

## **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**

#### **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](http://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.

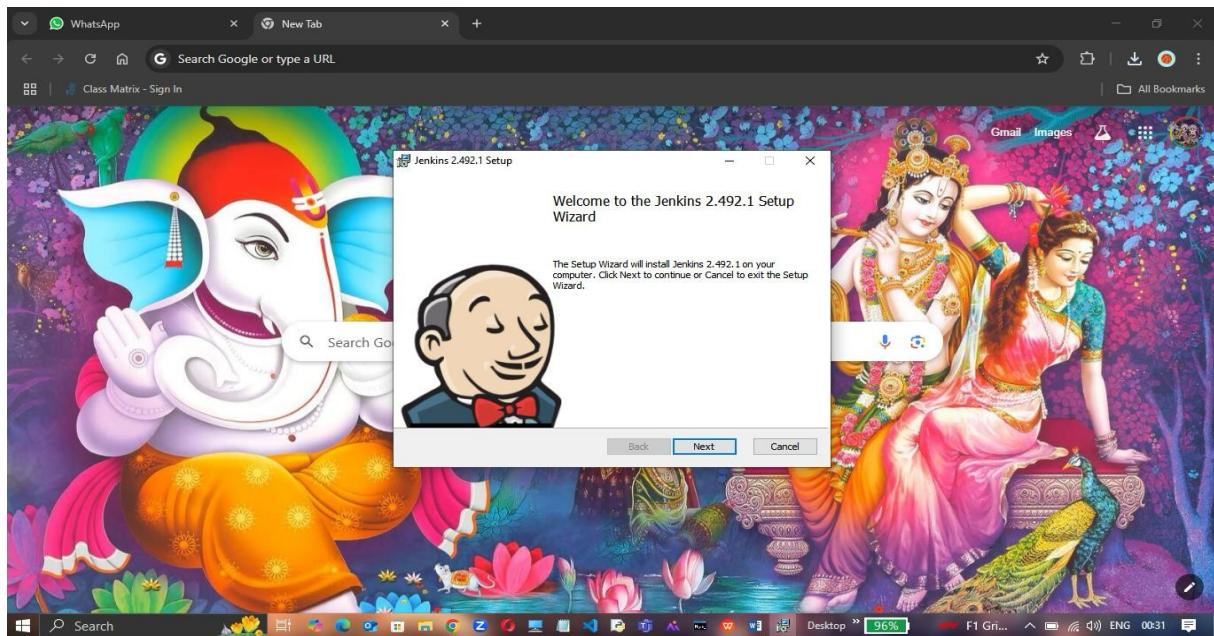
# Software Engineering & Project Management Lab

## Experiment No :- 04

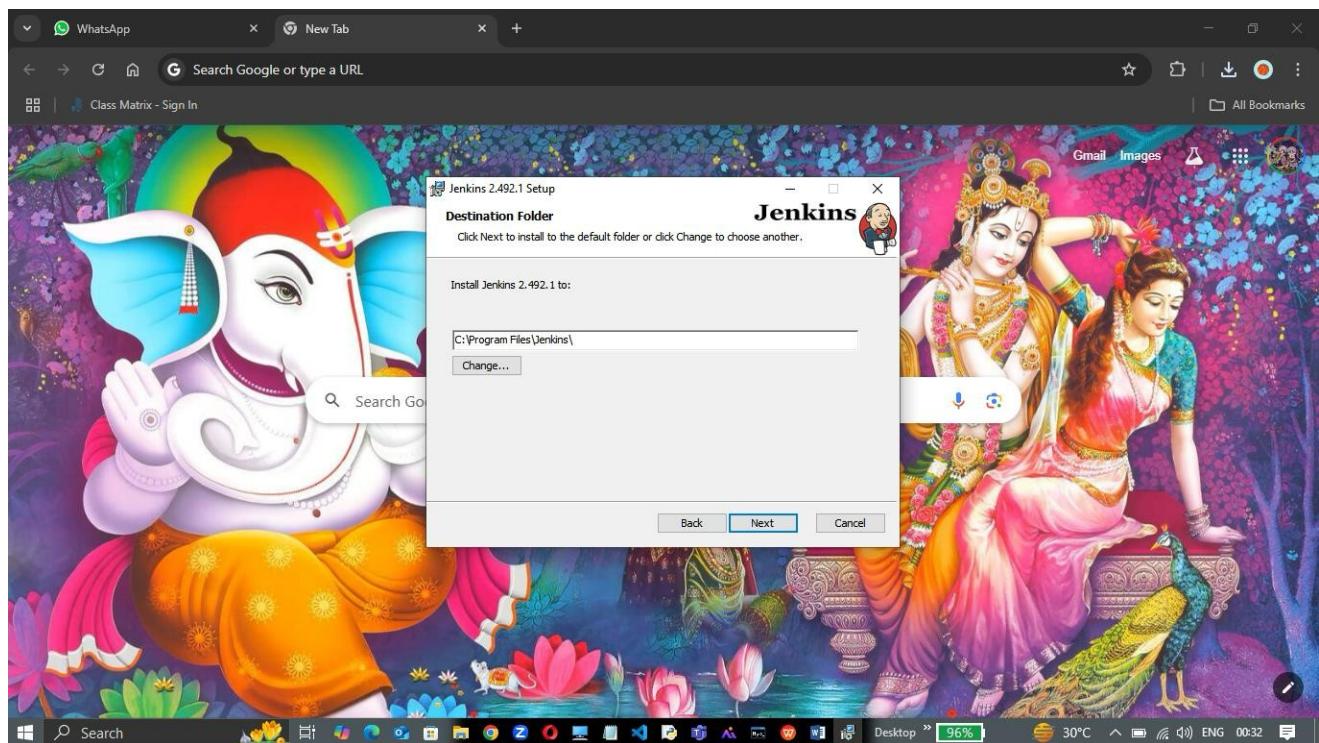
**Aim:** To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job

**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:

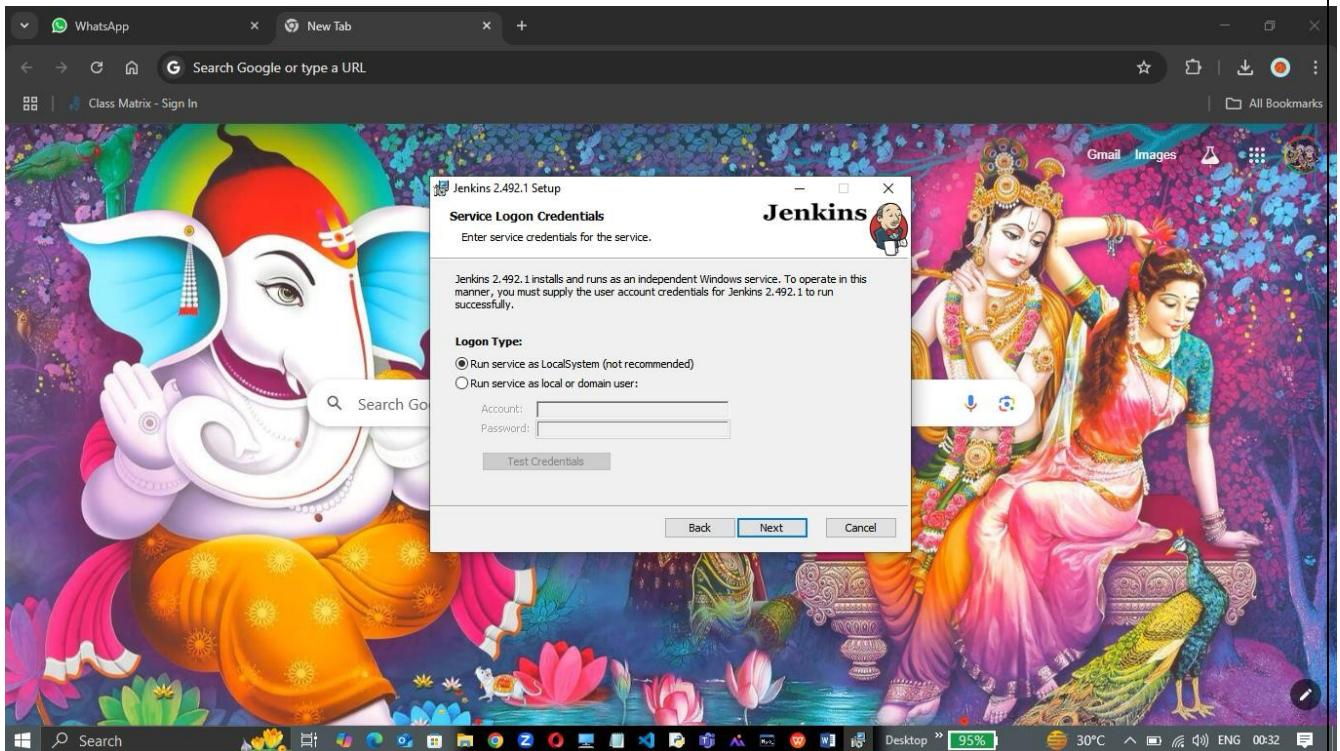


# Software Engineering & Project Management Lab

## 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.

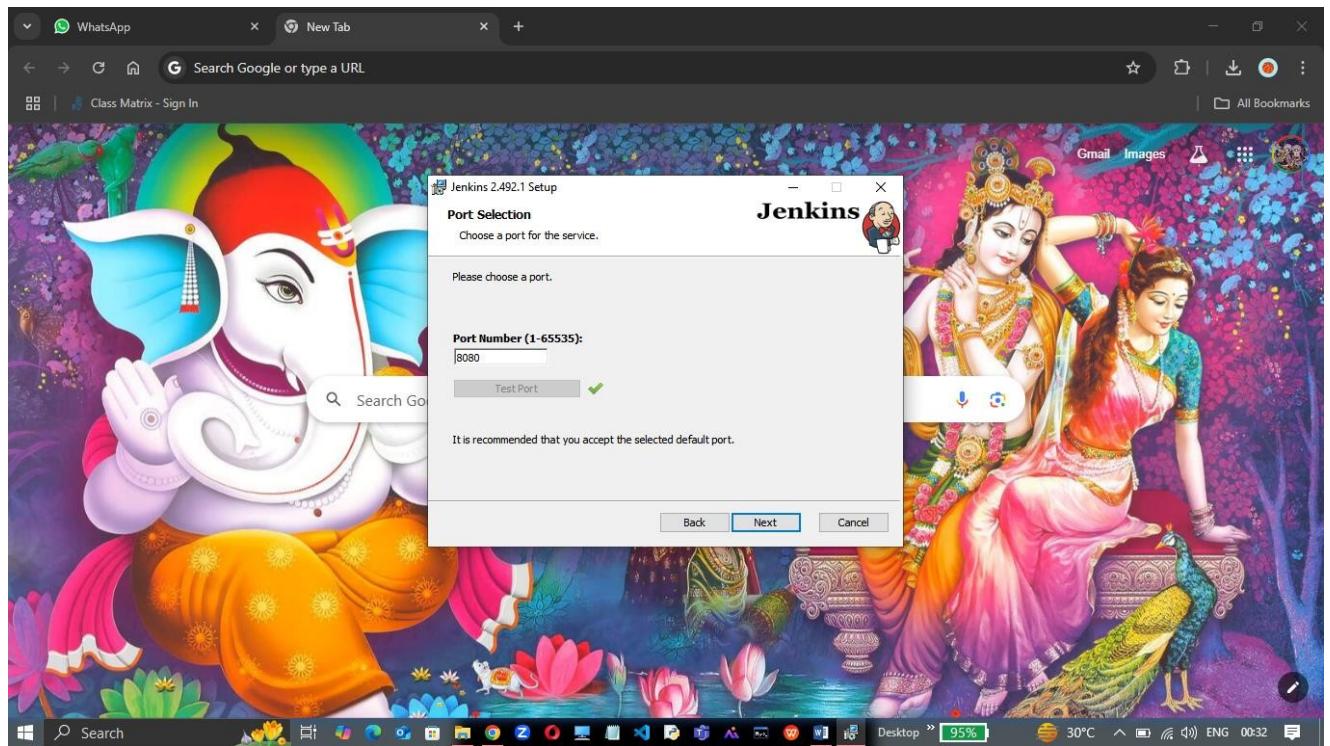


**Step 4:** Select the port 8080 and click Test Port button. The green tick will appear after which you can proceed to Next.

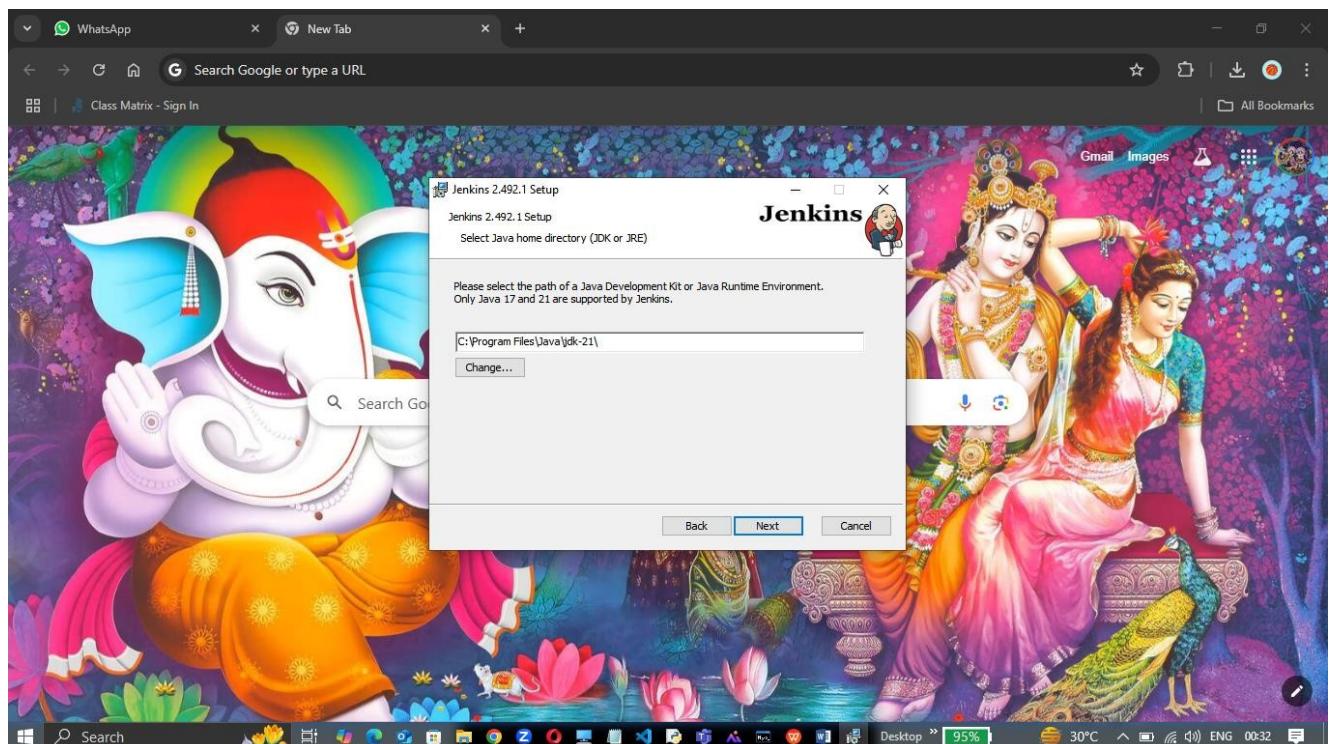
# Software Engineering & Project Management Lab

## 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:

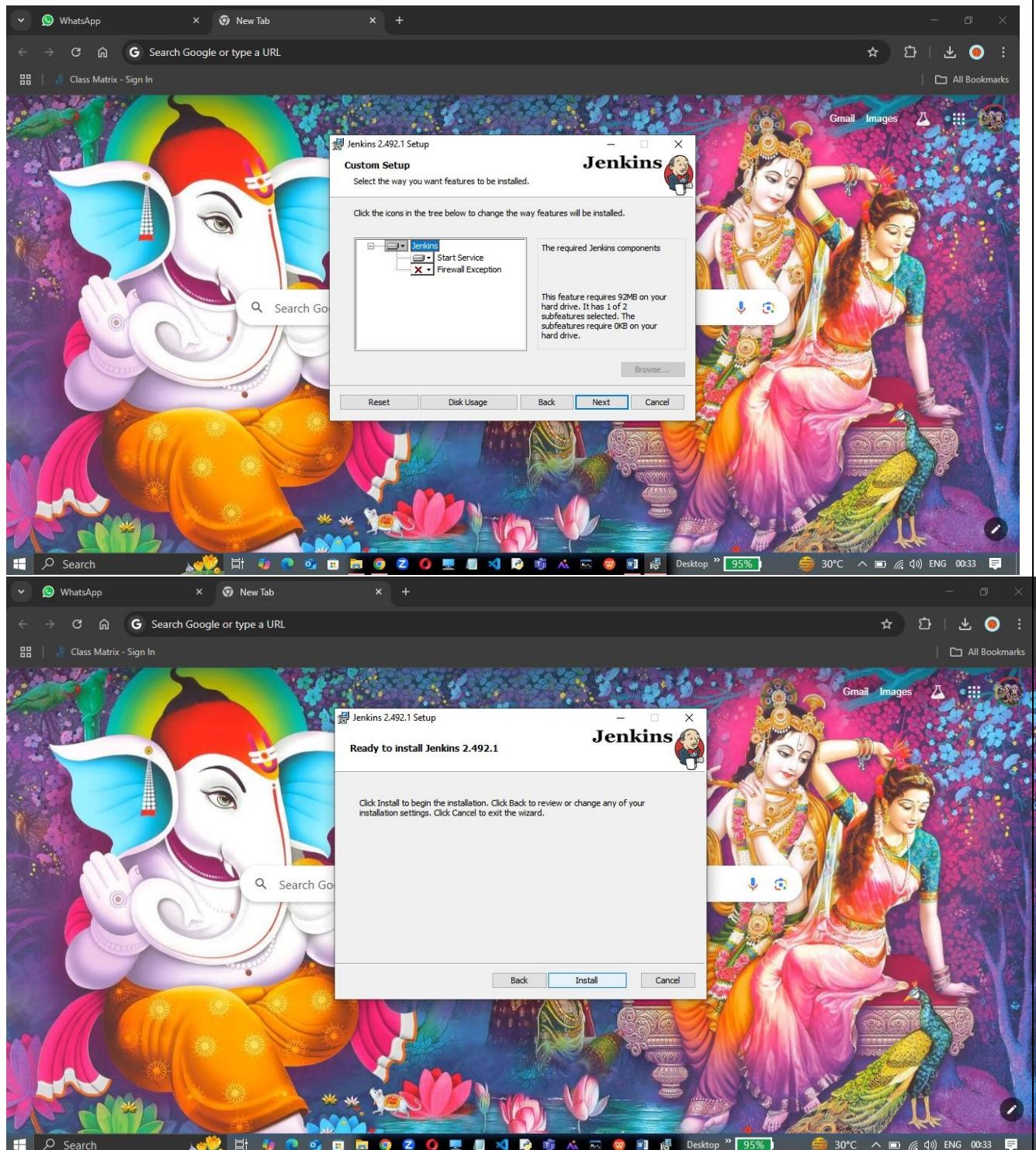


**Proceed to Next**

# 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**

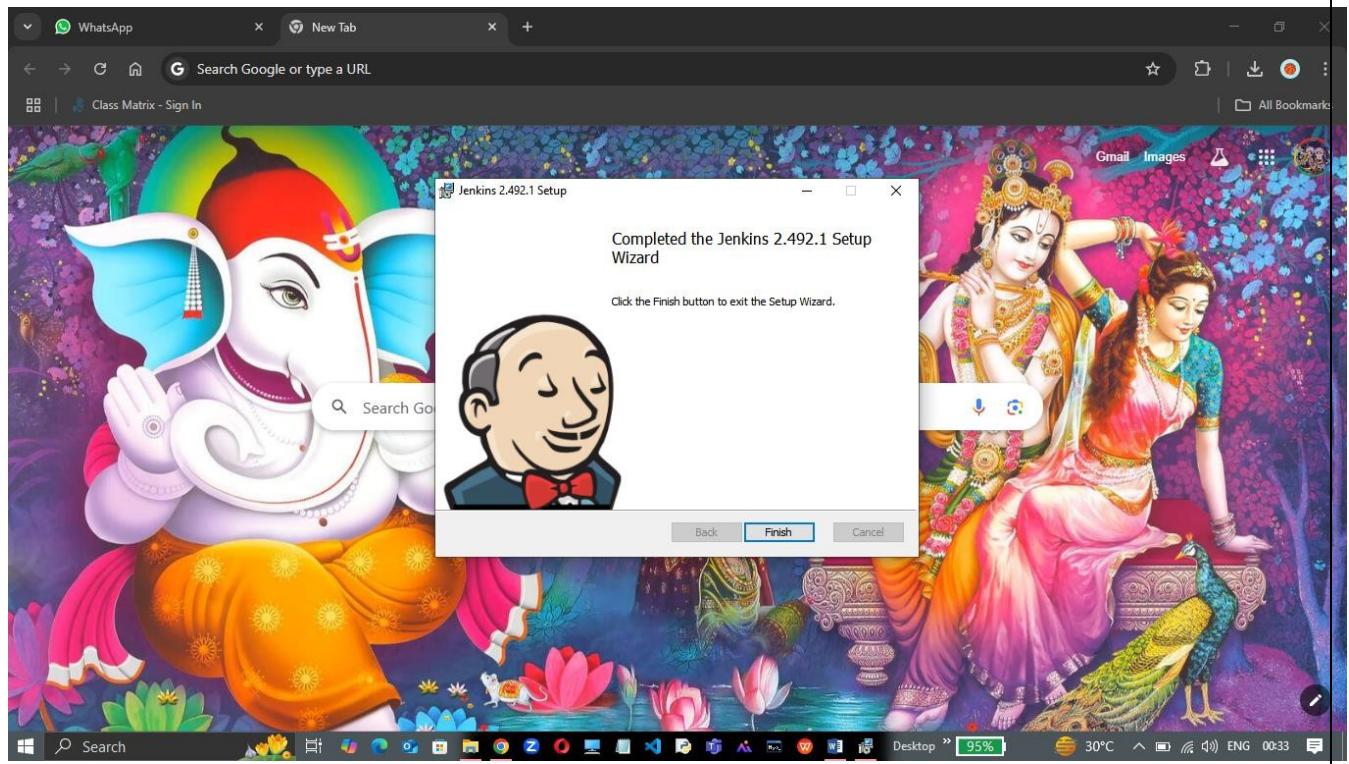


On clicking 'Install', installation is finished.

## **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**



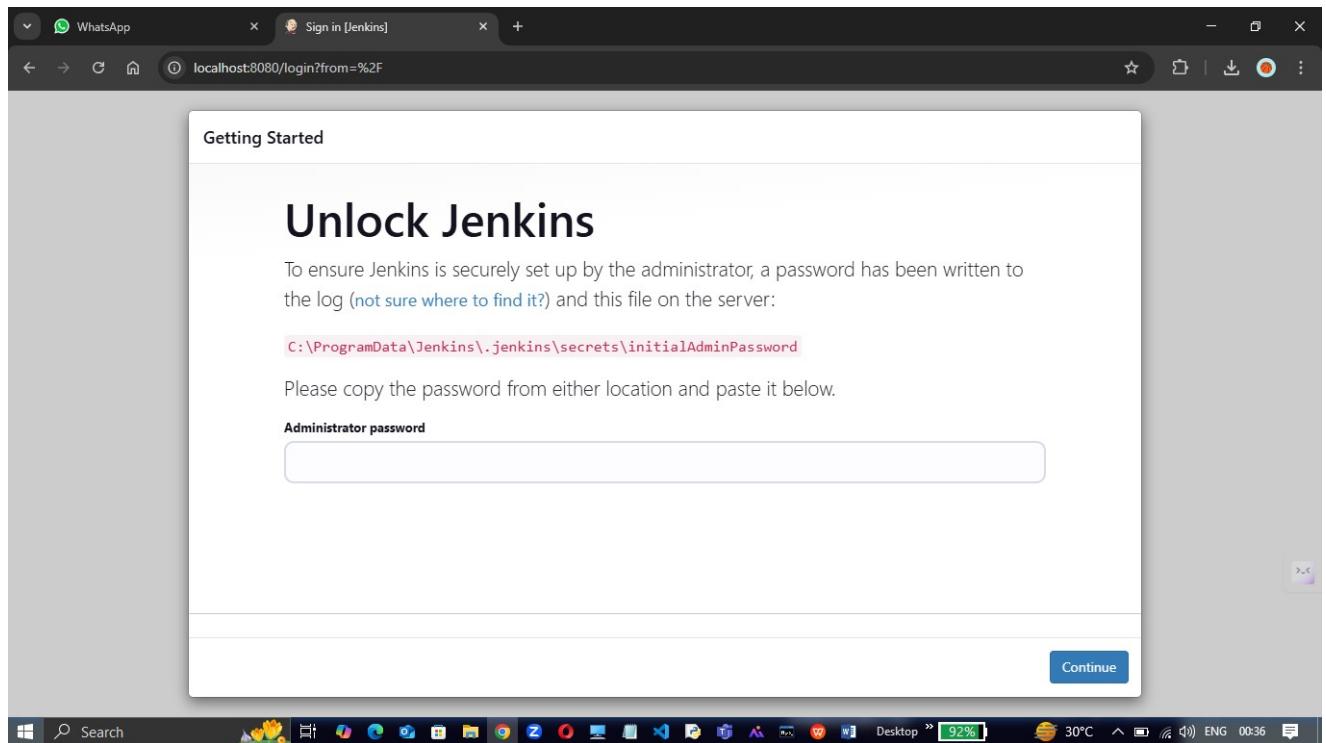
**Step 6:** Once Installation is done, you can test the Jenkins on <http://localhost:8080> on the browser.

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	** bouncycastle API ** Instance Identity ** JavaBeans Activation Framework (JAF) API ** JavaMail API ** Credentials ** Plain Credentials ** Gson API ** Trilead API ** SSH Credentials Credentials Binding ** SCH API ** Pipeline: API ** commons-lang3 v3.x Jenkins API Timestamper ** 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
✓ 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

?

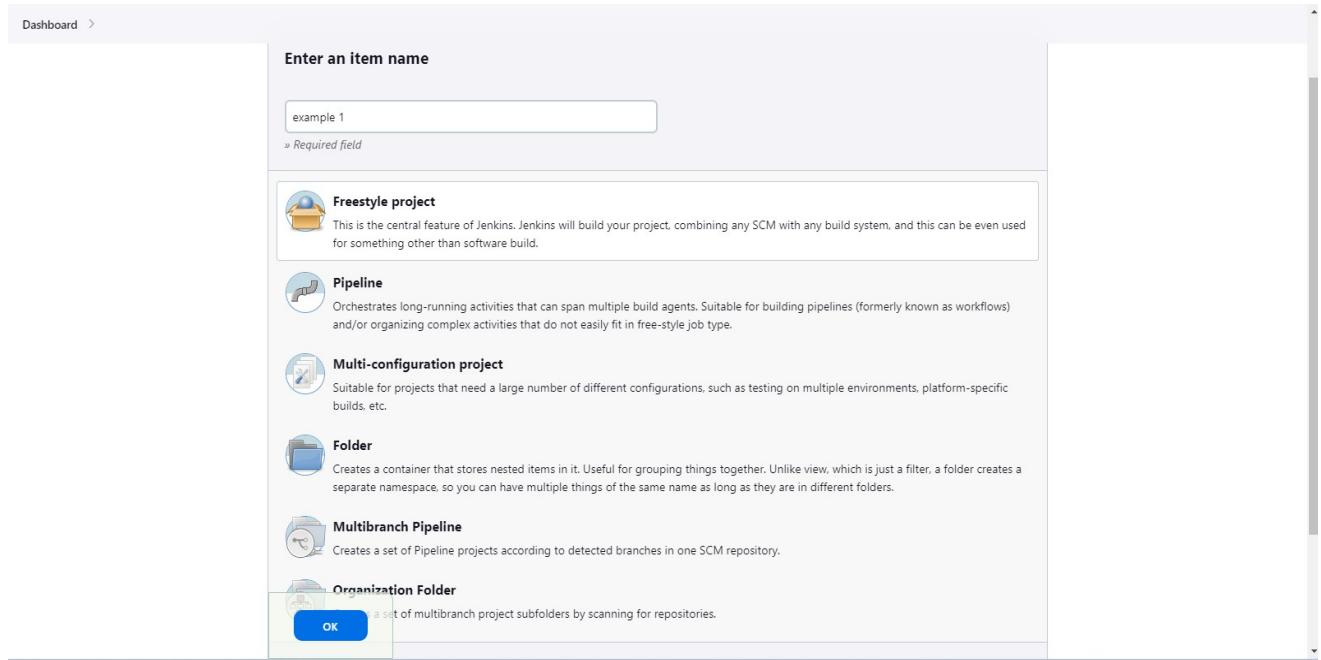
Learn more about distributed builds

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**



The screenshot shows the Jenkins 'New Item' creation interface. At the top, there is a header bar with 'Dashboard >' and a search bar labeled 'Enter an item name' containing the placeholder 'example 1'. Below this, a note says '» Required field'. A list of project types is displayed:

- 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**: A set of multibranch project subfolders by scanning for repositories.

At the bottom right of the list, there is a blue 'OK' button.

# 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 'New Item' creation interface. A red validation error message 'This field cannot be empty, please enter a valid name' is displayed above the 'Enter an item name' input field. A tooltip 'SEPM LAB' is shown above the 'SEPM LAB2' option in the dropdown menu. Below the input field, there are several project types listed:

- 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.

An 'OK' button is located at the bottom of the form.

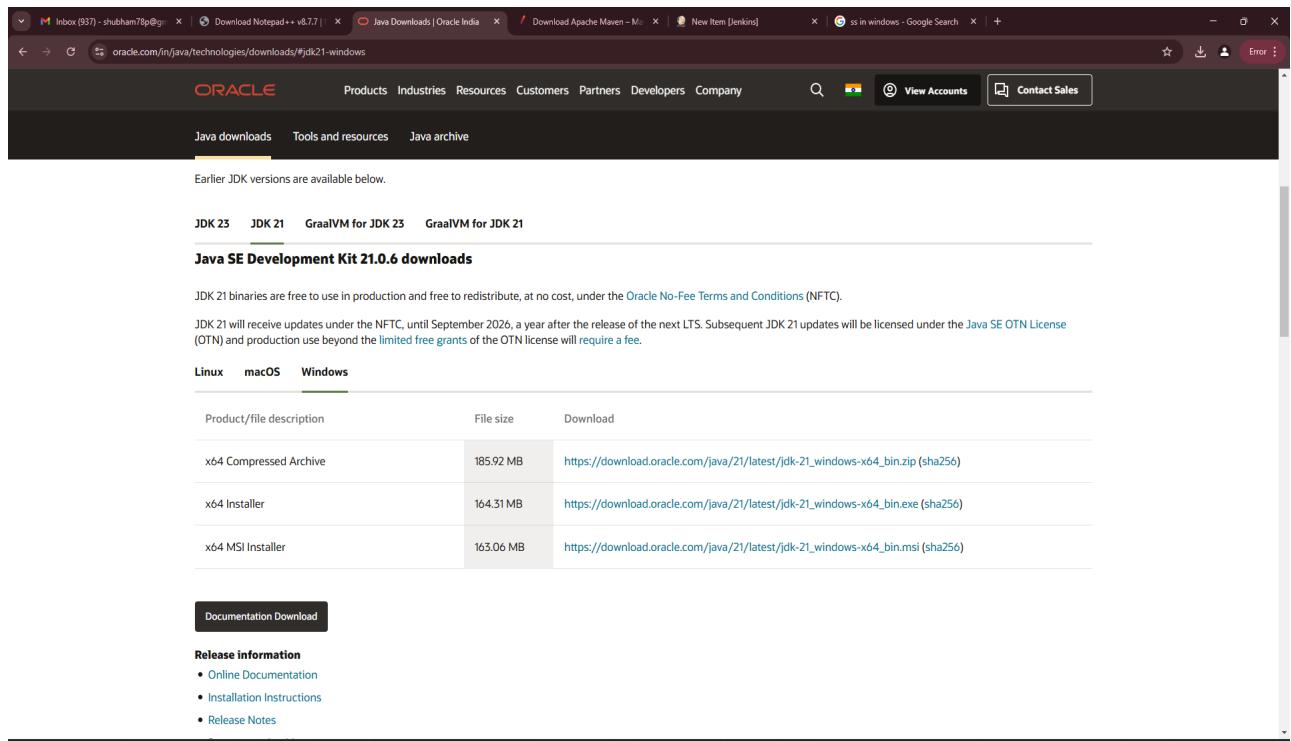
  

This screenshot is identical to the one above, showing the 'New Item' creation interface with the same validation error message and list of project types.

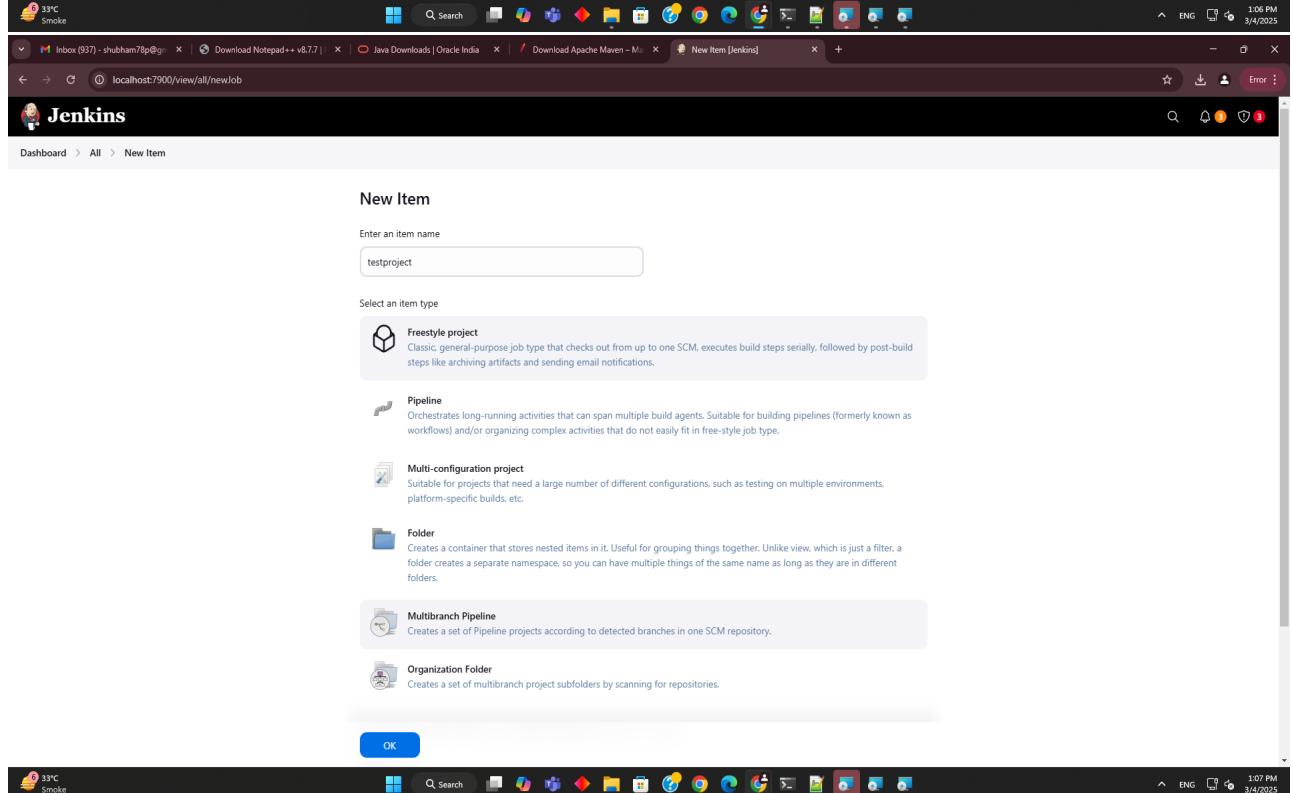
# Software Engineering & Project Management Lab

## 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 Oracle Java Downloads page for JDK 21. It features a navigation bar with links for Products, Industries, Resources, Customers, Partners, Developers, Company, View Accounts, and Contact Sales. Below the navigation bar, there are links for Java downloads, Tools and resources, and Java archive. A note states "Earlier JDK versions are available below." Below this, there are tabs for JDK 23, JDK 21, GraalVM for JDK 23, and GraalVM for JDK 21. The "Java SE Development Kit 21.0.6 downloads" section is highlighted. It notes that JDK 21 binaries are free to use in production and to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions (NFTC). It also mentions that 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. Download links are provided for Linux, macOS, and Windows. The Windows tab is selected, showing three options: x64 Compressed Archive (185.92 MB), x64 Installer (164.31 MB), and x64 MSI Installer (163.06 MB), each with a download link.

The screenshot shows the Jenkins "New Item" configuration dialog. At the top, there is a search bar and a "New Item" button. Below the search bar, it says "Dashboard > All > New Item". The main area is titled "New Item" and has a sub-section "Enter an item name" with a text input field containing "testproject". Below this, there is a "Select an item type" section with several options: "Freestyle project" (selected), "Pipeline", "Multi-configuration project", "Folder", "Multibranch Pipeline", and "Organization Folder". Each option has a brief description. At the bottom of the dialog is an "OK" button.

# 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 General configuration page for a project named 'testproject'. The 'General' tab is selected. The 'Enabled' switch is turned on. In the 'Description' field, the text 'hi i am shubham pandey' is entered. Other settings include 'Discard old builds' (unchecked), 'This project is parameterized' (unchecked), 'Add Parameter' (button), 'Throttle builds' (unchecked), 'Number of builds' (set to 1), 'Time period' (set to 'Second'), and 'Allow user triggered builds to skip the rate limit' (unchecked). Buttons for 'Save' and 'Apply' are at the bottom.

# 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 configuration interface for a job named 'testproject'. The 'Build Steps' section is selected in the sidebar. A new 'Execute shell' step is being added, with the command 'echo "Shubham"' entered into the text area. The Jenkins interface includes a toolbar at the top and a status bar at the bottom indicating the date and time.

# 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 interface for a project named 'testproject'. The dashboard includes a status bar at the top with various links. Below the status bar, there's a header with the Jenkins logo and the project name. A sidebar on the left contains links for Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. The main content area has a 'Status' section with a green icon and the text 'testproject'. It also includes sections for 'Changes' (with a note 'hi i am shubham pandey'), 'Workspace' (with a note 'Permalinks'), and 'Builds' (which shows 'No builds'). There's an 'Edit description' link in the top right corner.

The screenshot shows the Jenkins interface for a project named 'SEPM LAB'. The dashboard includes a status bar at the top with various links. Below the status bar, there's a header with the Jenkins logo and the project name. A sidebar on the left contains links for Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. The main content area has a 'Status' section with a green checkmark icon and the text 'SEPM LAB'. It also includes sections for 'Changes' (with a note 'This is my first jenkins project.'), 'Workspace' (with a note 'Permalinks'), and 'Builds' (which shows a list of four builds: #3 (23 hr ago), #3 (23 hr ago), #3 (23 hr ago), and #3 (23 hr ago)). There's an 'Edit description' link in the top right corner. The status bar at the bottom indicates the date and time as 3/4/2025 and 1:07 PM.

This screenshot is identical to the one above, showing the Jenkins interface for the 'SEPM LAB' project. It displays the same dashboard layout, build history, and status information. The status bar at the bottom indicates the date and time as 3/4/2025 and 1:24 PM.

# 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 three screenshots of a Jenkins interface. The top screenshot displays the 'Configuration' page for a project named 'testproject'. It includes sections for 'General' (with throttle builds set to 1 build every 60 minutes), 'Source Code Management' (set to 'None'), 'Triggers' (with 'Build after other projects are built' checked and a 'Projects to watch' field empty), and 'Advanced' settings. The middle screenshot shows the 'testproject' dashboard, featuring a status bar with '33°C Smoke', a message from 'hi i am shubham pandey', and a 'Permalinks' section. The bottom screenshot shows the Jenkins header bar with various icons and the text 'REST API Jenkins 2.49.2'.

**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**

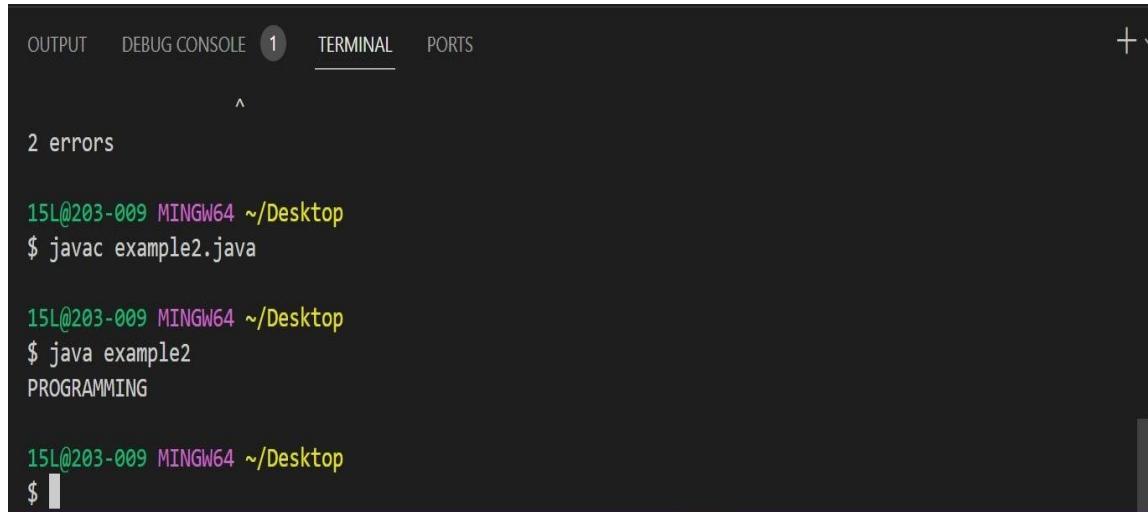
The screenshot shows a Jenkins console output page for a project named 'testproject'. The top navigation bar includes links for Gmail, Notepad+, Java Downloads, Apache Maven, testproject #1 Console, and an inbox. The main header has a Jenkins logo and the title 'Console Output'. The left sidebar lists 'Status', 'Changes', 'Console Output' (which is selected), 'Edit Build Information', and 'Delete build #1'. The right sidebar contains download, copy, and plain text options. The central content area displays the build log:

```
Started by user unknown or anonymous  
Running as SYSTEM  
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\testproject  
[testproject] $ cmd /c call C:\WINDOWS\TEMP\jenkins585211950519870491.bat  
  
C:\ProgramData\Jenkins\.jenkins\workspace\testproject>echo "shubham pandey"  
"shubham pandey"  
  
C:\ProgramData\Jenkins\.jenkins\workspace\testproject>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**

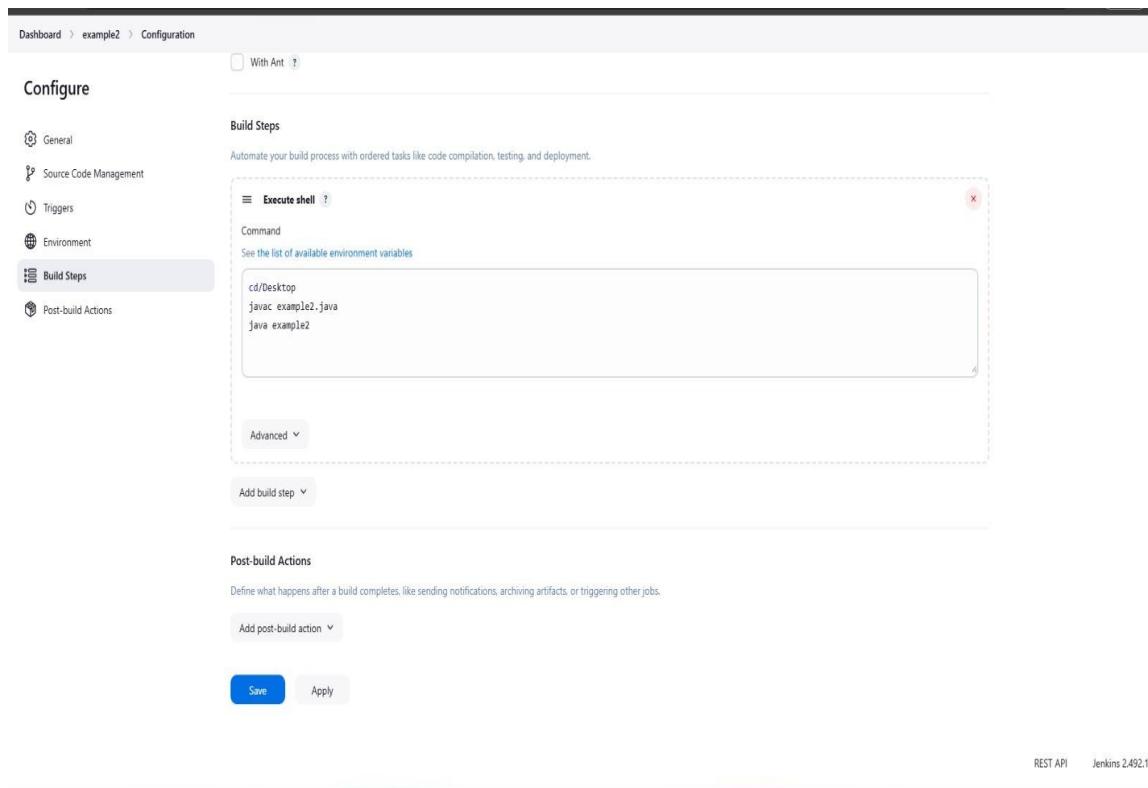


The screenshot shows a terminal window with the following output:

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

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

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



The screenshot shows the Jenkins job configuration page for 'example2'. The 'Build Steps' section is expanded, showing an 'Execute shell' step with the following commands:

```
cd/Desktop
javac example2.java
java example2
```

# 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 interface for a build named 'test1'. The build number is '#5'. The 'Console Output' tab is selected. The output window displays the following command-line session:

```
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:\>javac example2.java

C:\>java example2
PROGRAMMING

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

The screenshot shows the Jenkins interface for a build named 'test1'. The build number is '#4'. The 'Console Output' tab is selected. The output window displays the following command-line session:

```
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:\>echo "Your Name is 3"
"Your Name is 3"

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

The screenshot shows the Jenkins interface for a build named 'test1'. The build number is '#3'. The 'Console Output' tab is selected. The output window displays the following command-line session:

```
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

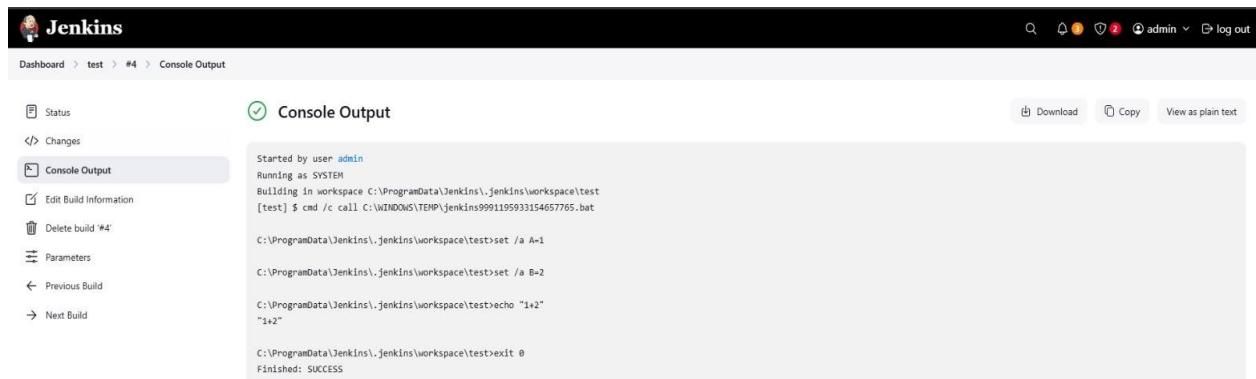
C:\>echo "Your Name is 12+34"
"Your Name is 12+34"

C:\>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 console output for build #4. The log starts with the build being started by user 'admin' as 'SYSTEM'. It shows the workspace path and a command being run to calculate the sum of 1 and 2. The final message indicates the build was successful.

```
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 console output for build #3. The log starts with the build being started by user 'admin' as 'SYSTEM'. It shows the workspace path and a command being run to echo 'ABC and DEF'. The final message indicates the build was successful.

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\test
[test] $ cmd /c call C:\WINDOWS\TEMP\jenkins2368247137534955462.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.