Experiment No. 1.5

Student Name: Parvinder Singh   **UID:** 22MCC20043
Branch: MCA - CCD   **Section/Group:** 22MCD-1/ Grp B
Semester: III
Subject Name: DevOps Process Automation Lab   **Subject Code:** 22CAP-745

1. **Aim/Overview of the practical:**

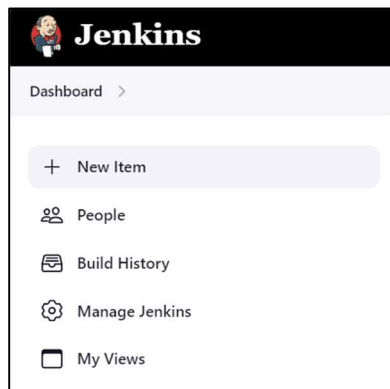   **Create a Jenkins Freestyle Project which consists of Maven Build jobs.**
   Note: Trigger the builds of all the Maven 9 Goals.

2. **Code for experiment/practical:**

Jenkins freestyle projects allow users to automate simple jobs, such as running tests, creating and packaging applications, producing reports, or executing commands.

Step 1: How to Set up a Build Job in Jenkins

- Click the New Item link on the left-hand side of the Jenkins dashboard.

- Enter the new project's name in the Enter an item name field and select the Freestyle project type. Click OK to continue.

- Under the General tab, add a project description in the Description field.

General   Enabled ⊘

Description

This job is created to execute mvn clean command.

Step 2: How to Setup Source code management

- Click on Source code management.
- Select Git. Provide Git repository link, Branch to build.

Source Code Management

○ None

● Git ?

Repositories ?

Repository URL ?   ✕

https://github.com/parvinder20043/Teamcity.git

Step 3: Add Build trigger

Note: Make sure you have one job available previously for using this feature.

- Click on build trigger.
- Select Build after other projects are built.
- You can add additional properties according to project requirement

Step 4: Add a Build Step

- Scroll down to the Build section.
- Open the Add build step drop-down menu and select Invoke top-level Maven targets.
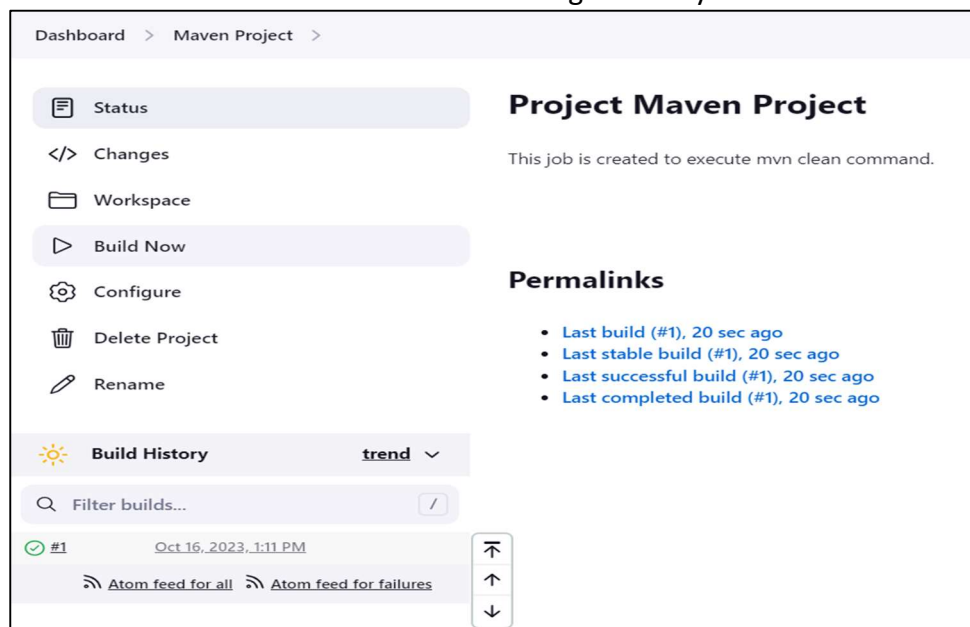- Provide Maven version and Goals to execute.

Build Steps

≡ **Invoke top-level Maven targets** ?   ✕

Maven Version

M1

Goals

clean   ▾

Advanced ⌄

Add build step ▾

Step 5: Save and apply

- Click the Save button to save changes to the project.

Save   Apply

Step 6: Build the Project

- Click the Build Now link on the left-hand side of the new project page.
- Click the link to the latest project build in the Build History section.
- Click the Console Output link on the left-hand side to display the output for the commands you entered.
- The console output indicates that Jenkins is successfully executing the commands, displaying the current version of Java and Jenkins working directory.

Dashboard > Maven Project >

📄 Status	**Project Maven Project**
</> Changes	This job is created to execute mvn clean command.
🗀 Workspace	
▷ Build Now	
⚙ Configure	**Permalinks**
🗑 Delete Project	• Last build (#1), 20 sec ago
✏ Rename	• Last stable build (#1), 20 sec ago
	• Last successful build (#1), 20 sec ago
	• Last completed build (#1), 20 sec ago

☼ **Build History**   trend ⌄

🔍 Filter builds...   /

✅ #1   Oct 16, 2023, 1:11 PM

🔊 Atom feed for all   🔊 Atom feed for failures

Trigger the builds of all the Maven 9 Goals

- Repeat all steps to create maven 9 Goal jobs except Step 3.
- In Step 3 you have to provide different Build trigger to execute one another.
  For example: After mvn clean -> mvn validate.

mvn clean

Build Triggers

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

☐ Build after other projects are built ?

☐ Build periodically ?

☐ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

→

mvn validate

Build Triggers

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

☑ Build after other projects are built ?

Projects to watch

Maven Project

Maven Project

🔘 Trigger only if build is stable

⚪ Trigger even if the build is unstable

- Repeat this build trigger step in every job one another one.

Clean, Validate, Compile, Test, Package, Integration-Test, Verify, Install, Deploy.

- After Creating and building all these 9 jobs you have successfully implemented freestyle CI/CD pipeline