the user to copy the password from the log and paste it into the "Administrator password" input field. The input field contains several dots, indicating a long password. A "Continue" button is at the bottom right of the dialog. The browser's taskbar shows other open tabs like "java 21 download - Search" and "Java Downloads | Oracle". The system tray shows the date and time as "11/23/2024 12:16 AM".







Creating pipeline for Jenkins Web app

The screenshot shows the Jenkins Dashboard at localhost:8080. The main area displays a table of pipelines:

S	W	Name	Last Success	Last Failure	Last Duration
Green circle	Sun icon	DeclarativePipeline_MavenJava	4 min 53 sec #712	1 day 2 hr #690	30 sec
Red circle	Cloud icon	Mavenprojectbuildtrain	N/A	3 days 11 hr #2	5.6 sec
Red circle	Cloud icon	Pipeline	3 days 11 hr #2	6 min 53 sec #56	25 sec
Red circle	Cloud icon	SampleMavenProjectTest	N/A	3 days 20 hr #1	15 ms

On the left sidebar, there are links for 'Build History', 'Manage Jenkins', and 'My Views'. Below the dashboard, there are sections for 'Build Queue' and 'Build Executor Status' (1 of 2 executors busy). At the bottom right, there are links for 'REST API' and 'Jenkins 2.479.2'.

The screenshot shows the Jenkins Plugin Manager at localhost:8080/manage/pluginManager/available. The 'Available plugins' tab is selected. A search bar at the top is empty. The list of available plugins includes:

Install	Name	Released
<input type="checkbox"/>	JavaMail API 1.6.2-10	6 mo 9 days ago
<input type="checkbox"/>	Command Agent Launcher 116vb85919c54a_d6	16 days ago
<input type="checkbox"/>	Oracle Java SE Development Kit Installer 80.0fa_dee3ed6f0	3 mo 24 days ago
<input type="checkbox"/>	Pipeline: REST API 2.34	1 yr 0 mo ago
<input type="checkbox"/>	Pipeline: Stage View 2.34	1 yr 0 mo ago
<input type="checkbox"/>	JSch dependency 0.2.16-85.v42ee10f08d4b_	11 mo ago

On the left sidebar, there are links for 'Updates', 'Available plugins' (selected), 'Installed plugins', and 'Advanced settings'. At the bottom, there is a Windows taskbar with various icons and a status bar showing 'Watchlist ideas', 'ENG', '25-11-2024', and '1506'.

The screenshot shows the Jenkins Plugins page. The left sidebar has links for Updates, Available plugins (which is selected), Installed plugins, and Advanced settings. The main area has a search bar and a 'Install' button. It lists several plugins:

- Maven Integration** 3.24 (Build Tools) - Released 1 mo 6 days ago. Description: This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTs as well as the automated configuration of various Jenkins publishers such as JUnit.
- Build Pipeline** 2.0.2 (User Interface, Build Tools, Other Post-Build Actions) - Released 6 mo 15 days ago. Description: This plugin renders upstream and downstream connected jobs that typically form a build pipeline. In addition, it offers the ability to define manual triggers for jobs that require intervention prior to execution, e.g. an approval process outside of Jenkins. A warning message states: "Warning: This plugin version may not be safe to use. Please review the following security notices:
 - Stored XSS vulnerability"
- Webhook Step** 342.v620877eff14 (Allows build pipelines to wait for notification from an external system before continuing.) - Released 10 mo ago.

At the bottom, there's a 'Pipeline timeline' link and a 'Watchlist ideas' button.

The screenshot shows the Jenkins Plugins page with the 'Download progress' section selected in the sidebar. The main area displays the download status for various plugins:

Plugin	Status
Preparation	Success
Javadoc	Success
JSch dependency	Pending
Maven Integration	Pending
Parameterized Trigger	Pending
jQuery	Pending
Build Pipeline	Pending
Loading plugin extensions	Pending

Below the table, there are two links: "Go back to the top page (you can start using the installed plugins right away)" and "Restart Jenkins when installation is complete and no jobs are running".

InPrivate Available plugins - Plugins [jenk... localhost:8083/WebApp/ Tessera... +

localhost:8080/manage/pluginManager/available

Plugins

Updates Available plugins Installed plugins Advanced settings Download progress

Install Name ↴ Released

Pipeline Utility Steps 2.18.0 pipeline Build Tools Miscellaneous Utility steps for pipeline jobs. 1 mo 18 days ago

Copy Artifact 757405365383a_455 Build Parameters Build Tools Adds a build step to copy artifacts from another project. 1 mo 5 days ago

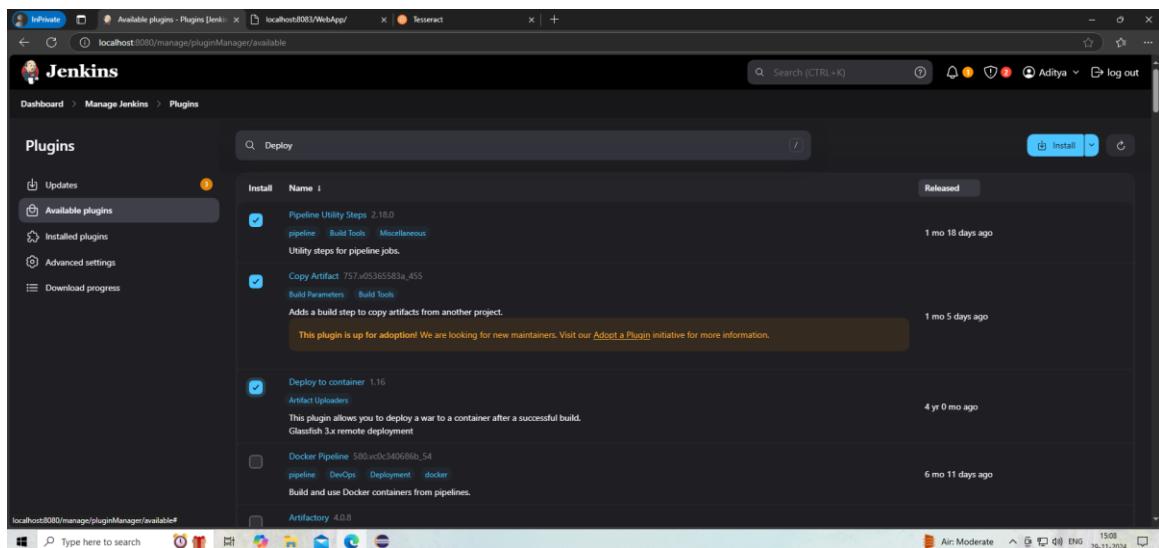
This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.

Deploy to container 1.16 Artifactory Uploaders This plugin allows you to deploy a war to a container after a successful build. Glassfish 3.x remote deployment 4 yr 0 mo ago

Docker Pipeline 580.v0c340696b_54 pipeline DevOps Deployment docker Build and use Docker containers from pipelines. 6 mo 11 days ago

Artifactory 4.0.8

localhost:8080/manage/pluginManager/available Type here to search Air: Moderate 29-11-2024



InPrivate Download progress - Plugins [jenk... localhost:8083/WebApp/ Tessera... +

localhost:8080/manage/pluginManager/updates/

Plugins

Updates Available plugins Installed plugins Advanced settings Download progress

Download progress

Preparation

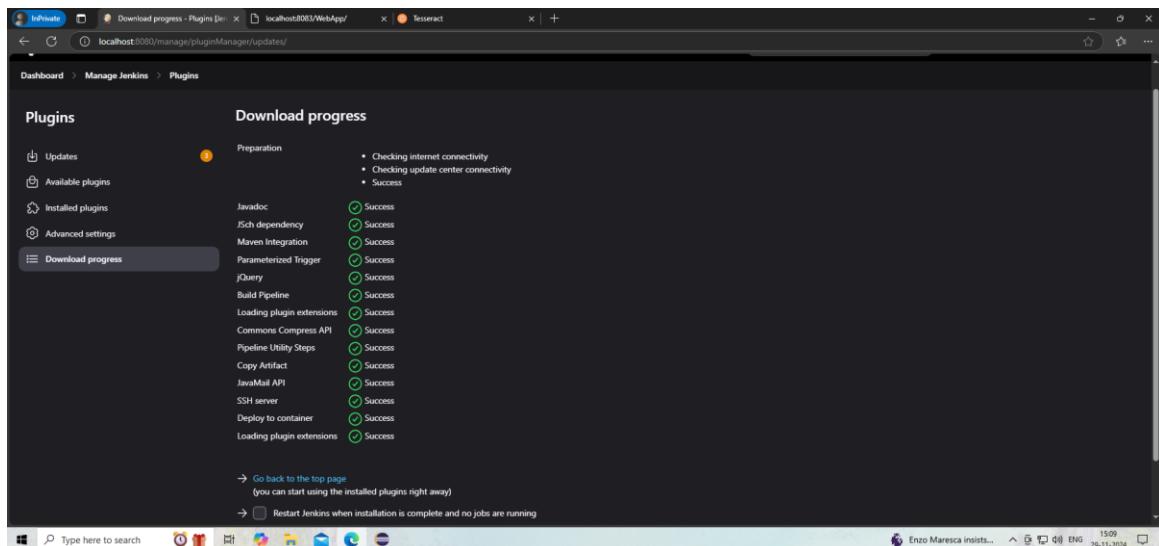
- Checking internet connectivity
- Checking update center connectivity
- Success

Plugin	Status
Javadoc	Success
JSch dependency	Success
Maven Integration	Success
Parameterized Trigger	Success
jQuery	Success
Build Pipeline	Success
Loading plugin extensions	Success
Commons Compress API	Success
Pipeline Utility Steps	Success
Copy Artifact	Success
JavaMail API	Success
SSH server	Success
Deploy to container	Success
Loading plugin extensions	Success

→ Go back to the top page
(you can start using the installed plugins right away)

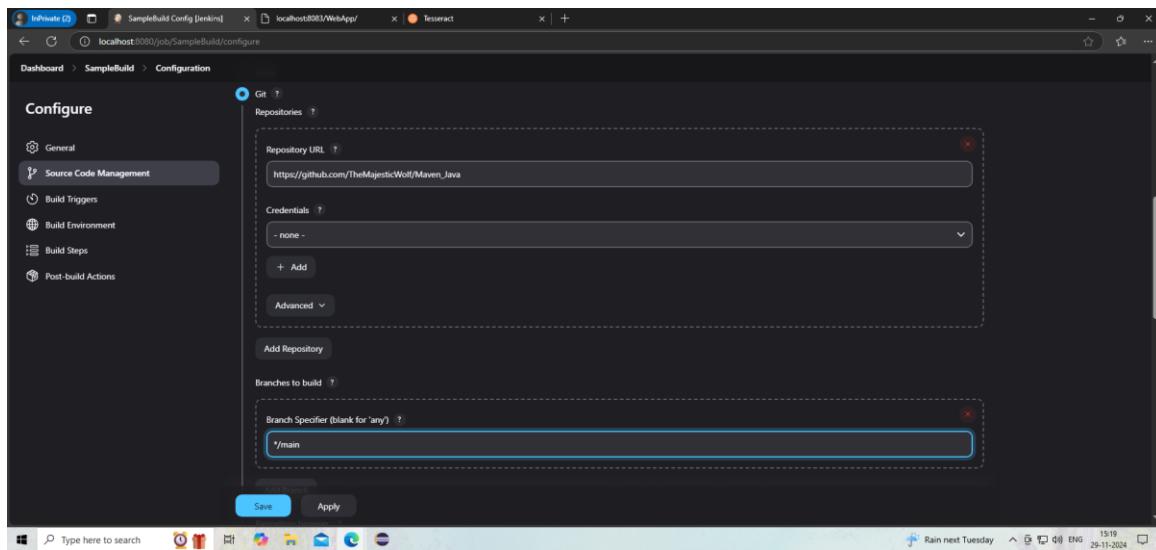
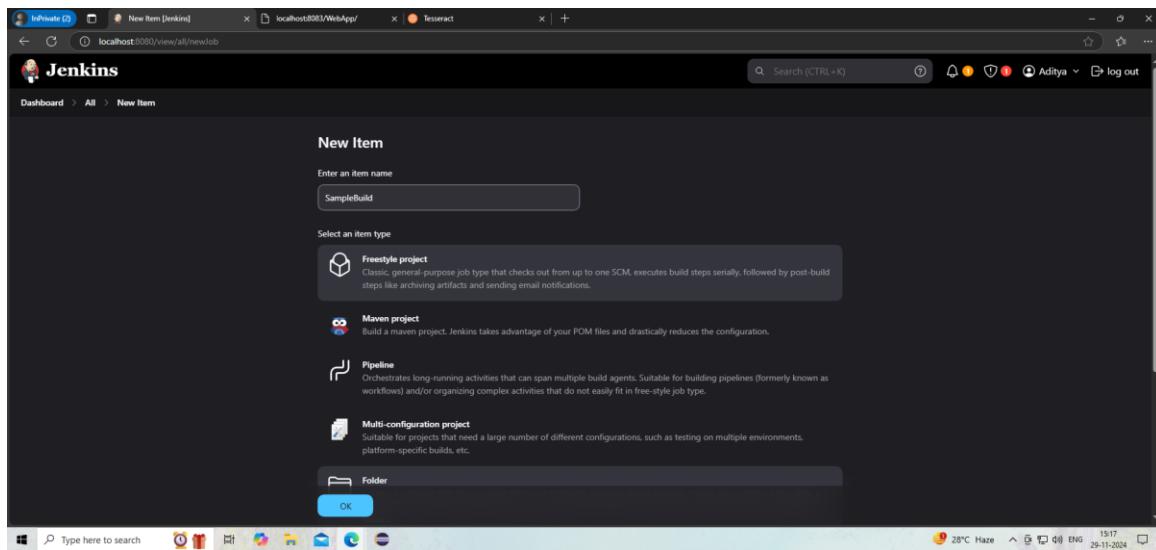
→ Restart Jenkins when installation is complete and no jobs are running

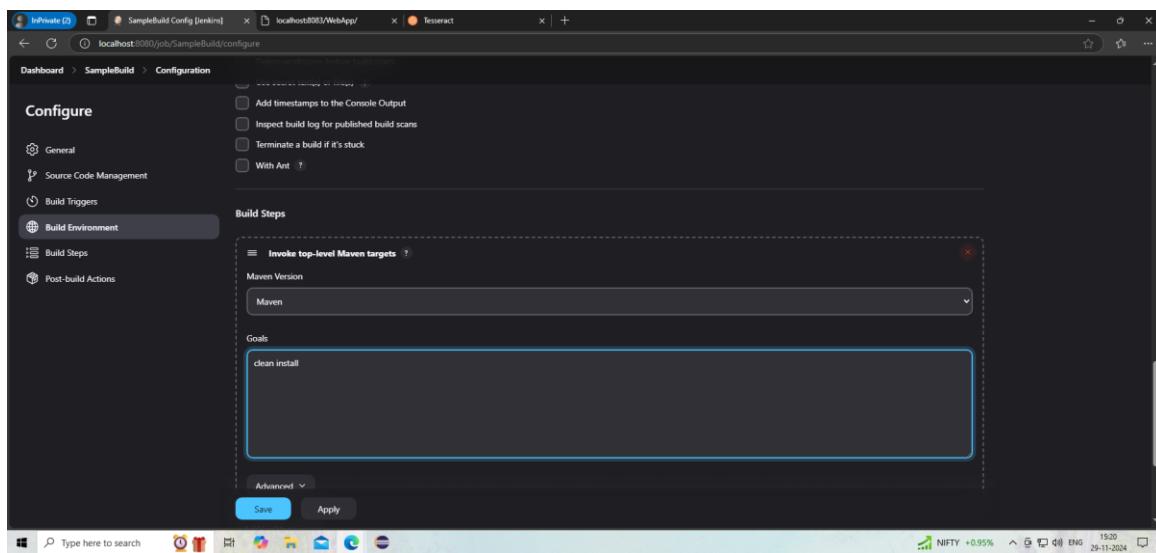
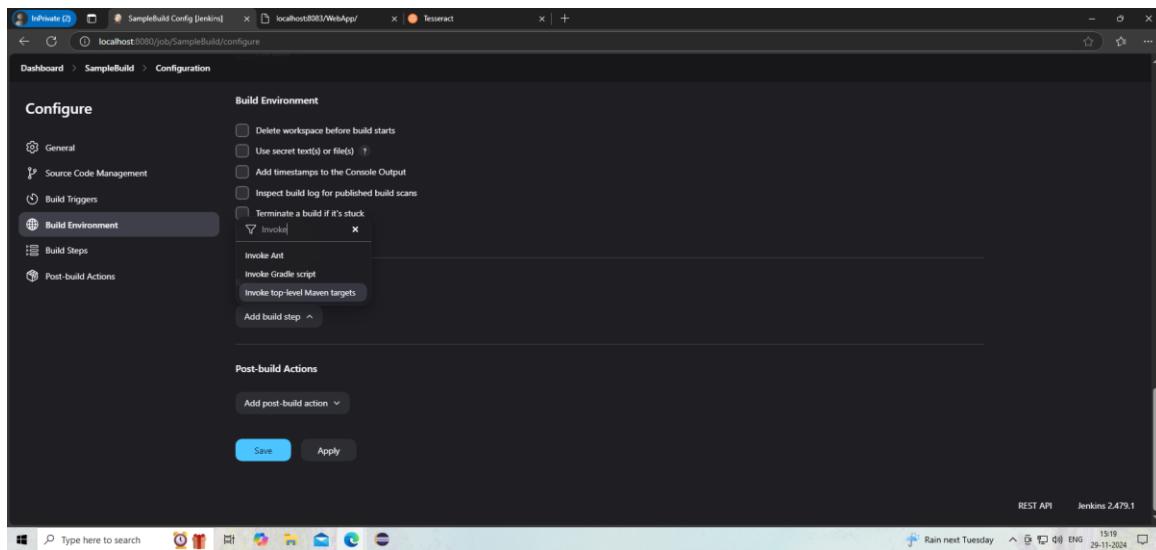
Enzo Maresca insists... Air: Moderate 29-11-2024

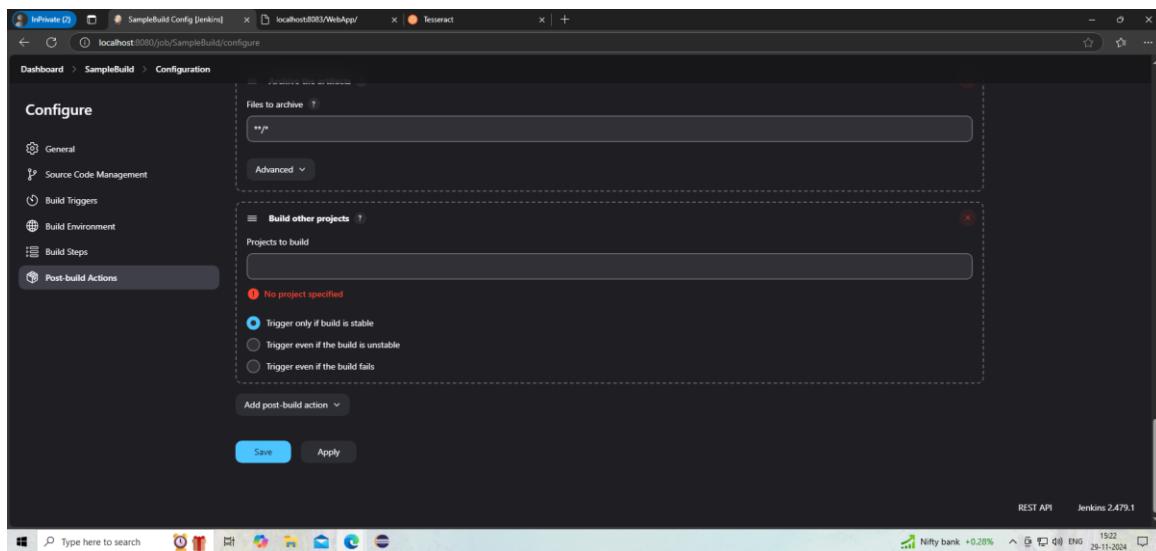
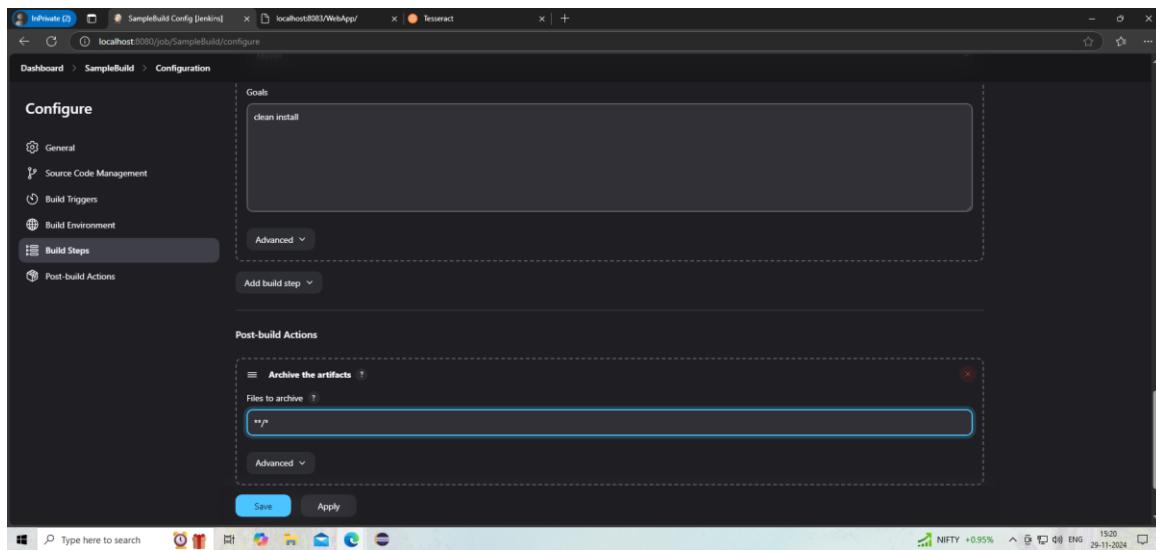


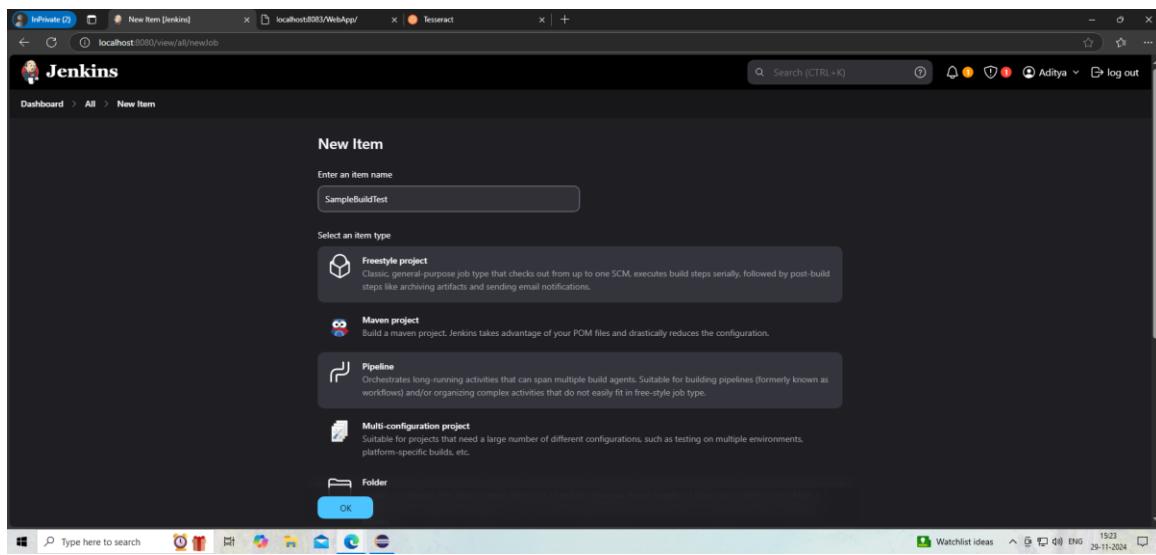
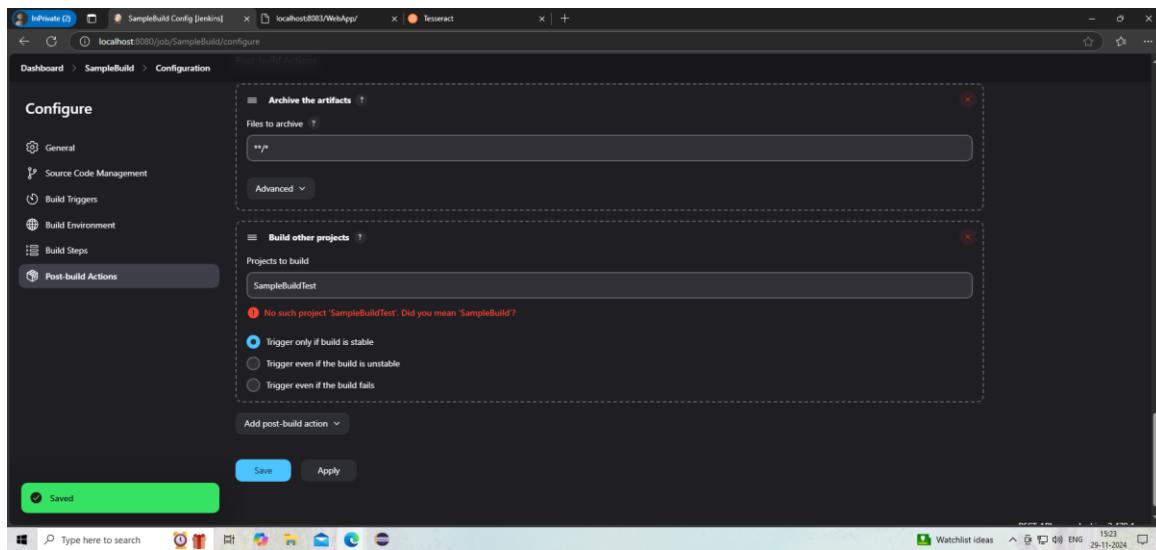
The screenshot shows the Jenkins 'Tools' configuration page. Under 'JDK installations', a new entry named 'JDK21' is being configured. The 'Name' field contains 'JDK21'. The 'JAVA_HOME' field is set to 'C:\Program Files\Java\jdk-21'. The 'Install automatically' checkbox is unchecked. At the bottom are 'Save' and 'Apply' buttons.

The screenshot shows the Jenkins 'Tools' configuration page. Under 'Maven installations', a new entry named 'Maven' is being configured. The 'Name' field contains 'Maven'. The 'MAVEN_HOME' field is set to 'C:\Users\VritMachine\Downloads\apache-maven-3.9.9-bin\apache-maven-3.9.9'. The 'Install automatically' checkbox is unchecked. At the bottom are 'Add Maven', 'Save', and 'Apply' buttons. The Jenkins version '2.479.1' is visible at the bottom right.



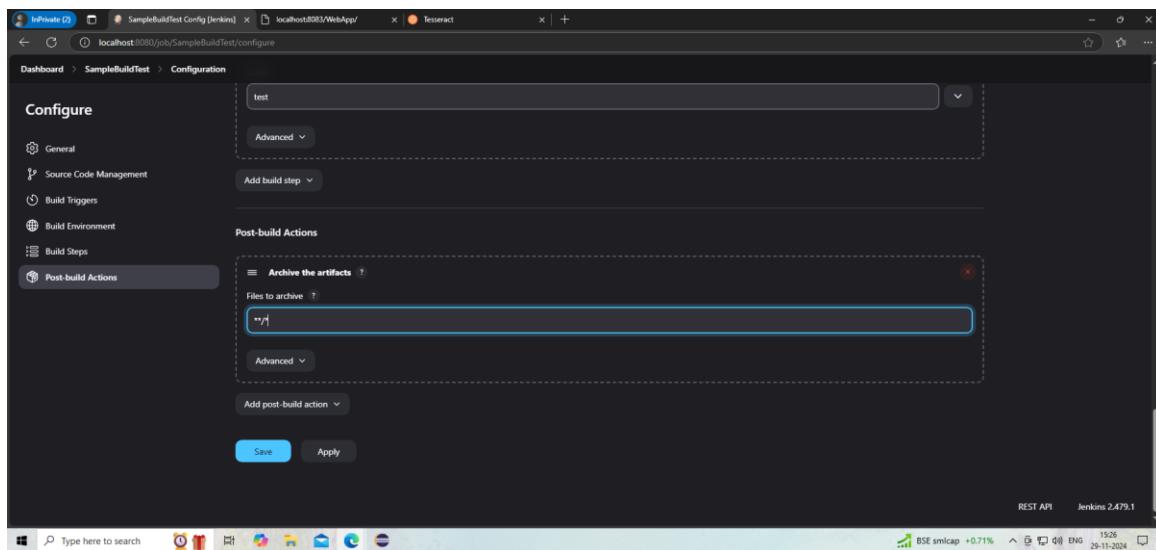
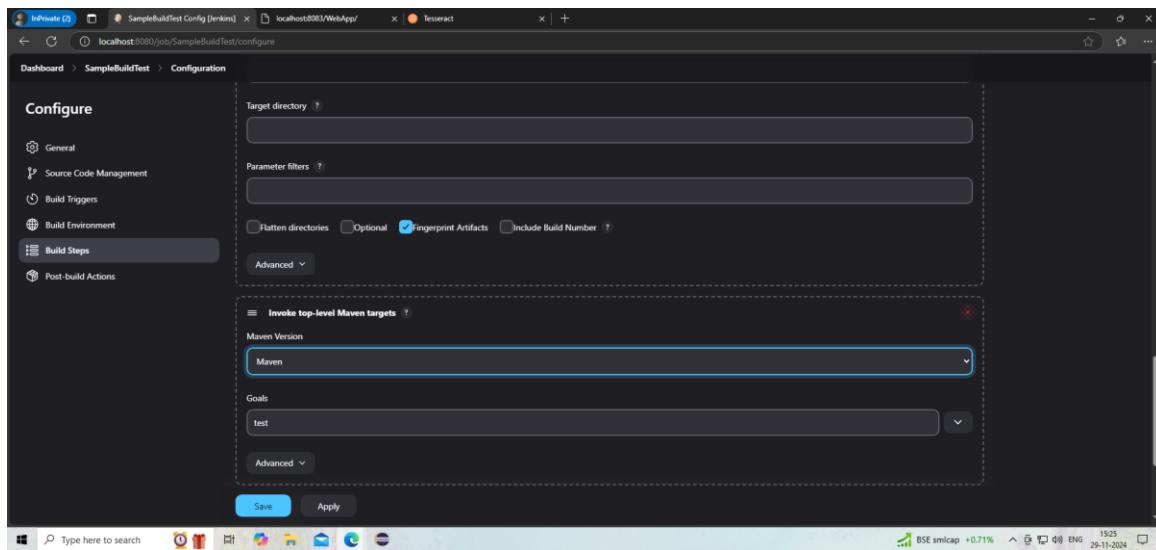






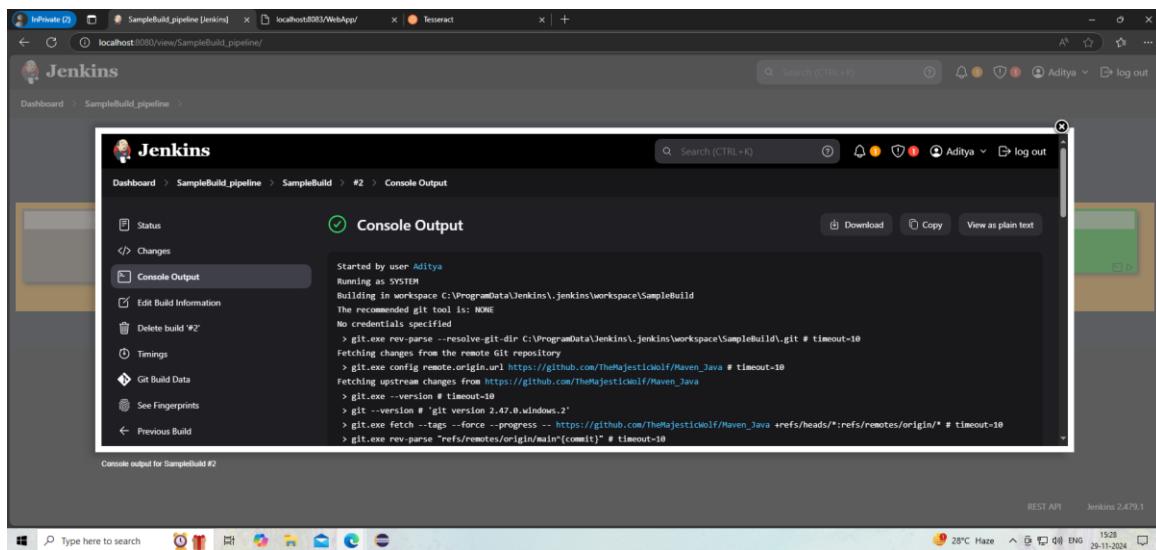
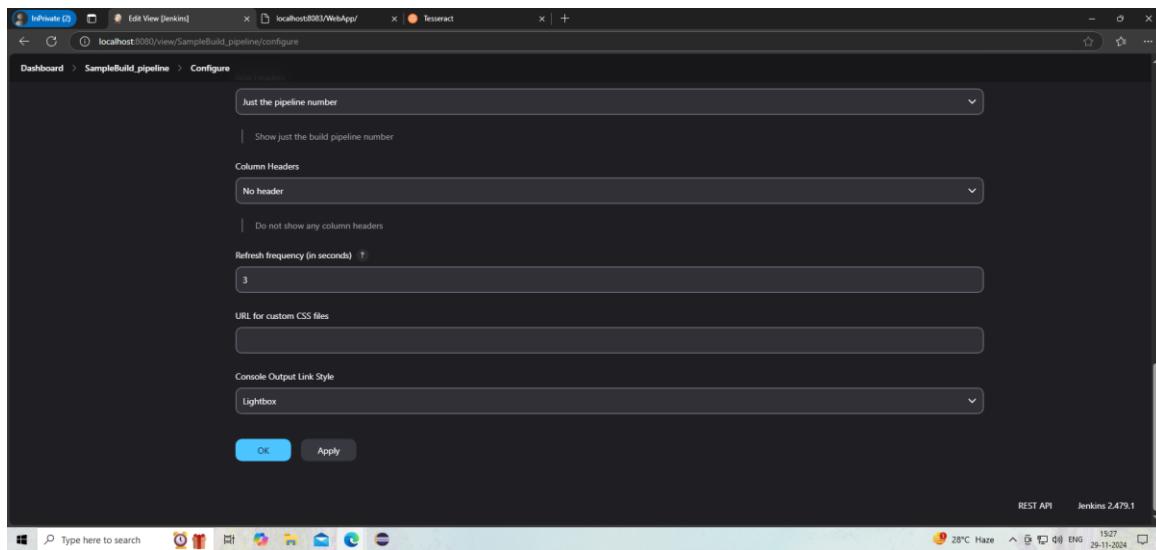
The screenshot shows the Jenkins configuration page for a job named "SampleBuildTest". The left sidebar has sections: General, Source Code Management, Build Triggers (selected), Build Environment, Build Steps, and Post-build Actions. Under Build Triggers, there are options like "Build periodically", "GitHub hook trigger for GITScm polling", and "Poll SCM". Under Build Environment, "Delete workspace before build starts" is checked. Under Build Steps, there is a button "Add build step". Under Post-build Actions, there is a "Save" button.

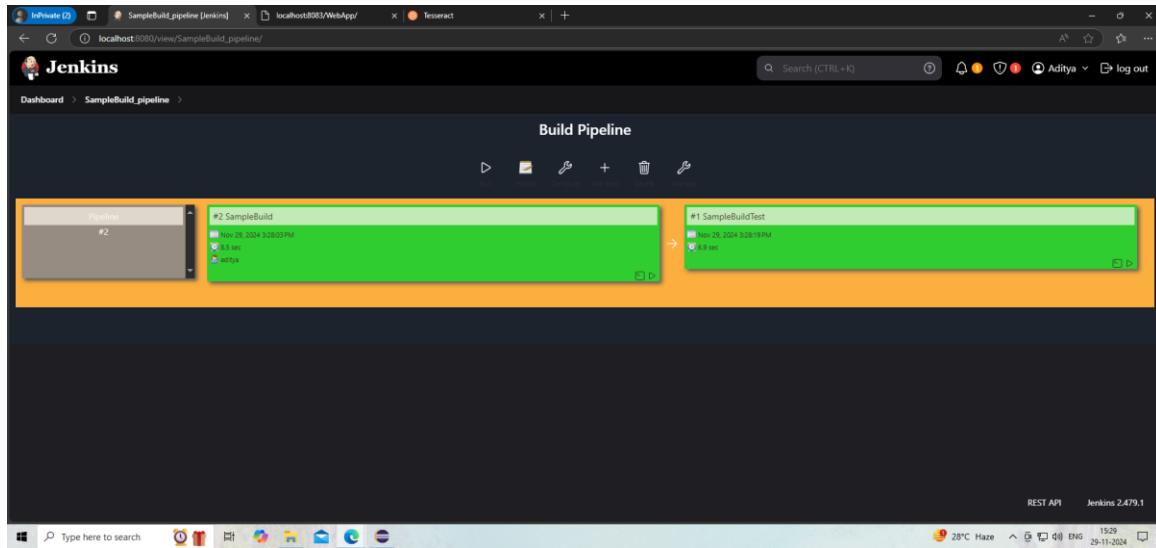
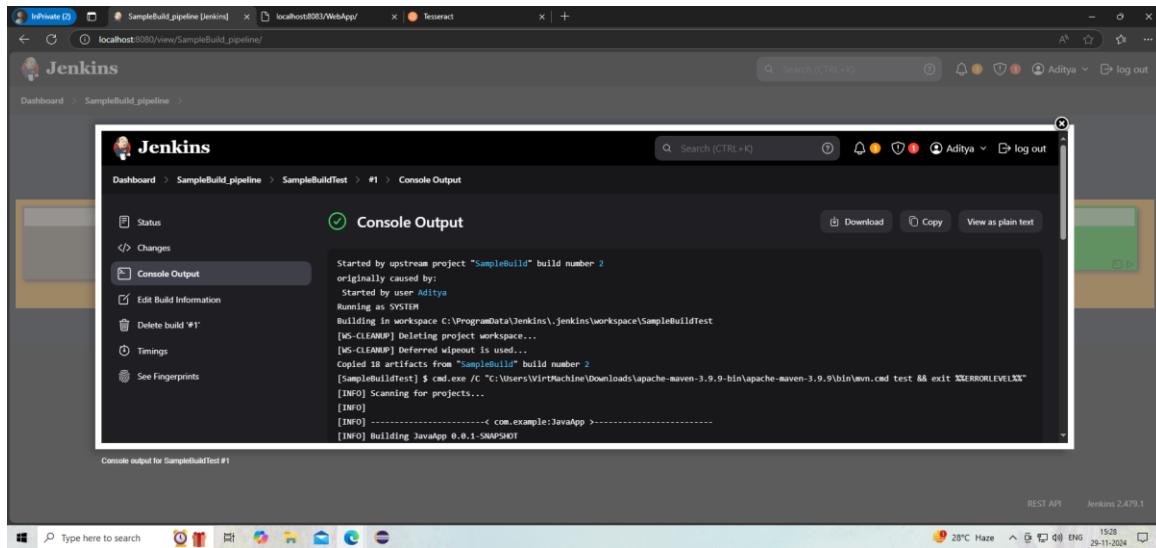
The screenshot shows the Jenkins configuration page for a job named "SampleBuildTest". The left sidebar has sections: General, Source Code Management, Build Triggers, Build Environment, Build Steps (selected), and Post-build Actions. Under Build Steps, a "Copy artifacts from another project" step is configured. It sets the "Project name" to "SampleBuild" and "Which build" to "Latest successful build". The "Stable build only" checkbox is unchecked. Under "Artifacts to copy", the pattern "**/*" is specified. There is also a section for "Artifacts not to copy" and a "Target directory". At the bottom are "Save" and "Apply" buttons.



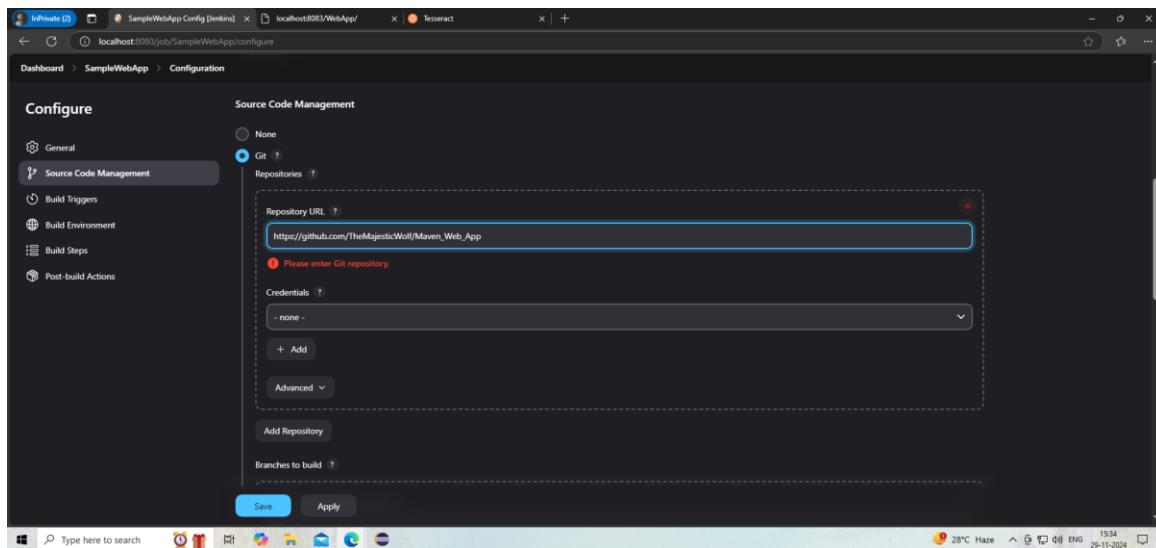
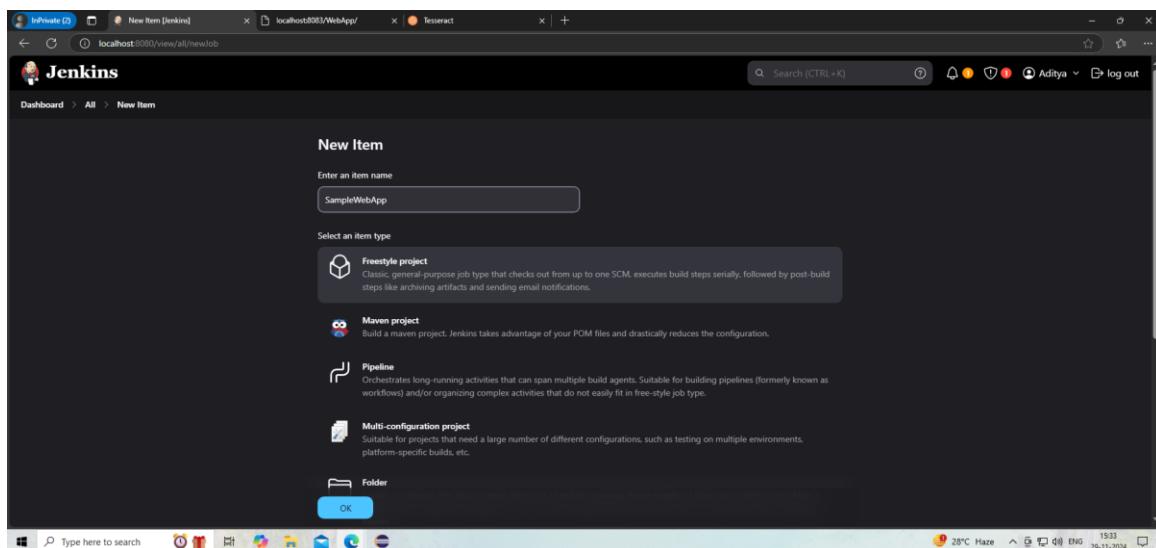
The screenshot shows the Jenkins 'New view' configuration page. The 'Name' field is set to 'SampleBuild_pipeline'. The 'Type' section is set to 'Build Pipeline View', which is described as showing the jobs in a build pipeline view. There are three other options: 'List View' (Shows items in a simple list format) and 'My View' (This view automatically displays all the jobs that the current user has an access to). A 'Create' button is at the bottom.

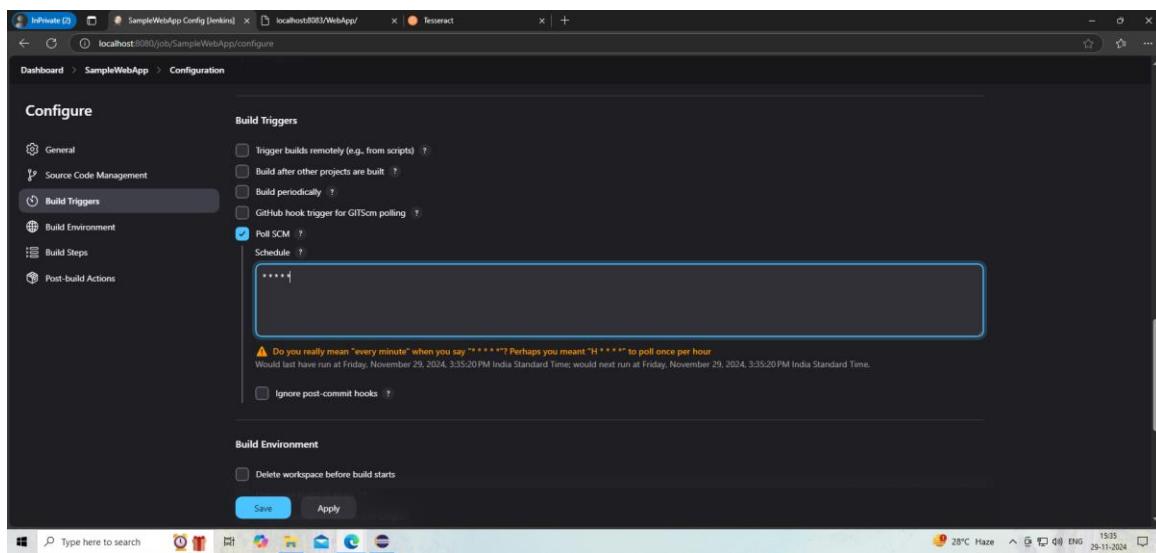
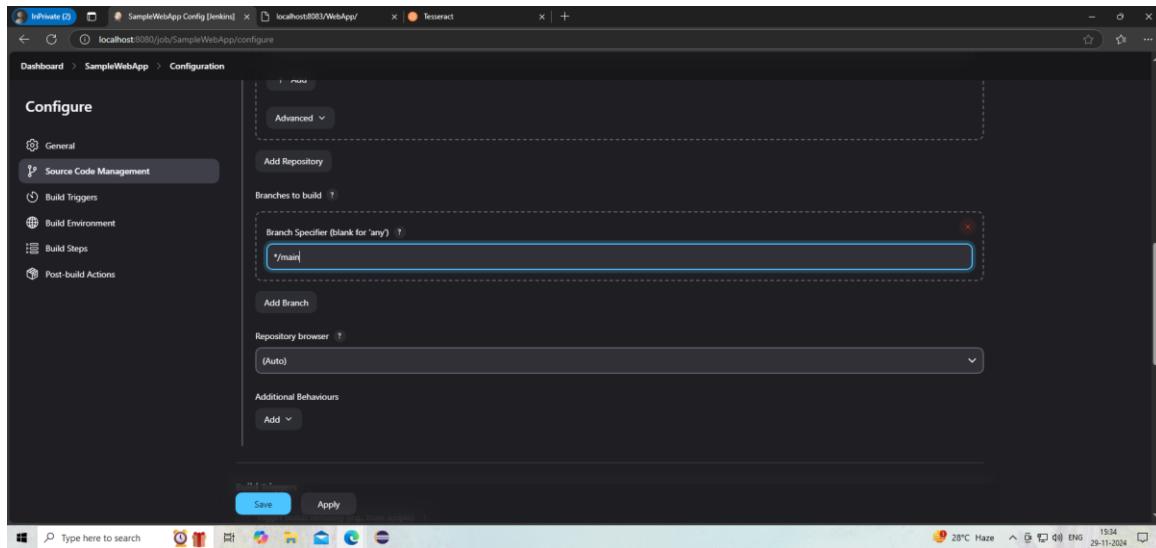
The screenshot shows the Jenkins 'Configure' page for the 'SampleBuild_pipeline' view. Under 'Pipeline Flow', the 'Layout' is set to 'Based on upstream/downstream relationship'. In the 'Upstream / downstream config' section, 'Select Initial Job' is set to 'SampleBuild'. Other options listed in the dropdown are 'SampleBuild' and 'SampleBuildTest'. At the bottom are 'OK' and 'Apply' buttons.

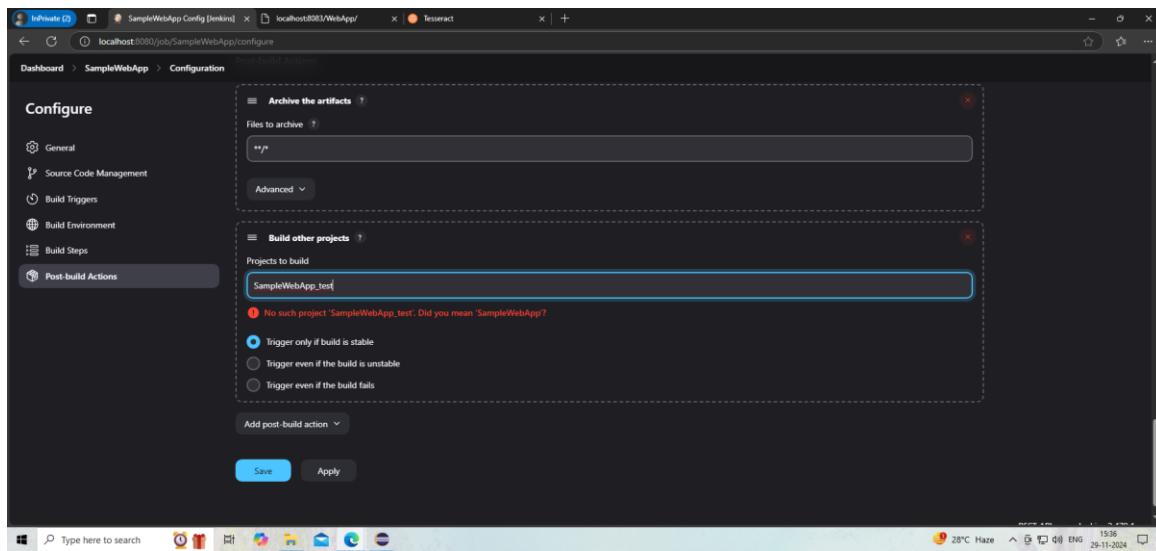
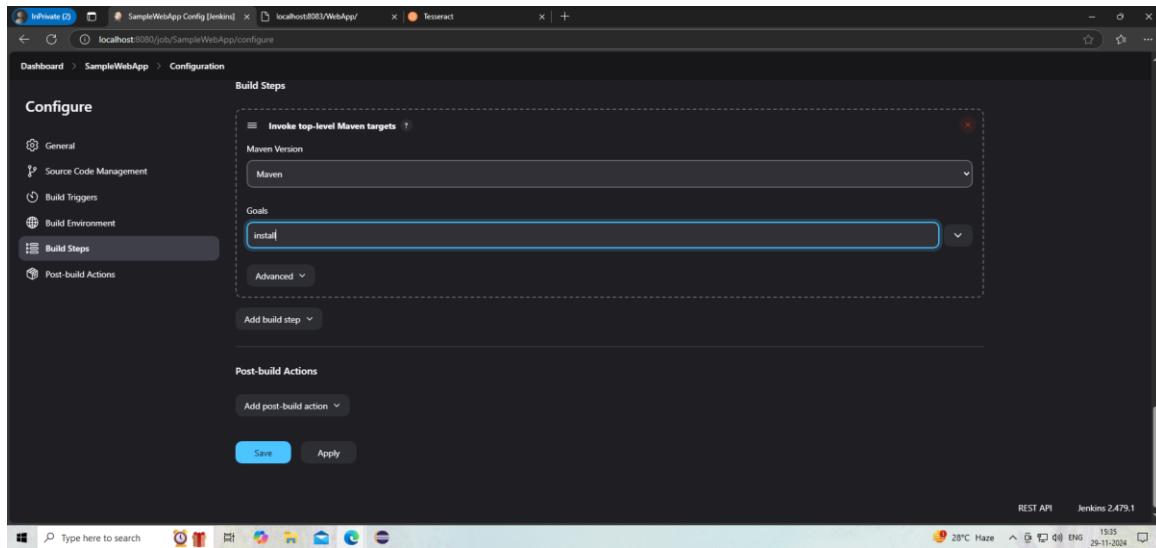


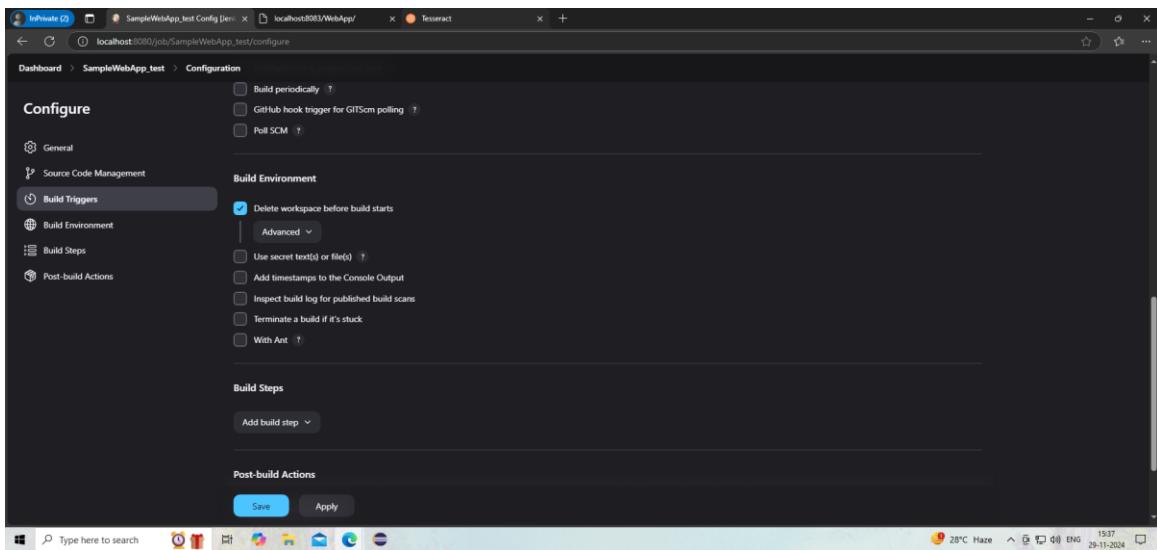
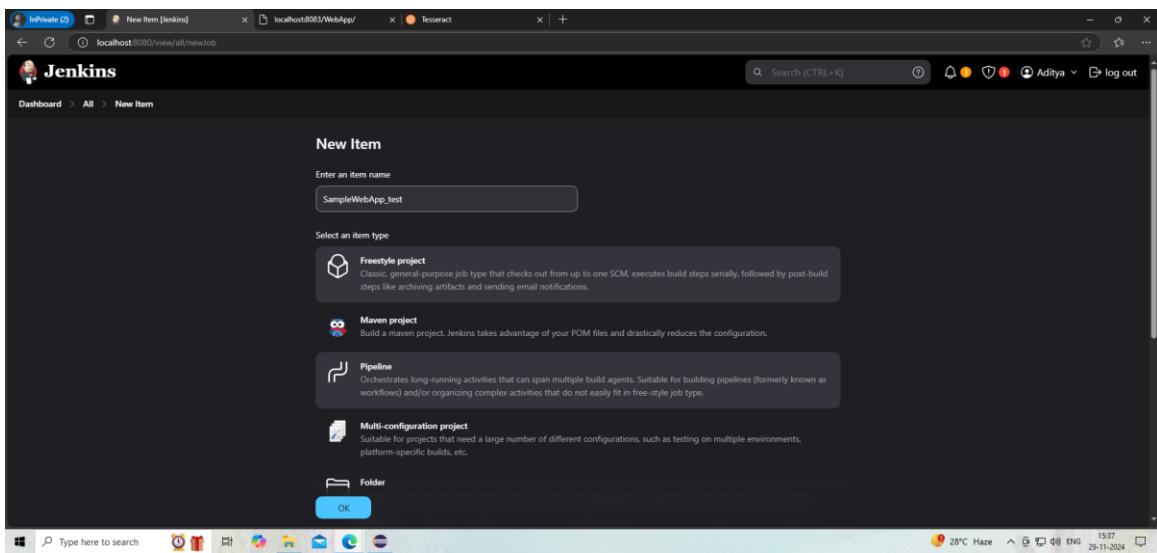


Creating pipeline for Jenkins Web app









InPrivate (2) SampleWebApp_test Config [Edit] localhost:8080/job/SampleWebApp,test/configure Tesseract

Dashboard > SampleWebApp_test > Configuration

Configure

Build Steps

General Source Code Management Build Triggers Build Environment Build Steps Post-build Actions

Copy artifacts from another project

Project name: SampleWebApp

Which build: Latest successful build

Stable build only

Artifacts to copy: **/*

Artifacts not to copy:

Target directory:

Save Apply

Type here to search 28°C Haze 19:38 ENG 29-11-2024

This screenshot shows the 'Build Steps' section of a Jenkins job configuration. It is specifically configured to copy artifacts from another project named 'SampleWebApp'. The 'Which build' dropdown is set to 'Latest successful build'. The 'Artifacts to copy' field contains '**/*', indicating all artifacts should be copied. There are also fields for 'Artifacts not to copy' and 'Target directory'.

InPrivate (2) SampleWebApp_test Config [Edit] localhost:8080/job/SampleWebApp,test/configure Tesseract

Dashboard > SampleWebApp_test > Configuration

Configure

General Source Code Management Build Triggers Build Environment Build Steps Post-build Actions

Invoke top-level Maven targets

Maven Version: Maven

Goals: test

Advanced

Add build step

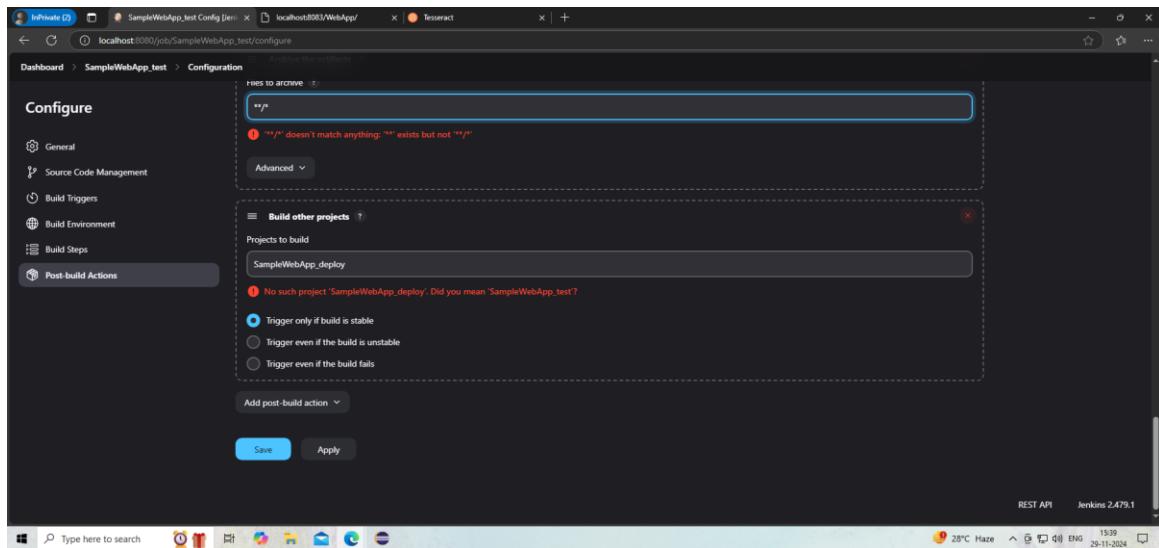
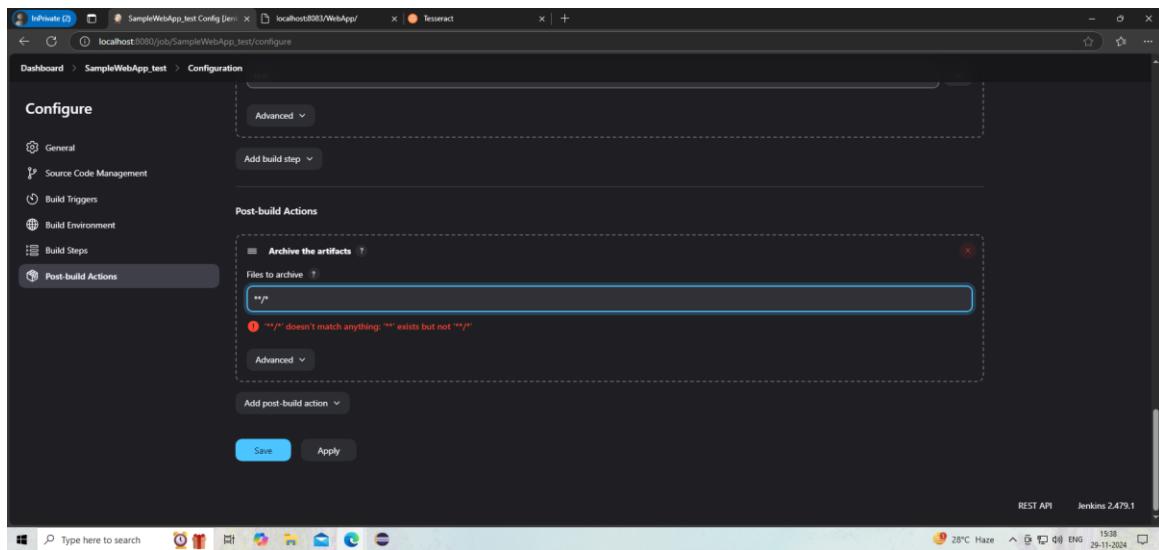
Post-build Actions

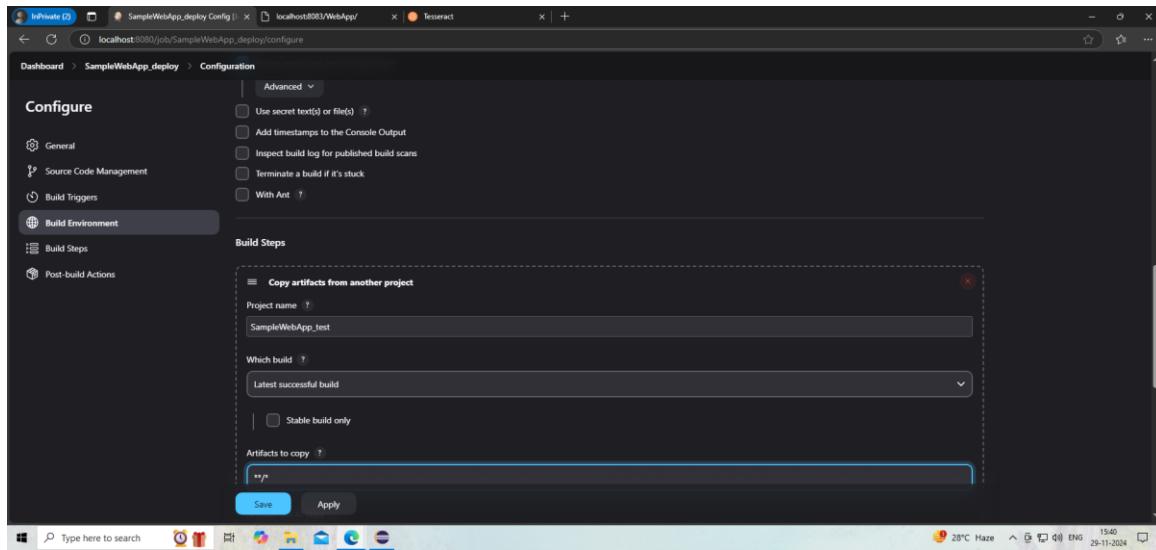
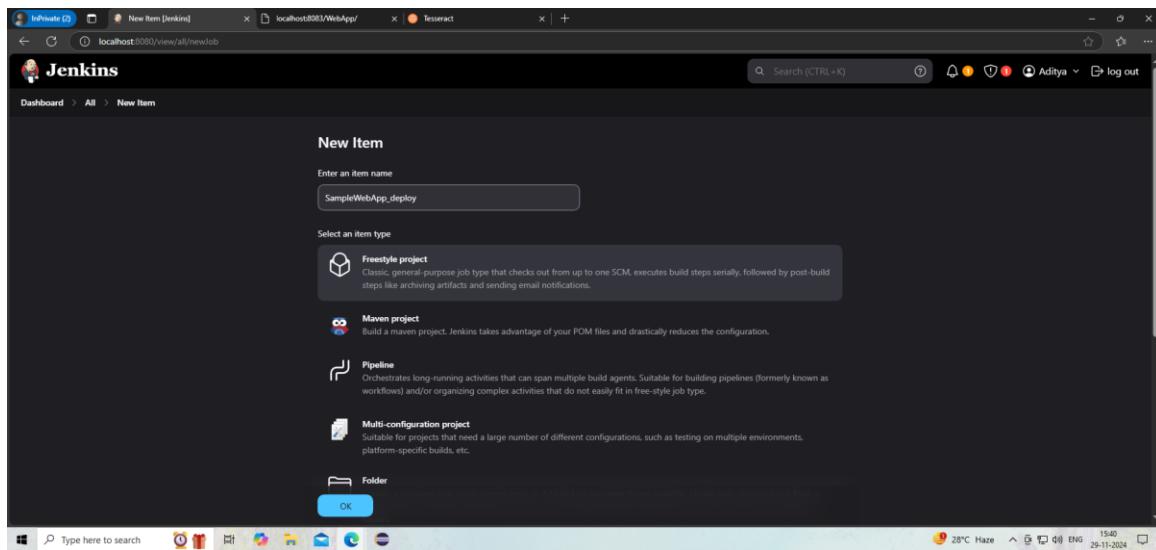
Add post-build action

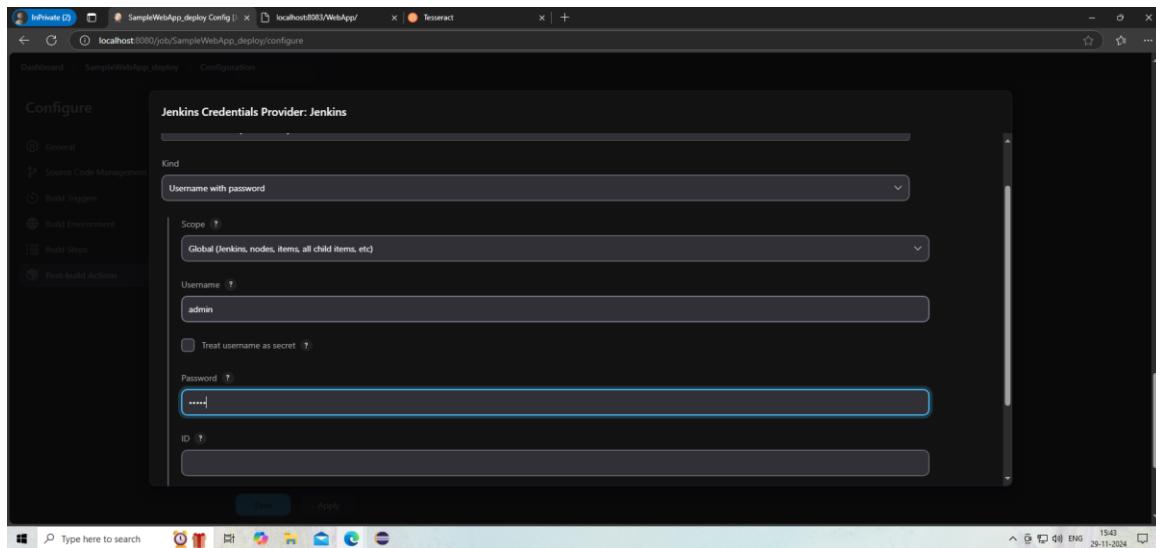
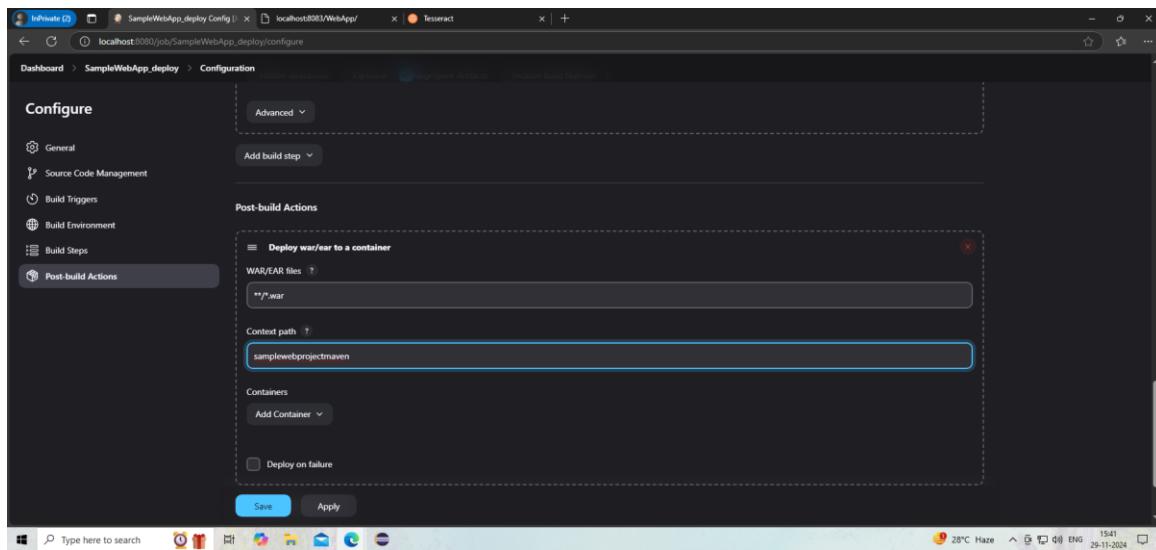
Save Apply

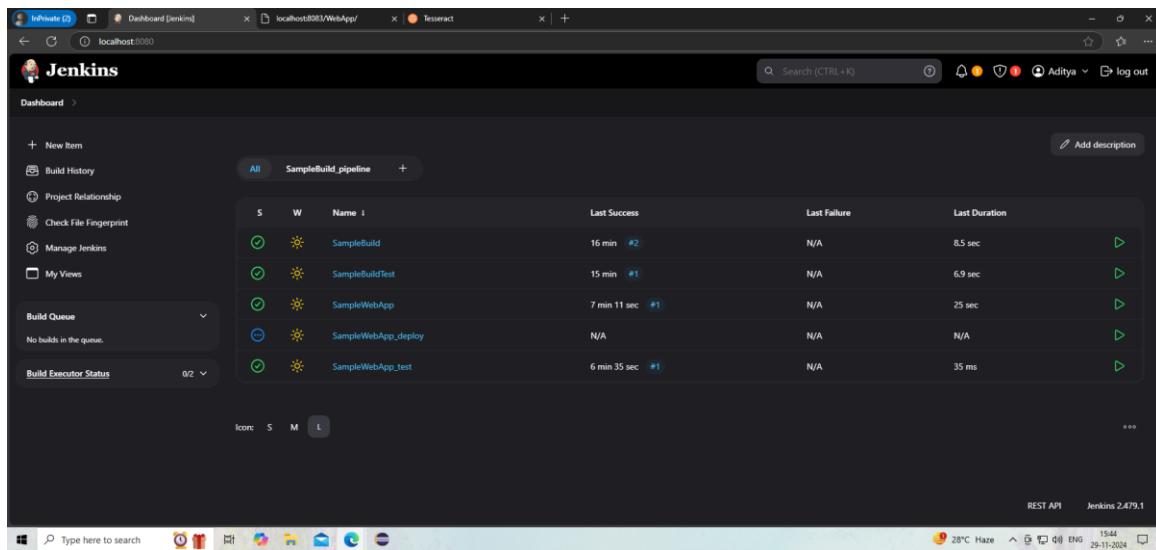
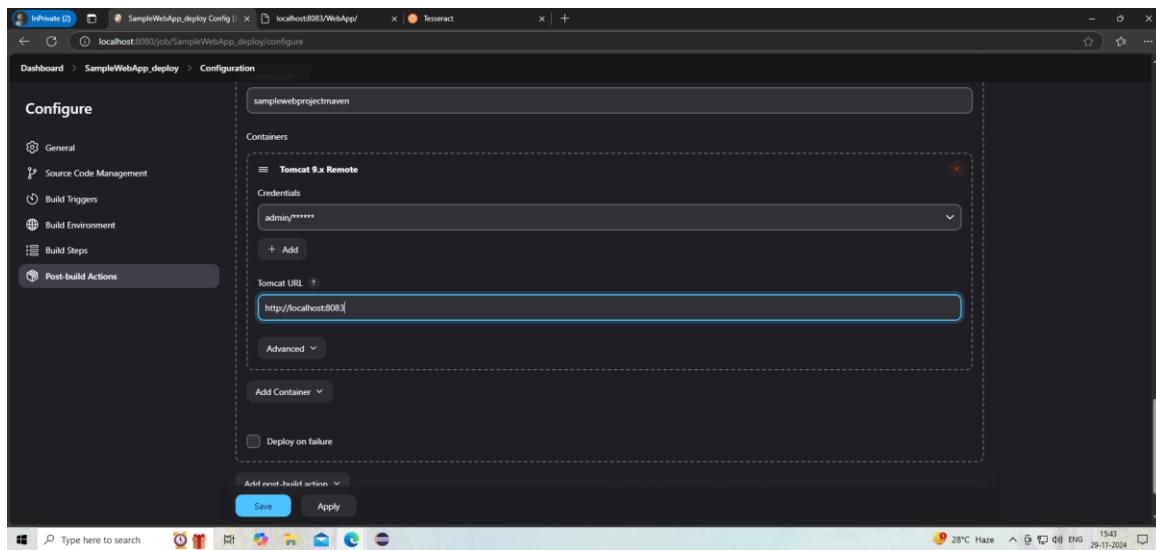
REST API Jenkins 2.479.1 Type here to search 28°C Haze 19:38 ENG 29-11-2024

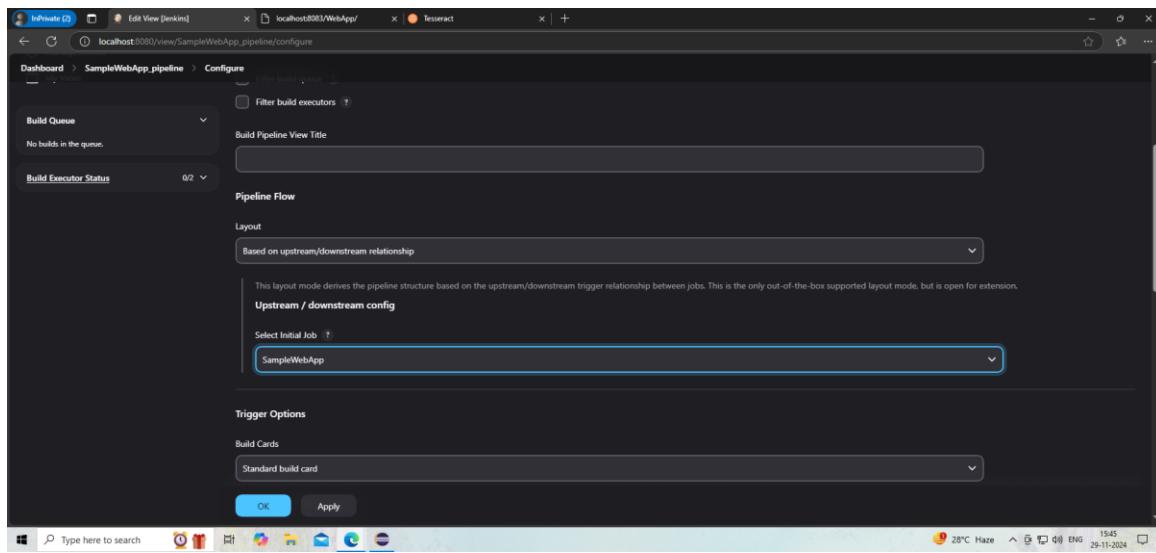
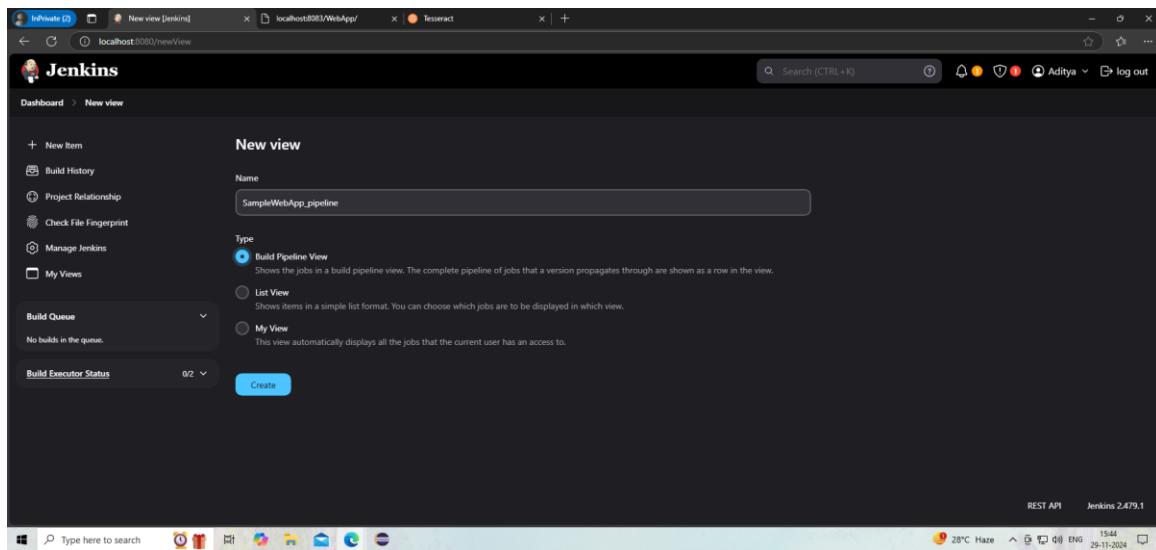
This screenshot shows the 'Build Steps' section of a Jenkins job configuration. It is configured to invoke top-level Maven targets, with 'Maven' selected as the version and 'test' specified as the goals. There is an 'Advanced' button and an 'Add build step' link below the main configuration area. The 'Post-build Actions' section is visible at the bottom.

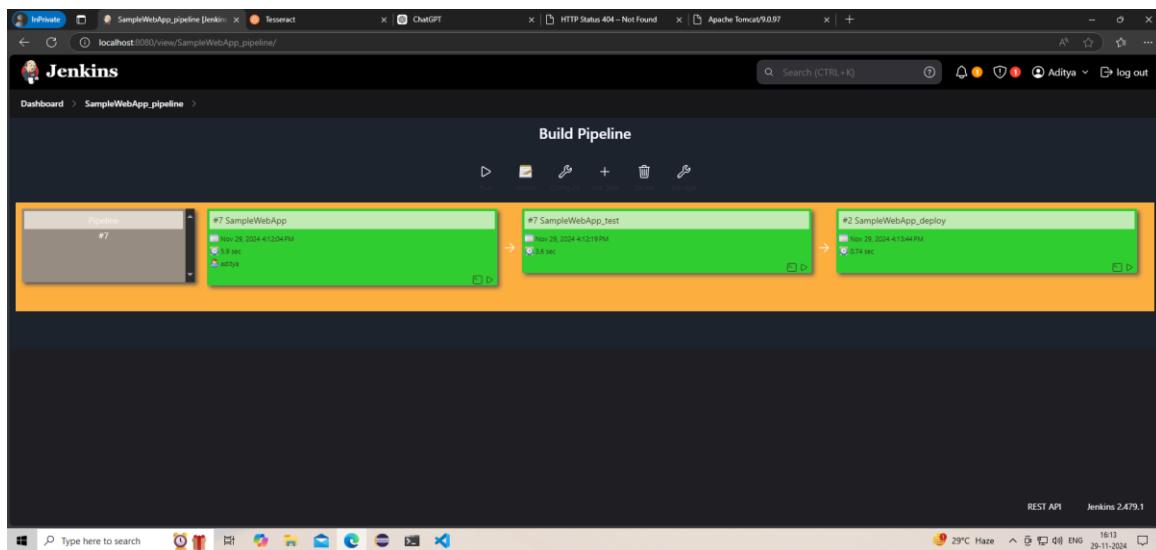
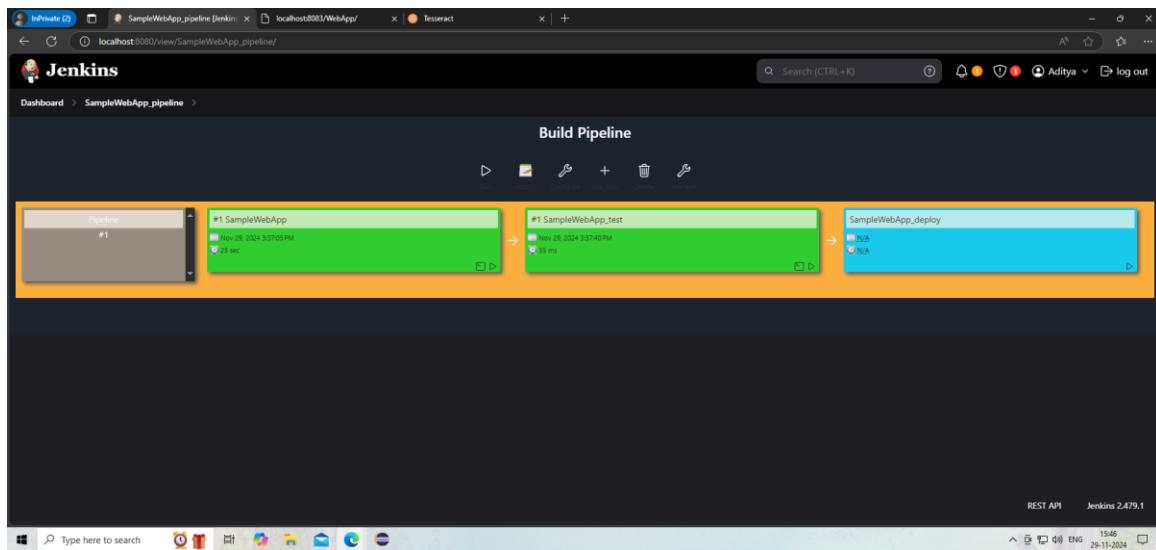


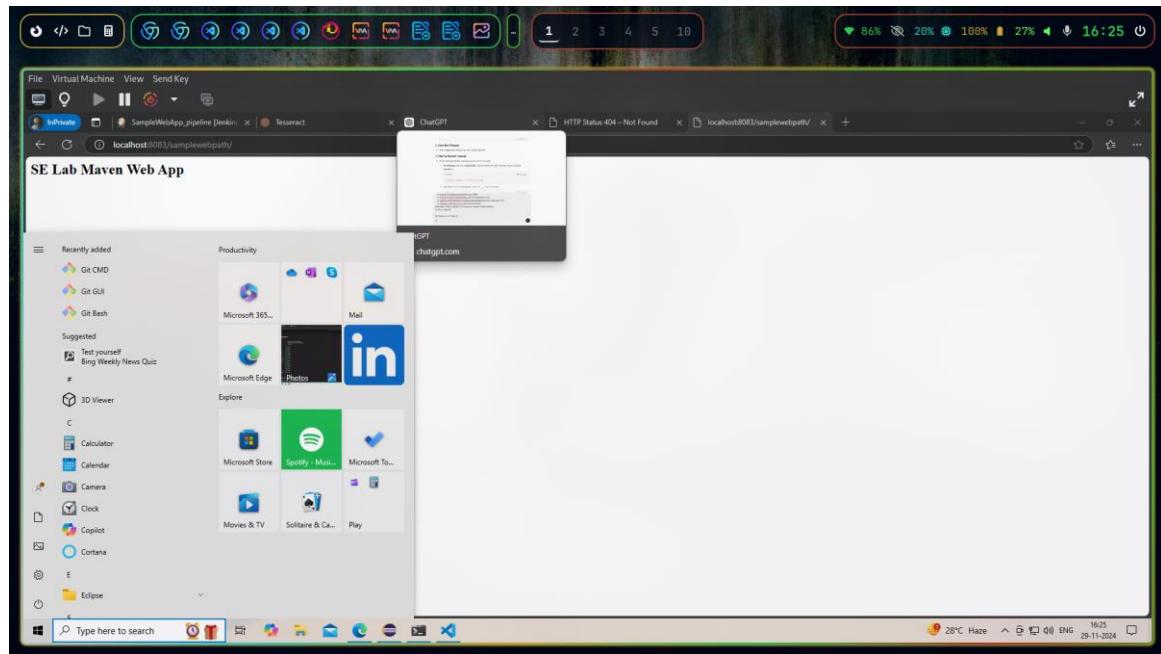
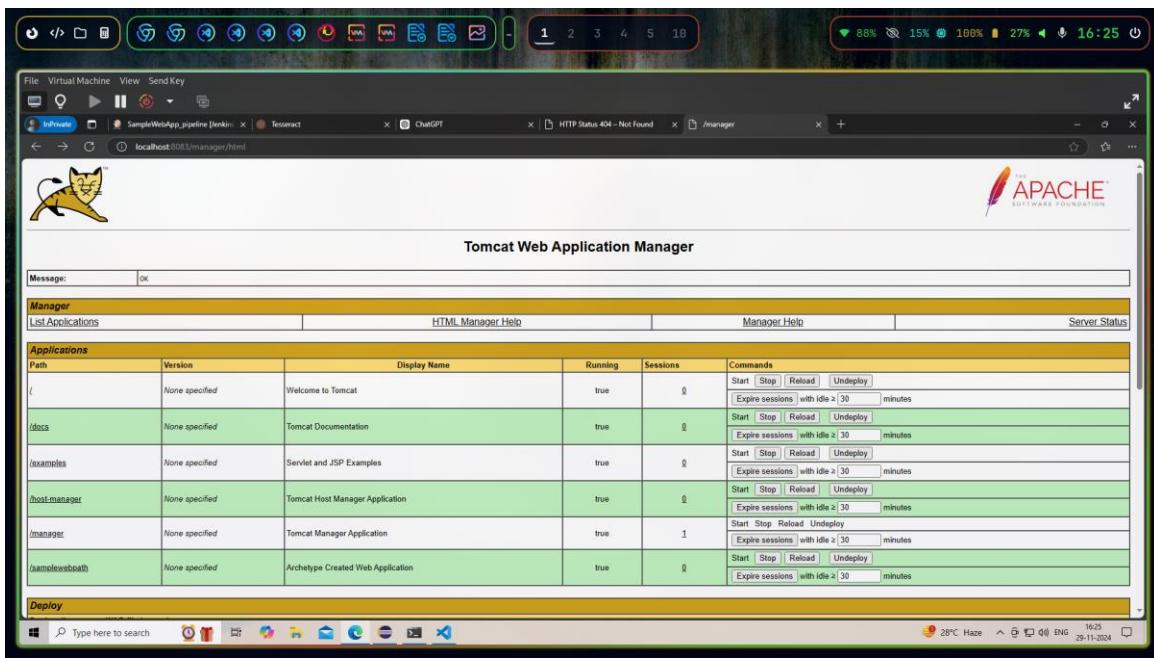


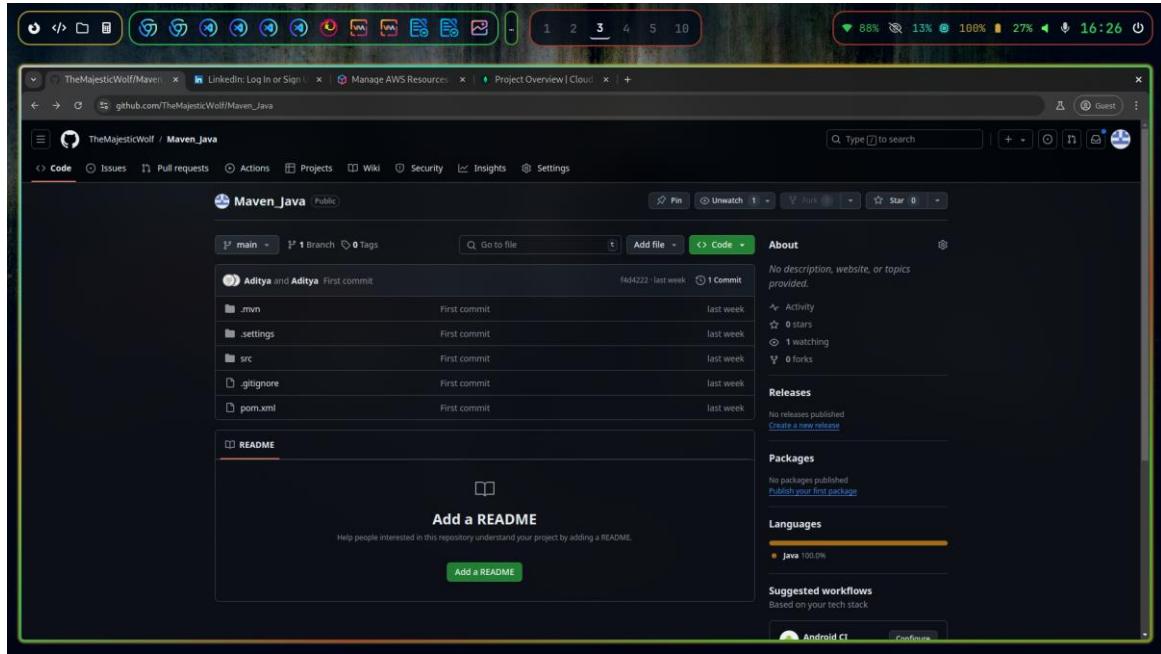
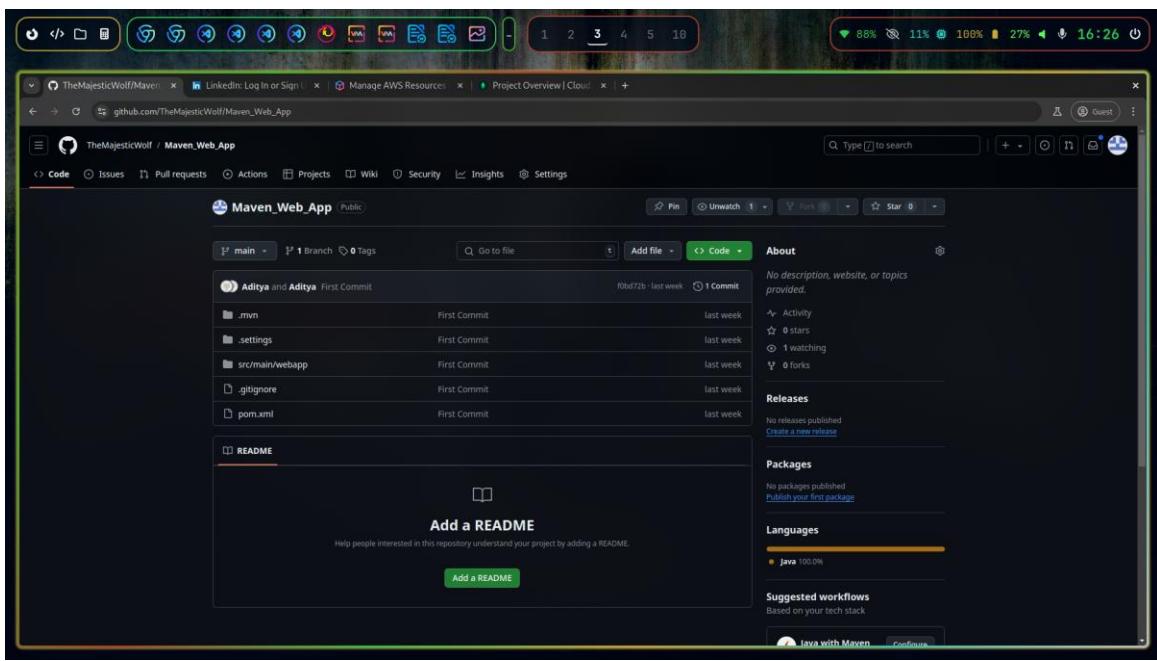












Jenkins Scripted Pipeline with Poll SCM

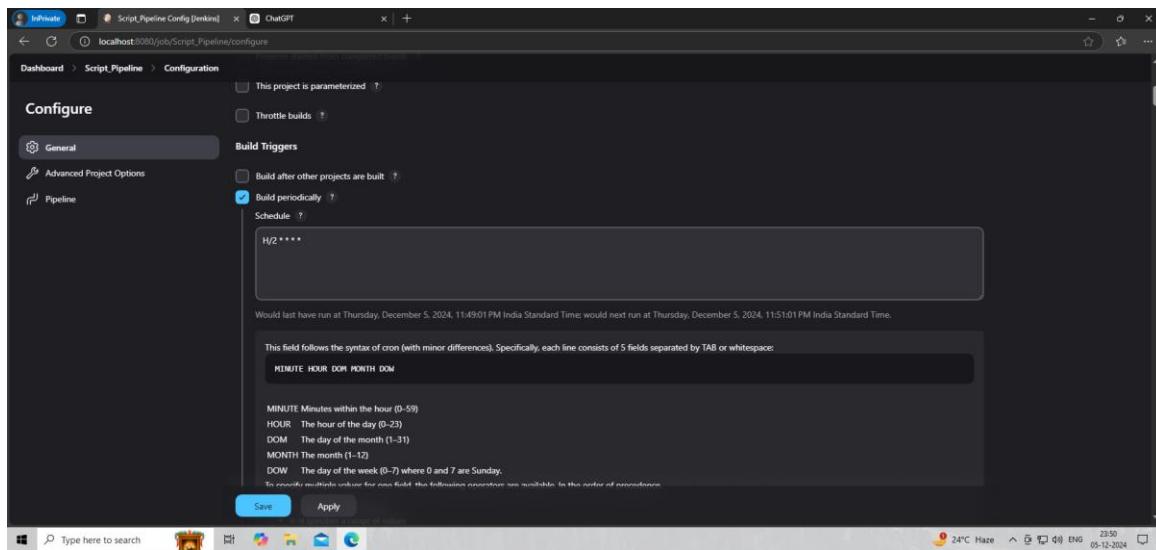
The screenshot shows the Jenkins dashboard with the following details:

- Left Sidebar:** Includes links for "Build History", "Project Relationship", "Check File Fingerprint", "Manage Jenkins", and "My Views".
- Build Queue:** Shows "No builds in the queue."
- Build Executor Status:** Shows "0/2" available executors.
- Job List:** A table with columns: S (Status), W (Last Build), Name, Last Success, Last Failure, and Last Duration. The jobs listed are:
 - SampleBuild (Status: Green, Last Success: 1 day 12 hr #4, Last Failure: N/A, Duration: 16 sec)
 - SampleBuildTest (Status: Green, Last Success: 1 day 12 hr #3, Last Failure: N/A, Duration: 5.8 sec)
 - SampleWebApp (Status: Green, Last Success: 1 day 12 hr #10, Last Failure: N/A, Duration: 10 sec)
 - SampleWebApp_deploy (Status: Green, Last Success: 1 day 12 hr #5, Last Failure: 6 days 7 hr #1, Duration: 0.45 sec)
 - SampleWebApp_test (Status: Green, Last Success: 1 day 12 hr #10, Last Failure: N/A, Duration: 3.5 sec)
- Bottom Status Bar:** Shows "localhost:8080/view/all/newJob" in the address bar, search bar with "Type here to search", system tray icons, and a status bar with "GBP/INR +0.30%", "23:44", and "05-12-2024".

The screenshot shows the "New Item" creation dialog with the following steps:

- Step 1:** "Enter an item name" field containing "Script_Pipeline".
- Step 2:** "Select an item type" dropdown:
 - Freestyle project**: Description: "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." (disabled)
 - Pipeline**: Description: "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." (selected)
 - Maven project**: Description: "Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration." (disabled)
 - Multi-configuration project**: Description: "Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc." (disabled)
- Step 3:** "Folder" button and "OK" button.

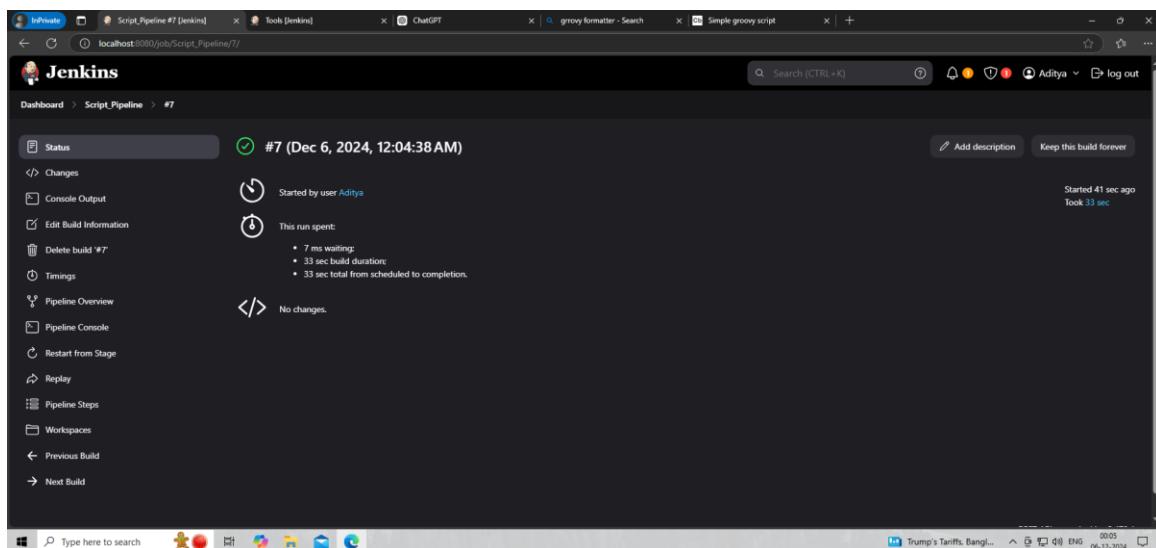
The bottom status bar shows "localhost:8080/view/all/newJob" in the address bar, search bar with "Type here to search", system tray icons, and a status bar with "GBP/INR +0.30%", "23:45", and "05-12-2024".



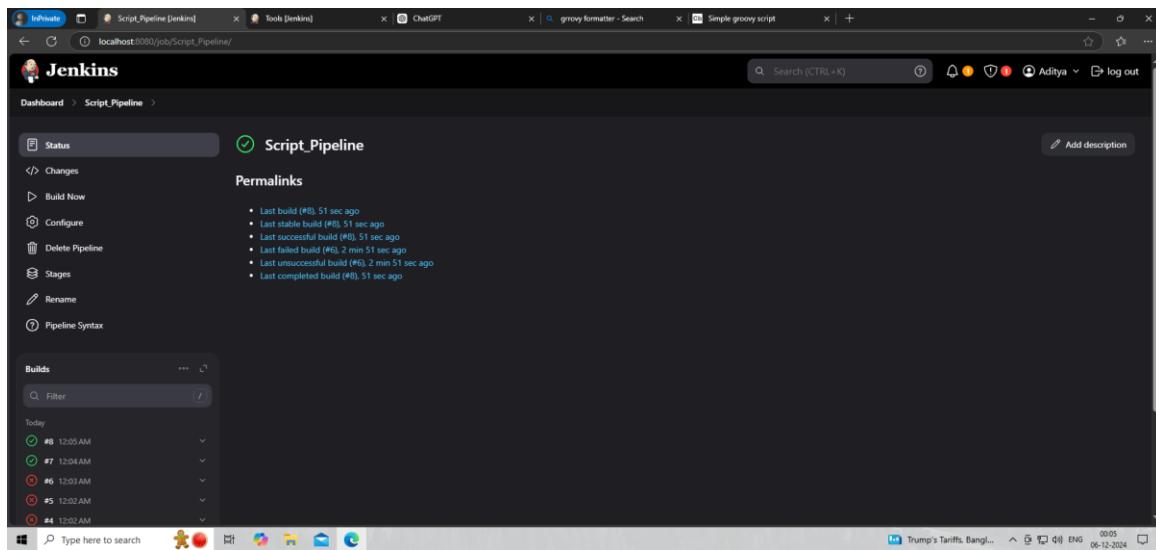
The screenshot shows the Jenkins configuration interface for a 'Script_Pipeline' job. Under the 'Pipeline' tab, the 'Definition' dropdown is set to 'Pipeline script'. The script content is as follows:

```
1 pipeline {
2     agent any
3     tools {
4         maven 'Maven'
5     }
6     stages {
7         stage('git repo & clean') {
8             steps {
9                 if (fileExists 'Maven_Java') {
10                     deleteDir()
11                 }
12                 git 'https://github.com/TheMajesticWolf/Maven_Java.git'
13             }
14         }
15         stage('install') {
16             steps {
17                 maven 'Maven_Java'
18             }
19         }
20         stage('test') {
21             steps {
22                 maven 'Maven_Java'
23             }
24         }
25         stage('package') {
26             steps {
```

A checkbox for 'Use Groovy Sandbox' is checked. At the bottom, there are 'Save' and 'Apply' buttons.



```
[INFO] Running com.example.JavaApp.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.078 s -- in com.example.JavaApp.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.4.2:jar (default-jar) @ JavaApp ---
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 4.554 s
[INFO] Finished at: 2024-12-06T00:05:12+05:30
[INFO] -----
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```



Result: Jenkins was installed and successfully configured for CI/CD, automating the build and deployment processes for the Maven projects.

Experiment-7: Local Deployment of Project using Docker, Kubernetes and Monitoring using Nagios tool

Aim: To explore Docker CLI commands, create and push Docker images and files to DockerHub, run and manage containers using Docker Compose, deploy and scale applications with Minikube, and implement monitoring systems using Nagios.

Introduction:

1. Docker: A platform for containerization that allows developers to package applications and their dependencies into lightweight containers.
 - o Docker CLI Commands: Used for managing containers, images, networks, and volumes.
 - o DockerHub: A cloud-based repository for sharing Docker images.
2. Docker Compose: A tool for defining and managing multi-container applications. It uses a docker-compose.yml file to configure and run multiple containers together.
3. Minikube: A lightweight Kubernetes implementation for deploying and scaling applications locally, simulating a Kubernetes cluster.
4. Nagios: An open-source monitoring system used to track the performance and health of applications, services, and infrastructure.

7A - Docker Commands Part-1

1. docker –version && docker

```
PS C:\Users\anand> docker --version
Docker version 27.3.1, build ce12230
```

2. docker images

```
PS C:\Users\anand> docker images
REPOSITORY          TAG      IMAGE ID   CREATED        SIZE
gcr.io/k8s-minikube/kicbase   v0.0.45  aeed0e1d4642  3 months ago  1.28GB
hello-world          latest   d2c94e258dcf  19 months ago  13.3kB
```

3. docker image rm hello-world

```
PS C:\Users\anand> docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
/
```

For more examples and ideas, visit:
<https://docs.docker.com/get-started/>

4. docker pull redis

```
PS C:\Users\anand> docker pull redis
Using default tag: latest
latest: Pulling from library/redis
9c0965b23a04: Pull complete
9501a6ec095f: Pull complete
98e7597530ef: Pull complete
75dfffa679c9b: Pull complete
3912a88e73c8: Pull complete
141f00d6fee8: Pull complete
4f4fb700ef54: Pull complete
3242f9d5b464: Pull complete
Digest: sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6ae0f83
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
PS C:\Users\anand> |
```

5. docker exec -it my-redis redis-cli

```
Windows PowerShell      X + 
PS C:\Users\anand> docker exec -it newredis redis-cli
127.0.0.1:6379> SET name "Abcdefg"
OK
127.0.0.1:6379> GET name
"Abcdefg"
127.0.0.1:6379> |
```

6. docker stop my-redis && docker start my-redis

7. docker rm my-redis && docker image rm redis:latest

```
PS D:\dockerpro\redis> docker rm newredis
newredis
PS D:\dockerpro\redis> |
```

7A - Docker Commands Part-2

1. docker build -t redisnew .

```
PS D:\redis\redis-docker> docker build -t redisnew .
[+] Building 2.6s (6/6) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 86B
=> [internal] load metadata for docker.io/library/redis:latest
=> [internal] load .dockerrigore
=> => transferring context: 2B
=> [1/1] FROM docker.io/library/redis:latest@sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6ae0f83
=> => resolve docker.io/library/redis:latest@sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6ae0f83
=> [auth] library/redis:pull token for registry-1.docker.io
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:ee573e1f6df4901e37acdc97884314eb51265ccb9c34333ca776a04cf167907a
=> => exporting config sha256:b811c987af0831dd5a1712170ea61a95f065f13a4948c8af664a8b547c4d1928
=> => exporting attestation manifest sha256:e10a4dc15ba704f1900ac8cb4448ec9042ba8a61af1d63504648c90ccaac5f9
=> => exporting manifest list sha256:52a4ad31f39d090c30449bae5e7ad639e0f312e2375d863111a5b543fad2dc5d
=> => naming to docker.io/library/redisnew:latest
=> => unpacking to docker.io/library/redisnew:latest
PS D:\redis\redis-docker> |
```

2. docker run –name myredisnew redisnew && docker ps

```
PS D:\redis\redis-docker> docker run --name myredisnew -d redisnew
038ca4fed49fd6b70ab2a67f3487fa81606e584ebb95e1b117acf97594cd69c4
PS D:\redis\redis-docker> |
```

3. docker stop myredisnew

```
PS D:\redis\redis-docker> docker stop myredisnew
myredisnew
PS D:\redis\redis-docker> |
```

4. docker login

```
PS D:\redis\redis-docker> docker login
Authenticating with existing credentials...
Login Succeeded
PS D:\redis\redis-docker> |
```

5. docker ps -a

```
PS D:\redis\redis-docker> docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS          NAMES
038ca4fed49f      redisnew           "docker-entrypoint.s..."   About a minute ago   Exited (0) 44 seconds ago   myredisnew
b320f5d01862      anand-app         "docker-entrypoint.s..."   36 minutes ago     Exited (0) 36 minutes ago   serene_montalcini
50b775d3a3cf      anand-app:latest  "docker-entrypoint.s..."   37 minutes ago     Exited (0) 37 minutes ago   romantic_kepler
4da780b7b41d      hello-world       "/hello"            3 hours ago       Exited (0) 3 hours ago    funny_panini
da58924546dd      hello-world       "/hello"            3 hours ago       Exited (0) 3 hours ago    naughty_bhaskara
19fb26e3b21f      hello-world       "/hello"            3 hours ago       Exited (0) 3 hours ago    blissful_curran
PS D:\redis\redis-docker> |
```

6. docker commit

```
PS D:\redis\redis-docker> docker commit myredisnew anand/redisnew
sha256:506b72d9a5db8754cf36792c148fe79fafff5213ddb8401905c16321e0be0c87
PS D:\redis\redis-docker> |
```

```
PS D:\redis\redis-docker> docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
anand/redisnew  latest   506b72d9a5db  17 seconds ago  173MB
anand/anand-app latest   8306550e6d38  41 minutes ago  1.35GB
anand/myimages  latest   8306550e6d38  41 minutes ago  1.35GB
anand-app       latest   8306550e6d38  41 minutes ago  1.35GB
redisnew        latest   52a4ad31f39d  2 months ago   173MB
redis           latest   ea96c435dc17  2 months ago   173MB
hello-world     latest   305243c73457  19 months ago  24.4kB
6. docker images
```

7. docker push

```
PS D:\redis\redis-docker> docker push anand/redisnew
Using default tag: latest
The push refers to repository [docker.io/anand/redisnew]
bc0965b23a04: Waiting
75dfffa679c9b: Waiting
141f00d6fee8: Waiting
4f4fb700ef54: Waiting
9501a6ec095f: Waiting
8242f9d5b464: Waiting
98e7597530ef: Waiting
8912a88e73c8: Waiting
```

8. docker rm myredisnew

```
PS D:\redis\redis-docker> docker rm myredisnew
myredisnew
PS D:\redis\redis-docker> |
```

9. docker image rm anand/redis1

```
PS D:\redis\redis-docker> docker rm myredisnew  
myredisnew  
PS D:\redis\redis-docker> |
```

9. docker ps -a

10. docker pull anand/redis1

11. docker run --name myredis -d anand/redis1:latest

12. Running inside the container

13. docker ps -a

14. Stopping container, deleting it and logging out

7B - Modify and Push image

1. docker pull ubuntu

```
> docker pull ubuntu  
Using default tag: latest  
latest: Pulling from library/ubuntu  
de44b265507a: Pull complete  
Digest: sha256:80dd3c3b9c6cecb9f1667e9290b3bc61b78c2678c02cbdae5f0fea92cc6734ab  
Status: Downloaded newer image for ubuntu:latest  
docker.io/library/ubuntu:latest  
[~]/De/A/T/U/A/Aditya/CSM A/III Year/I Semester/SE/Lab/DockerProjects/Redis
```

2. docker run -it --name newubuntu -d ubuntu:latest && docker ps

```
> docker run -it --name newubuntu -d ubuntu:latest  
c736bd017b56a045705a70008c046b483c8d4aae86bbe233c4a29f8f5ebd801  
> docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
c736bd017b56 ubuntu:latest "/bin/bash" 14 seconds ago Up 13 seconds  
bdd6dca08ffa cyber-detective-flask "python flask_server..." 10 days ago Up 18 hours 0.0.0.0:5000->5000/tcp, :::5000->5000/tcp  
f65f34b81661 cyber-detective-backend "docker-entrypoint.s..." 13 days ago Up 18 hours 0.0.0.0:6969->6969/tcp, :::6969->6969/tcp  
98425af9d1a0 cyber-detective-frontend "docker-entrypoint.s..." 13 days ago Up 18 hours 0.0.0.0:3000->3000/tcp, :::3000->3000/tcp  
[~]/De/A/T/U/A/Aditya/CSM A/III Year/I Semester/SE/Lab/DockerProjects/Redis
```

3. apt update && apt install git -y

```

> docker exec -it newubuntu /bin/bash
root@c736bd017b56:/# git --version
bash: git: command not found
root@c736bd017b56:/# apt update
Get:1 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [630 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [15.3 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [689 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [728 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [947 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [679 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [897 kB]
Get:16 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [19.7 kB]
Get:17 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [11.9 kB]
Fetched 26.7 MB in 40s (675 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@c736bd017b56:/# apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  adduser ca-certificates git-man krb5-locales less libbrotli1 libbsd0 libcurl3t64-gnutls libedit2 liberror-perl libexpat1 libfido2-1 libgdbm-compat4t64 libgdbm6t64
  libgssapi-krb5-2 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 libldap-common libldap2 libnghttp2-14 libperl5.38t64 libpsl5t64 librmp1 libssasl2-2 libssasl2-modules
  libssasl2-modules-db libssh-4 libx11-6 libx11-data libxau6 libxcb1 libxdmcp6 libxext6 libxmuu1 netbase openssh-client openssl patch perl perl-modules-5.38 publicsuffix xauth
root@c736bd017b56:/#

```

4. git --version

```

root@c736bd017b56:/# git --version
git version 2.43.0
root@c736bd017b56:/#

```

5. docker commit c736bd017b56 adityaprabhala0101/mynewubuntu

```

> docker commit c736bd017b56 adityaprabhala0101/newubuntu2024
sha256:ed460f6cf36374882584a4a5fe5d12f2ff3e75899dc4aee862ed3e7cb22142

```

6. docker images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
adityaprabhala0101/newubuntu2024	latest	ed460f6cf363	45 seconds ago	205MB
redisnew	latest	fdc482f12c19	32 minutes ago	117MB
cyber-detective-flask	latest	e59b14947f0c	10 days ago	15.3GB
cyber-detective-backend	latest	ac7ebaa8e368	13 days ago	4.81GB
<none>	<none>	0f1b8d1f03ba	13 days ago	4.81GB
<none>	<none>	0d2780e6c9ae	13 days ago	4.1GB
cyber-detective-frontend	latest	837f7a787eb6	13 days ago	2.09GB
<none>	<none>	6be93e52267c	13 days ago	4.1GB
<none>	<none>	3a589cb3f762	13 days ago	2.09GB
<none>	<none>	4369bee310cd	13 days ago	12.7GB
<none>	<none>	e76f32fc7083	2 weeks ago	1.45GB
<none>	<none>	8b636ed49a20	2 weeks ago	10.2GB
<none>	<none>	e7f387e58d46	2 weeks ago	2.09GB
ubuntu	latest	b1d9df8ab815	3 weeks ago	78.1MB
ovmf-vbios-patch	latest	b269970f4810	4 weeks ago	2.93GB

7. docker login && docker push adityaprabhala0101/mynewubuntu:latest

```

Login Succeeded
> docker push adityaprabhala0101/newubuntu2024:latest
The push refers to repository [docker.io/adityaprabhala0101/newubuntu2024]
6ecbd0c7d4c5: Pushed
687d50f2f6a6: Mounted from library/ubuntu
latest: digest: sha256:a8117272d9b8f753042f5c84e1da67d2b20cbfc176a2175db97ab36329dce4 size: 741
[~ /De/A/T/U/A/Aditya/CSM A/III Year/I Semester/SE/Lab/DockerProjects/Redts] ✓ < 58s < 15:18:09

```

8. Deleting containers and images

```

> docker rm newubuntu
newubuntu
> docker image rm adityaprabhala0101/newubuntu2024:latest
Untagged: adityaprabhala0101/newubuntu2024:latest
Untagged: adityaprabhala0101/newubuntu2024@sha256:7a8117272d9b8f753042f5c84e1da67d2b20cbfc176a2175db97ab36329dce4
Deleted: sha256:ed460f6cf36374882584ada5fe5d12f2ffe375899dc4aae862ed3e7cb22142
Deleted: sha256:c7acbfff4dc7bd207d4157abe0fb575459f3c49ff9d00b945bdda05d53faa893
[~ /De/A/T/U/A/Aditya/CSM A/III Year/I Semester/SE/Lab/DockerProjects/Redts] ✓ < 15:20:25

```

9. docker pull adityaprabhala0101/mynewubuntu:latest

```

> docker pull adityaprabhala0101/newubuntu2024
Using default tag: latest
latest: Pulling from adityaprabhala0101/newubuntu2024
de44b265507a: Already exists
c866e63c02f3: Pull complete
Digest: sha256:a8117272d9b8f753042f5c84e1da67d2b20cbfc176a2175db97ab36329dce4
Status: Downloaded newer image for adityaprabhala0101/newubuntu2024:latest
docker.io/adityaprabhala0101/newubuntu2024:latest
[~ /De/A/T/U/A/Aditya/CSM A/III Year/I Semester/SE/Lab/DockerProjects/Redts] ✓ < 59s < 15:22:21

```

10. Running the pulled container

```

> docker run -it --name newubuntu -d adityaprabhala0101/newubuntu2024:latest
6846206c06ceda461ce324d1ac007b9e7cf0f29df98ebc8832671d20d386d6
> docker exec -it newubuntu /bin/bash
root@6846206c06ce:/# git --version
git version 2.43.0
root@6846206c06ce:/#
[~ /De/A/T/U/A/Aditya/CSM A/III Year/I Semester/SE/Lab/DockerProjects/Redts]

```

11. Deleting the images and pulled container

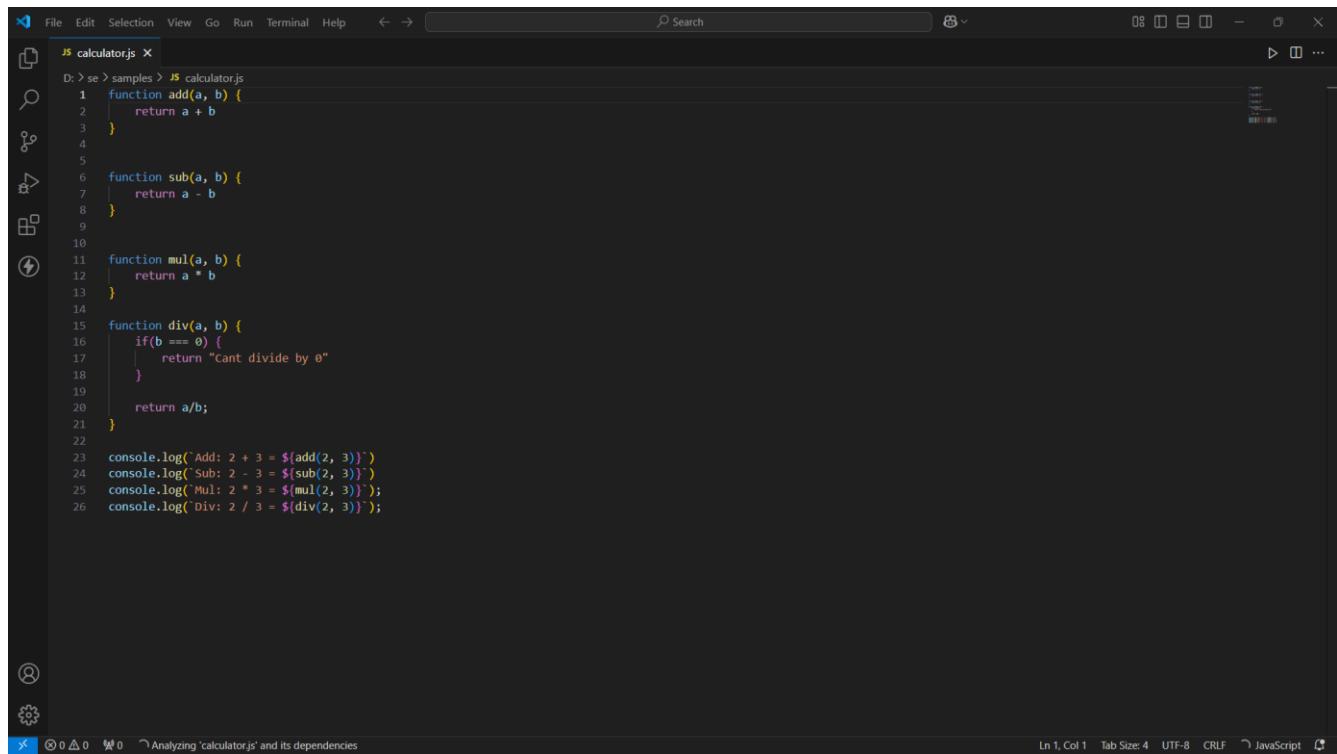
```

> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
6846206c06ce adityaprabhala0101/newubuntu2024:latest "/bin/bash" About a minute ago Up About a minute 0.0.0.0:5000->5000/tcp, :::5000->5000/tcp newubuntu
dd6dcca08fa cyber-detective-flask "python flask_server..." 10 days ago Up 18 hours 0.0.0.0:6969->6969/tcp, :::6969->6969/tcp cyber-detective-flask
65f34b81661 cyber-detective-backend "docker-entrypoint.s..." 13 days ago Up 18 hours 0.0.0.0:6969->6969/tcp, :::6969->6969/tcp cyber-detective-backend
end-1
08425af9d1a0 cyber-detective-frontend "docker-entrypoint.s..." 13 days ago Up 18 hours 0.0.0.0:3000->3000/tcp, :::3000->3000/tcp cyber-detective-frontend
ntend-1
> docker rm newubuntu
Error response from daemon: cannot remove container "/newubuntu": container is running: stop the container before removing or force remove
> docker stop newubuntu
newubuntu
> docker rm newubuntu
newubuntu
> docker image rm adityaprabhala0101/newubuntu2024:latest
Untagged: adityaprabhala0101/newubuntu2024:latest
Untagged: adityaprabhala0101/newubuntu2024@sha256:7a8117272d9b8f753042f5c84e1da67d2b20cbfc176a2175db97ab36329dce4
Deleted: sha256:ed460f6cf36374882584ada5fe5d12f2ffe375899dc4aae862ed3e7cb22142
Deleted: sha256:c7acbfff4dc7bd207d4157abe0fb575459f3c49ff9d00b945bdda05d53faa893
[~ /De/A/T/U/A/Aditya/CSM A/III Year/I Semester/SE/Lab/DockerProjects/Redts] ✓ < 15:26:20

```

7C - Create and Push Dockerfile image

1. Create calculator.js



```
File Edit Selection View Go Run Terminal Help ⏪ ⏩ ⏴ Search D: Y se > samples > JS calculator.js
1 function add(a, b) {
2   return a + b
3 }
4
5
6 function sub(a, b) {
7   return a - b
8 }
9
10
11 function mul(a, b) {
12   return a * b
13 }
14
15 function div(a, b) {
16   if(b === 0) {
17     return "cant divide by 0"
18   }
19
20   return a/b;
21 }
22
23 console.log("Add: 2 + 3 = ${add(2, 3)}")
24 console.log("Sub: 2 - 3 = ${sub(2, 3)})")
25 console.log("Mul: 2 * 3 = ${mul(2, 3)})")
26 console.log("Div: 2 / 3 = ${div(2, 3)})");
```

The screenshot shows a code editor window with a tab labeled 'calculator.js'. The code in the editor is a JavaScript file containing functions for addition, subtraction, multiplication, and division. It also includes several console.log statements to output the results of these operations. The code editor interface includes a toolbar with various icons, a search bar at the top, and a status bar at the bottom indicating file analysis and encoding information.

2. Create Dockerfile



Building docker image

```
> docker build -t simple-calc .
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 3.072kB
Step 1/4 : FROM node:22
22: Pulling from library/node
fdf894e782a2: Pull complete
5bd71677db44: Pull complete
551df7f94f9c: Downloading [=====] 40.31MB/64.39MB
ce82e98d553d: Downloading [=====] 25.84MB/211.3MB
551df7f94f9c: Pull complete
ce82e98d553d: Pull complete
9207a94a3132: Pull complete
788306d1e45b: Pull complete
a6916b58c81c: Pull complete
e650db35f240: Pull complete
Digest: sha256:35a5dd72bcac4bce4326640b58a02be6ff0b6098ffa6f5435aeea980a8951d7
Status: Downloaded newer image for node:22
--> 5dc60a49c105
Step 2/4 : WORKDIR /app
--> Running in 4fd2a0a672df
--> Removed intermediate container 4fd2a0a672df
--> 06fd5e7b58cd
Step 3/4 : COPY calculator.js .
--> a851f86efee2
Step 4/4 : CMD ["node", "calculator.js"]
--> Running in 6995cc9eb96f
--> Removed intermediate container 6995cc9eb96f
--> 76dd9c8fe4f3
Successfully built 76dd9c8fe4f3
Successfully tagged simple-calc:latest
```

4. docker run simple-calc

```
> docker run simple-calc:latest
Add: 2 + 3 = 5
Sub: 2 - 3 = -1
Mul: 2 * 3 = 6
Div: 2 / 3 = 0.6666666666666666
[~/De/A/T/U/A/Aditya/CSM A/III Year/I Semester/SE/Lab/DockerProjects/simplejs] < 15:47:23
```

5. docker tag

```
Login Succeeded
> docker tag simple-calc:latest adityaprabhala0101/simple-calc
[~/De/A/T/U/A/Aditya/CSM A/III Year/I Semester/SE/Lab/DockerProjects/simplejs] < 15:48:43
```

6. Pushing to docker hub

```
> docker push adityaprabhala0101/simple-calc:latest
The push refers to repository [docker.io/adityaprabhala0101/simple-calc]
614c2b6afbc1: Pushed
41526607e3a7: Pushed
e0dba9f2a2ef: Mounted from library/node
b37454b23b8f: Mounted from library/node
fb6c888e3ccc: Mounted from library/node
952f5ee3867b: Mounted from library/node
0aeeeb7c293d: Mounted from library/node
c81d4fdbb7fc: Mounted from library/node
0e82d78b3ea1: Mounted from library/node
301c1bb42cc0: Mounted from library/node
latest: digest: sha256:949adad465edb676117097ced57d333f58a6c151af1231910d31df2eb03c990d size: 2417
[~/De/A/T/U/A/Aditya/CSM A/III Year/I Semester/SE/Lab/DockerProjects/simplejs] < 27s < 15:49:41
```

7. Deleting container and images

```

> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
956c8552a2b0 simple-calc:latest "docker-entrypoint.s..." 2 minutes ago Exited (0) 2 minutes ago nervous_kepler
7b1b8eb2bef d2c94e258dcf "/hello" 6 hours ago Exited (0) 6 hours ago modest_ritchie
3a3fc21e89b0 d2c94e258dcf "/hello" 6 days ago Exited (0) 6 days ago frosty_mayer
f6dad014956b d2c94e258dcf "/hello" 6 days ago Exited (0) 6 days ago festive_margulis
40effa25bd57 cyber-detective-flask:latest "/bin/bash" 10 days ago Exited (130) 10 days ago romantic_lamarr
c3ca95b4a566 cyber-detective-flask:latest "/bin/bash" 10 days ago Exited (130) 10 days ago sad_edison
00146e90875d cyber-detective-flask:latest "/bin/bash" 10 days ago Exited (130) 10 days ago wizardly_goldwasser
bdd60ccaa8fa cyber-detective-flask "python flask_server..." 10 days ago Up 19 hours 0.0.0.0:5000->5000/tcp, :::5000->5000/tcp cyber-detective-flask-1
fafd287f1036 cyber-detective-flask:latest "/bin/bash" 10 days ago Exited (130) 10 days ago musing_hodgkin
4fe86df7c7a9 4369bee310cd "/bin/bash" 10 days ago Exited (130) 10 days ago modest_albattani
24d760a7ba12 cyber-detective-backend:latest "docker-entrypoint.s..." 12 days ago Exited (130) 12 days ago tender_ardinghell
f65f34b81661 cyber-detective-backend "docker-entrypoint.s..." 13 days ago Up 19 hours 0.0.0.0:6969->6969/tcp, :::6969->6969/tcp cyber-detective-backend-1
98425a79d1a0 cyber-detective-frontend "docker-entrypoint.s..." 13 days ago Up 19 hours 0.0.0.0:3000->3000/tcp, :::3000->3000/tcp cyber-detective-frontend-1
510016cecf83 4369bee310cd "/bin/bash" 13 days ago Exited (130) 13 days ago festive_dhawan
a0c13f3125d8 8b636ed49a20 "/bin/bash" 13 days ago Exited (130) 13 days ago thirsty_ishizaka
164ef8118688 e76f32fc7083 "docker-entrypoint.s..." 13 days ago Exited (130) 13 days ago goofy_mendeleev
6834539a01d5 e76f32fc7083 "docker-entrypoint.s..." 13 days ago Exited (130) 13 days ago inspiring_pascal
d67e53c52dd4 e76f32fc7083 "/bin/bash" 2 weeks ago Exited (130) 2 weeks ago beautiful_herschel
0ba8b37feff8 8b636ed49a20 "/bin/bash" 2 weeks ago Exited (130) 2 weeks ago nifty_meitner
84ae0dfdb07a 8b636ed49a20 "-c /bin/bash" 2 weeks ago Created 5000/tcp heuristic_elgamal
b5bcae073c46 ovmf-vbios-patch "/bin/bash" 4 weeks ago Exited (129) 3 weeks ago sleepy_faraday
58150a0c376d5 c46c12b9e2be "/bin/bash" 4 weeks ago Exited (130) 4 weeks ago suspicious_sinoussi
5e07734db381 c46c12b9e2be "/bin/bash" 4 weeks ago Exited (130) 4 weeks ago busy_carver
5cd0d040598a c46c12b9e2be "/bin/bash" 4 weeks ago Exited (130) 4 weeks ago lucid_gould
ad300d33c535 c46c12b9e2be "/bin/bash" 4 weeks ago Exited (130) 4 weeks ago hungry_archimedes
0a6175ae0274 c46c12b9e2be "/bin/bash" 4 weeks ago Exited (130) 4 weeks ago gallant_perlmutter
4db0082f3032b beec17136783 "/bin/bash" 4 weeks ago Exited (130) 4 weeks ago objective_archimedes
5e6f3164b611 a21e51a99310 "/bin/bash" 4 weeks ago Exited (130) 4 weeks ago quirky_montalcini
8ff35a8ac051e ollama/ollama "/bin/ollama serve" 5 weeks ago Exited (0) 3 weeks ago ollama-gpu
159511a6b078 ollama/ollama "/bin/ollama serve" 5 weeks ago Exited (0) 3 weeks ago ollama-cpu
> docker rm 956c8552a2b0 && docker image rm adityaprabhala0101/simple-calc:latest
956c8552a2b0
Untagged: adityaprabhala0101/simple-calc:latest
Untagged: adityaprabhala0101/simple-calc@sha256:949adad465edb676117097ced57d333f58a6c151af1231910d31df2eb03c990d

```

8. docker pull adityaprabhala0101/simple-calc && docker run simple-calc

```

> docker pull adityaprabhala0101/simple-calc
Using default tag: latest
latest: Pulling from adityaprabhala0101/simple-calc
Digest: sha256:949adad465edb676117097ced57d333f58a6c151af1231910d31df2eb03c990d
Status: Downloaded newer image for adityaprabhala0101/simple-calc:latest
docker.io/adityaprabhala0101/simple-calc:latest
> docker run simple-calc:latest
Add: 2 + 3 = 5
Sub: 2 - 3 = -1
Mul: 2 * 3 = 6
Div: 2 / 3 = 0.6666666666666666

```

9. Deleting the containers and removing images

```

> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
05a5b95f51a4 simple-calc:latest "docker-entrypoint.s..." 46 seconds ago Exited (0) 45 seconds ago sleep_euclid
7b1bd8eb2fef d2c94e258dcb "/hello" 6 hours ago Exited (0) 6 hours ago modest_ritchie
3a3fc21e89b0 d2c94e258dcb "/hello" 6 days ago Exited (0) 6 days ago frosty_mayer
f6dada1495b6 d2c94e258dcb "/hello" 6 days ago Exited (0) 6 days ago festive_margulis
40effa25bd57 cyber-detective-flask:latest "/bin/bash" 10 days ago Exited (130) 10 days ago romantic_lamarr
c3ca95b4b5e6 cyber-detective-flask:latest "/bin/bash" 10 days ago Exited (130) 10 days ago sad_edison
00146e90875d cyber-detective-flask:latest "/bin/bash" 10 days ago Created wizardly_goldwasser
bdd6dcfa08fa cyber-detective-flask "python flask_server..." 10 days ago Up 19 hours 0.0.0.0:5000->5000/tcp, :::5000->5000/tcp
fafd287f1036 cyber-detective-flask:latest "/bin/bash" 10 days ago Exited (130) 10 days ago cyber-detective-flask-1
4fe86df7c7a9 4369bee310cd "/bin/bash" 10 days ago Exited (130) 10 days ago musing_hodgkin
0.0.0.0:6969->6969/tcp, :::6969->6969/tcp
0.0.0.0:3000->3000/tcp, :::3000->3000/tcp
cyber-detective-backend-1
cyber-detective-frontend-1
festive_dhawan
thirsty_iszhaka
goofy_mendelev
inspiring_pascal
beautiful_herschel
nifty_meitner
heuristic_elgamal
sleepy_faraday
suspicious_sinoussi
busy_carver
lucid_gould
hungry_archimedes
gallant_perlmans
objective_archimedes
quirky_montalcini
ollama-gpu
ollama-cpu
24d760a7ba12 cyber-detective-backend:latest "docker-entrypoint.s..." 12 days ago Exited (130) 12 days ago
f65f34ab1661 cyber-detective-backend "docker-entrypoint.s..." 13 days ago Up 19 hours
98425af9d1a0 cyber-detective-frontend "docker-entrypoint.s..." 13 days ago Up 19 hours
518016eceef83 4369bee310cd "/bin/bash" 13 days ago Exited (130) 13 days ago
a0c13f3125d8 8b636ed49a20 "/bin/bash" 13 days ago Exited (130) 13 days ago
164ef81186a8 e76f32fc7883 "/docker-entrypoint.s..." 13 days ago Exited (130) 13 days ago
6834539a01d5 e76f32fc7883 "/docker-entrypoint.s..." 13 days ago Exited (130) 13 days ago
d67e53c52d4 e76f32fc7883 "/docker-entrypoint.s..." 2 weeks ago Exited (130) 2 weeks ago
0baa8b37feff8 8b636ed49a20 "/bin/bash" 2 weeks ago Exited (130) 2 weeks ago
84ae0fd807a 8b636ed49a20 "-c /bin/bash" 2 weeks ago Created
b5bcae073c6a6 ovmf-vbios-patch "/bin/bash" 4 weeks ago Exited (129) 3 weeks ago
58150a0c376d5 c46c12b9e2be "/bin/bash" 4 weeks ago Exited (130) 4 weeks ago
5e07734d3b81 c46c12b9e2be "/bin/bash" 4 weeks ago Exited (130) 4 weeks ago
5cdda040590a c46c12b9e2be "/bin/bash" 4 weeks ago Exited (130) 4 weeks ago
ad30cd33c535 c46c12b9e2be "/bin/bash" 4 weeks ago Exited (130) 4 weeks ago
0a6175aea0274 c46c12b9e2be "/bin/bash" 4 weeks ago Exited (130) 4 weeks ago
4db082f3032b beeec17136783 "/bin/bash" 4 weeks ago Exited (130) 4 weeks ago
5e6f316486b11 a21e51a49310 "/bin/bash" 4 weeks ago Exited (130) 4 weeks ago
8ff35a8c051e ollama/ollama "/bin/ollama serve" 5 weeks ago Exited (0) 3 weeks ago
159511a6a0878 ollama/ollama "/bin/ollama serve" 5 weeks ago Exited (0) 3 weeks ago
> docker rm 05a5b95f51a4 && docker image rm simple-calc:latest && docker logout
Untagged: simple-calc:latest
Removing login credentials for https://index.docker.io/v1/

```

10. DockerHub

The screenshot shows the Docker Hub homepage. At the top, there's a navigation bar with various icons and a search bar. Below the header, there are tabs for 'Explore', 'Repositories' (which is selected), 'Organizations', and 'Usage'. A search bar labeled 'Search Docker Hub' is positioned above the repository list.

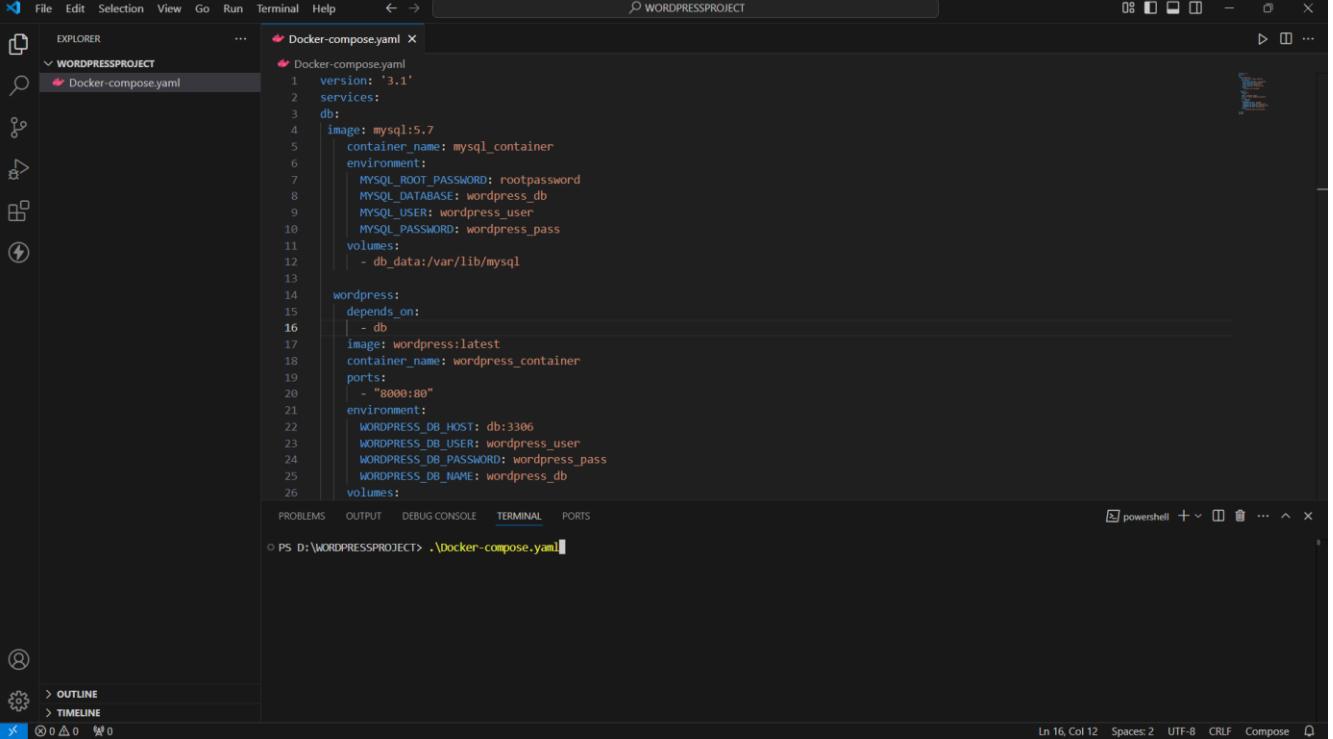
The main content area shows a table of repositories:

Name	Last Pushed	Contains	Visibility	Scout
adityaprabhala0101/simple-calc	7 minutes ago	IMAGE	Public	Inactive
adityaprabhala0101/newubuntu2024	38 minutes ago	IMAGE	Public	Inactive
adityaprabhala0101/redis1	about 1 hour ago	IMAGE	Public	Inactive

At the bottom of the table, it says '1-3 of 3'. To the right of the table, there's a large blue button labeled 'Create a repository'. Below that, there's a section titled 'Create an organization' with a small icon. Further down, there's another section with the Docker logo and some text.

7D – Multiple containers using Docker Compose

1. Create a new directory
2. Open the directory in a text editor
3. Create a file “docker-compose.yaml”



The screenshot shows a code editor window titled "WORDPRESSPROJECT". The left sidebar is labeled "EXPLORER" and shows a folder named "WORDPRESSPROJECT" containing a file named "Docker-compose.yaml". The main editor area displays the following YAML configuration:

```
version: '3.1'
services:
  db:
    image: mysql:5.7
    container_name: mysql_container
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
      MYSQL_DATABASE: wordpress_db
      MYSQL_USER: wordpress_user
      MYSQL_PASSWORD: wordpress_pass
    volumes:
      - db_data:/var/lib/mysql

  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    container_name: wordpress_container
    ports:
      - "8000:80"
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress_user
      WORDPRESS_DB_PASSWORD: wordpress_pass
      WORDPRESS_DB_NAME: wordpress_db
    volumes:
```

Below the editor, there are tabs for "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", "TERMINAL", and "PORTS". A terminal window at the bottom shows the command: "PS D:\WORDPRESSPROJECT> .\Docker-compose.yaml". The status bar at the bottom right indicates "Ln 16, Col 12" and "Compose".

4. In the terminal type docker compose up -d

The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the command `docker-compose up -d` being run, followed by a list of successful operations:

```
PS D:\WORDPRESSPROJECT> docker-compose up -d
[+] Running 3/2
  ✓ db Pulled
  ✓ wordpress Pulled
[+] Running 4/4
  ✓ Network wordpressproject_default Created
  ✓ Volume "wordpressproject_db_data" Created
  ✓ Container mysql_container Started
  ✓ Container wordpress_container Started
D:\WORDPRESSPROJECT>
```

The status bar at the bottom indicates the terminal is using powershell, has 30 lines, 1 column, 2 spaces, UTF-8 encoding, and CRLF line endings.

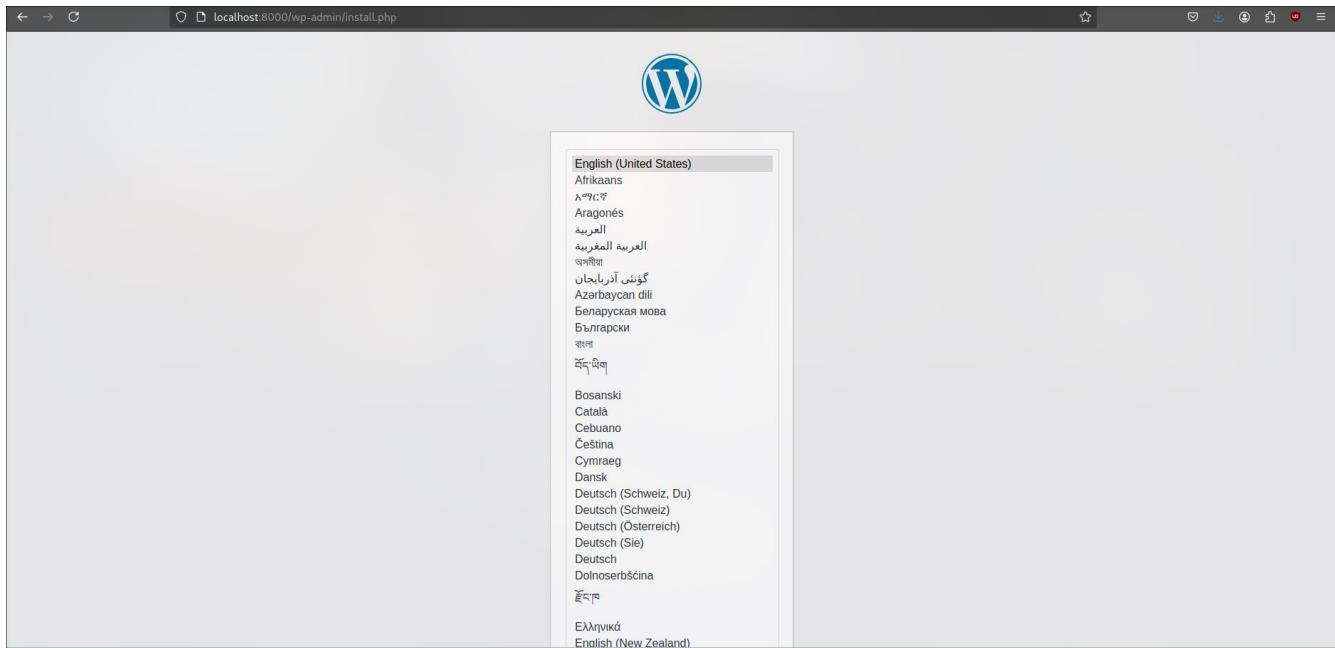
5. It means everything is working

This screenshot is identical to the one above, showing the same terminal output for the command `docker-compose up -d`. The terminal window displays the command and a list of successful operations, indicating that the Docker Compose project is running correctly.

```
PS D:\WORDPRESSPROJECT> docker-compose up -d
[+] Running 3/2
  ✓ db Pulled
  ✓ wordpress Pulled
[+] Running 4/4
  ✓ Network wordpressproject_default Created
  ✓ Volume "wordpressproject_db_data" Created
  ✓ Container mysql_container Started
  ✓ Container wordpress_container Started
D:\WORDPRESSPROJECT>
```

The status bar at the bottom indicates the terminal is using powershell, has 30 lines, 1 column, 2 spaces, UTF-8 encoding, and CRLF line endings.

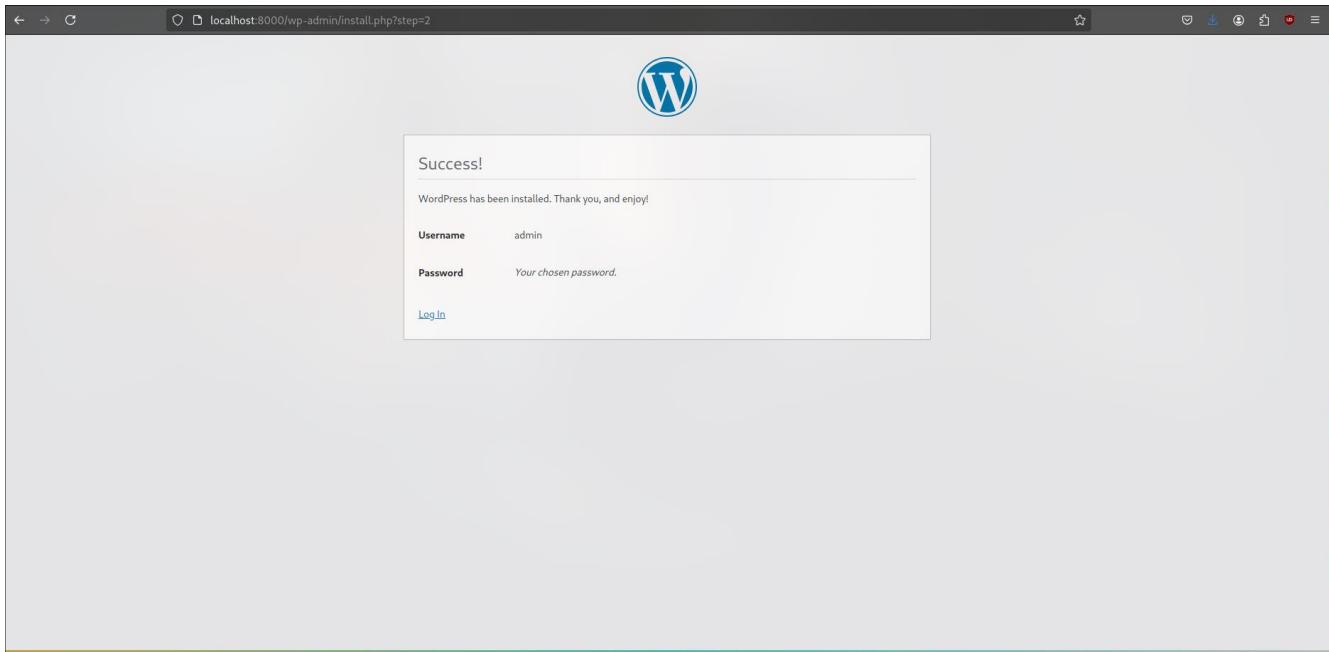
6. Go to localhost:8000



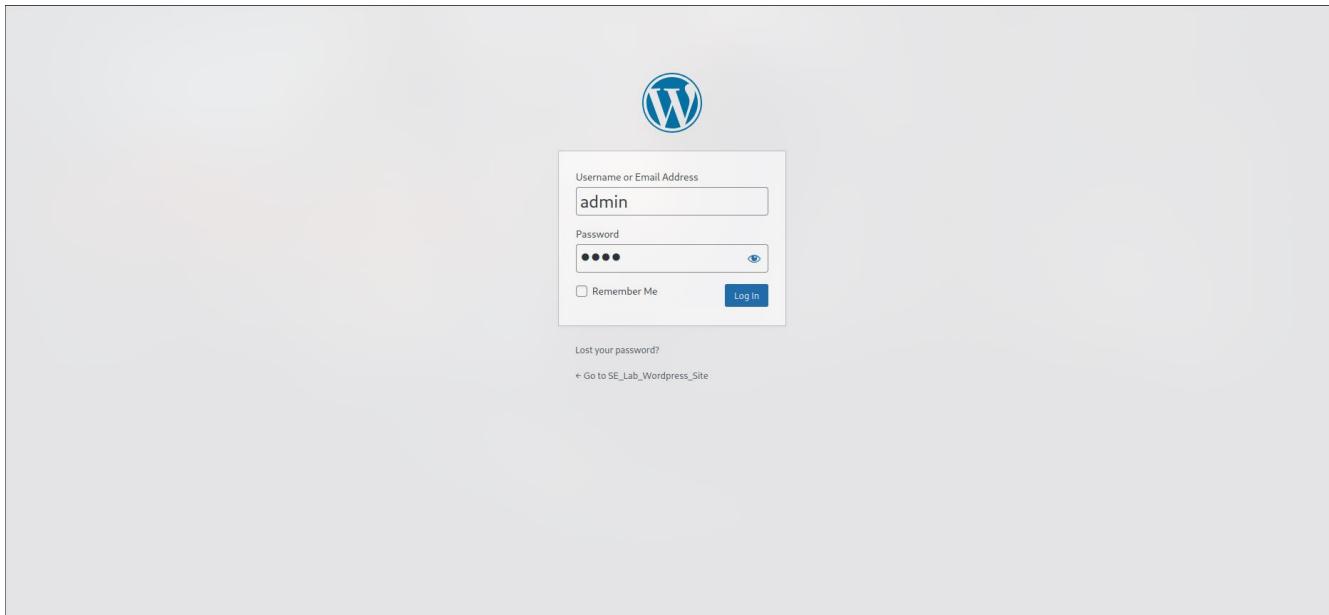
7. Complete the setup

A screenshot of a web browser showing the first step of the WordPress setup process at localhost:8000/wp-admin/install.php?step=1. The page title is 'Welcome'. It says: '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.' Below this is a section titled 'Information needed' with instructions: 'Please provide the following information. Do not worry, you can always change these settings later.' The form fields are: 'Site Title' (SE_Lab_Wordpress_Site), 'Username' (admin), 'Password' (1234, color-coded as 'Very weak'), 'Confirm Password' (checkbox checked), 'Your Email' (fobexef405@bawsny.com), and 'Search engine visibility' (checkbox unchecked). At the bottom is a blue 'Install WordPress' button.

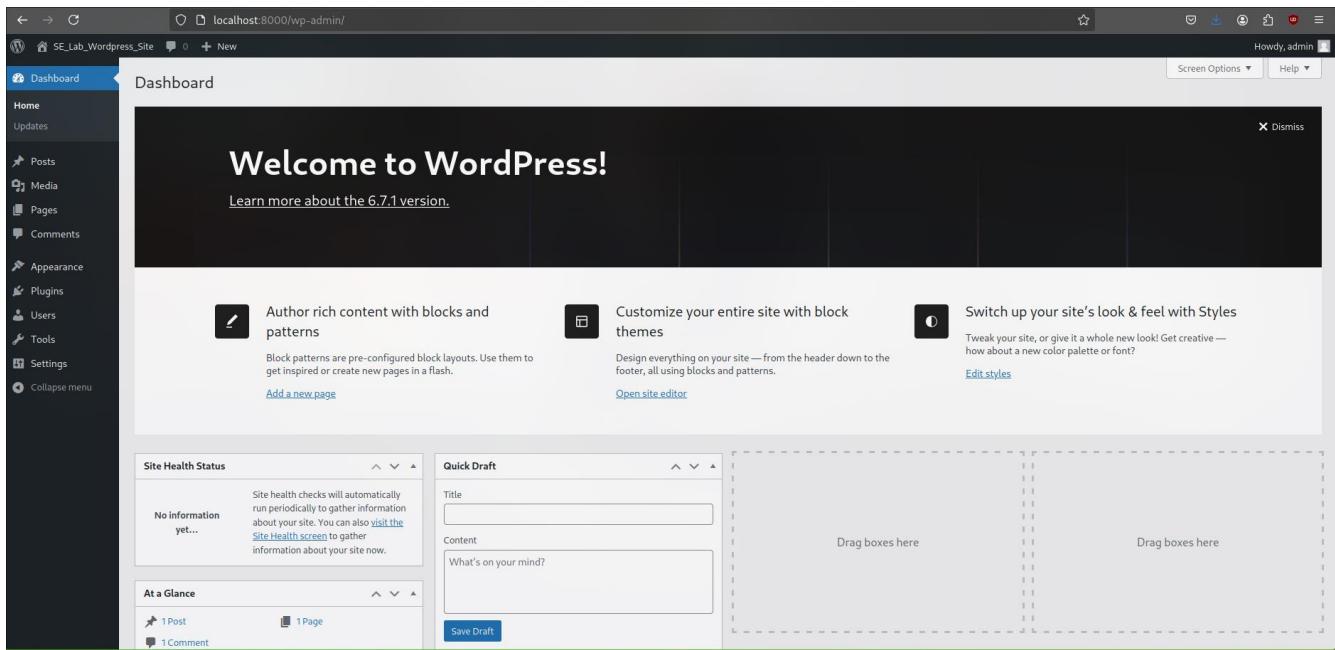
8. Setup complete



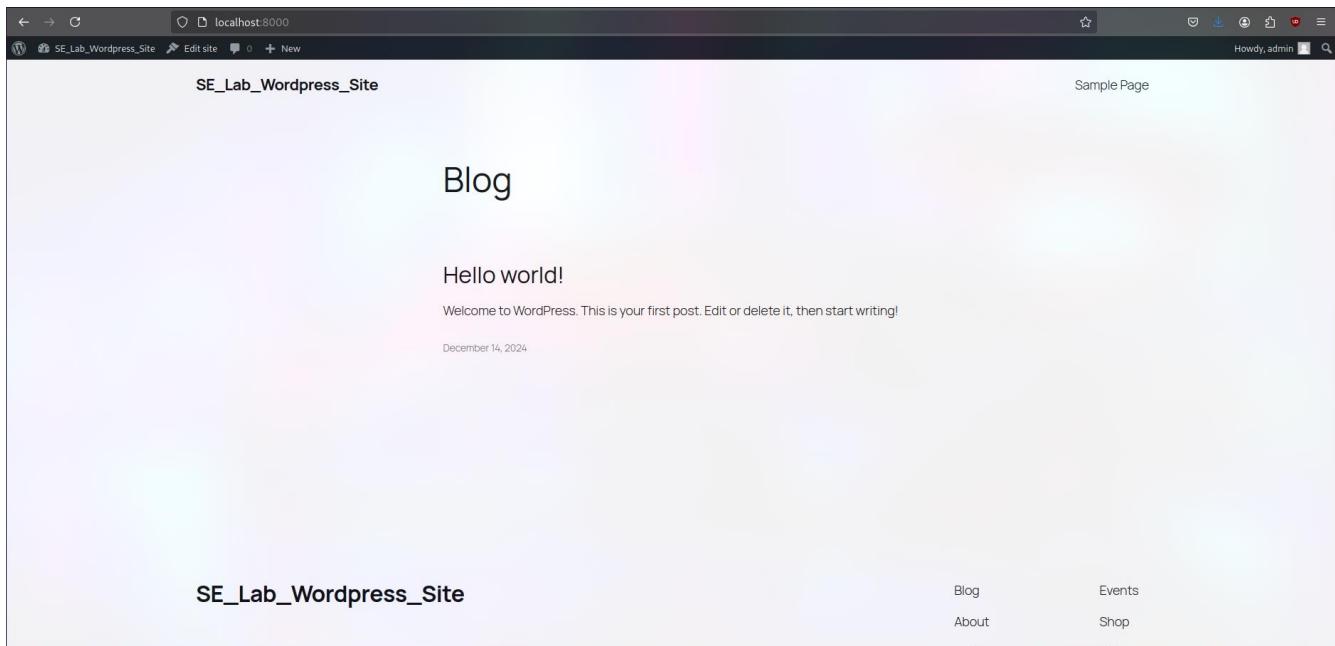
9. Login to wordpress



10. After login



11. Page in Browser



12. docker compose stop

The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the command 'docker compose stop' being run, followed by a list of stopped containers. The 'WORDPRESSPROJECT' workspace is selected in the Explorer sidebar.

```
WORDPRESSPROJECT
  docker-compose.yaml
    services:
      db:
      wordpress:
        depends_on:
          - db
        image: wordpress:latest
        container_name: wordpress_container
        ports:
```

```
[+] Stopping 2/2
  ✓ Container wordpress_container Stopped
  ✓ Container mysql_container Stopped
```

Terminal output:

```
[17/Dec/2024:16:42:59 +0000] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 530 "http://localhost:8000/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:42:59 +0000] "GET /wp-admin/index.php HTTP/1.1" 200 18961 "http://localhost:8000/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:43:04 +0000] "POST /wp-admin/admin-ajax.php?action=dashboard-widgets&widget=dashboard_primary&pagenow=dashbaord" 200 929 "http://localhost:8000/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:42:14 +0000] "OPTIONS * HTTP/1.0" 200 126 "-" "Apache/2.4.62 (Debian) PHP/8.2.26 (internal dummy connection)"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:42:14 +0000] "GET /wp-admin/edit.php?post_type=page HTTP/1.1" 200 12670 "http://localhost:8000/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:42:14 +0000] "GET /wp-admin/js/tags-suggest.min.js?ver=6.7.1 HTTP/1.1" 200 1429 "http://localhost:8000/wp-admin/edit.php?post_type=page" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:42:14 +0000] "GET /wp-admin/js/inline-edit-post.min.js?ver=6.7.1 HTTP/1.1" 200 3653 "http://localhost:8000/wp-admin/edit.php?post_type=page" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:42:14 +0000] "GET /wp-admin/load-styles.php?c=&dir=ltr&load%5Bchunk_1%5D=site-icon,lion_buttons,wp-auth-check&ver=6.7.1 HTTP/1.1" 200 101733 "http://localhost:8000/wp-admin/edit.php?post_type=page" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:42:14 +0000] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 530 "http://localhost:8000/wp-admin/edit.php?post_type=page" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:43:05 +0000] "GET /wp-admin/index.php HTTP/1.1" 200 18961 "http://localhost:8000/wp-admin/edit.php?post_type=page" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:43:04 +0000] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 530 "http://localhost:8000/wp-admin/edit.php?post_type=page" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:43:09 +0000] "GET /wp-admin/index.php HTTP/1.1" 200 18961 "http://localhost:8000/wp-admin/edit.php?post_type=page" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
Gracefully stopping... (press Ctrl+C again to force)
```

13. docker compose start

The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the command 'docker compose start' being run, followed by a list of started containers. The 'WORDPRESSPROJECT' workspace is selected in the Explorer sidebar.

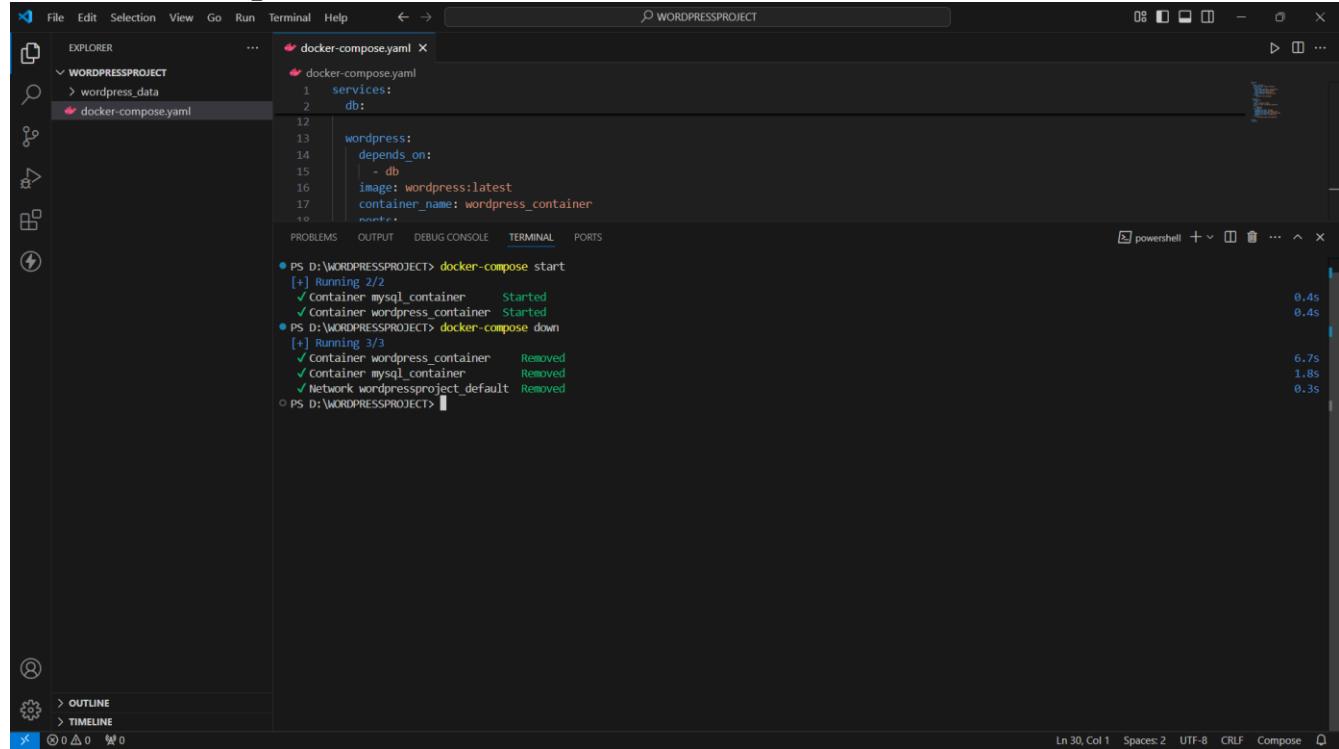
```
WORDPRESSPROJECT
  docker-compose.yaml
    services:
      db:
      wordpress:
        depends_on:
          - db
        image: wordpress:latest
        container_name: wordpress_container
        ports:
```

```
[+] Running 2/2
  ✓ Container mysql_container Started
  ✓ Container wordpress_container Started
```

Terminal output:

```
[17/Dec/2024:16:43:07 +0000] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 530 "http://localhost:8000/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:43:07 +0000] "GET /wp-admin/index.php HTTP/1.1" 200 18961 "http://localhost:8000/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:43:07 +0000] "POST /wp-admin/admin-ajax.php?action=dashboard-widgets&widget=dashboard_primary&pagenow=dashbaord" 200 929 "http://localhost:8000/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:43:07 +0000] "OPTIONS * HTTP/1.0" 200 126 "-" "Apache/2.4.62 (Debian) PHP/8.2.26 (internal dummy connection)"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:43:07 +0000] "GET /wp-admin/edit.php?post_type=page HTTP/1.1" 200 12670 "http://localhost:8000/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:43:07 +0000] "GET /wp-admin/js/tags-suggest.min.js?ver=6.7.1 HTTP/1.1" 200 1429 "http://localhost:8000/wp-admin/edit.php?post_type=page" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:43:07 +0000] "GET /wp-admin/js/inline-edit-post.min.js?ver=6.7.1 HTTP/1.1" 200 3653 "http://localhost:8000/wp-admin/edit.php?post_type=page" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:43:07 +0000] "GET /wp-admin/load-styles.php?c=&dir=ltr&load%5Bchunk_1%5D=site-icon,lion_buttons,wp-auth-check&ver=6.7.1 HTTP/1.1" 200 101733 "http://localhost:8000/wp-admin/edit.php?post_type=page" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:43:07 +0000] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 530 "http://localhost:8000/wp-admin/edit.php?post_type=page" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:43:07 +0000] "GET /wp-admin/index.php HTTP/1.1" 200 18961 "http://localhost:8000/wp-admin/edit.php?post_type=page" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
[wordpress container] | 172.19.0.1 - - [17/Dec/2024:16:43:07 +0000] "POST /wp-admin/admin-ajax.php?action=dashboard-widgets&widget=dashboard_primary&pagenow=dashbaord" 200 929 "http://localhost:8000/wp-admin/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
Gracefully starting... (press Ctrl+C again to force)
```

14. docker compose down



The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the command `docker-compose down` being run in a PowerShell session. The output shows the containers being stopped and removed:

```
PS D:\WORDPRESSPROJECT> docker-compose start
[+] Running 2/2
  ✓ Container mysql_container    Started
  ✓ Container wordpress_container Started
PS D:\WORDPRESSPROJECT> docker-compose down
[+] Running 3/3
  ✓ Container wordpress_container Removed
  ✓ Container mysql_container     Removed
  ✓ Network wordpressproject_default Removed
PS D:\WORDPRESSPROJECT>
```

The status bar at the bottom indicates the file is saved (green dot) and shows the current file is `docker-compose.yaml`.

7E – Deploying and Scaling using Minikube

1. minikube start

```
PS C:\Users\anand> minikube start
* minikube v1.34.0 on Microsoft Windows 11 Home Single Language 10.0.22631.4602 Build 22631.4602
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.45 ...
* docker "minikube" container is missing, will recreate.
* Creating docker container (CPUs=2, Memory=4000MB) ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
- Generating certificates and keys ...
- Booting up control plane ...
- Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
- Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\Users\anand> |
```

2. minikube kubectl -- create deployment mynginx --image=nginx

```
PS C:\Users\anand> minikube start
* minikube v1.34.0 on Microsoft Windows 11 Home Single Language 10.0.22631.4602 Build 22631.4602
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.45 ...
* docker "minikube" container is missing, will recreate.
* Creating docker container (CPUs=2, Memory=4000MB) ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
- Generating certificates and keys ...
- Booting up control plane ...
- Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
- Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\Users\anand> kubectl create deployment mynginx --image=nginx
deployment.apps/mynginx created
PS C:\Users\anand>
```

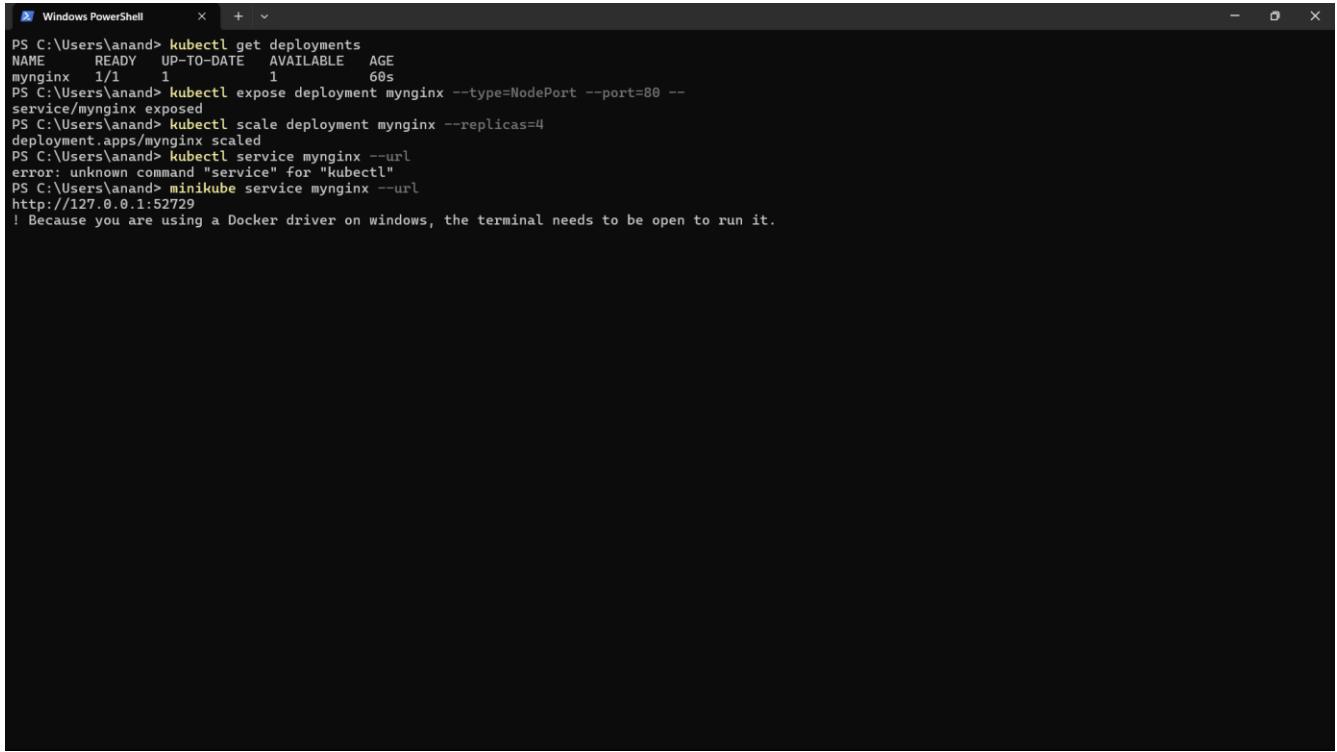
3. minikube kubectl -- get deployments

```
Windows PowerShell
PS C:\Users\anand> kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mynginx   1/1     1           1           60s
PS C:\Users\anand>
```

4. kubectl expose deployment mynginx --type=NodePort --port=80 --target-port=80

```
Windows PowerShell
PS C:\Users\anand> kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mynginx   1/1     1           1           60s
PS C:\Users\anand> kubectl expose deployment mynginx --type=NodePort --port=80 --
service/mynginx exposed
PS C:\Users\anand>
```

5. minikube kubectl -- scale deployment mynginx --replicas=4

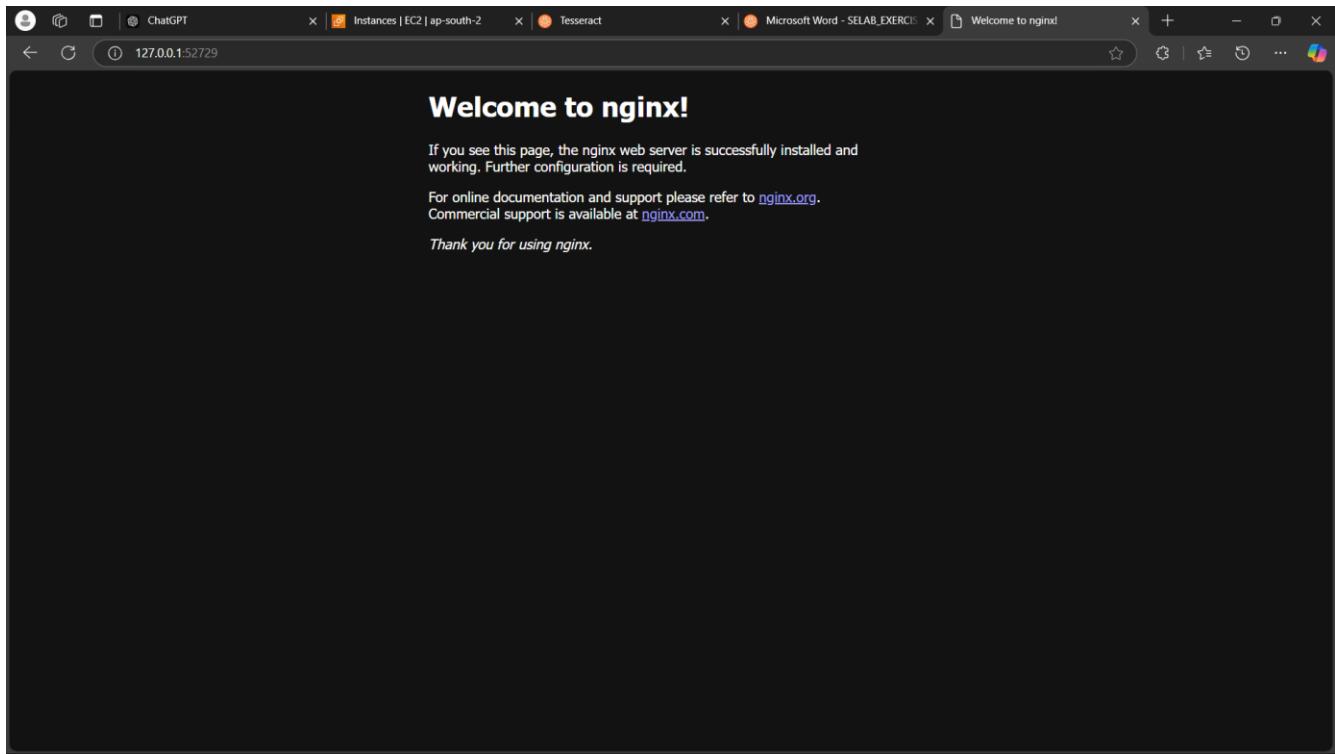


```
PS C:\Users\anand> kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mynginx   1/1     1           1           60s
PS C:\Users\anand> kubectl expose deployment mynginx --type=NodePort --port=80 --
service/mynginx exposed
PS C:\Users\anand> kubectl scale deployment mynginx --replicas=4
deployment.apps/mynginx scaled
PS C:\Users\anand> kubectl service mynginx --url
error: unknown command "service" for "kubectl"
PS C:\Users\anand> minikube service mynginx --url
http://127.0.0.1:52729
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```

6. minikube kubectl -- port-forward svc/mynginx 8081:80

```
> minikube kubectl -- port-forward svc/mynginx 8081:80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
[]
```

7. Go to localhost:8081 in browser



8. minikube kubectl -- delete deployment mynginx

```
Windows PowerShell      +  x
PS C:\Users\anand> kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mynginx  1/1     1           1           60s
PS C:\Users\anand> kubectl expose deployment mynginx --type=NodePort --port=80 --
service/mynginx exposed
PS C:\Users\anand> kubectl scale deployment mynginx --replicas=4
deployment.apps/mynginx scaled
PS C:\Users\anand> kubectl service mynginx --url
error: unknown command "service" for "kubectl"
PS C:\Users\anand> minikube service mynginx --url
http://127.0.0.1:52729
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
PS C:\Users\anand> kubectl delete deployment mynginx
deployment.apps "mynginx" deleted
PS C:\Users\anand> |
```

9. minikube kubectl -- delete service mynginx

```
Windows PowerShell      +  x
PS C:\Users\anand> kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mynginx  1/1     1           1           60s
PS C:\Users\anand> kubectl expose deployment mynginx --type=NodePort --port=80 --
service/mynginx exposed
PS C:\Users\anand> kubectl scale deployment mynginx --replicas=4
deployment.apps/mynginx scaled
PS C:\Users\anand> kubectl service mynginx --url
error: unknown command "service" for "kubectl"
PS C:\Users\anand> minikube service mynginx --url
http://127.0.0.1:52729
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
PS C:\Users\anand> kubectl delete deployment mynginx
deployment.apps "mynginx" deleted
PS C:\Users\anand> |
```

7F – Continuous Monitoring using Nagios

1. docker pull jasonrivers/nagios:latest

```

❯ docker pull jasonrivers/nagios:latest
latest: Pulling from jasonrivers/nagios
ff65dd9f9395b: Pull complete
785b9873bd4: Pull complete
0ef9446ba5cc: Pull complete
53af188bab4: Pull complete
d72f192e9533: Pull complete
706ed7d4ce0a: Pull complete
d3245570f968: Pull complete
e58e184b986a: Pull complete
eeb77e6ddee3e: Pull complete
0bd0f15795eab: Pull complete
71bbfb306f8cb: Pull complete
738fc7520889: Pull complete
fe8a6b2c74e3: Pull complete
e6f8fab512d1: Pull complete
15f36d0b0439: Pull complete
a2fc4187e3b4: Pull complete
3d7585144815: Pull complete
566cdc02555d: Pull complete
c700be87d617: Pull complete
4f4f1b700ef154: Pull complete
b69c76bd2b6b: Pull complete
d5aa2a3a6539: Pull complete
8fb30af17153: Pull complete
9ffe54c5c139: Pull complete
279b28aeaf10: Pull complete
a900dfcceeb38: Pull complete
9a90645e352c: Pull complete
8e911c59da28: Pull complete
c219d58c3f9: Pull complete
b0e280e9aa8c: Pull complete
8c389e58e67: Pull complete
Digest: sha256:2a7c2b2d0118ba92b47b69a3901e68dd7664617801b94e560bc4d6564d6ae54
Status: Downloaded newer image for jasonrivers/nagios:latest
docker.io/jasonrivers/nagios:latest

~/De/Aditya/Tranberg USB/USB Drive/Aditya new/Aditya/CSM A/III Year/I Semester
[✓] 1m 10s < 23:20:19

```

2. docker run --name nagiosdemo -p 8888:80 jasonrivers/nagios:latest

```

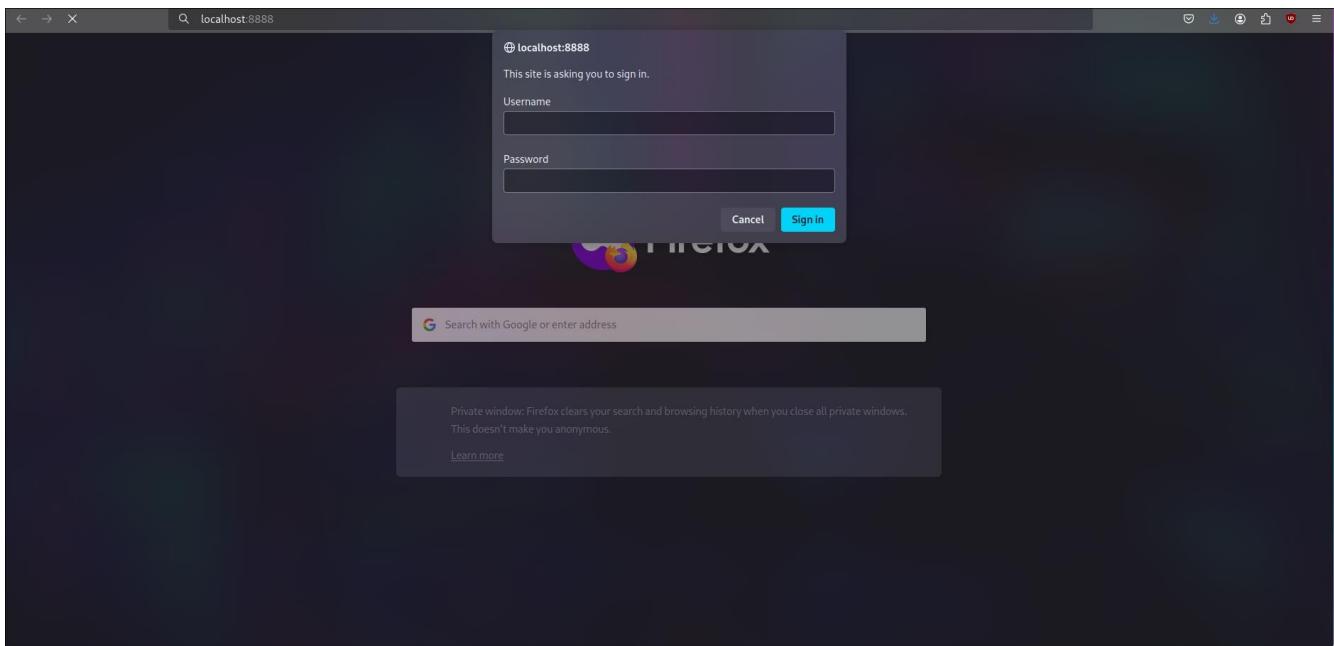
❯ docker run --name nagiosdemo -p 8888:80 jasonrivers/nagios:latest
Adding password for user nagiosadmin
chown: warning: '-' should be ':' 'nagios.nagios'
Started runsvdir; PID 13
checking permissions for nagios & nagiosgraph
rsyslogd: [origin software="rsyslogd" swVersion="8.2312.0" x-pid="21" x-info="https://www.rsyslog.com"] start

Nagios Core 4.5.7
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-10-24
License: GPL

Website: https://www.nagios.org
Nagios 4.5.7 starting... (PID=23)
Local time is Sun Dec 22 17:52:29 UTC 2024
nagios: Nagios 4.5.7 starting... (PID=23)
nagios: Local time is Sun Dec 22 17:52:29 UTC 2024
nagios: LOG VERSION: 2.0
nagios: qh: Socket '/opt/nagios/var/rw/nagios.qh' successfully initialized
nagios: qh: core query handler registered
nagios: qh: echo service query handler registered
wproc: Successfully registered manager as @wproc with query handler
nagios: qh: help for the query handler registered
nagios: wproc: Successfully registered manager as @wproc with query handler
wproc: Registry request: name=Core Worker 37;pid=37
wproc: Registry request: name=Core Worker 38;pid=38
wproc: Registry request: name=Core Worker 39;pid=39
wproc: Registry request: name=Core Worker 40;pid=40
wproc: Registry request: name=Core Worker 41;pid=41
wproc: Registry request: name=Core Worker 42;pid=42
nagios: wproc: Registry request: name=Core Worker 37;pid=37
nagios: wproc: Registry request: name=Core Worker 38;pid=38
nagios: wproc: Registry request: name=Core Worker 39;pid=39
nagios: wproc: Registry request: name=Core Worker 40;pid=40
nagios: wproc: Registry request: name=Core Worker 41;pid=41
nagios: wproc: Registry request: name=Core Worker 42;pid=42
wproc: Registry request: name=Core Worker 43;pid=43
nagios: wproc: Registry request: name=Core Worker 43;pid=43
nagios: wproc: Registry request: name=Core Worker 44;pid=44
wproc: Registry request: name=Core Worker 45;pid=45
nagios: wproc: Registry request: name=Core Worker 48;pid=48
wproc: Registry request: name=Core Worker 49;pid=49
nagios: wproc: Registry request: name=Core Worker 49;pid=49
wproc: Registry request: name=Core Worker 50;pid=50

```

3. Go to localhost:8888



4. Nagios dashboard

Nagios

Version 4.5.7 October 24, 2024

A new version of Nagios Core is available!
Visit nagios.org to download Nagios 4.5.9.

Check for updates

General

Home Documentation

Current Status

Tactical Overview

Map

Hosts

Services

Host Groups

Summary Grid

Service Groups

Summary Grid

Problems

Services (Unhandled)

Hosts (Unhandled)

Network Outages

Quick Search:

Reports

Availability Trends Alerts History Summary Histogram Notifications Event Log

System

Comments Downtime Process Info Performance Info Scheduling Queue Configuration

Meet Nagios Core Services Platform

The next generation of Open Source powered monitoring with advanced dashboards, monitoring wizards, and much more!

Learn More Newsletter Sign-Up

Copyright © 2010-2024 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Nagios Core is licensed under the GPL (General Public License) and is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN.

Page Tour

5. Monitoring the host

The screenshot shows the Nagios web interface at localhost:8888. The left sidebar contains navigation links for General, Hosts, Services, Host Groups, Grid, Service Groups, Grid, Problems, Reports, Availability, Trends, Alerts, Event Log, and System. The main content area displays 'Host Information' for the host 'localhost'. Key details include:

- Last Updated: Sun Dec 22 17:56:55 UTC 2024
- Updated every 90 seconds
- Nagios version: 5.0.0 (stable) - www.nagios.org
- Logged in as nagiosadmin
- Host Status: UP [for 0d 0h 4m 24s]
- Status Information: PING OK - Packet loss = 0%, RTA = 0.04 ms
- Performance Data: rta=0.035000ms;3000.000000;5000.000000;0.000000 pl=0%;80;100;0
- Current Attempt: 1/10 (HARD state)
- Last Check Time: 12-22-2024 17:54:37
- Check Type: ACTIVE
- Check Latency / Duration: 0.000 / 4.069 seconds
- Next Scheduled Active Check: 12-22-2024 17:59:37
- Last State Change: 12-22-2024 17:52:31
- Last Notification: N/A (Notification 0)
- Is This Host Flapping? NO (0.00% state change)
- In Scheduled Downtime? NO
- Last Update: 12-22-2024 17:56:49 (0d 0h 0m 6s ago)
- Active Checks: ENABLED
- Passive Checks: ENABLED
- Obsessing: ENABLED
- Notifications: ENABLED
- Event Handler: ENABLED
- Flap Detection: ENABLED

The right side of the interface shows 'Host Commands' with various options like Locate host on map, Disable active checks, and Stop accepting passive checks for this host.

6. Stopping the container

```
> docker stop nagiosdemo
nagiosdemo
```

7. Deleting the container

```
~|De/Aditya/Tranberg USB/USB Drive/Aditya new/Aditya/CSM A/III Year/I Semester
[ ] docker rm nagiosdemo
```

Result: Local Deployment of Project using Docker, Kubernetes and Monitoring using Nagios tool were demonstrated

Experiment-8: Deployment of a project into AWS Cloud using EC2 Instance

Aim: To create an AWS Learning Account, deploy a sample HTML file on AWS EC2, and deploy the Maven web application created in a previous experiment onto AWS EC2.

Introduction:

1. AWS (Amazon Web Services): A leading cloud computing platform offering scalable and reliable infrastructure for deploying applications and services.
 - o AWS Learning Account: Provides access to AWS services for educational purposes, enabling hands-on experience with cloud computing.
2. AWS EC2 (Elastic Compute Cloud): A service that provides virtual servers (instances) for running applications in the cloud.
 - o Instances can host web servers, databases, or any software, offering scalability and cost-efficiency.

AWS – Account Creation

Outlook

Home View Help

New email Delete Archive Move to Report Reply all Read / Unread Flag / Unflag Discover groups Undo ...

Favourites

- Inbox 2350
- Sent Items
- Drafts 8

anandsaisubrahma...

- Inbox 2350
- Junk Email 4
- Drafts 8
- Sent Items
- Deleted Items
- Archive 1
- Outbox

anand26042005@g...

- Inbox 2667
- Junk Email 12
- Drafts 4
- Sent Items
- Deleted Items
- Archive 1

Inbox

Change in TER - Sch... Mon 16/12 NOTICE FOR CHANGE IN THE BA...

Coding Ninjas E&CT, IIT Guwahati i... Mon 16/12 Click here to know more. ...

AWS Academy Course Invitation Mon 16/12 You've been invited to participat...

AWS Academy Course Invitation Mon 16/12 You've been invited to participat...

AWS Academy Course Invitation Mon 16/12 You've been invited to participat...

AWS Academy Course Invitation Mon 16/12 You've been invited to participat...

Team Unstop 1000 students apply... Mon 16/12 View details ...

LinkedIn Job Alerts "data scientist": Opt... Mon 16/12 ₹2M-₹2.5M / year salary ...

Course Invitation

AWS Academy<notifications@instructure.com> To: You Mon 16/12/2024 16:33

You've been invited to participate in a class at AWS Academy . The class is called AWS Academy Learner Lab [90659]. Course role: Student

Name: anand26042005@gmail.com
Email: anand26042005@gmail.com
Username: none

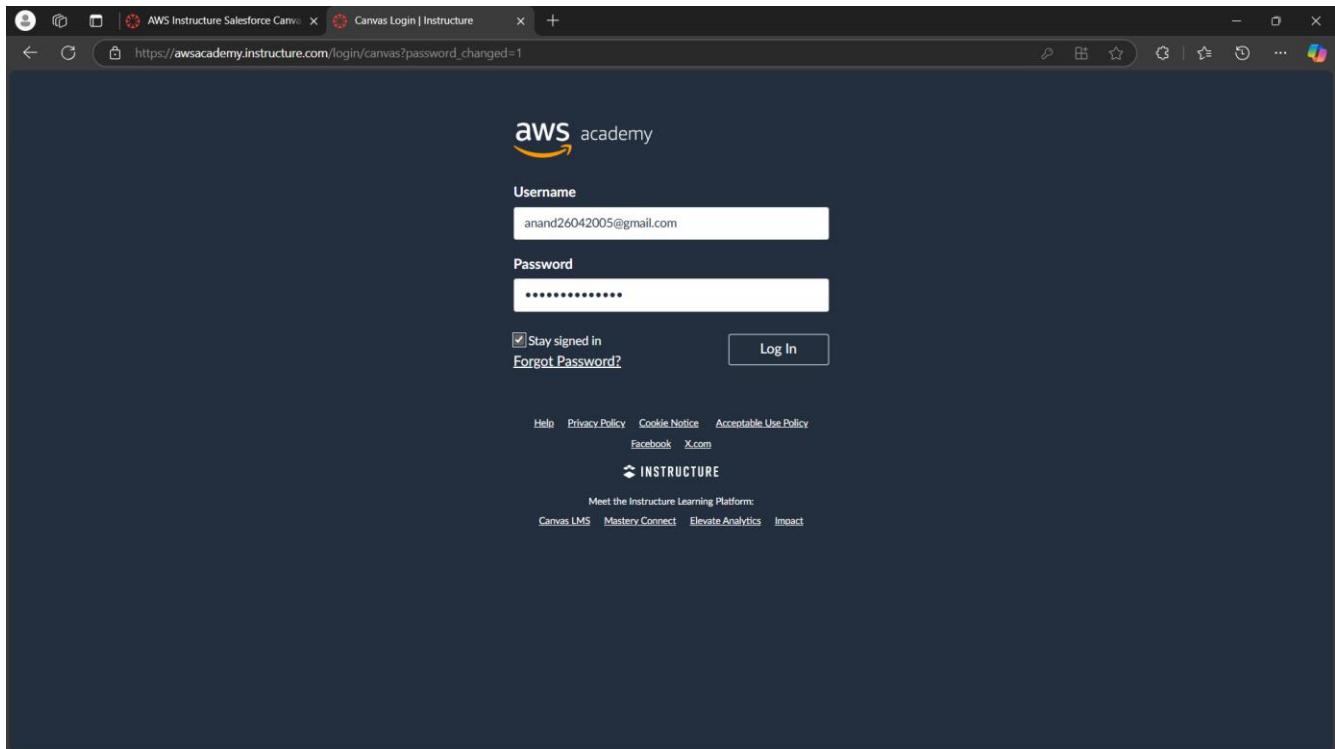
You'll need to register with Canvas before you can participate in the class.

Get Started

CANVAS

Reply Forward

This screenshot shows the Microsoft Outlook inbox interface. On the left, there's a sidebar with 'Favourites' and a list of accounts. The main pane shows the 'Inbox' with several emails listed. One email from 'AWS Academy' is selected, displaying its contents. The email subject is 'Course Invitation'. It includes a message from 'AWS Academy<notifications@instructure.com>' to 'You' dated 'Mon 16/12 16:33'. The message text says: 'You've been invited to participate in a class at AWS Academy . The class is called AWS Academy Learner Lab [90659]. Course role: Student'. Below this, it lists 'Name: anand26042005@gmail.com', 'Email: anand26042005@gmail.com', and 'Username: none'. A note at the bottom states: 'You'll need to register with Canvas before you can participate in the class.' At the bottom of the email view, there are 'Get Started' and 'CANVAS' buttons, along with 'Reply' and 'Forward' links. The overall interface is clean and modern, typical of Microsoft's software.



The screenshot shows the AWS Academy dashboard. At the top, there is a notification bar with the message: "Notifications. Tell us how and when you would like to be notified of events in Canvas." and a link to "Notification Preferences". Below the notification bar, the title "Dashboard" is displayed. On the left side, there is a sidebar with icons for Account, Dashboard (selected), Courses, Calendar, Inbox, History, and Help. The main content area features a course card for "AWS Academy Learner Lab [90659]" with the identifier "ALLv2EN-US-LTI13-90659". To the right of the course card, there are sections for "To Do" (Nothing for now) and "Recent Feedback" (Nothing for now). A "View Grades" button is also present. At the bottom of the page, there are links for Privacy Policy, Cookie Notice, Acceptable Use Policy, Facebook, and X.com.

The screenshot shows the "Course Modules" page for the AWS Academy Learner Lab. The left sidebar includes icons for Account, Dashboard (selected), Courses (selected), Calendar, Inbox, History, and Help. The main content area displays the "AWS Academy Learner Lab" module structure. It includes sections for "Course Welcome and Overview" (with "Pre-Course Survey" and "AWS Academy Learner Lab Student Guide"), "AWS Academy Learner Lab Compliance and Security" (with "Learn how to effectively use the AWS Academy Learner Lab" and "Module Knowledge Check" worth 100 pts), "AWS Academy Learner Lab" (with "Launch AWS Academy Learner Lab"), and "AWS Academy Learner Lab Resources" (with "Demo - How to Access Learner Lab"). A "Complete All Items" button is located near the top of the "AWS Academy Learner Lab Compliance and Security" section.

The screenshot shows the AWS Academy Learner Lab interface. On the left is a sidebar with various navigation options: Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main area has a top navigation bar with tabs for AWS, Start Lab, End Lab, AWS Details, Readme, and Reset. Below the navigation is a terminal window showing the command `eee_W_3943750@runweb155631:~$`. To the right of the terminal is a dropdown menu set to EN-US. The main content area is titled "Learner Lab" and contains a list of instructions:

- 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
- SSH Access from Windows
- SSH Access from a Mac

At the bottom of the content area, it says "Instructions last updated: 2024-12-18".

This screenshot is identical to the one above, showing the AWS Academy Learner Lab interface. The main difference is the terminal output, which now displays the dimensions `117x34`.

The screenshot shows the AWS Console Home page. At the top, there are tabs for "AWS Infrastructure SalesForce Canvas", "Launch AWS Academy Learner Lab", "Console Home | Console Home", "Tessellator 4.0 - Dashboard", and "Pre-FS-2026-23-12-2024". The URL is https://us-east-1.console.aws.amazon.com/console/home?region=us-east-1#. The header includes the AWS logo, a search bar, and navigation icons. The main content area has several cards:

- Recently visited**: Shows a placeholder icon and text: "No recently visited services". Buttons for "View all services" and "Go to myApplications".
- Applications**: Shows a placeholder icon and text: "Region: US East (N. Virginia)". Buttons for "Create application" and "Go to myApplications".
- Welcome to AWS**: Includes a "Getting started with AWS" section with a rocket icon and text: "Learn the fundamentals and find valuable information to".
- AWS Health**: Shows a placeholder icon.
- Cost and usage**: Shows a placeholder icon.

At the bottom, there are links for "CloudShell", "Feedback", "© 2024, Amazon Web Services, Inc. or its affiliates.", "Privacy", "Terms", and "Cookie preferences".

The screenshot shows the AWS Academy Learner Lab interface. The left sidebar includes a navigation tree: "ALLv2EN-US-LTI13-90... > Modules > AWS Academy Learner... > Launch AWS Academy Learner Lab". Other menu items include "Home", "Modules", "Discussions", "Grades", "Lucid (Whiteboard)", "Calendar", "Inbox", "History", and "Help".

The main area has two panes:

- Terminal Session:** A terminal window showing a user's session:

```
eee_w_3940856@runweb155541:~$ ls
eee_w_3940856@runweb155541:~$ users
eee_w_3940856@runweb155541:~$ users -a
users: invalid option -- 'a'
Try `users --help' for more information.
eee_w_3940856@runweb155541:~$ who
eee_w_3940856@runweb155541:~$ pwd
/rnt/vocvor02/work/eee_w_3940856/asn3382139_1/asn3382140_1/work
eee_w_3940856@runweb155541:~$ ]
```
- Learner Lab:** A panel titled "Learner Lab" containing "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", "SSH Access from Windows", and "SSH Access from a Mac". It also includes a note: "Instructions last updated: 2024-12-18".

At the bottom, there are "Previous" and "Next" navigation buttons.

AWS – Deploying index.html

1. Login to AWS dashboard

The screenshot shows the AWS Console Home dashboard. It includes sections for Recently visited (EC2, AWS Organizations), Applications (0), AWS Health (0 open issues, 0 scheduled changes), and Cost and usage (\$0.00 current month costs, \$0.00 forecasted month end costs). There are also links to Welcome to AWS, View all services, and Go to myApplications.

2. Click on launch instance

The screenshot shows the AWS EC2 home page. The left sidebar lists navigation items such as Dashboard, Instances, Images, Elastic Block Store, and Network & Security. The main content area includes sections for Resources (listing Instances, Auto Scaling Groups, Capacity Reservations, Dedicated Hosts, Elastic IPs, Key pairs, Load balancers, Security groups, Snapshots, and Volumes), Launch instance (with a prominent 'Launch instance' button), Service health (showing Region: US East (N. Virginia) and Status: This service is operating normally), and Explore AWS (sections on Best Price-Performance with AWS Graviton2, 10 Things You Can Do Today to Reduce AWS Costs, and Amazon GuardDuty Malware Protection).

3. Select your choices

4. Create an SSH key pair

Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.

SE_Lab_AWS_Learners

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

RSA
RSA encrypted private and public key pair

ED25519
ED25519 encrypted private and public key pair

Private key file format

.pem
For use with OpenSSH

.ppk
For use with PuTTY

⚠️ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Create key pair

5. Allow traffic from everywhere

The screenshot shows the AWS EC2 'Launch an instance' wizard. The 'Instance type' section is set to 't2.micro' (Free tier eligible). The 'Key pair (login)' section has 'SELab' selected. The 'Network settings' section includes 'Auto-assign public IP' (Enable) and 'Additional charges apply when outside of free tier allowance'. The 'Summary' section shows 1 instance, software image (AMI) as Canonical, Ubuntu, 24.04, amd64, storage as 1 volume(s) - 8 GiB, and a note about the Free tier. The 'Launch instance' button is at the bottom right.

Instance type

t2.micro Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

All generations

Compare instance types

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

SELab

Create new key pair

▼ Network settings [Info](#)

Network [Info](#)
vpc-073bb02e2913ee977

Subnet [Info](#)
No preference (Default subnet in any availability zone)

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

Additional charges apply when outside of free tier allowance

▼ Summary

Number of instances [Info](#)
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64...[read more](#)
ami-0e2c8caa4b6378d8c

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of

Cancel

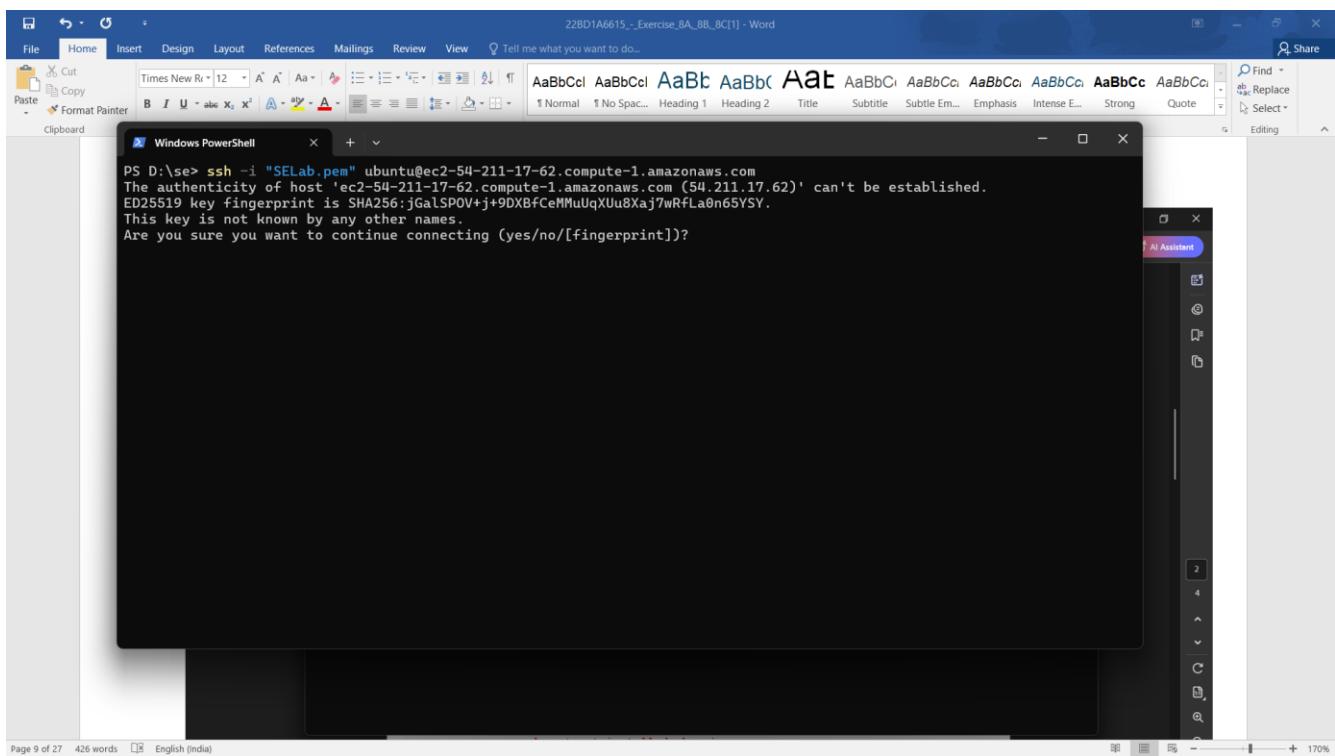
Launch instance

Preview code

6. Use 8GB default storage

The screenshot shows the AWS EC2 Launch Wizard interface. On the left, the 'Configure security group' section is visible, featuring options to 'Create security group' or 'Select existing security group'. It lists three selected inbound rules: 'Allow SSH traffic from anywhere', 'Allow HTTPS traffic from the internet', and 'Allow HTTP traffic from the internet'. A note at the bottom of this section cautions against allowing all IP addresses and recommends setting security group rules. Below this is the 'Configure storage' section, which shows a single 8 GiB gp3 root volume (not encrypted). A note indicates that free-tier eligible customers can get up to 30 GB of EBS storage. At the bottom of the wizard, there's a summary box showing one instance being launched with the Canonical, Ubuntu 24.04 AMI, and a t2.micro instance type. The summary also includes a note about the free tier and provides 'Launch instance' and 'Preview code' buttons.

7. SSH into the instance



8. Install tools like Docker, git, vim, nano

```
ubuntu@ip-172-31-25-189:~ + ~ PS D:\se> icacls "D:\se\SELab.pem" /remove "NT AUTHORITY\Authenticated Users"
processed file: D:\se\SELab.pem
Successfully processed 1 files; Failed processing 0 files
PS D:\se> icacls "D:\se\SELab.pem" /remove "BUILTIN\Users"
processed file: D:\se\SELab.pem
Successfully processed 1 files; Failed processing 0 files
PS D:\se> ssh -i "D:/se/SELab.pem" ubuntu@ec2-54-211-17-62.compute-1.amazonaws.com
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

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

System information as of Mon Dec 23 16:12:24 UTC 2024

System load:  0.08      Processes:          104
Usage of /:   24.7% of 6.71GB  Users logged in:     0
Memory usage: 20%           IPv4 address for enX0: 172.31.25.189
Swap usage:   0%

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-172-31-25-189:~$ |
```

9. In your local system, create a folder and inside it create an index.html

The screenshot shows the Visual Studio Code interface. In the top left, there's a tree view with a file named "index.html". The main editor area contains the following HTML code:

```
1 <html>
2   <head>
3     <title>My WebPage</title>
4   </head>
5   <body>
6     <h1>Hello from AWS!</h1>
7   </body>
8 </html>
```

Below the editor is a tab bar with PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected, showing a PowerShell session with the following commands:

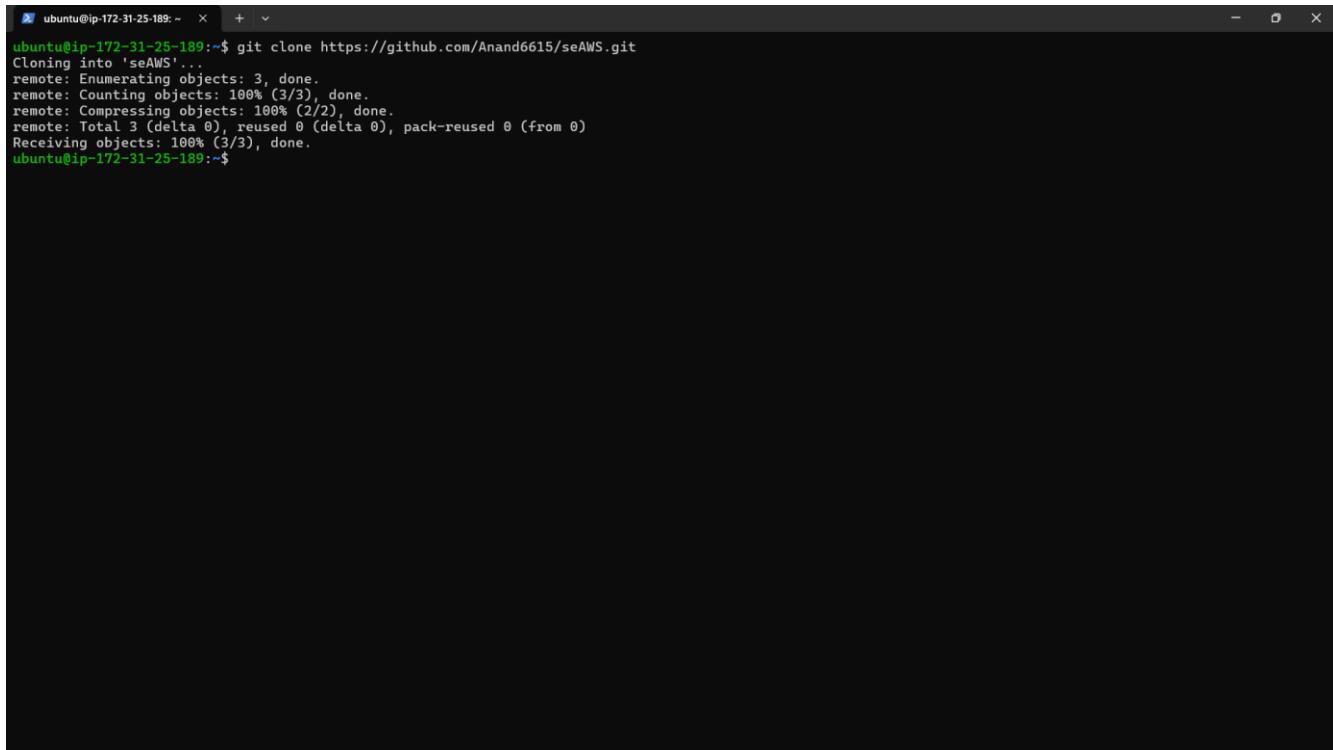
```
PS D:\se> git init
>>
Reinitialized existing Git repository in D:/se/.git/
PS D:\se> git remote add origin https://github.com/Anand6615/seAWS.git
>>
error: remote origin already exists.
PS D:\se> git remote -v
>>
origin https://github.com/Anand6615/seAWS.git (fetch)
origin https://github.com/Anand6615/seAWS.git (push)
PS D:\se>
```

The status bar at the bottom indicates "Ln 10, Col 8" and "Spaces: 4 CRLF HTML".

10. Initialise a git repo and commit your changes

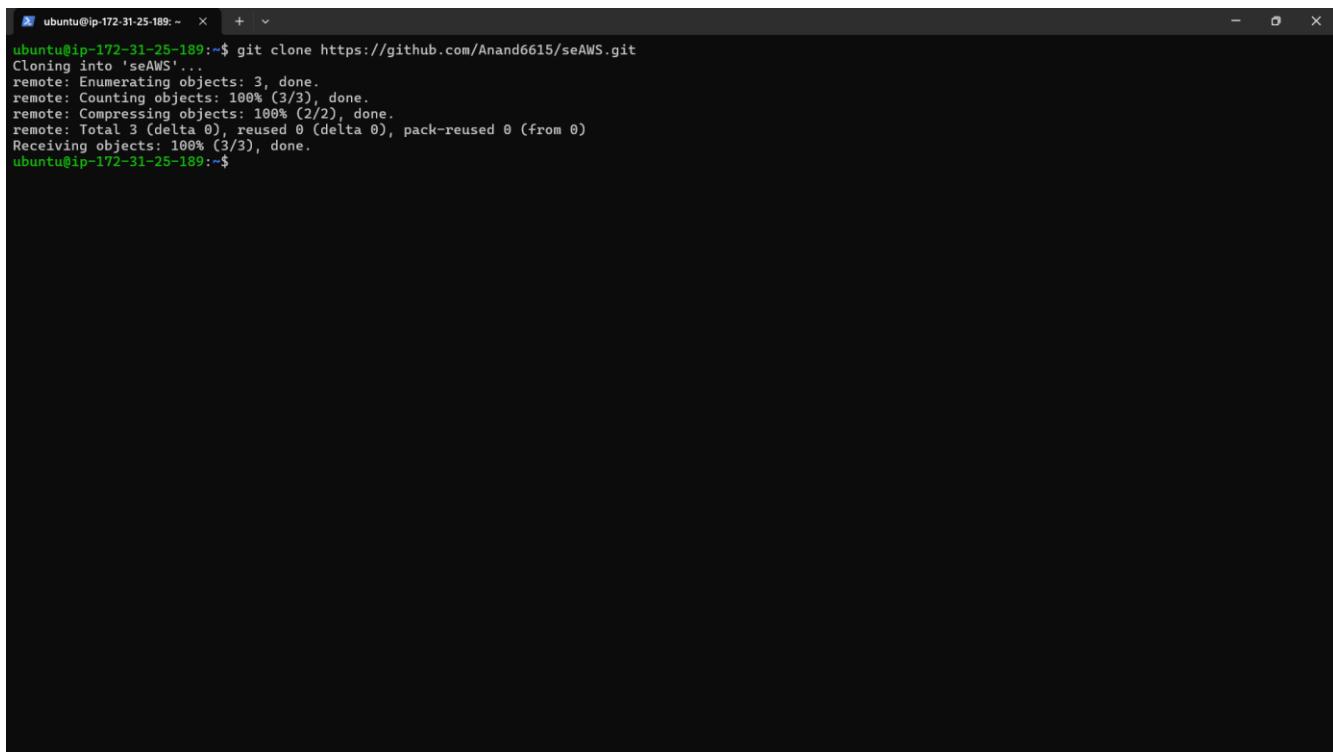
This screenshot is identical to the one above, showing the same file content, terminal history, and status bar information.

11. Push into github



```
ubuntu@ip-172-31-25-189:~$ git clone https://github.com/Anand6615/seAWS.git
Cloning into 'seAWS'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
ubuntu@ip-172-31-25-189:~$
```

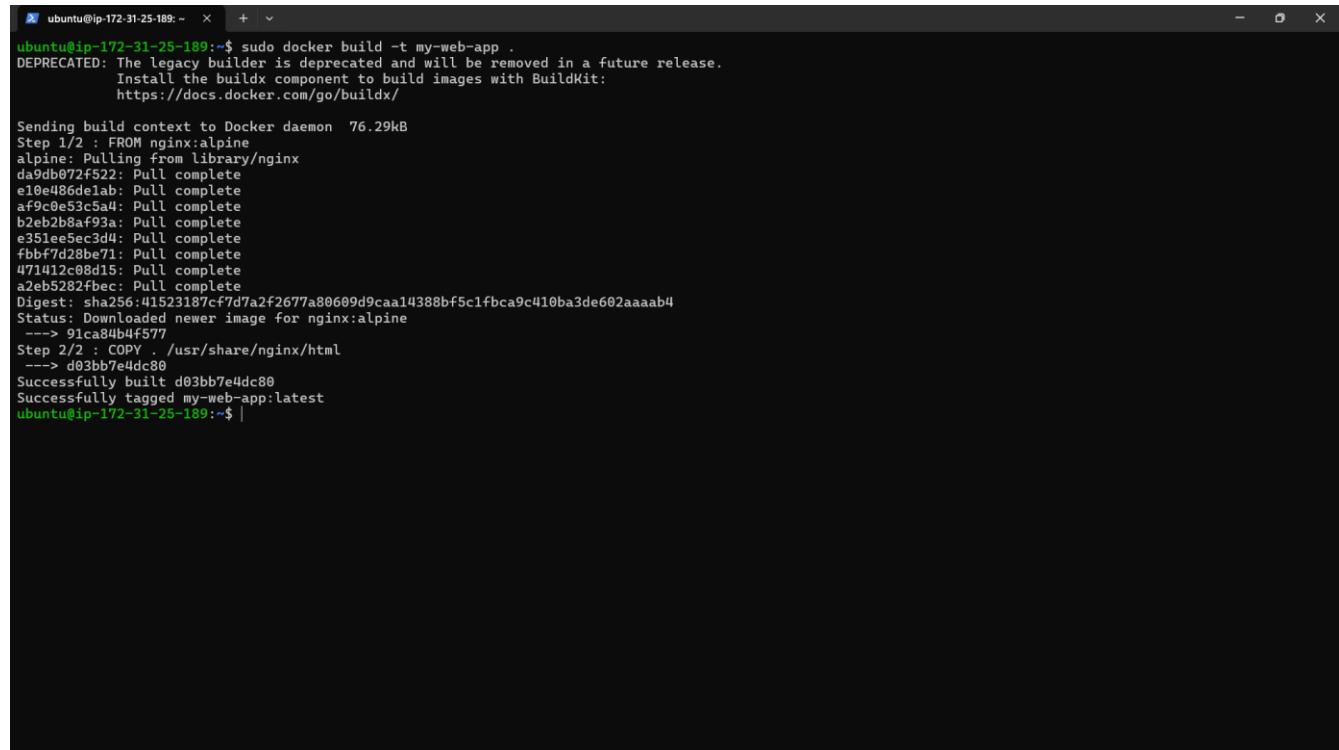
12. In the EC2 instance, clone your repo



```
ubuntu@ip-172-31-25-189:~$ git clone https://github.com/Anand6615/seAWS.git
Cloning into 'seAWS'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
ubuntu@ip-172-31-25-189:~$
```

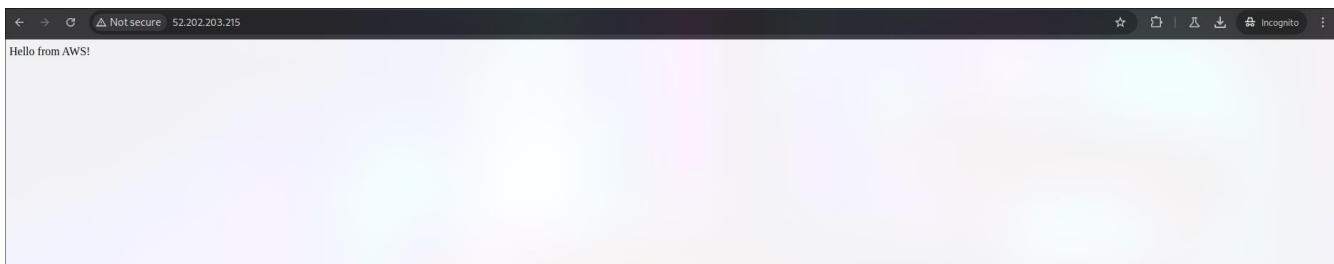
13. Create a dockerfile and add these lines

14. Create docker image and run the container

A screenshot of a terminal window titled "ubuntu@ip-172-31-25-189:~". The window shows the command "sudo docker build -t my-web-app .". The output indicates that the legacy builder is deprecated and will be removed in a future release. It shows the pulling of the "nginx:alpine" image, which is successfully built and tagged as "my-web-app:latest".

```
ubuntu@ip-172-31-25-189:~$ sudo docker build -t my-web-app .
DEPRECATION: 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 76.29kB
Step 1/2 : FROM nginx:alpine
alpine: Pulling from library/nginx
da9db072f522: Pull complete
e10e486de1ab: Pull complete
af9c0e53c5a4: Pull complete
b2eb2b8af93a: Pull complete
e351ee5ec3d4: Pull complete
fbff7fd28be71: Pull complete
471412c08d15: Pull complete
a2eb5282fbec: Pull complete
Digest: sha256:41523187cf7d7a2f2677a80609d9caa14388bf5c1fbca9c410ba3de602aaaab4
Status: Downloaded newer image for nginx:alpine
--> 91ca81b4f577
Step 2/2 : COPY ./usr/share/nginx/html
--> d03bb7e4dc80
Successfully built d03bb7e4dc80
Successfully tagged my-web-app:latest
ubuntu@ip-172-31-25-189:~$ |
```

15. In your EC2 dashboard, copy your public IP address and open in browser



16. Finally stop your container

```
ubuntu@ip-172-31-82-208:~/SE_AWS$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
7a0e7802c8a8 my-web-app "/docker-entrypoint..." 2 minutes ago Up 2 minutes 0.0.0.0:80->80/tcp, :::80->80/tcp graculous_hopper
ubuntu@ip-172-31-82-208:~/SE_AWS$ sudo docker stop 7a0e7802c8a8
7a0e7802c8a8
ubuntu@ip-172-31-82-208:~/SE_AWS$
```

AWS – Deploying Maven Web Project

1. Login to AWS dashboard

A screenshot of the AWS Console Home dashboard. The top navigation bar shows 'Search [Alt+S]', a user icon, and 'N. Virginia'. The main area is titled 'Console Home' with a 'Reset to default layout' button and a '+ Add widgets' button. It features several cards: 'Recently visited' (EC2, AWS Organizations), 'Applications (0)' (Create application), 'Welcome to AWS' (Getting started with AWS, Training and certification), 'AWS Health' (Open issues: 0, Past 7 days), and 'Cost and usage' (Current month costs: \$0.00, Forecasted month end costs: \$0.00, Savings opportunities). The bottom of the screen shows a footer with links for CloudShell, Feedback, and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates.' and links to Privacy, Terms, and Cookie preferences.

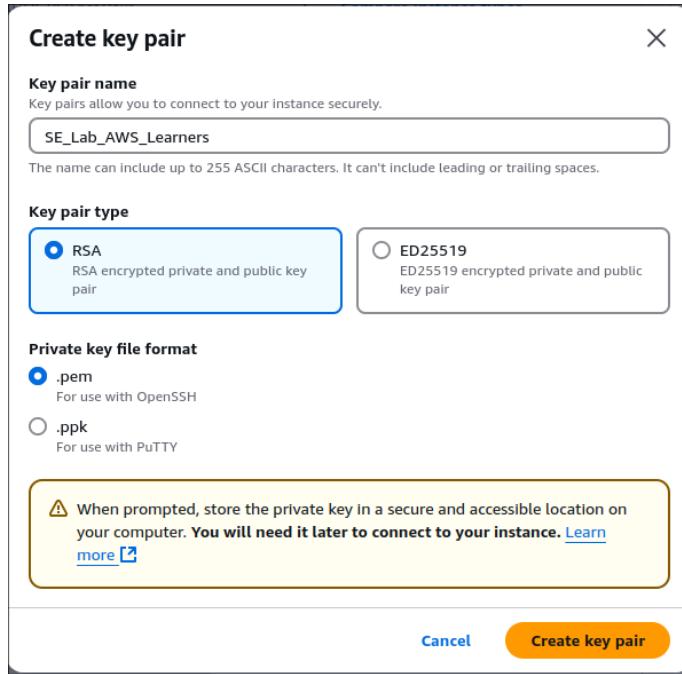
2. Click on launch instance

The screenshot shows the AWS EC2 'Launch an instance' configuration page. The 'Name and tags' section has 'MavenWebProjectServer' entered. The 'Software Image (AMI)' section shows 'Amazon Linux 2023 AMI 2023.6.2...' selected. The 'Virtual server type (instance type)' is set to 't2.micro'. The 'Storage (volumes)' section indicates 1 volume(s) - 8 GiB. A summary box on the right shows 1 instance selected. A tooltip for the 'Free tier' is displayed, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of'. The 'Launch instance' button is visible at the bottom right.

3. Select your choices

This screenshot is identical to the previous one, showing the same configuration for launching an instance. The tooltip for the 'Free tier' now states: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of'. The 'Launch instance' button is also present.

4. Create an SSH key pair



5. Allow traffic from everywhere

The screenshot shows the AWS EC2 Launch Wizard interface. The left pane displays the configuration steps:

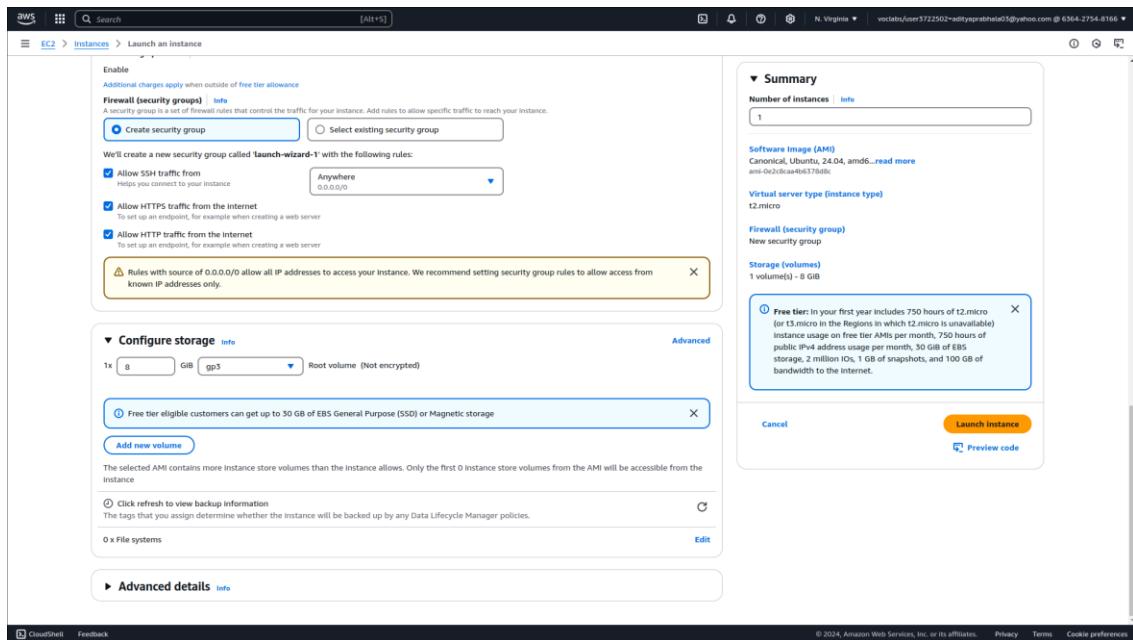
- Firewall (security groups)**: A new security group named 'launch-wizard-2' is being created. It includes three inbound rules:
 - Allow SSH traffic from Anywhere (0.0.0.0/0)
 - Allow HTTPS traffic from the internet
 - Allow HTTP traffic from the internetA note states: "Rules of source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only."
- Configure storage**: A root volume of 8 GiB (gp3) is selected. A note indicates: "Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage". An "Add new volume" button is available.

The right pane shows the summary and launch options:

- Summary**: Number of instances: 1
- Software Image (AMI)**: Canonical, Ubuntu, 24.04, amd64... (ami-0e2c8caa4b6378d8c)
- Virtual server type (instance type)**: t2.micro
- Firewall (security group)**: New security group
- Storage (volumes)**: 1 volume(s) - 8 GiB
- Free tier**: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of data transfer.
- Launch instance** button

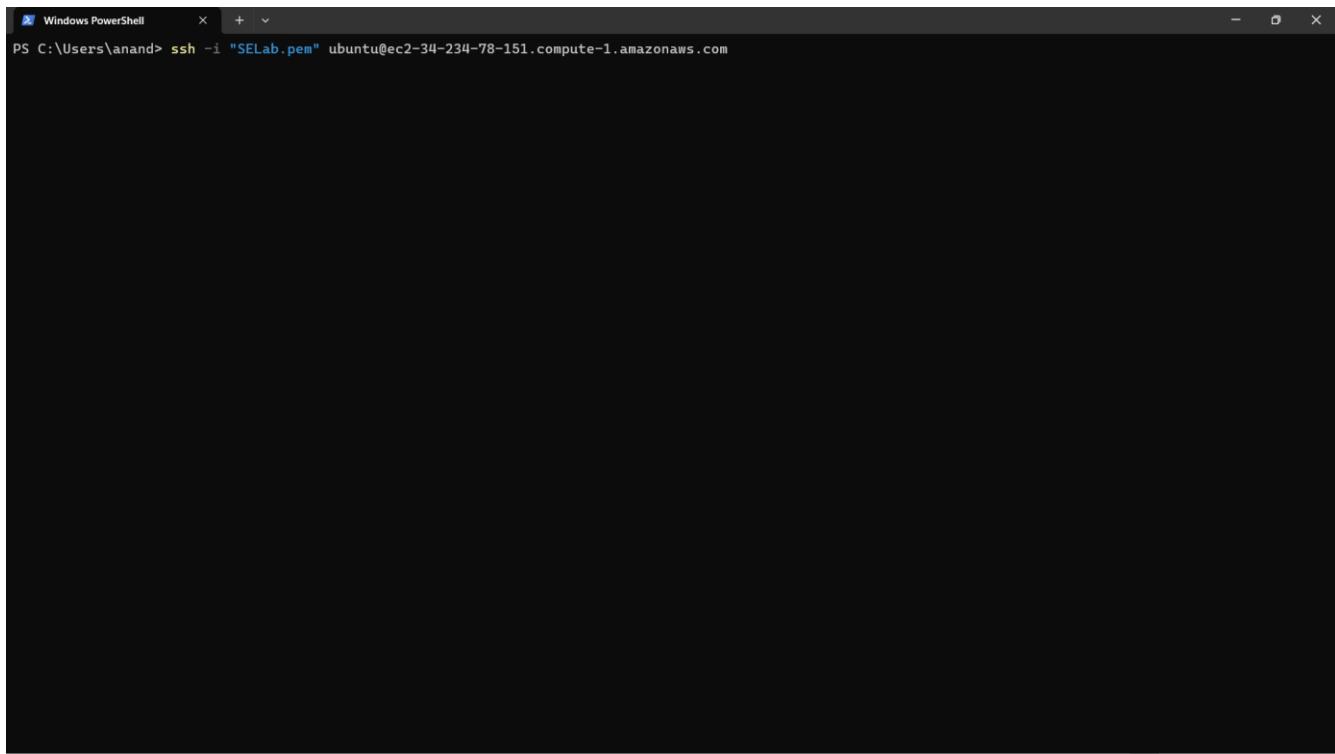
At the bottom, there are links for CloudShell, Feedback, and Copyright information: © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

6. Use 8GB default storage



7. SSH into the instance

```
Windows PowerShell
PS C:\Users\anand> ssh -i "SELab.pem" ubuntu@ec2-34-234-78-151.compute-1.amazonaws.com
```



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The window shows a command being run in the terminal:

```
PS C:\Users\anand> ssh -i "SELab.pem" ubuntu@ec2-34-234-78-151.compute-1.amazonaws.com
```

The terminal is currently empty, indicating no output has been received from the remote host.

8. Install tools like Docker, git, vim, nano

```
ubuntu@ip-172-31-28-123:~$ sudo apt-get install docker.io -y
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 53 not upgraded.
Need to get 80.1 MB of archives.
After this operation, 304 MB of additional disk space will be used.
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 1.1.12-0ubuntu3.1 [8599 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.19+really1.7.12-0ubuntu4.2 [38.6 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dns-root-data all 2023112702-willsync1 [4450 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dnsmasq-base amd64 2.90-2build2 [375 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 26.1.3-0ubuntu1~24.04.1 [32.4 MB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
Fetched 80.1 MB in 1s (66.3 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 76605 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_1.1.12-0ubuntu3.1_amd64.deb ...
Unpacking runc (1.1.12-0ubuntu3.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.7.19+really1.7.12-0ubuntu4.2_amd64.deb ...
Unpacking containerd (1.7.19+really1.7.12-0ubuntu4.2) ...
Selecting previously unselected package dns-root-data.
Preparing to unpack .../4-dns-root-data_2023112702-willsync1_all.deb ...
Unpacking dns-root-data (2023112702-willsync1) ...
Selecting previously unselected package dnsmasq-base.
Preparing to unpack .../5-dnsmasq-base_2.90-2build2_amd64.deb ...
Unpacking dnsmasq-base (2.90-2build2) ...
Selecting previously unselected package docker.io.
Preparing to unpack .../6-docker.io_26.1.3-0ubuntu1~24.04.1_amd64.deb ...
```

9. Clone Maven Web App

```
ubuntu@ip-172-31-82-208:~$ git clone https://github.com/TheMajesticWolf/Maven_Web_App.git
Cloning into 'Maven_Web_App'...
remote: Enumerating objects: 19, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 19 (delta 0), reused 19 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (19/19), done.
ubuntu@ip-172-31-82-208:~$ █
```

10. Create docker image and then run the container

```
ubuntu@ip-172-31-28-123:~/ > mv dockerfile Dockerfile
mv: cannot stat 'dockerfile': No such file or directory
ubuntu@ip-172-31-28-123:~/ > nano Dockerfile
ubuntu@ip-172-31-28-123:~/ > nano Docker
ubuntu@ip-172-31-28-123:~/ > mv Docker Dockerfile
ubuntu@ip-172-31-28-123:~/ > sudo docker build -t maven-web-project .
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 70.14kB
Step 1/2 : FROM tomcat:9-jdk21
9-jdk21: Pulling from library/tomcat
de44b265507a: Pull complete
4c2af91a87d: Pull complete
89e9bbcfa697: Pull complete
11be3e613582: Pull complete
1b9d1e181a2a: Pull complete
fc68c0117916: Pull complete
4f4fb700ef54: Pull complete
c9f74ce6014e: Extracting [>
] 163.8kB/13.66MB
```

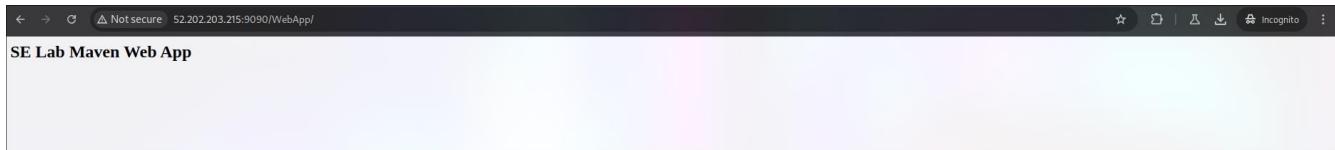
11. Add inbound rule to allow traffic

The screenshot shows a browser window with multiple tabs open, including AWS Infrastructure, Launch AWS Academy, ModifyInboundSec, Tessellator 4.0, GitHub, ChatGPT, and Welcome to nginx!. The main content area is the AWS Management Console, specifically the EC2 > Security Groups page. A security group named "sg-0bf9eec5cddf1cc33" is selected, and the "Edit inbound rules" section is active. The table lists four existing rules and one new rule being added:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-07d045fee30cebbe2	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-061f4692a0c803754	SSH	TCP	22	Custom	0.0.0.0/0
sgr-064ac7f0d9db6689e	HTTPS	TCP	443	Custom	0.0.0.0/0
-	Custom TCP	TCP	9090	Anyw...	0.0.0.0/0

A tooltip at the bottom of the table area states: "⚠️ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." At the bottom of the page, there are links for CloudShell, Feedback, and a footer with copyright information.

12. In browser, go to <http://<public-ip>:9090/<project-name>>



13. Stop your container

```
ubuntu@ip-172-31-82-208:~/Maven_Web_App$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
14b6a01c7b8        maven-web-project   "catalina.sh run"   4 minutes ago    Up 4 minutes      0.0.0.0:9090->8080/tcp, :::9090->8080/tcp   heuristic_villani
14b6a01c7b8
ubuntu@ip-172-31-82-208:~/Maven_Web_App$ sudo docker stop 14b6a01c7b8
14b6a01c7b8
ubuntu@ip-172-31-82-208:~/Maven_Web_App$
```

Result: An AWS Learning Account was created, a sample HTML file was deployed on AWS EC2, and the Maven web application was successfully hosted.