

**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**

### **Theory:**

Continuous Integration (CI) is a DevOps practice where code changes are automatically built, tested, and integrated into a shared repository multiple times a day. It helps in early detection of errors, reduces integration problems, and improves software quality.

### **Jenkins: An Overview**

Jenkins is an open-source CI/CD automation tool used for building, testing, and deploying applications. It allows developers to automate software development workflows and ensures a seamless integration process. Jenkins supports various build tools like **Maven**, **Ant**, and **Gradle** to compile and package applications.

### **Installing and Configuring Jenkins**

#### **1. Download and Install Jenkins**

- Install Java (JDK) as a prerequisite.
- Download Jenkins from the official website and install it on the server.
- Start Jenkins and configure initial setup using an administrator password.

#### **2. Installing Build Tools**

- Install **Maven**, **Ant**, or **Gradle** depending on project requirements.
- Configure Jenkins to recognize the installed build tool.

#### **3. Creating a Build Job in Jenkins**

- Navigate to **Jenkins Dashboard** → **New Item** → **Freestyle**

## Software Engineering & Project Management Lab

### Experiment No :- 04

**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**

- **Project/Pipeline.**
  - Configure the **Git repository URL** to fetch the source code.
  - Select the **Build Tool (Maven/Ant/Gradle)** and define the build command.
  - Set up triggers (e.g., Git webhooks) for automatic build execution.
  - Save and trigger the build job to verify the setup.

To install Jenkins following software packages are required:

- 1) GIT (git-scm.com)
- 2) Notepad++ (<https://notepad-plus-plus.org/downloads/>)
- 3) Latest Java development kit (JDK)
- 4) Jenkins
- 5) Apache Maven (Optional)

Step 1 -: Install GIT

Step 2 -: Install Notepad++

Step 3 -: Install Java

Step 4 -: Install Jenkins

Step 5 -: Install Maven

Jenkins is an open source automation tool written in Java with plugins built for Continuous Integration purpose. Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. It also allows you to continuously deliver your software by integrating with a large number of testing and deployment technologies.

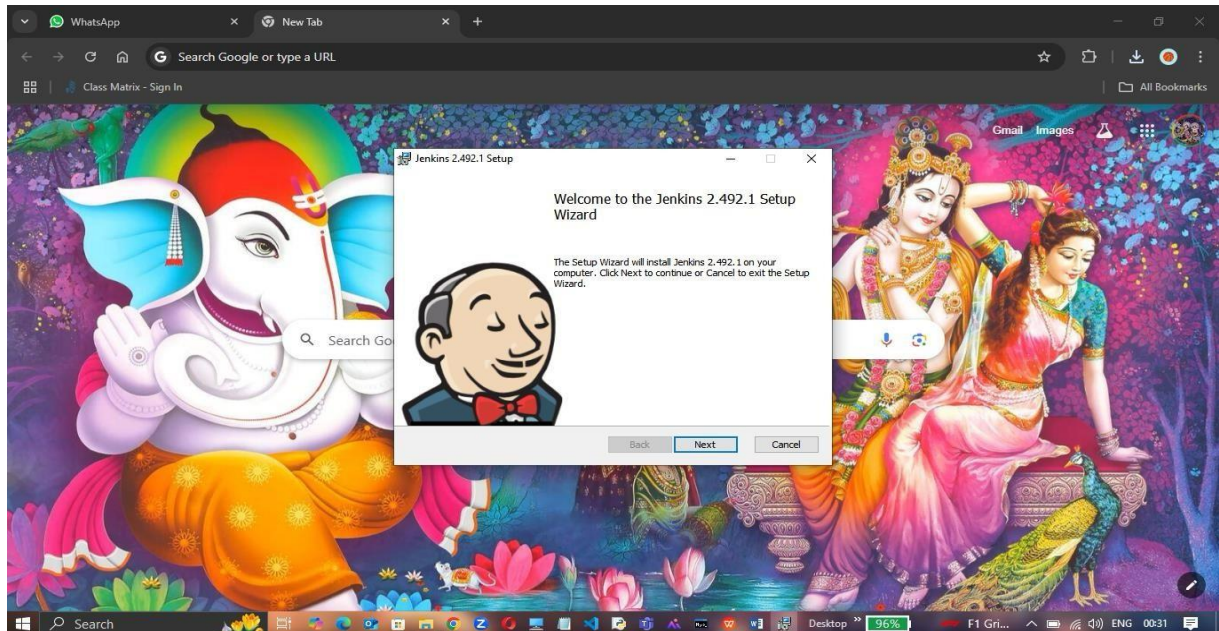
## Software Engineering & Project Management Lab

### Experiment No :- 04

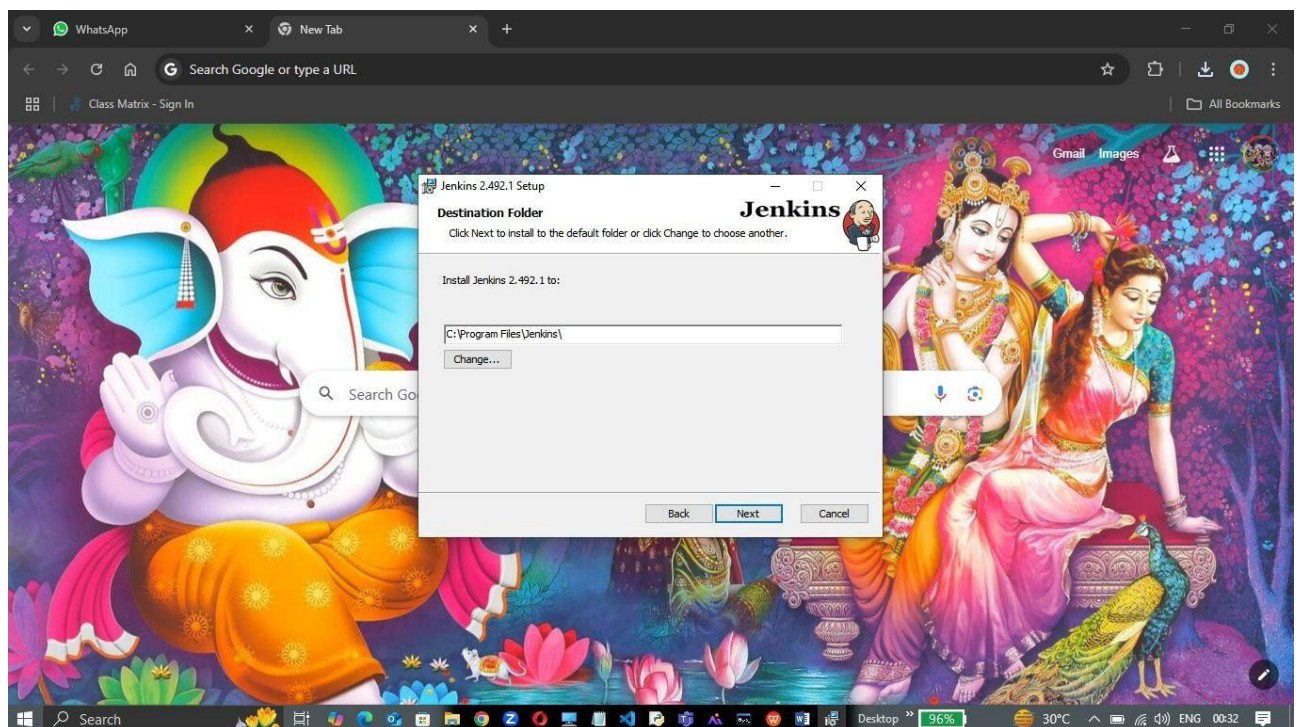
**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**

**Step 1:-** Open <https://www.jenkins.io/doc/book/installing/windows/> and install Jenkins.

Open the installed .exe setup



**Step 2:** Locate the folder where you want to install Jenkins in the location path:

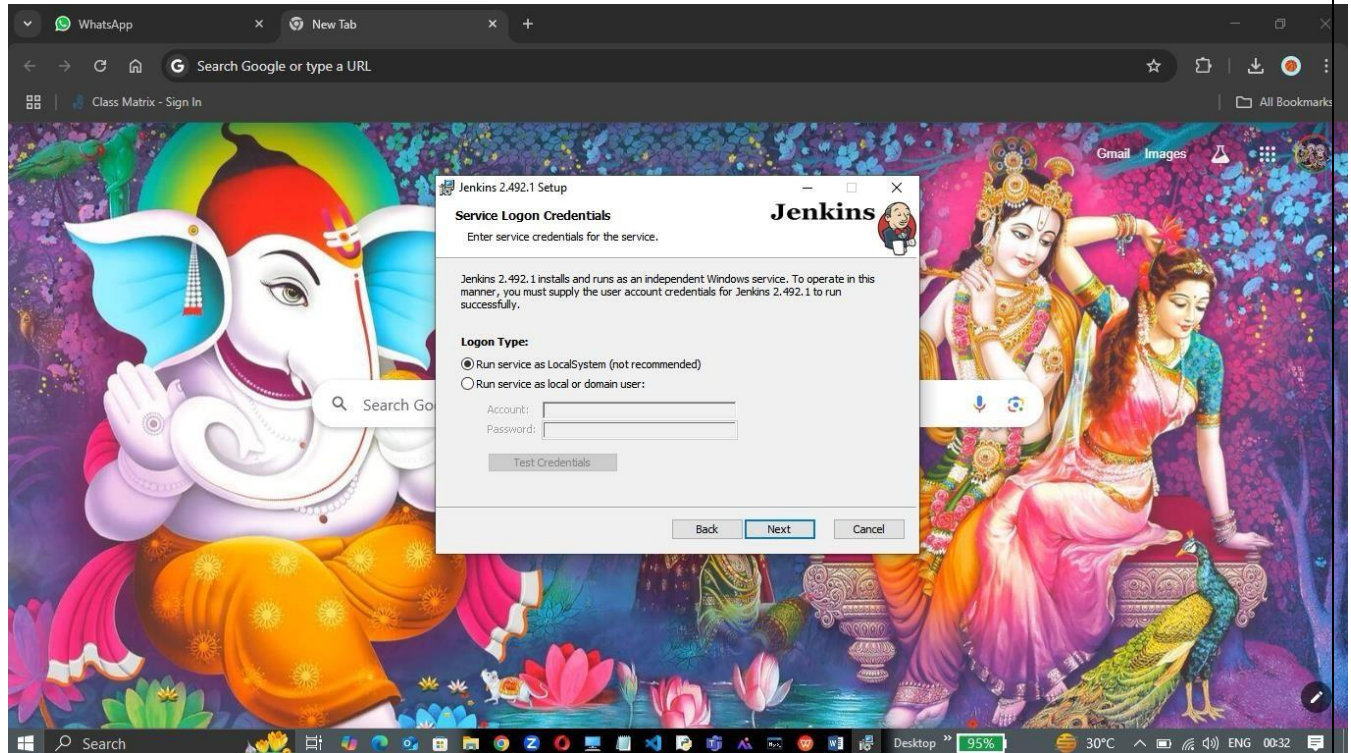


## Software Engineering & Project Management Lab

### Experiment No :- 04

**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**

**Step 3:** Select service as Local System and proceed to Next.



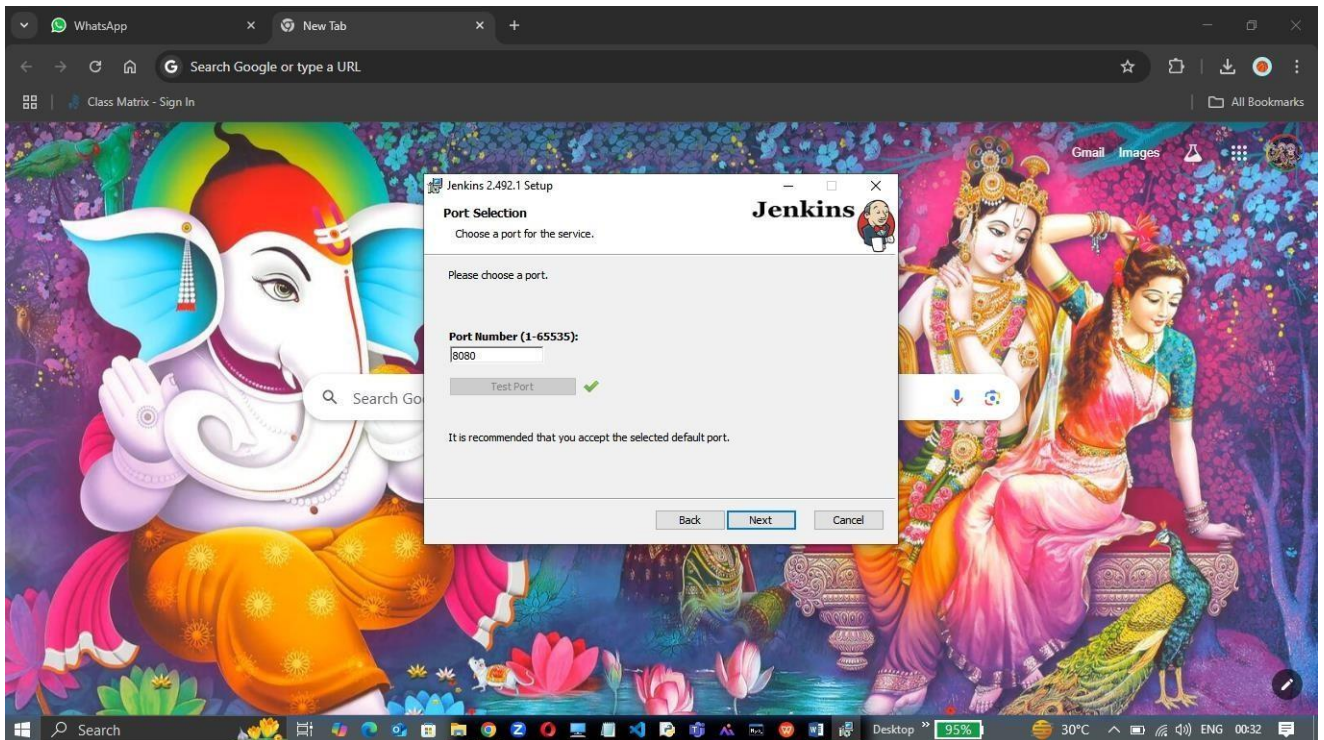
**Step 4:** Select the port 8080 and click Test Port button. The green tick will appear after which you can proceed to Next.



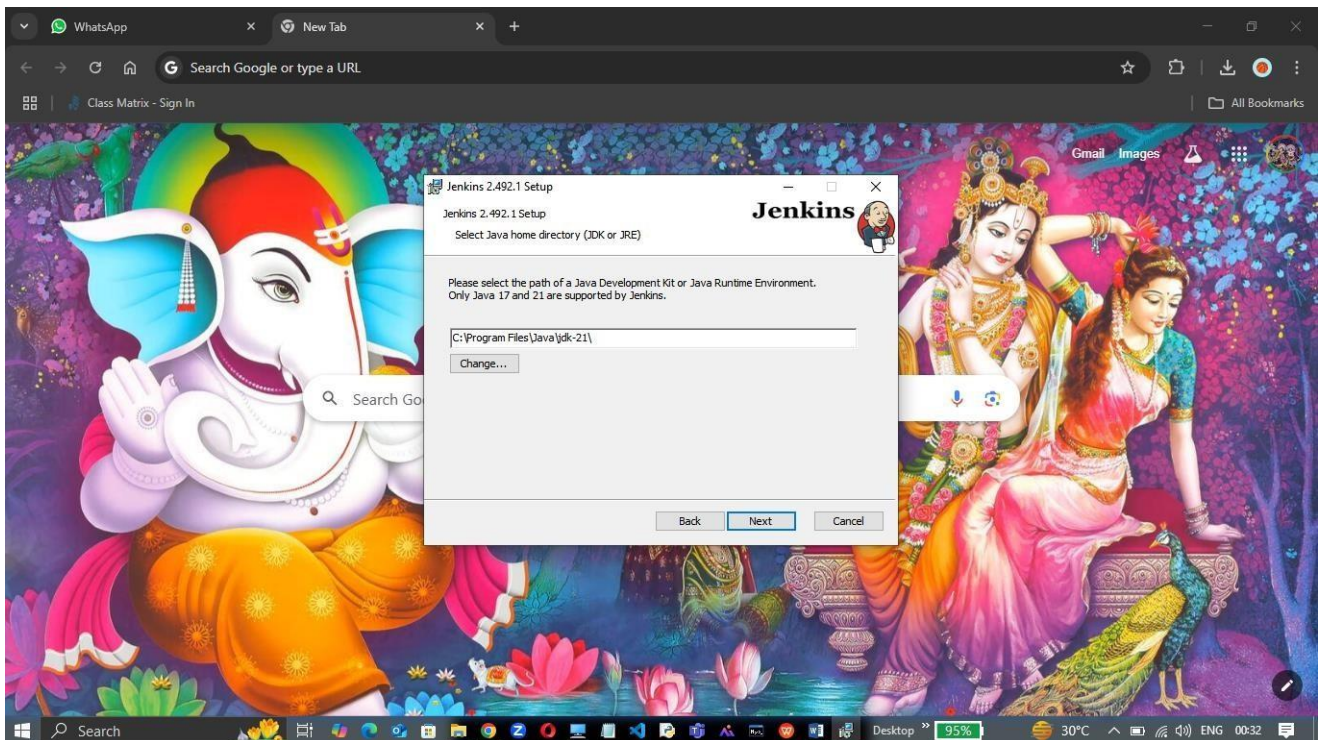
## Software Engineering & Project Management Lab

### Experiment No :- 04

**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**



**Step 5:** Locate the folder where you have installed JDK in the location path:



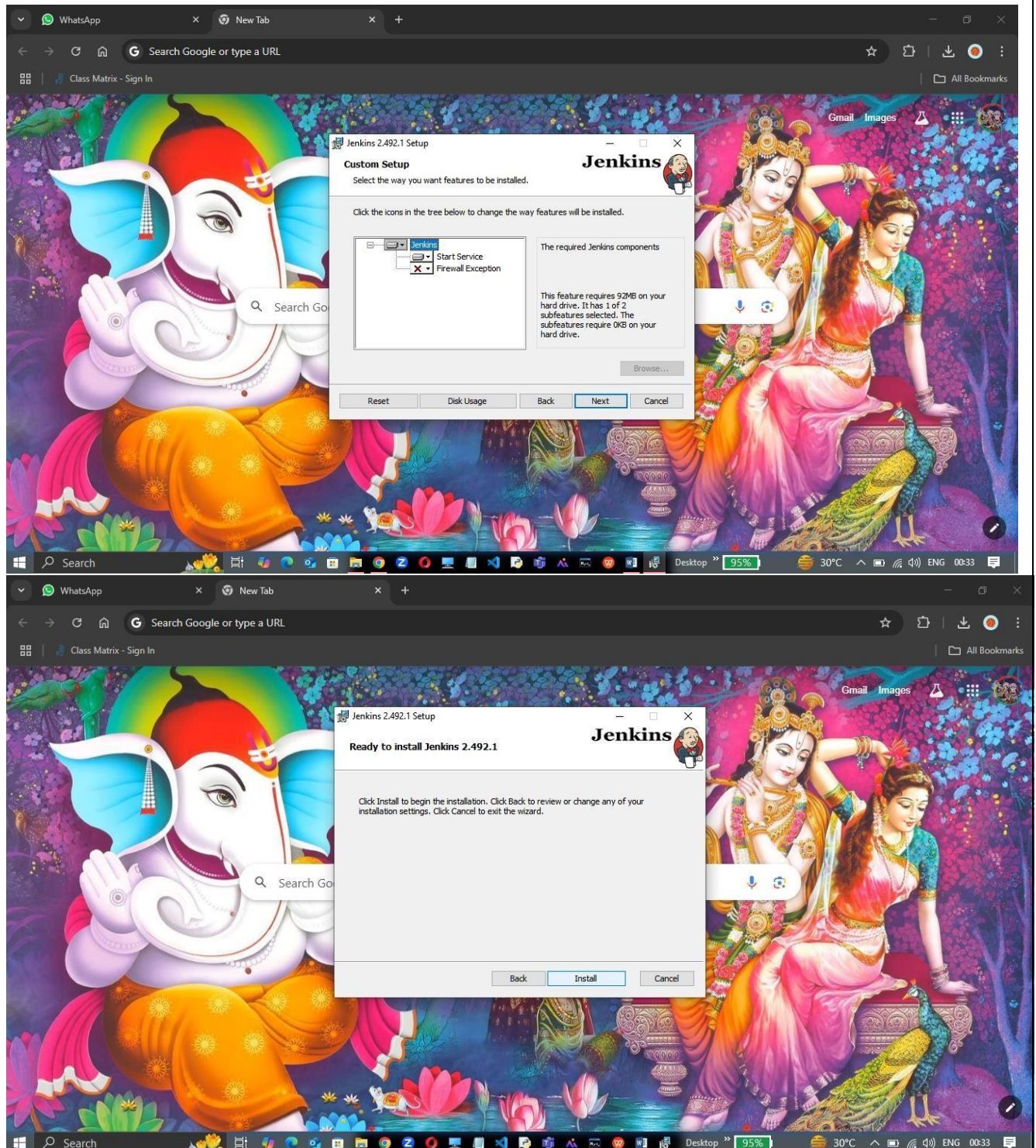
**Proceed to Next**



## Software Engineering & Project Management Lab

### Experiment No :- 04

**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**

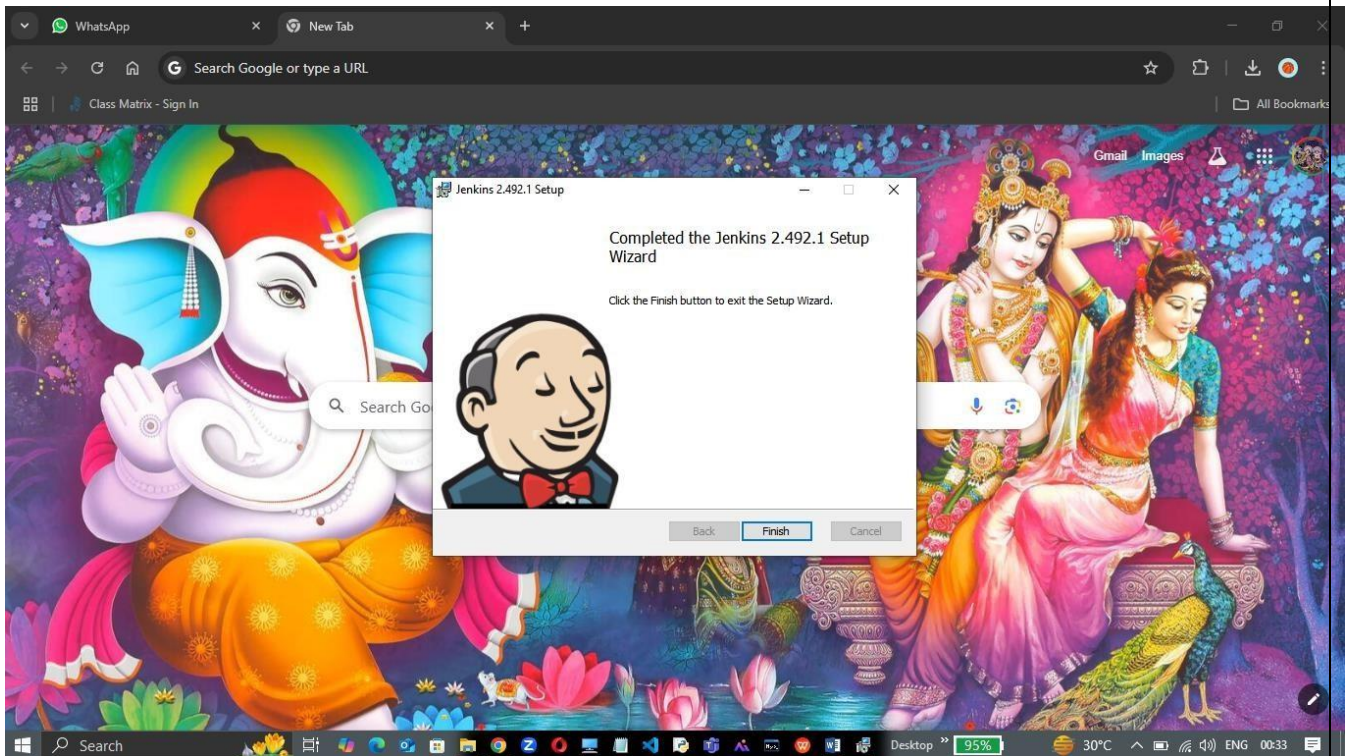


**On clicking 'Install', installation is finished.**

## Software Engineering & Project Management Lab

### Experiment No :- 04

**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**



**Step 6:** Once Installation is done, you can test the Jenkins on <http://localhost:8080> on the browser.

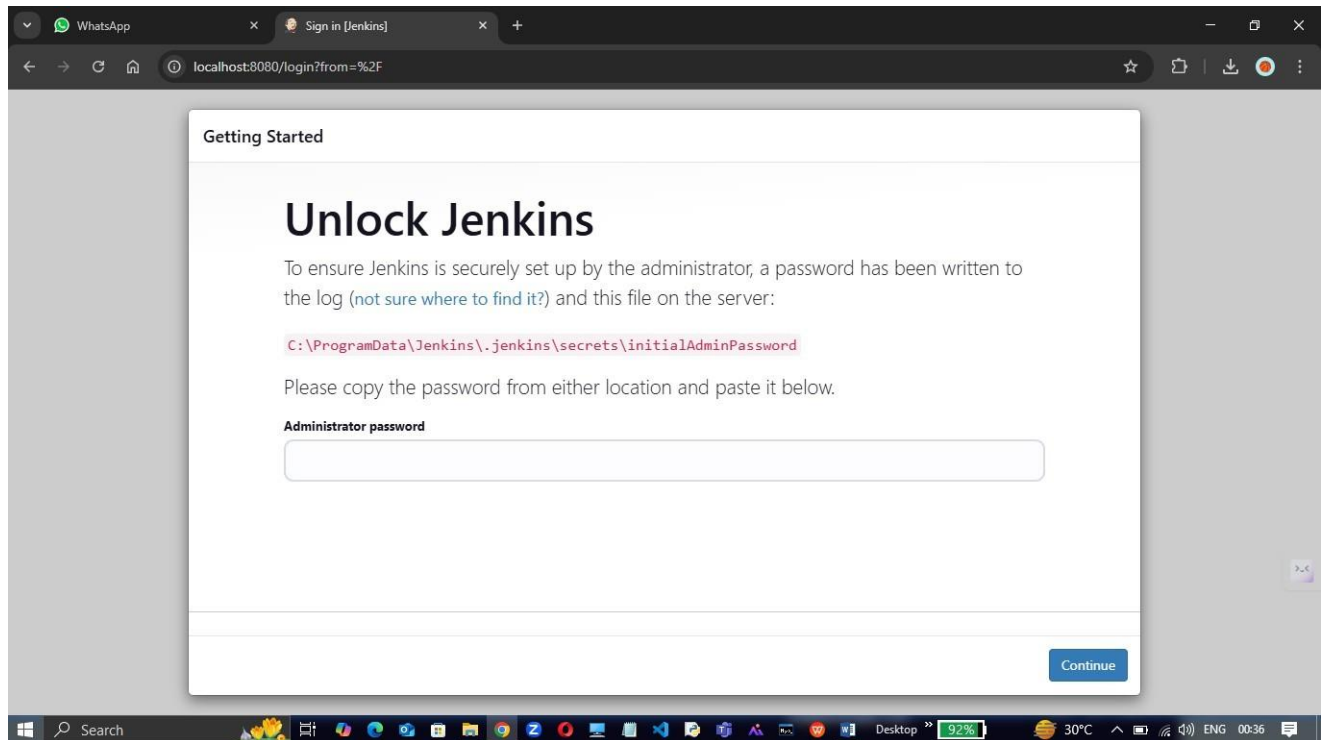
First time, when you open Jenkins portal it will ask to put admin default password which is stored in `/var/lib/jenkins/secrets/initialAdminPassword` file.



## Software Engineering & Project Management Lab

### Experiment No :- 04

**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**



**Step 7:** On entering the password, you can continue to choose “Install Suggested Plugins”



Once plugins are installed, click on next and specify the admin details along with the new password for Jenkins admin and click on finish to complete the installation.



## Software Engineering & Project Management Lab

### Experiment No :- 04

**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**

After filling the details, click on Save & Continue, you will be redirected to the dashboard.

### Getting Started

# Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	<pre>** bouncycastle API ** Instance Identity ** JavaBeans Activation Framework (JAF) API ** JavaMail API ** Credentials ** Plain Credentials ** Gson API ** Trilead API ** SSH Credentials Credentials Binding ** SCM API ** Pipeline: API ** commons-lang3 v3.x Jenkins API Timestamp ** Caffeine API ** Script Security ** JAXB ** SnakeYAML API ** Jackson 2 API ** commons-text API ** Pipeline: Supporting APIs ** Plugin Utilities API ** Font Awesome API ** Bootstrap 5 API ** JQuery3 API ** - required dependency</pre>
✓ Timestamper	⌚ Workspace Cleanup	⌚ Ant	⌚ Gradle	
⌚ Pipeline	⌚ GitHub Branch Source	⌚ Pipeline: GitHub Groovy Libraries	⌚ Pipeline: Stage View	
⌚ Git	⌚ SSH Build Agents	⌚ Matrix Authorization Strategy	⌚ PAM Authentication	
⌚ LDAP	⌚ Email Extension	⌚ Mailer		

Jenkins 2.426.3

### Dashboard

+ New Item

👤 People

📅 Build History

⚙️ Manage Jenkins

📌 My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

### Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

#### Start building your software project

Create a job +

#### Set up a distributed build

Set up an agent 🖨

Configure a cloud ☁

Learn more about distributed builds ?

REST API

Jenkins 2.426.3

# Software Engineering & Project Management Lab

## Experiment No :- 04


**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**


Dashboard >


Enter an item name


example 1


» Required field


**Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

**Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

**Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

**Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**Multibranch Pipeline**  
Creates a set of Pipeline projects according to detected branches in one SCM repository.

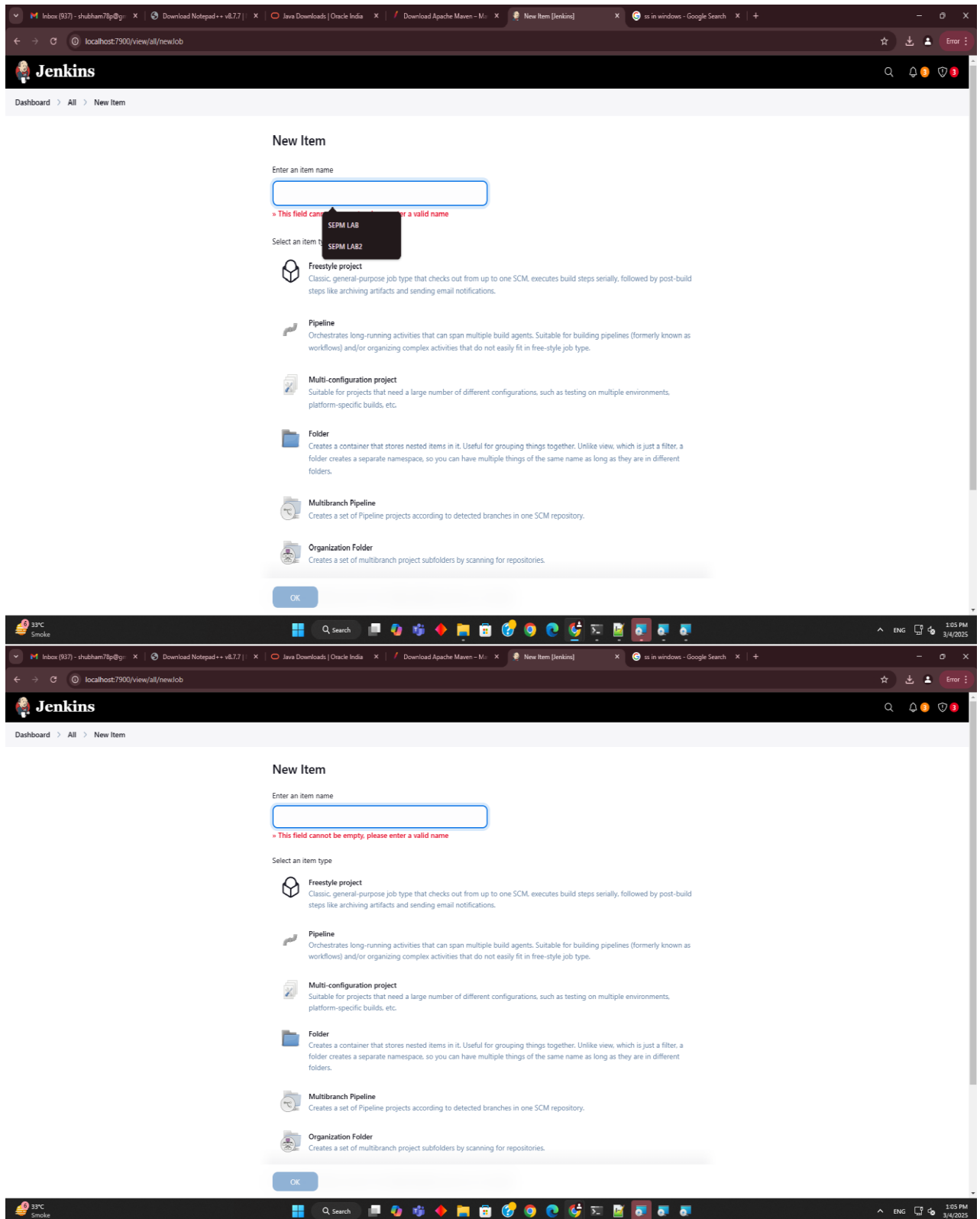
**Organization Folder**  
Creates a set of multibranch project subfolders by scanning for repositories.

OK

# Software Engineering & Project Management Lab

## Experiment No :- 04

**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**





# Software Engineering & Project Management Lab

## Experiment No :- 04

**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**

The image shows two screenshots from a computer screen. The top screenshot is of the Oracle Java Downloads page for JDK 21 on Windows. It lists download links for x64 Compressed Archive, x64 Installer, and x64 MSI Installer. The bottom screenshot is of the Jenkins 'New Item' configuration page, where 'testproject' is entered as the item name, and 'Freestyle project' is selected as the item type.

**Oracle Java Downloads Page (JDK 21 Windows):**

Earlier JDK versions are available below.

**JDK 23   JDK 21   GraalVM for JDK 23   GraalVM for JDK 21**

**Java SE Development Kit 21.0.6 downloads**

JDK 21 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions \(NFTC\)](#).

JDK 21 will receive updates under the NFTC, until September 2026, a year after the release of the next LTS. Subsequent JDK 21 updates will be licensed under the [Java SE OTN License \(OTN\)](#) and production use beyond the [limited free grants](#) of the OTN license will [require a fee](#).

**Linux   macOS   Windows**

Product/file description	File size	Download
x64 Compressed Archive	185.92 MB	<a href="https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.zip">https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.zip</a> (sha256)
x64 Installer	164.31 MB	<a href="https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe">https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe</a> (sha256)
x64 MSI Installer	163.06 MB	<a href="https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.msi">https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.msi</a> (sha256)

**Documentation Download**

**Release information**

- [Online Documentation](#)
- [Installation Instructions](#)
- [Release Notes](#)

**Jenkins New Item Configuration:**

Enter an item name:

Select an item type:

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

**OK**

# Software Engineering & Project Management Lab

## Experiment No :- 04

**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**

The image displays two screenshots of the Jenkins web interface, specifically the 'Configuration' page for a job named 'testproject'. The top screenshot shows the 'General' tab with the 'Description' field empty. The bottom screenshot shows the same page with the 'Description' field containing the text 'hi i am shubham pandey'.

**Jenkins Configuration - General Tab**

**Configuration**

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

**General** (Enabled)

Description

Plain text [Preview](#)

☐ Discard old builds ?

☐ This project is parameterized ?

[Add Parameter](#) ▾

☐ Throttle builds ?

Number of builds ?

1

Approximately 60 minutes between builds

Time period ?

Second ▾

☐ Allow user triggered builds to skip the rate limit ?

☐ Execute concurrent builds if necessary ?

[Save](#) [Apply](#)

**Jenkins Configuration - General Tab**

**Configuration**

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

**General** (Enabled)

Description

hi i am shubham pandey

Plain text [Preview](#)

☐ Discard old builds ?

☐ This project is parameterized ?

[Add Parameter](#) ▾

☐ Throttle builds ?

Number of builds ?

1

Approximately 60 minutes between builds

Time period ?

Second ▾

☐ Allow user triggered builds to skip the rate limit ?

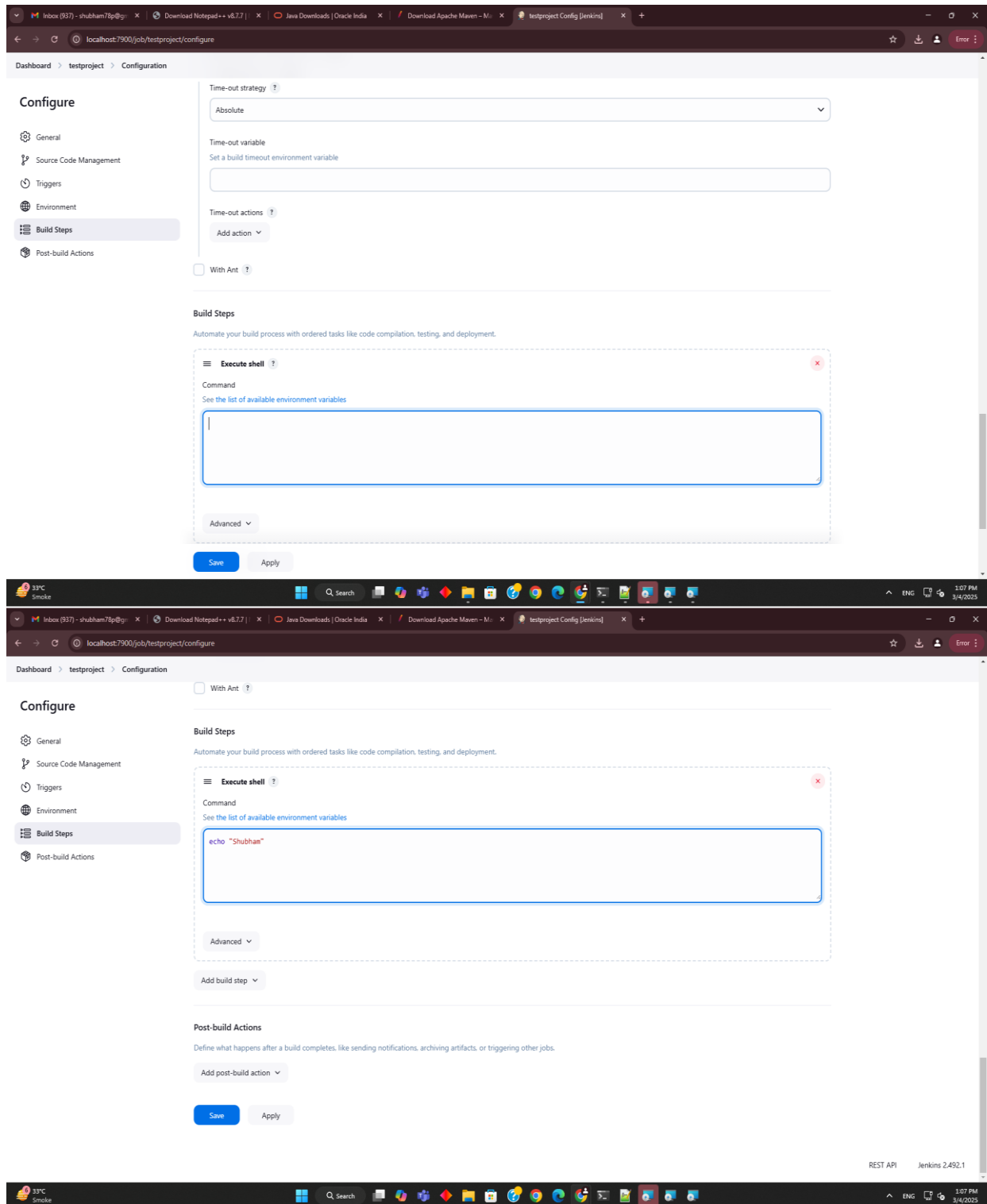
☐ Execute concurrent builds if necessary ?

[Save](#) [Apply](#)

# Software Engineering & Project Management Lab

## Experiment No :- 04

**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**

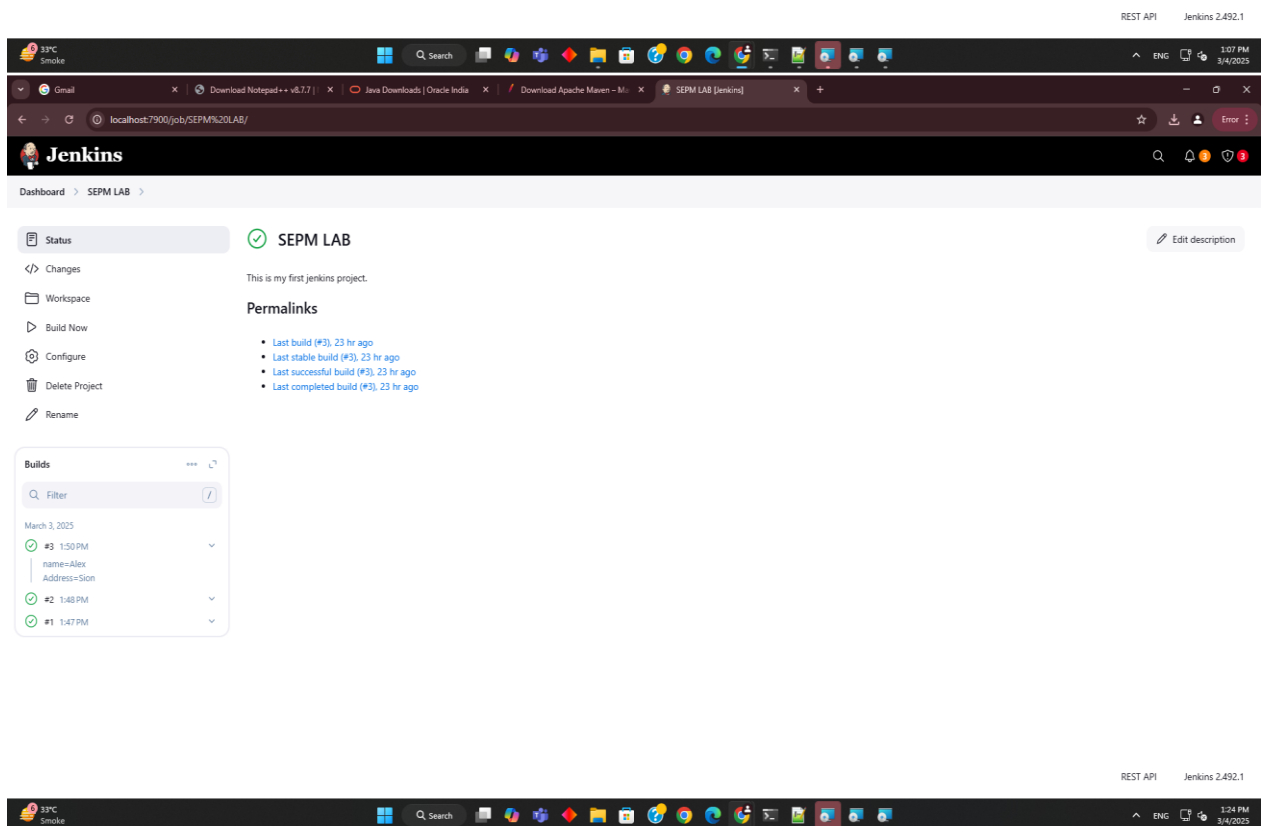
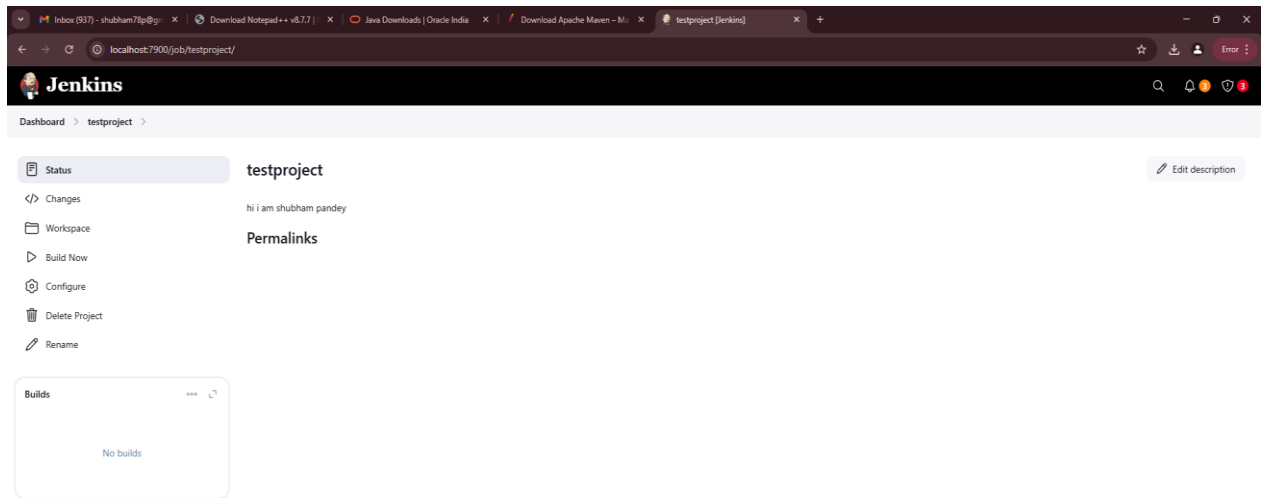




# Software Engineering & Project Management Lab

## Experiment No :- 04

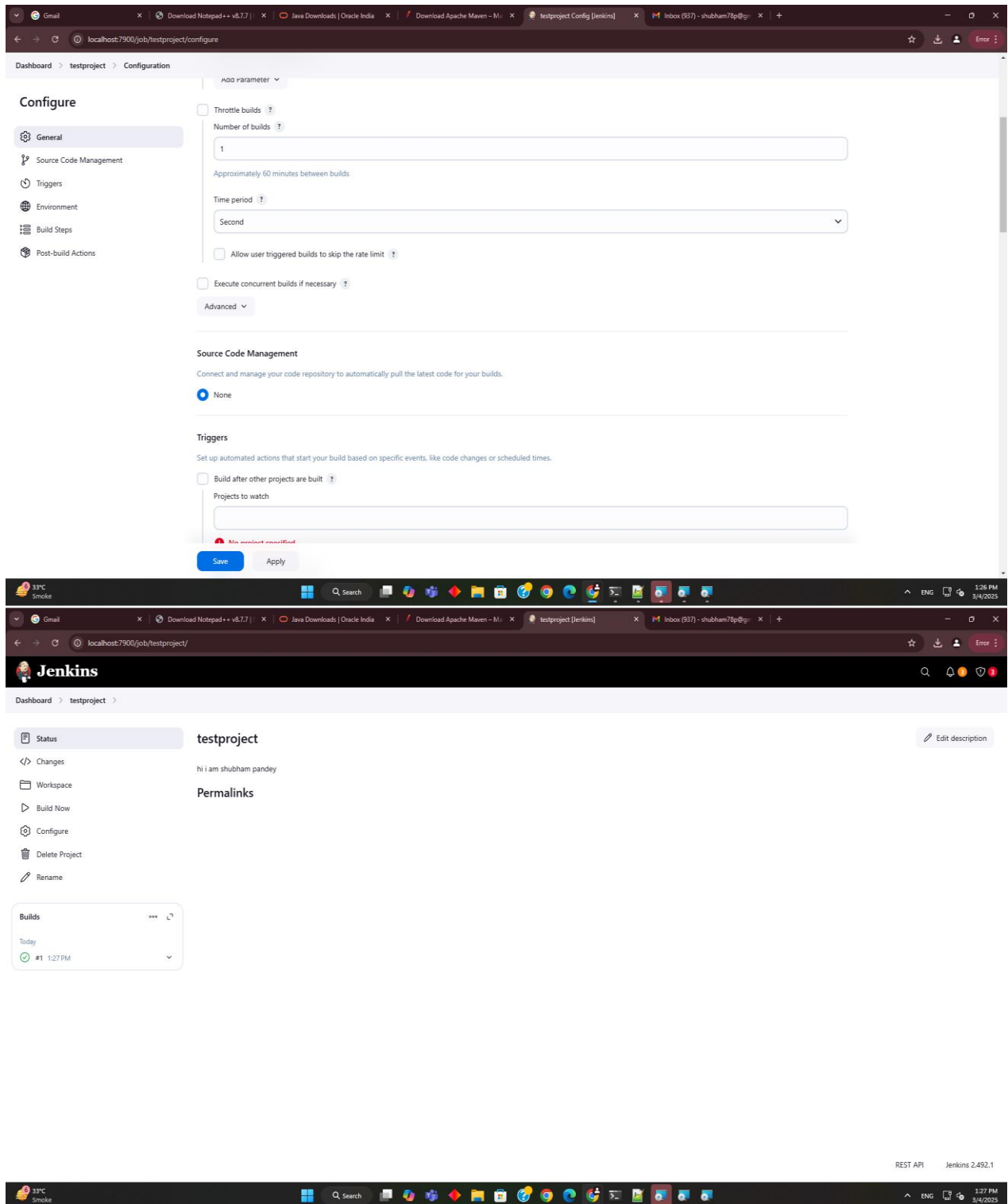
**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**



# Software Engineering & Project Management Lab

## Experiment No :- 04

**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**

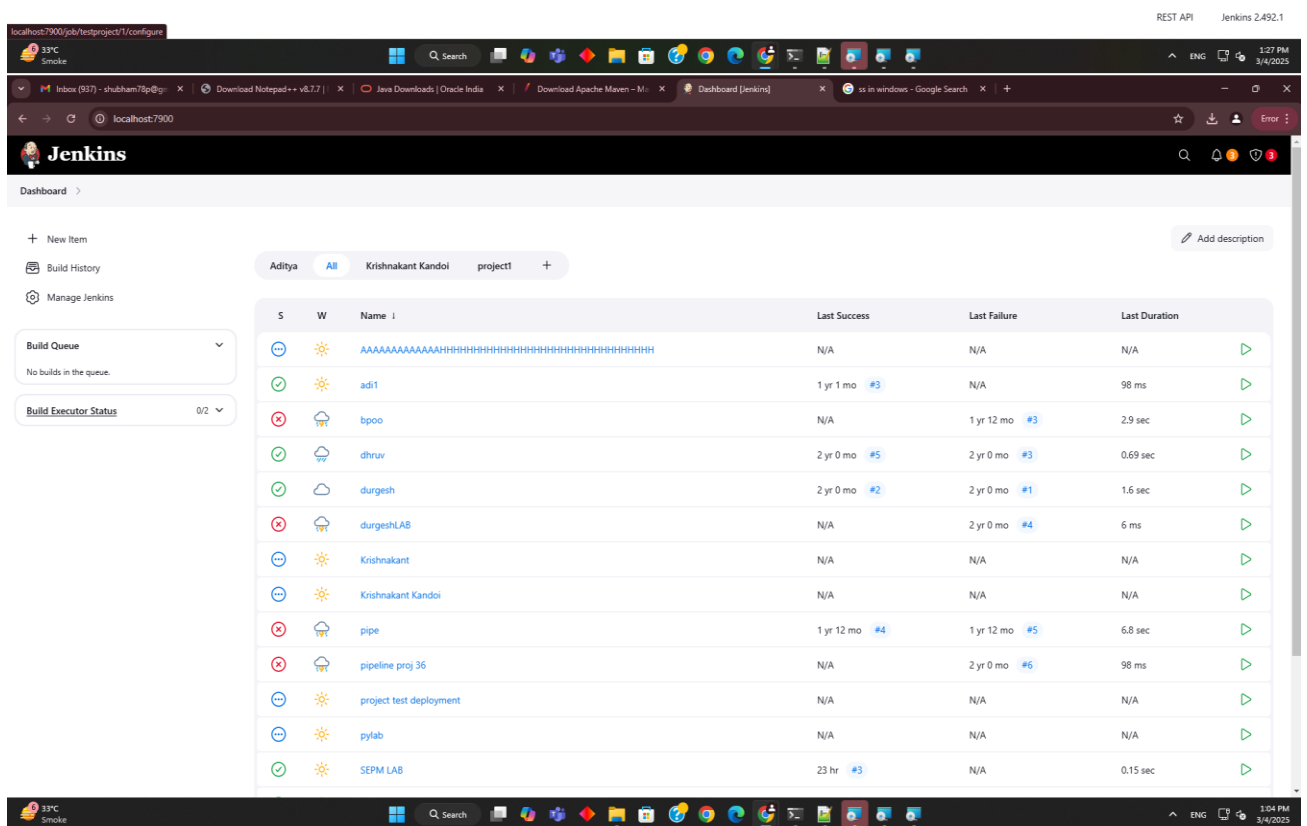
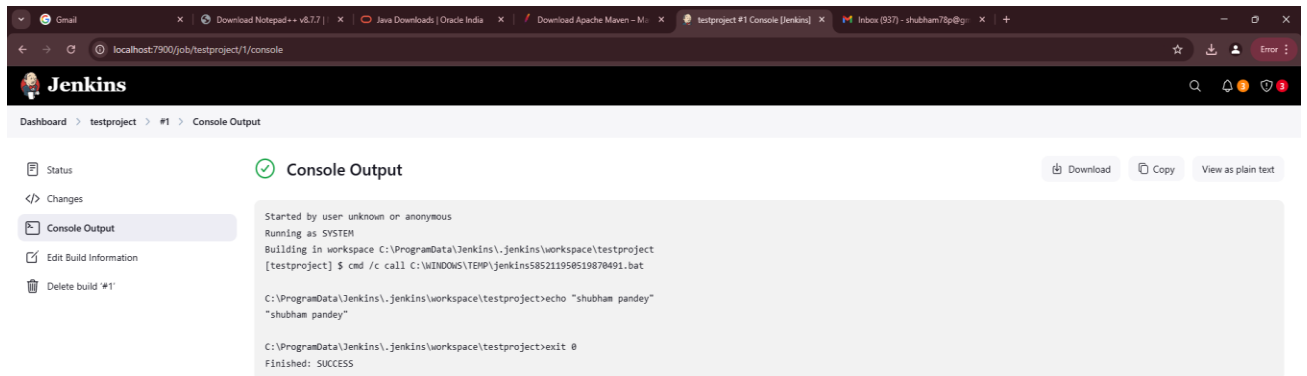


The image shows two screenshots of the Jenkins web interface. The top screenshot displays the 'Configure' page for a job named 'testproject'. The left sidebar contains a 'Configure' section with sub-items: General, Source Code Management, Triggers, Environment, Build Steps, and Post-build Actions. The 'General' tab is selected, showing options for 'Throttle builds' (Number of builds: 1, Time period: Second), 'Execute concurrent builds if necessary', and 'Source Code Management' (set to None). The 'Triggers' section is partially visible. The bottom screenshot shows the Jenkins 'Dashboard' for the 'testproject' job. It includes a 'Status' section with 'testproject' and a description 'hi i am shubham pandey'. Below this are 'Permalinks' and a 'Builds' section showing a single build (1) completed at 1:27 PM. The bottom status bar indicates 'REST API' and 'Jenkins 2.492.1'.

## Software Engineering & Project Management Lab

### Experiment No :- 04

**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**

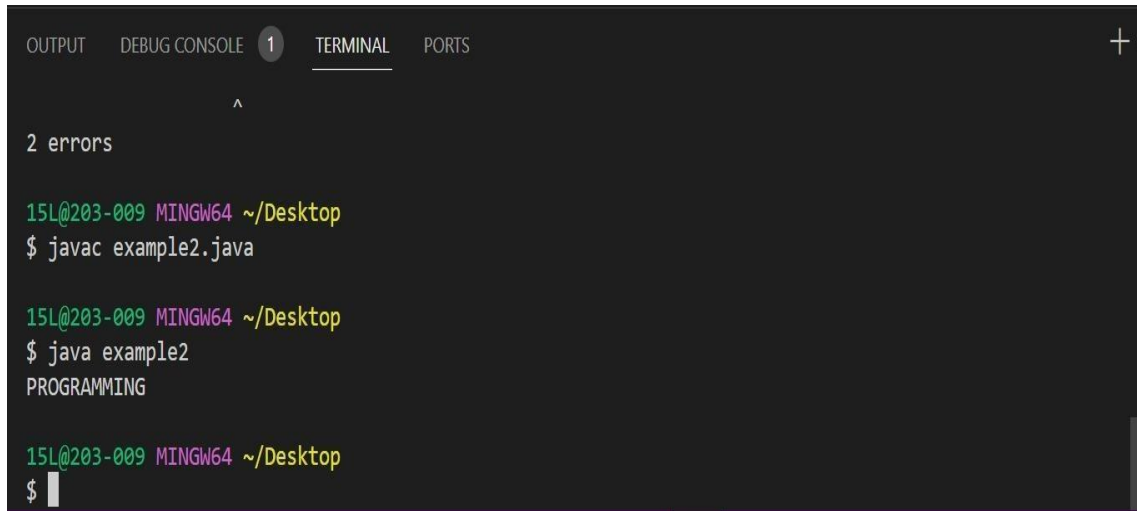




# Software Engineering & Project Management Lab

## Experiment No :- 04

**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**

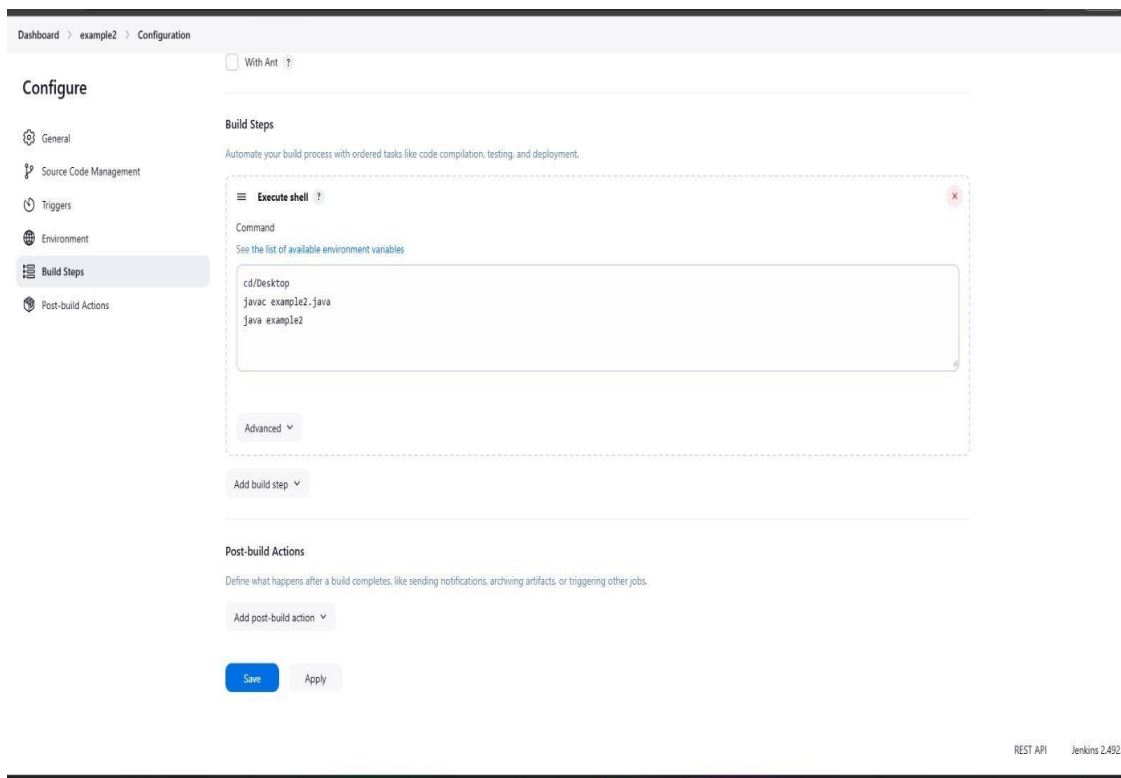


```
OUTPUT  DEBUG CONSOLE 1 TERMINAL  PORTS
^
2 errors

15L@203-009 MINGW64 ~/Desktop
$ javac example2.java

15L@203-009 MINGW64 ~/Desktop
$ java example2
PROGRAMMING

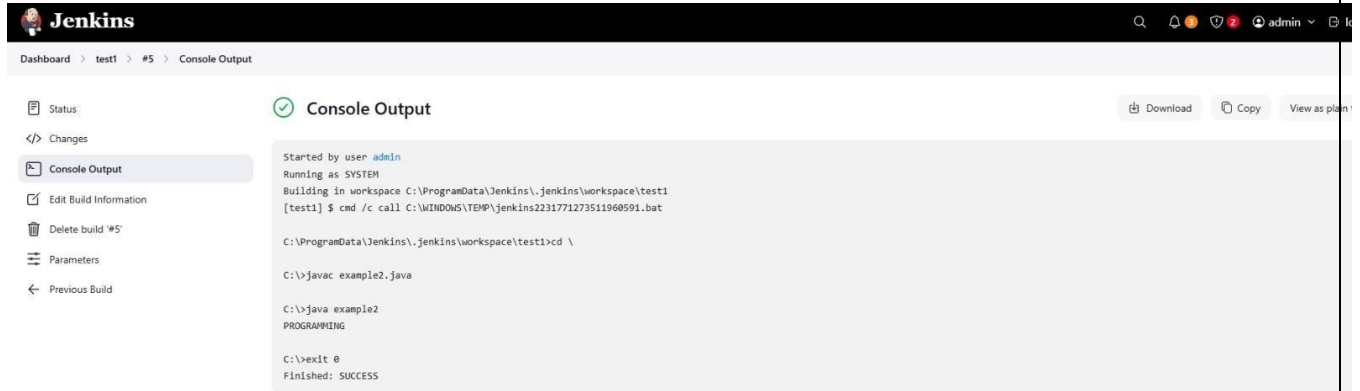
15L@203-009 MINGW64 ~/Desktop
$
```



# Software Engineering & Project Management Lab

## Experiment No :- 04

**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**



This screenshot shows the Jenkins web interface for build #5 of the 'test1' job. The left sidebar contains links for Status, Changes, Console Output (selected), Edit Build Information, Delete build #5, Parameters, and Previous Build. The main area displays the console output for build #5, which shows a successful build process. The output starts with 'Started by user admin' and 'Running as SYSTEM'. It then shows the workspace path 'C:\ProgramData\Jenkins\jenkins\workspace\test1' and the command '[test1] \$ cmd /c call C:\WINDOWS\TEMP\jenkins2231771273511960591.bat'. The build steps include 'cd \', 'javac example2.java', 'java example2 PROGRAMMING', and 'exit 0'. The build finished successfully.

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\test1
[test1] $ cmd /c call C:\WINDOWS\TEMP\jenkins2231771273511960591.bat

C:\ProgramData\Jenkins\jenkins\workspace\test1>cd \

C:\>javac example2.java

C:\>java example2
PROGRAMMING

C:\>exit 0
Finished: SUCCESS
```



This screenshot shows the Jenkins web interface for build #4 of the 'test1' job. The left sidebar contains links for Status, Changes, Console Output (selected), Edit Build Information, Delete build #4, Parameters, Previous Build, and Next Build. The main area displays the console output for build #4, which shows a successful build process. The output starts with 'Started by user admin' and 'Running as SYSTEM'. It then shows the workspace path 'C:\ProgramData\Jenkins\jenkins\workspace\test1' and the command '[test1] \$ cmd /c call C:\WINDOWS\TEMP\jenkins11493019808206271570.bat'. The build steps include 'set /a c=1+2', 'echo "Your Name is 3"', and 'exit 0'. The build finished successfully.

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\test1
[test1] $ cmd /c call C:\WINDOWS\TEMP\jenkins11493019808206271570.bat

C:\ProgramData\Jenkins\jenkins\workspace\test1>set /a c=1+2

C:\ProgramData\Jenkins\jenkins\workspace\test1>echo "Your Name is 3"
"Your Name is 3"

C:\ProgramData\Jenkins\jenkins\workspace\test1>exit 0
Finished: SUCCESS
```



This screenshot shows the Jenkins web interface for build #3 of the 'test1' job. The left sidebar contains links for Status, Changes, Console Output (selected), Edit Build Information, Delete build #3, Parameters, Previous Build, and Next Build. The main area displays the console output for build #3, which shows a successful build process. The output starts with 'Started by user admin' and 'Running as SYSTEM'. It then shows the workspace path 'C:\ProgramData\Jenkins\jenkins\workspace\test1' and the command '[test1] \$ cmd /c call C:\WINDOWS\TEMP\jenkins9536516207865739292.bat'. The build steps include 'set c=12+34', 'echo "Your Name is 12+34"', and 'exit 0'. The build finished successfully.

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\test1
[test1] $ cmd /c call C:\WINDOWS\TEMP\jenkins9536516207865739292.bat

C:\ProgramData\Jenkins\jenkins\workspace\test1>set c=12+34

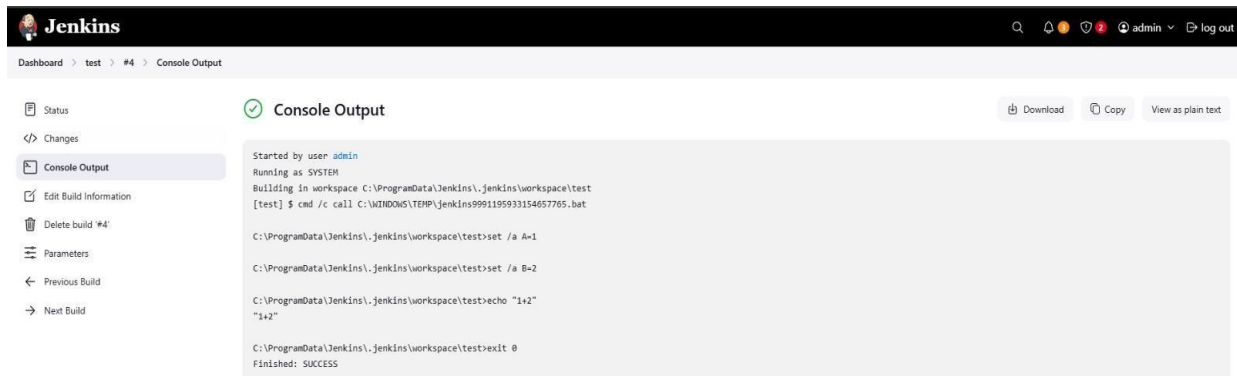
C:\ProgramData\Jenkins\jenkins\workspace\test1>echo "Your Name is 12+34"
"Your Name is 12+34"

C:\ProgramData\Jenkins\jenkins\workspace\test1>exit 0
Finished: SUCCESS
```

# Software Engineering & Project Management Lab

## Experiment No :- 04

**Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job**



The screenshot shows the Jenkins web interface for build #4. The left sidebar contains a menu with options: Status, Changes, Console Output (selected), Edit Build Information, Delete build #4, Parameters, Previous Build, and Next Build. The main area displays the 'Console Output' for build #4, which is a successful build. The output text is as follows:

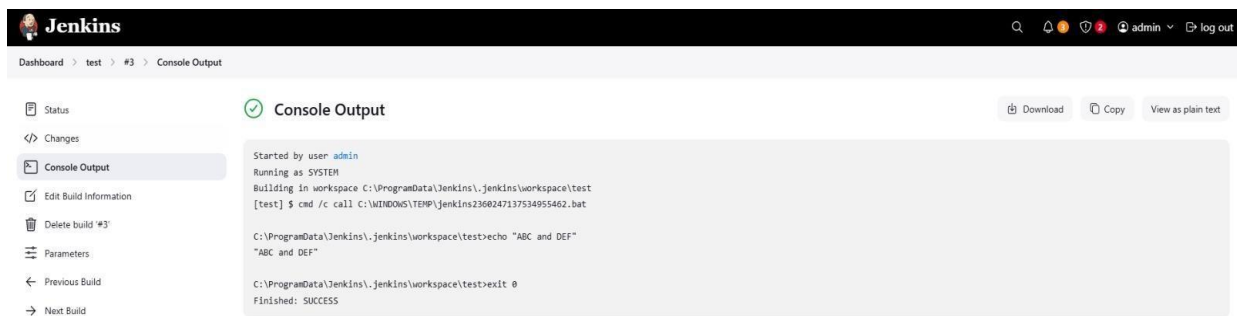
```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\test
[test] $ cmd /c call C:\WINDOWS\TEMP\jenkins9991195933154657765.bat

C:\ProgramData\Jenkins\jenkins\workspace\test>set /a A=1

C:\ProgramData\Jenkins\jenkins\workspace\test>set /a B=2

C:\ProgramData\Jenkins\jenkins\workspace\test>echo "1+2"
"1+2"

C:\ProgramData\Jenkins\jenkins\workspace\test>exit 0
Finished: SUCCESS
```



The screenshot shows the Jenkins web interface for build #3. The left sidebar contains a menu with options: Status, Changes, Console Output (selected), Edit Build Information, Delete build #3, Parameters, Previous Build, and Next Build. The main area displays the 'Console Output' for build #3, which is a successful build. The output text is as follows:

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\test
[test] $ cmd /c call C:\WINDOWS\TEMP\jenkins2360247137534955462.bat

C:\ProgramData\Jenkins\jenkins\workspace\test>echo "ABC and DEF"
"ABC and DEF"

C:\ProgramData\Jenkins\jenkins\workspace\test>exit 0
Finished: SUCCESS
```

**Conclusion:** Thus, we have successfully installed and configured Jenkins with Maven/Ant/Gradle to setup a build Job and learnt about the implementation of Jenkins in open source continuous integration.