<table>
<tr><td colspan="3">Name of Student: Pushkar Sane</td></tr>
<tr><td colspan="2">Roll Number: 45</td><td>Lab Assignment Number: 5</td></tr>
<tr><td colspan="3">Title of Lab Assignment: To implement Jenkins pipeline using scripted/ declarative pipeline.</td></tr>
<tr><td colspan="2">DOP: 21-02-2024</td><td>DOS: 23-02-2024</td></tr>
<tr><td>CO Mapped:
CO3</td><td>PO Mapped:
PO2, PO3, PO5, PSO1, PSO2</td><td>Signature:</td></tr>
</table>

Practical No. 5

Aim: To implement Jenkins pipeline using scripted/ declarative pipeline.

Steps to implement Jenkins pipeline using declarative pipeline:

Introduction:

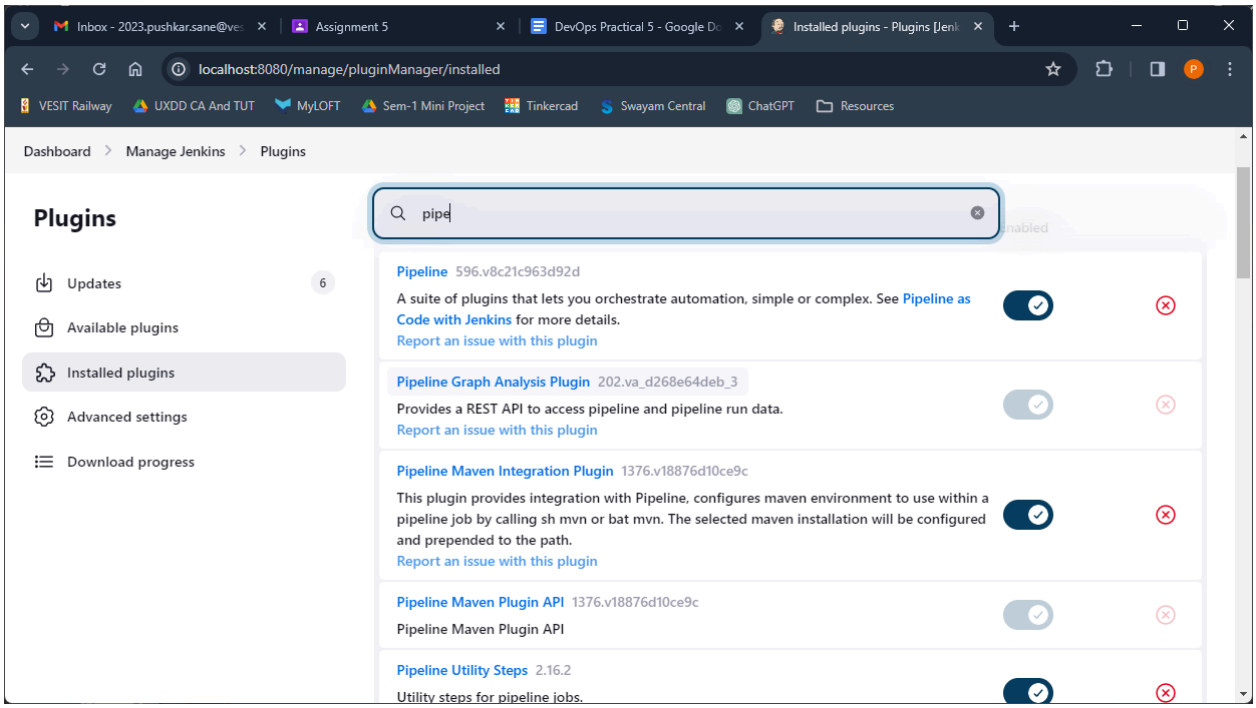
Jenkins pipeline is a powerful tool for automating software delivery pipelines. It allows users to define their entire software delivery process as code, enabling continuous integration and continuous delivery (CI/CD). This detailed guide aims to walk users through the process of implementing Jenkins pipelines using both scripted and declarative syntaxes.

Step 1: Setting up Jenkins:

1. Install Jenkins on your system by downloading the latest version from the official Jenkins website or using a package manager.
2. Access the Jenkins dashboard by navigating to http://localhost:8080 (or the appropriate URL if Jenkins is hosted elsewhere).
3. Follow the instructions to complete the initial setup, including creating an admin user and installing recommended plugins.

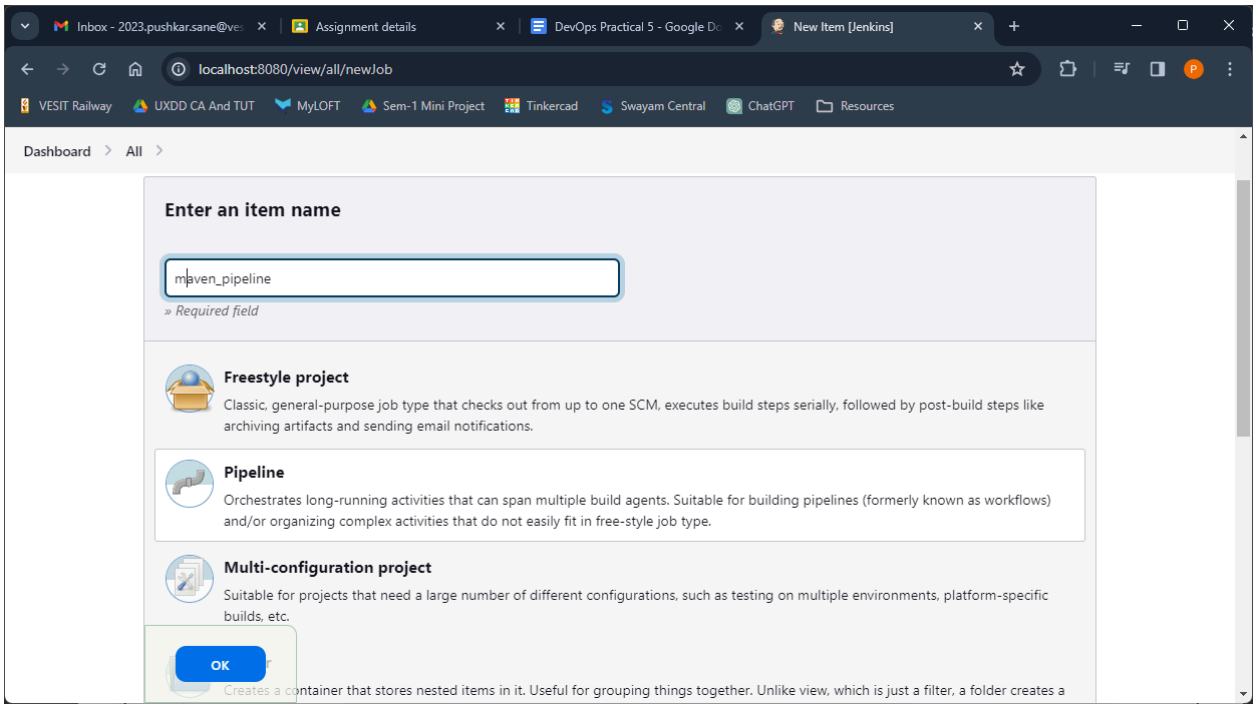
Step 2: Installing Pipeline Plugins:

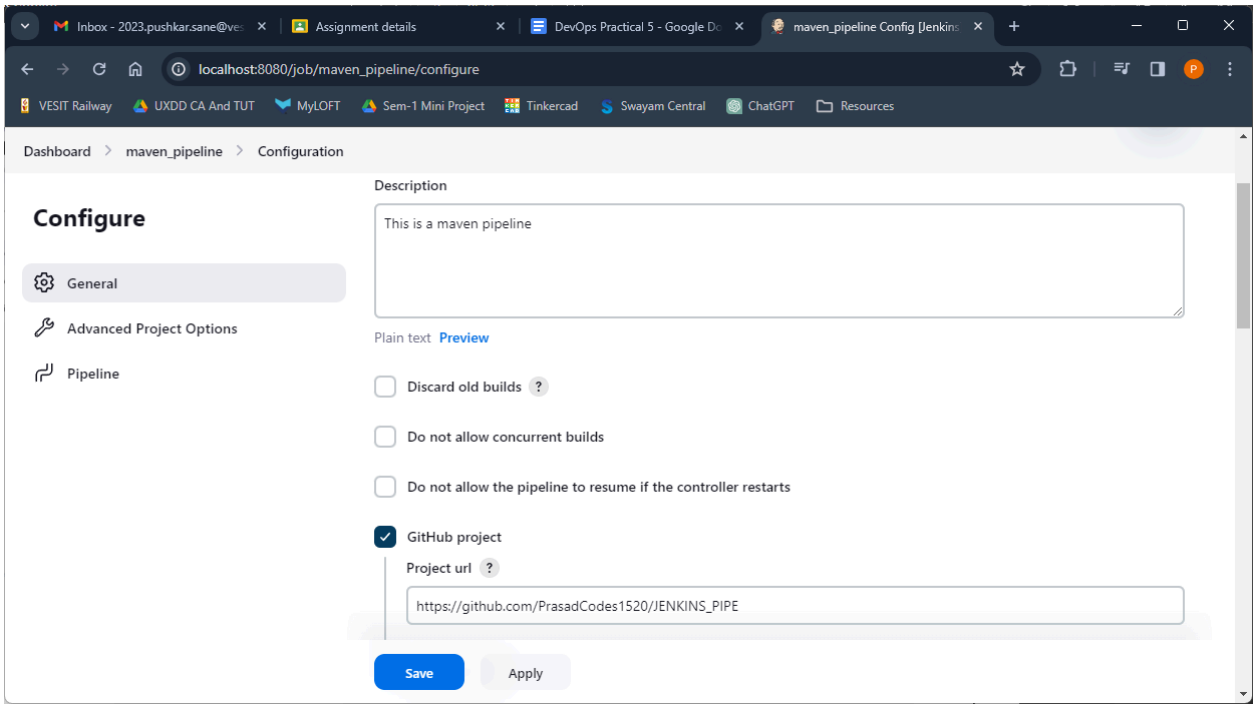
1. Navigate to the Jenkins dashboard and click on "Manage Jenkins" in the left sidebar.
2. Select "Manage Plugins" from the dropdown menu.
3. Go to the "Available" tab and search for "Pipeline" plugins.
4. Install the following plugins:
    - Pipeline
    - Pipeline: GitHub Integration
    - Pipeline Maven Integration
    - Pipeline Utility
    - Maven integration

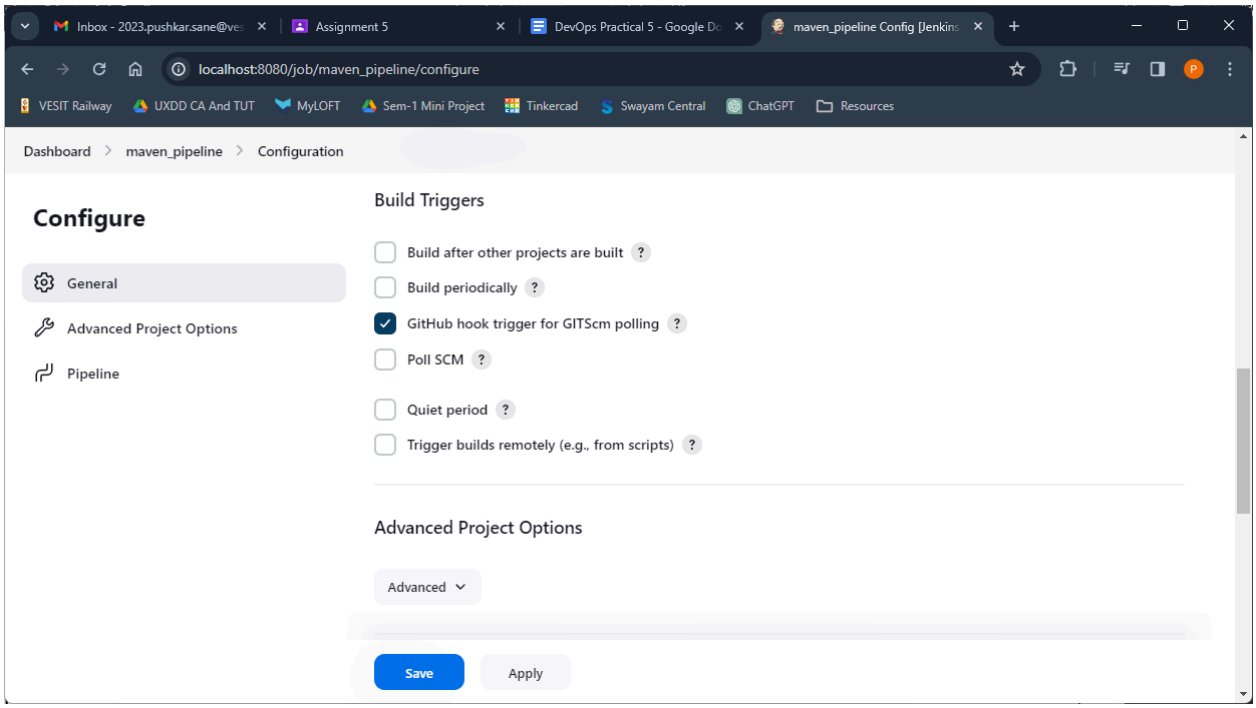


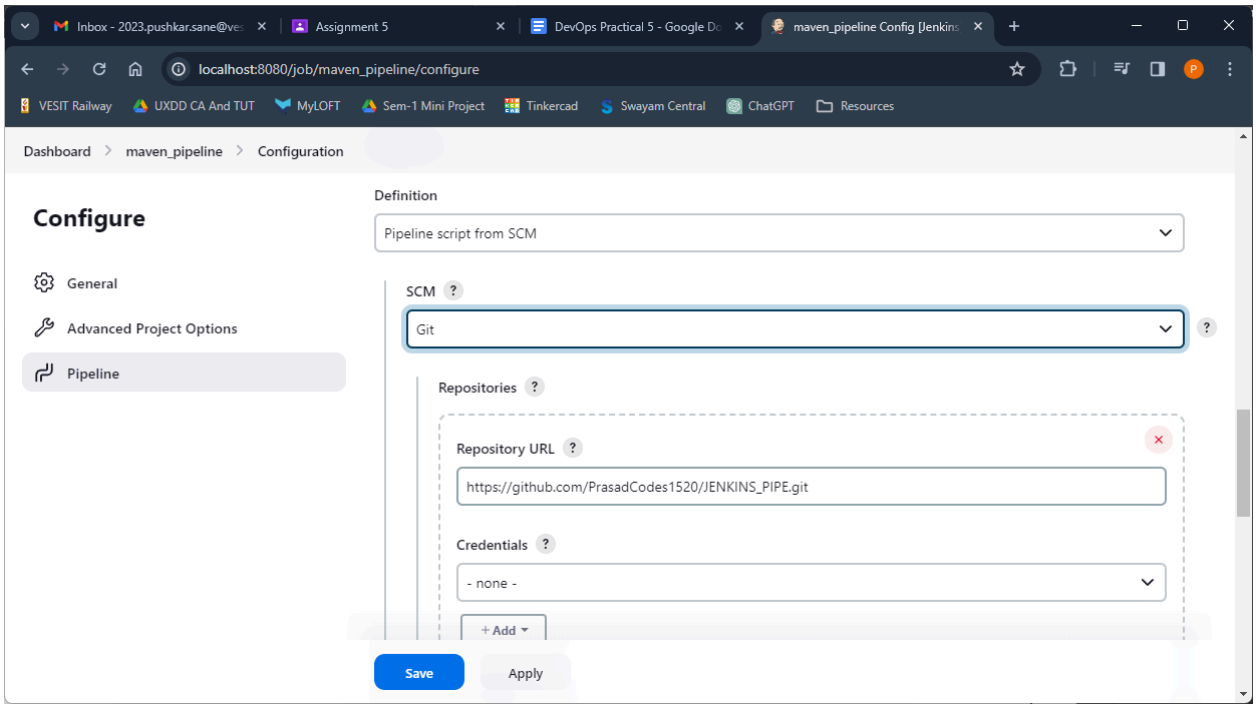
Step 3: Creating a New Pipeline Job and Configuration:

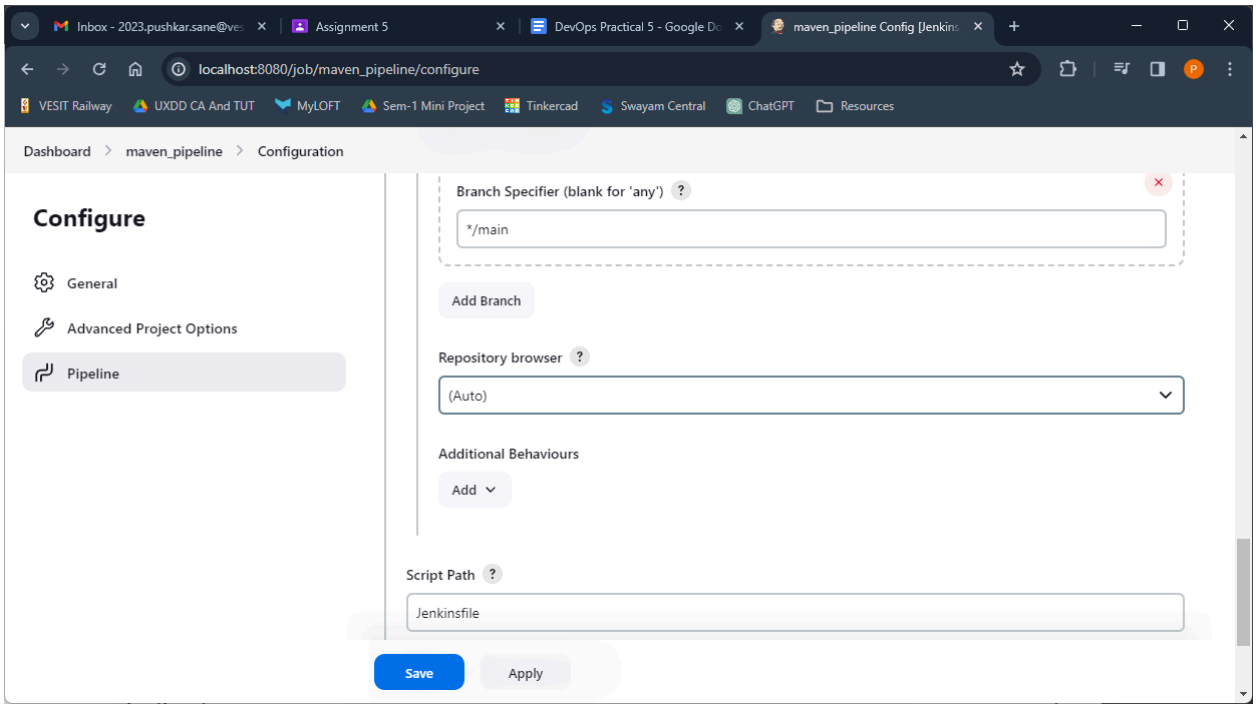
1. Click on "New Item" in the Jenkins dashboard.
2. Enter a name for your pipeline job and select "Pipeline" as the job type.
3. Click on "OK" to create the job.




Inbox - 2023.pushkar.sane@ves
Assignment details
DevOps Practical 5 - Google Do
maven_pipeline Config [Jenkins
localhost:8080/job/maven_pipeline/configure
VESIT Railway
UXDD CA And TUT
MyLOFT
Sem-1 Mini Project
Tinkercad
Swayam Central
ChatGPT
Resources
Dashboard > maven_pipeline > Configuration
Configure
General
Advanced Project Options
Pipeline
Description
This is a maven pipeline
Plain text Preview
Discard old builds
Do not allow concurrent builds
Do not allow the pipeline to resume if the controller restarts
GitHub project
Project url
https://github.com/PrasadCodes1520/JENKINS_PIPE
Save
Apply


Inbox - 2023.pushkar.sane@ves
Assignment 5
DevOps Practical 5 - Google Do
maven_pipeline Config [Jenkins
localhost:8080/job/maven_pipeline/configure
VESIT Railway
UXDD CA And TUT
MyLOFT
Sem-1 Mini Project
Tinkercad
Swayam Central
ChatGPT
Resources
Dashboard > maven_pipeline > Configuration
Configure
General
Advanced Project Options
Pipeline
Build Triggers
Build after other projects are built
Build periodically
GitHub hook trigger for GITScm polling
Poll SCM
Quiet period
Trigger builds remotely (e.g., from scripts)
Advanced Project Options
Advanced
Save
Apply





Step 4: Writing a Scripted Pipeline:

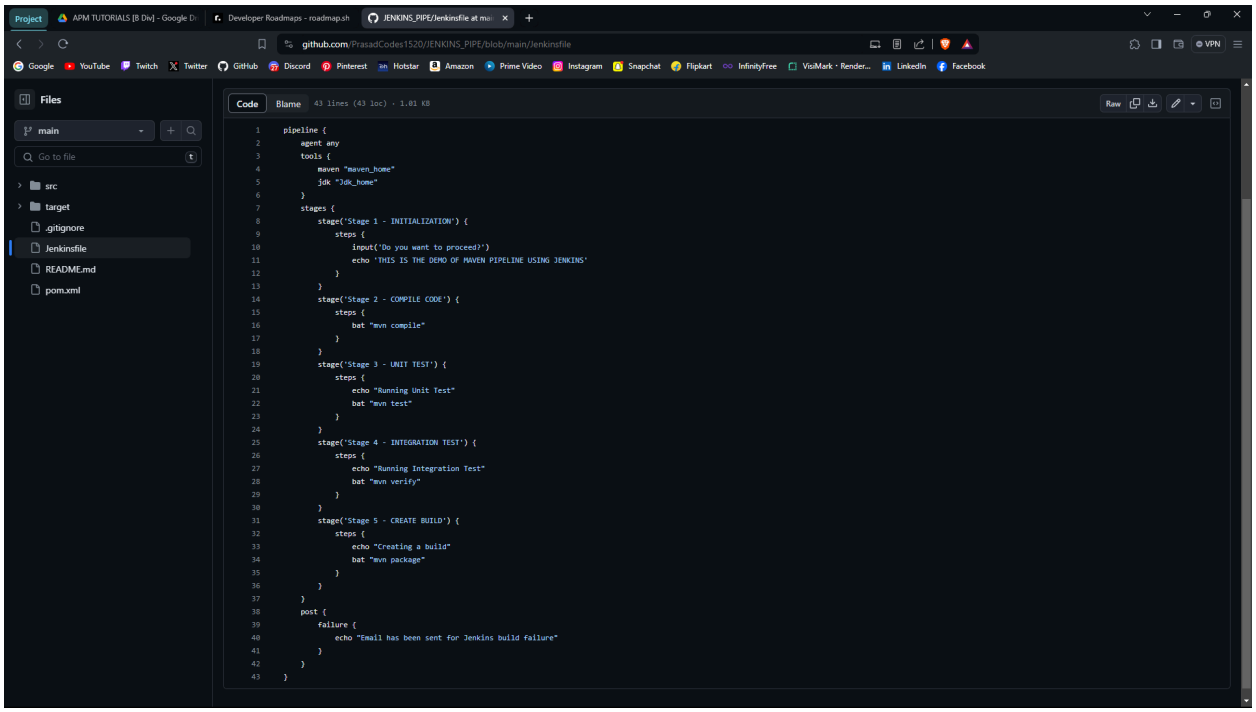
1. In the pipeline job configuration page, scroll down to the "Pipeline" section.
2. Select "Pipeline script" from the definition dropdown.
3. Write your scripted pipeline code in the Script box.

4. Example of scripted pipeline:

```
pipeline {
    agent any
    stages {
        stage('Build') {
            steps {
                echo 'Building the project...'
            }
        }
        stage('Test') {
            steps {
                echo 'Running tests...'
            }
        }
        stage('Deploy') {
            steps {
                echo 'Deploying to production...'
            }
        }
    }
}
```

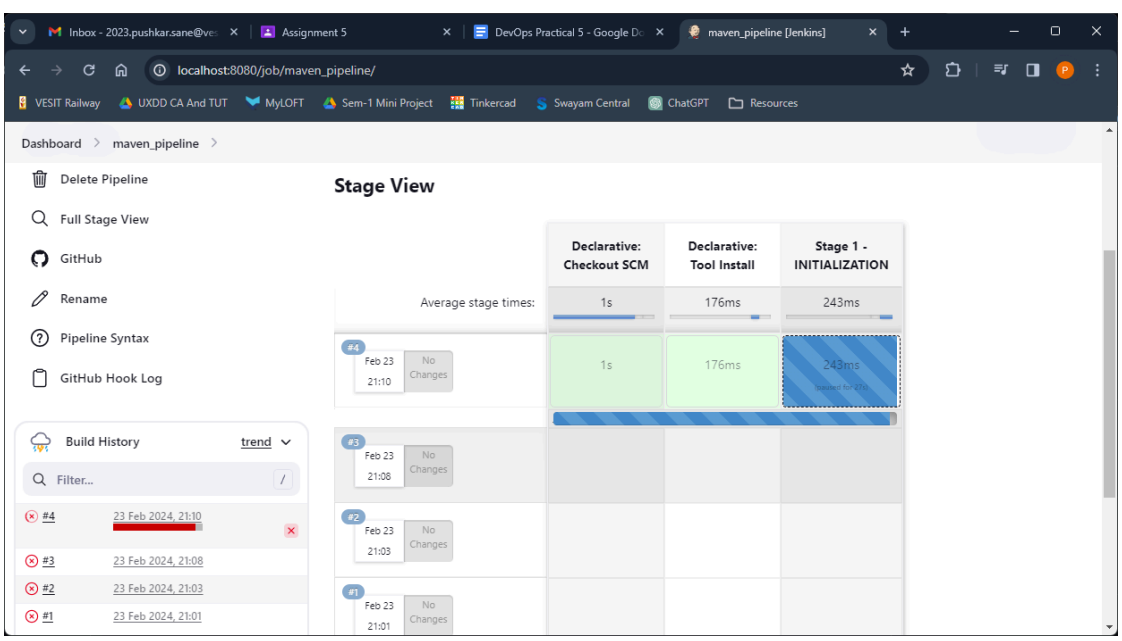
Step 5: Writing a Declarative Pipeline:

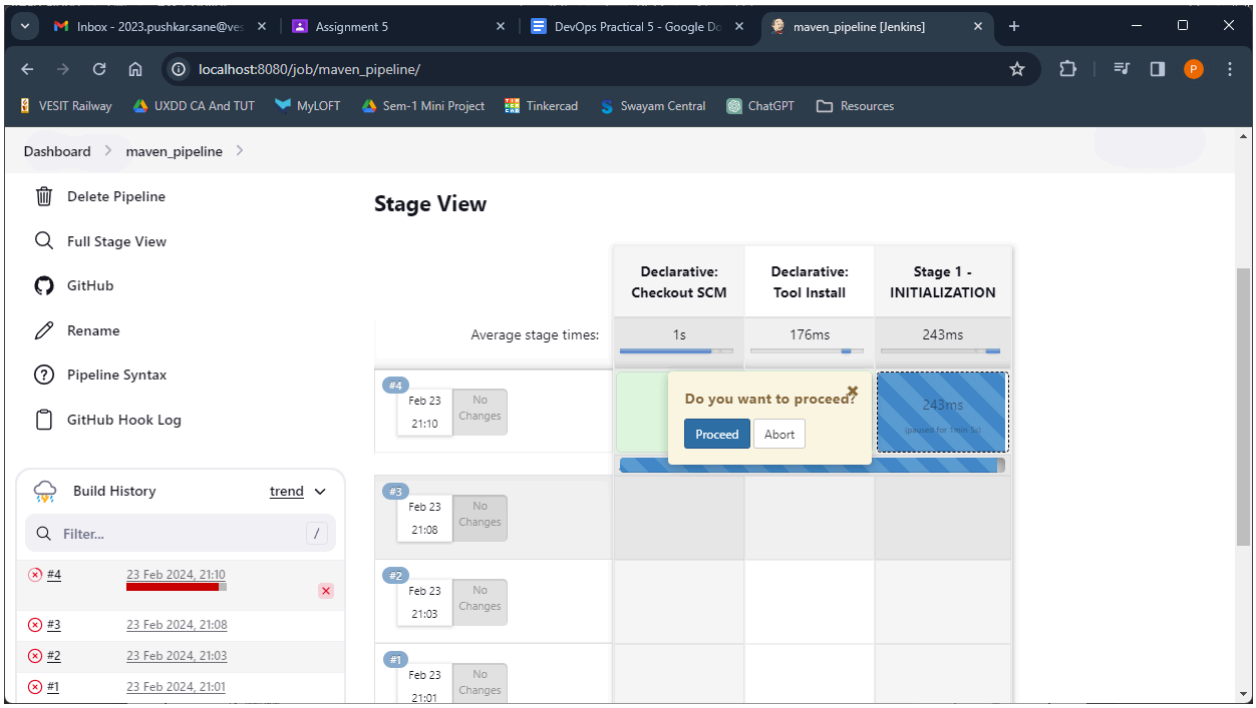
1. In the pipeline job configuration page, select "Pipeline script from SCM" from the definition dropdown.
2. Choose your SCM repository (e.g., Git) and provide the necessary repository URL and credentials.
3. Create a Jenkinsfile in your repository with the declarative pipeline syntax.
4. For example:

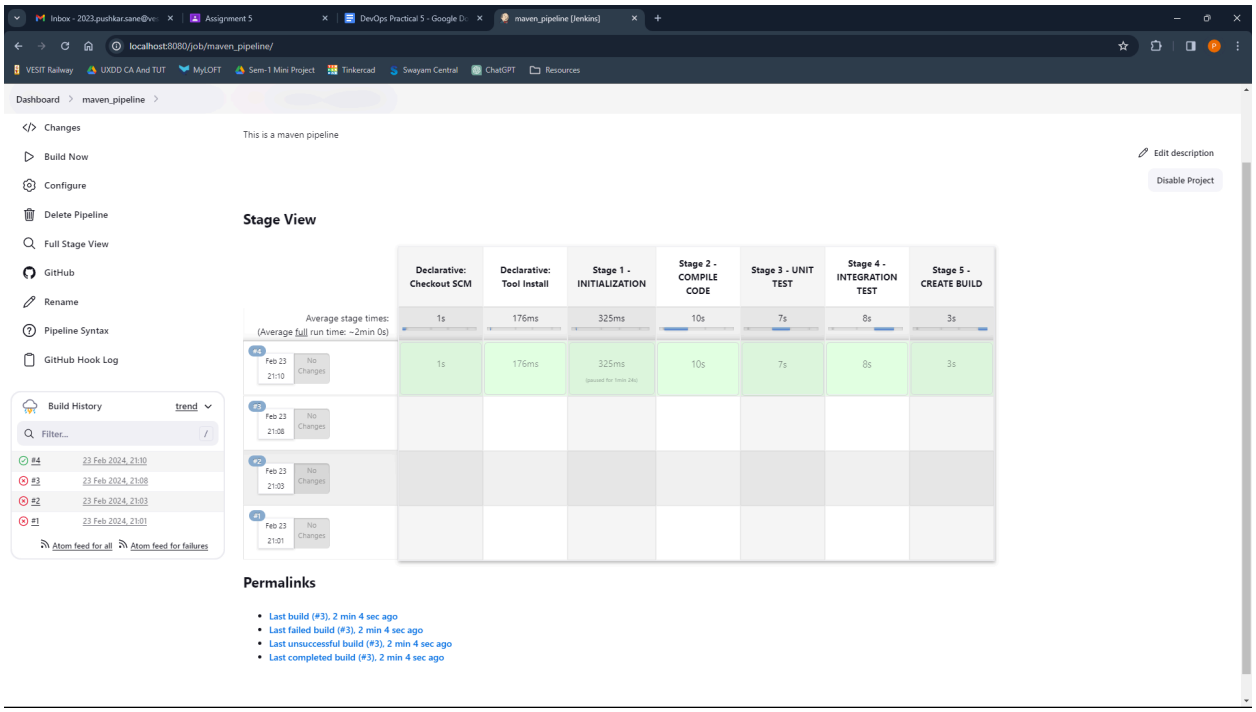


Step 6: Running the Pipeline:

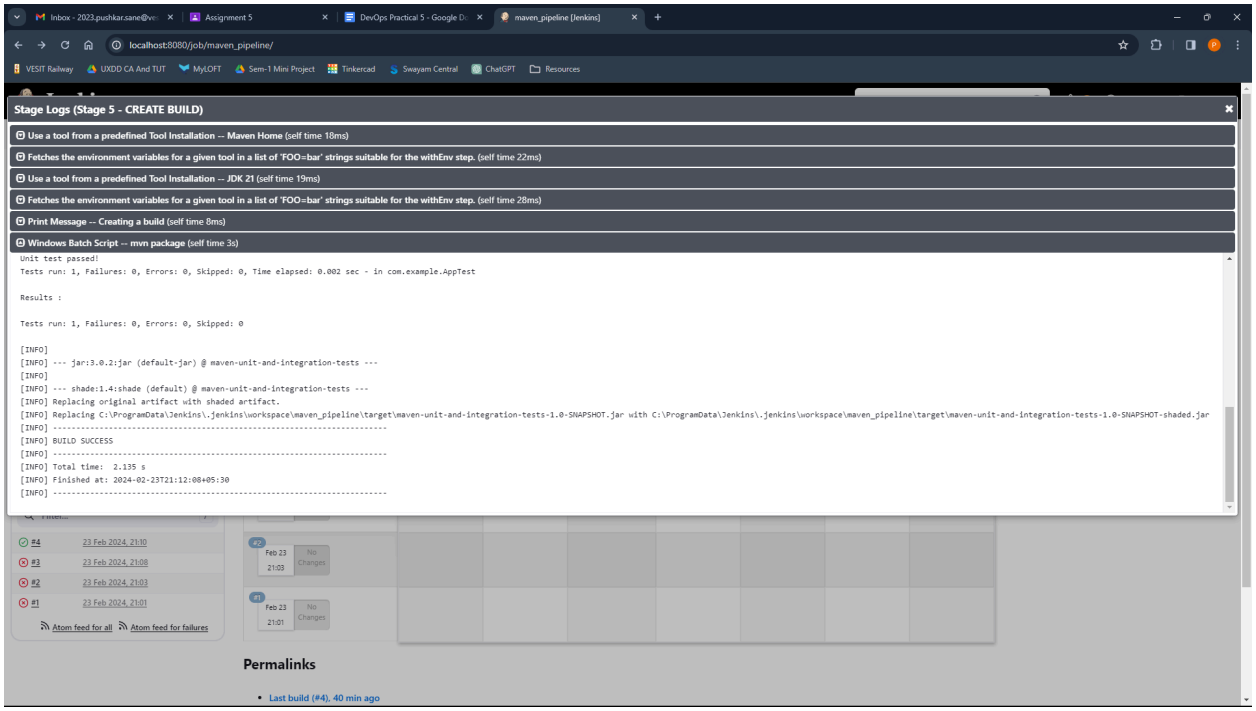
1. Save your pipeline job configuration.
2. Trigger a build for the pipeline job.
3. Monitor the pipeline execution in the Jenkins dashboard.
4. View the console output and logs to track the progress of the pipeline stages.



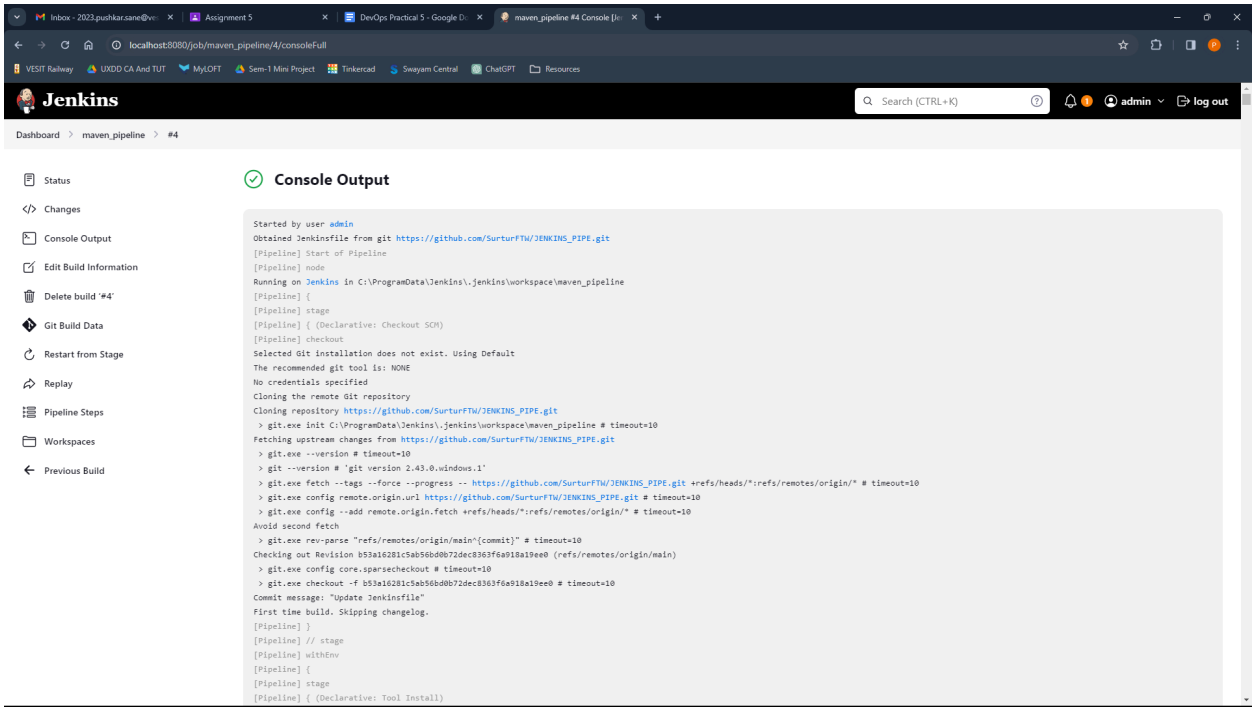

Inbox - 2023.pushkar.sane@ves
Assignment 5
DevOps Practical 5 - Google Do
maven_pipeline [Jenkins]
localhost:8080/job/maven_pipeline/
VESIT Railway
UXDD CA And TUT
MyLOFT
Sem-1 Mini Project
Tinkercad
Swayam Central
ChatGPT
Resources
Dashboard
maven_pipeline
Delete Pipeline
Full Stage View
GitHub
Rename
Pipeline Syntax
GitHub Hook Log
Stage View
Declarative: Checkout SCM
Declarative: Tool Install
Stage 1 - INITIALIZATION
Average stage times:
1s
176ms
243ms
Do you want to proceed?
Proceed
Abort
243ms
Build History
trend
Filter...
#4 23 Feb 2024, 21:10
#3 23 Feb 2024, 21:08
#2 23 Feb 2024, 21:03
#1 23 Feb 2024, 21:01


Dashboard
maven_pipeline
Changes
Build Now
Configure
Delete Pipeline
Full Stage View
GitHub
Rename
Pipeline Syntax
GitHub Hook Log
This is a maven pipeline
Edit description
Disable Project
Stage View
Declarative: Checkout SCM
Declarative: Tool Install
Stage 1 - INITIALIZATION
Stage 2 - COMPILE CODE
Stage 3 - UNIT TEST
Stage 4 - INTEGRATION TEST
Stage 5 - CREATE BUILD
Average stage times:
(Average full run time: ~2min 0s)
1s
176ms
325ms
10s
7s
8s
3s
Build History
trend
Filter...
#4 23 Feb 2024, 21:10
#3 23 Feb 2024, 21:08
#2 23 Feb 2024, 21:03
#1 23 Feb 2024, 21:01
Atom feed for all
Atom feed for failures
Permalinks
Last build (#3), 2 min 4 sec ago
Last failed build (#3), 2 min 4 sec ago
Last unsuccessful build (#3), 2 min 4 sec ago
Last completed build (#3), 2 min 4 sec ago

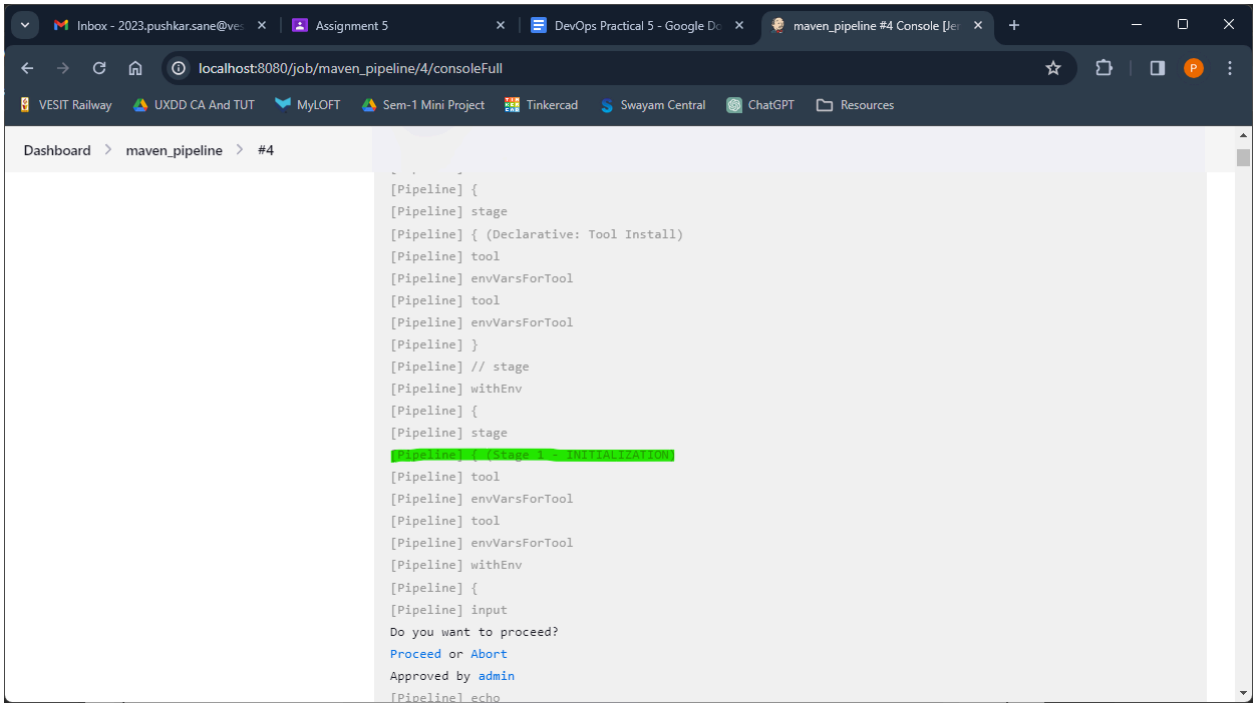
Click on the Logs to check the pipeline logs if there are any failures



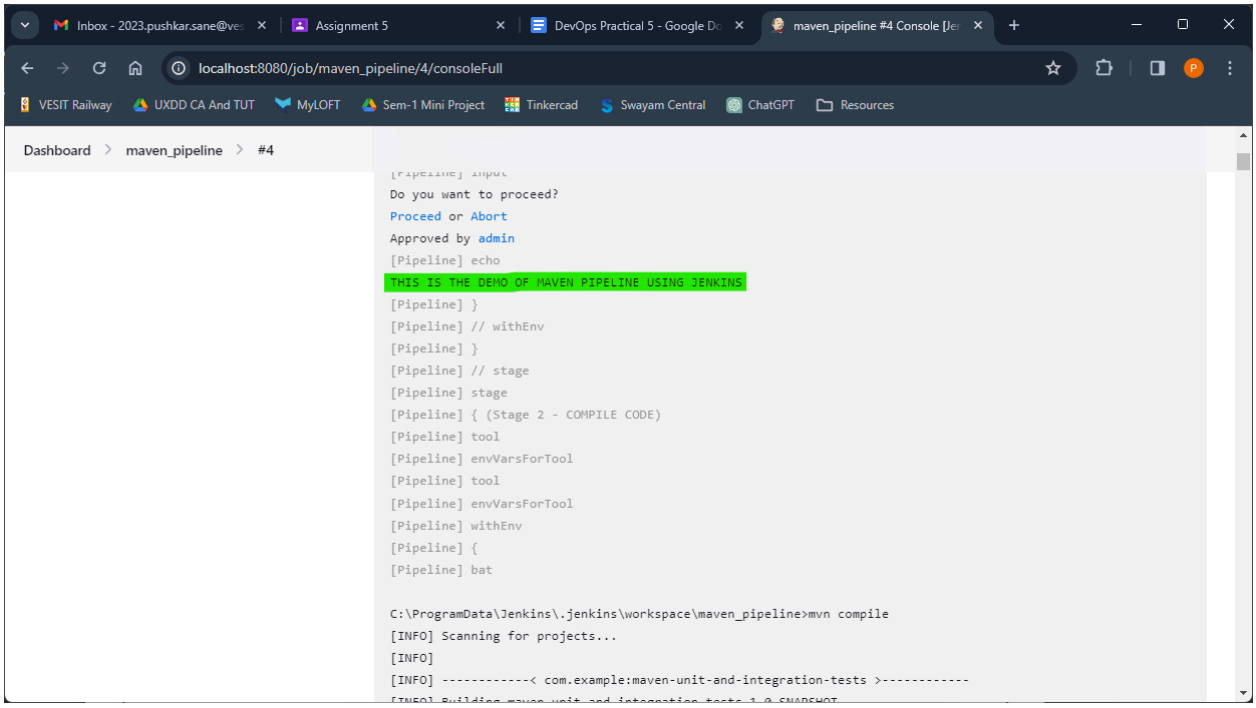
Console Output:



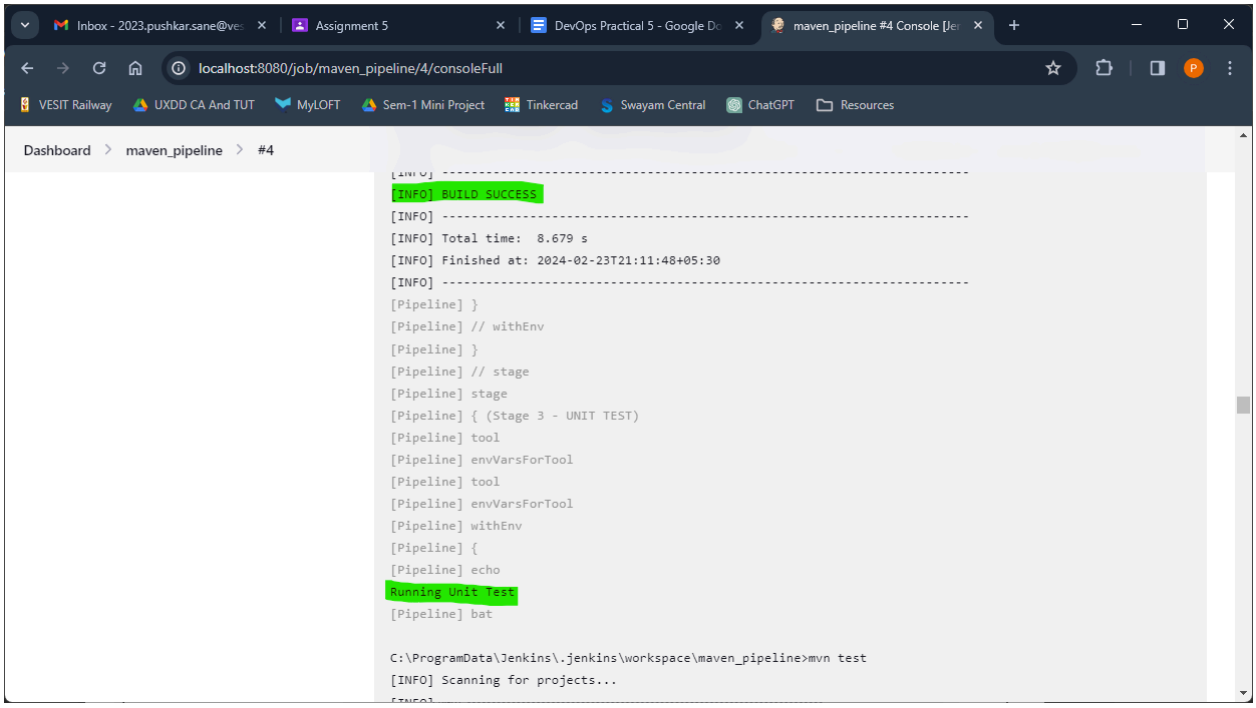
Initialization Stage:

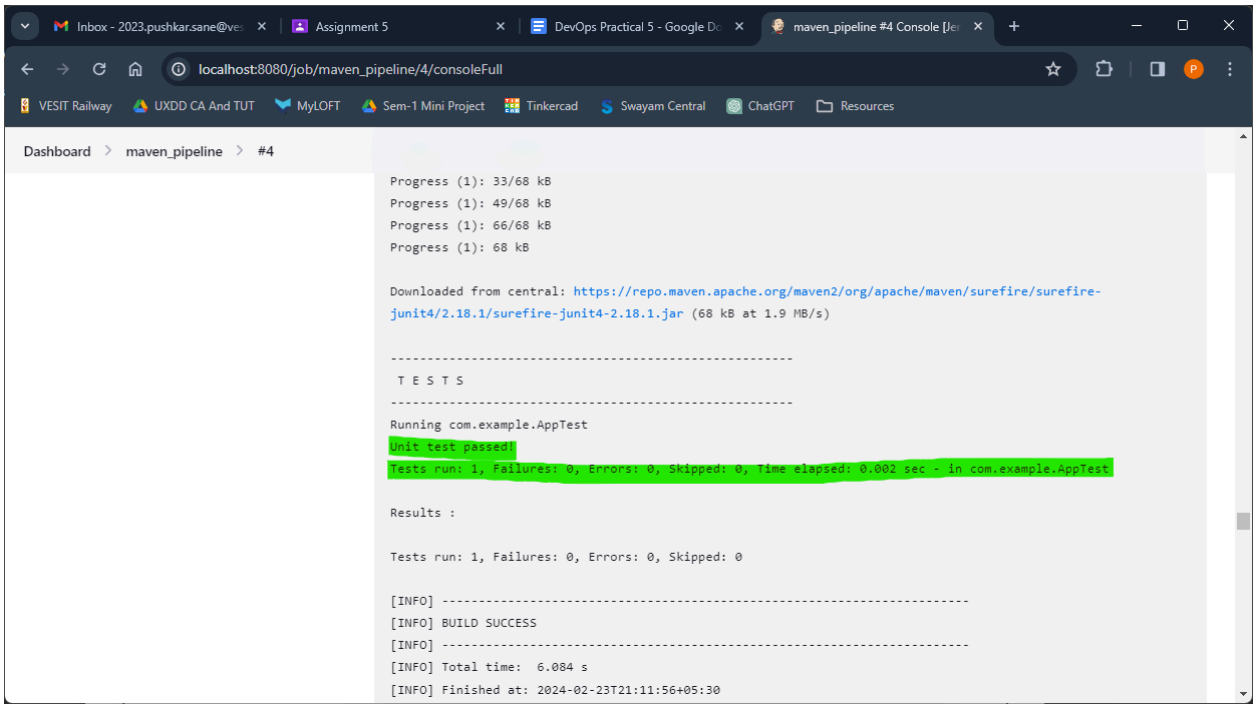


Compilation Stage:

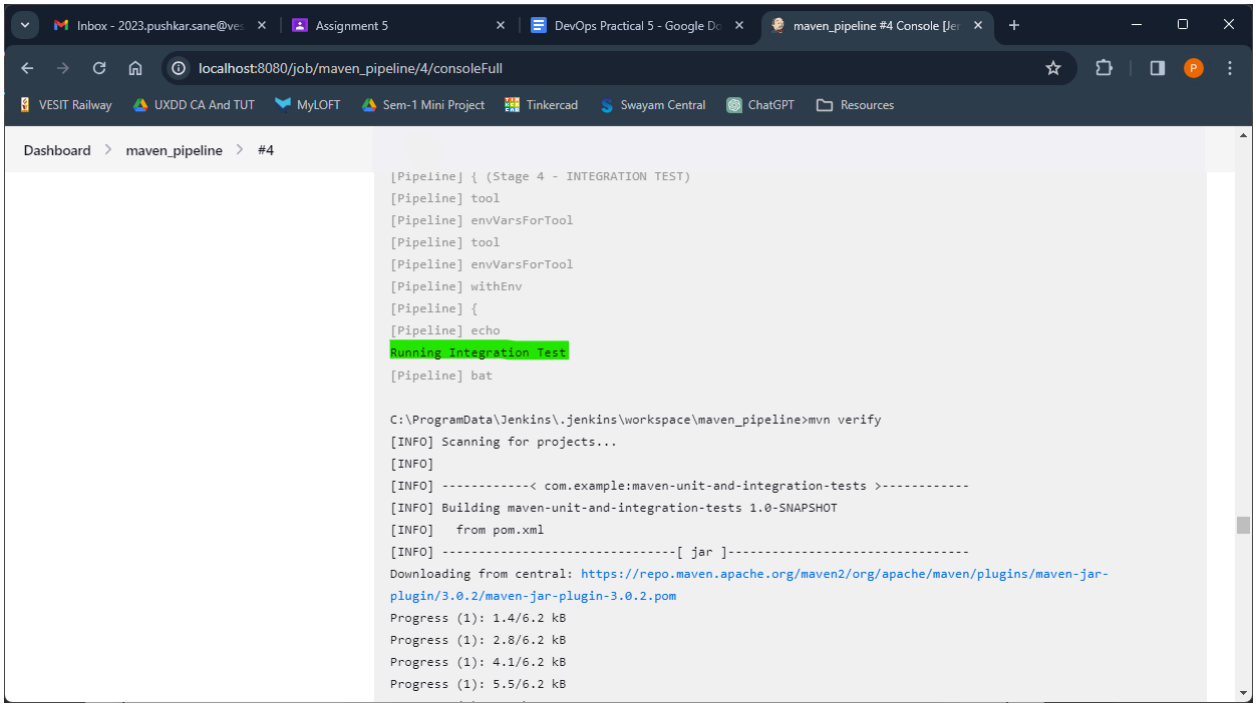


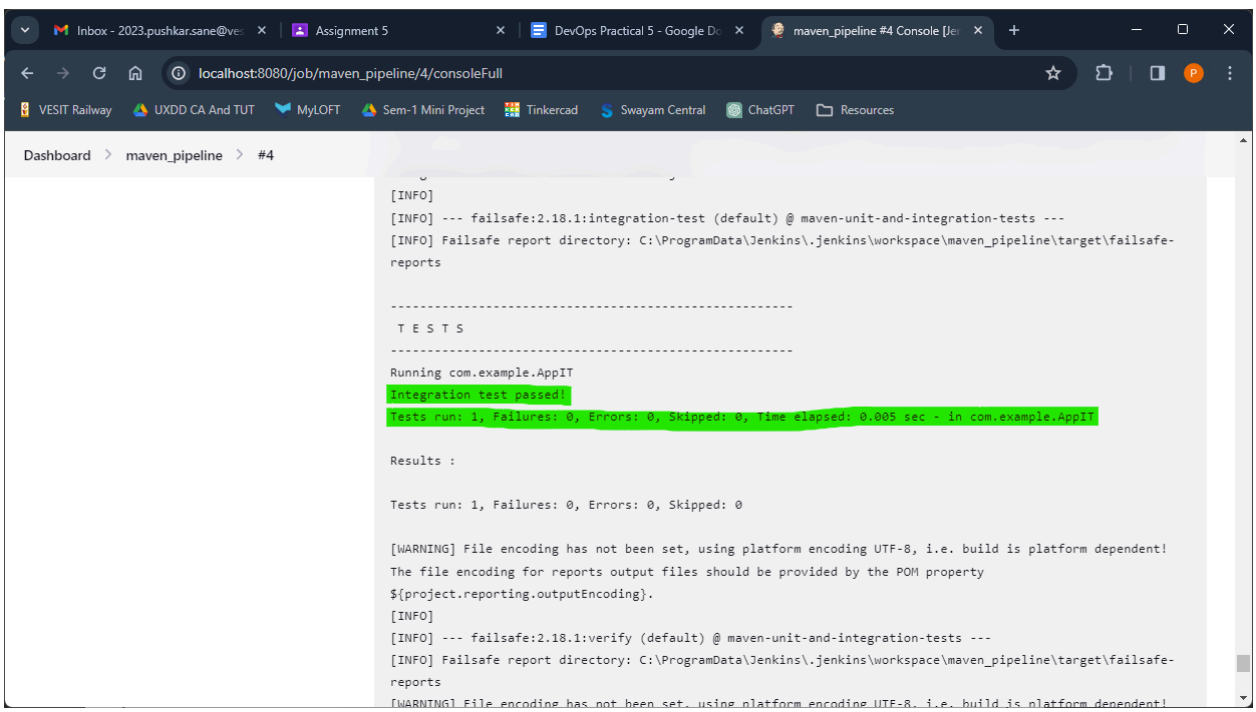
Unit Test Stage:



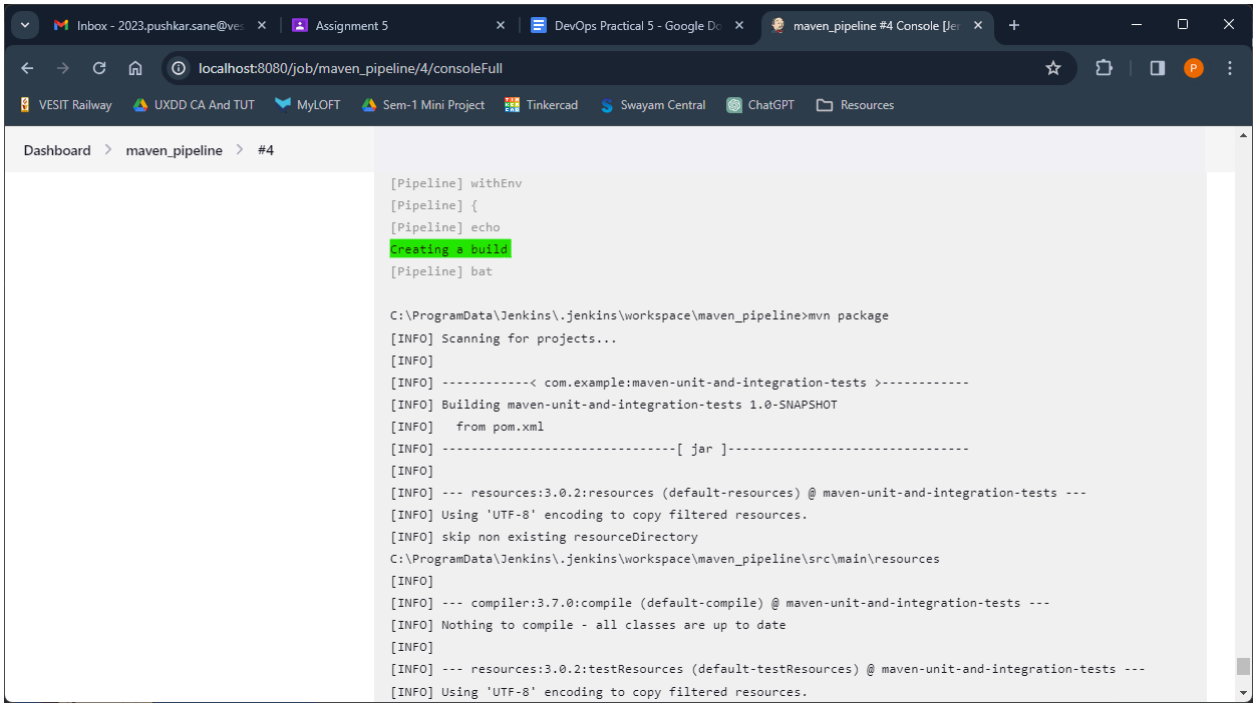


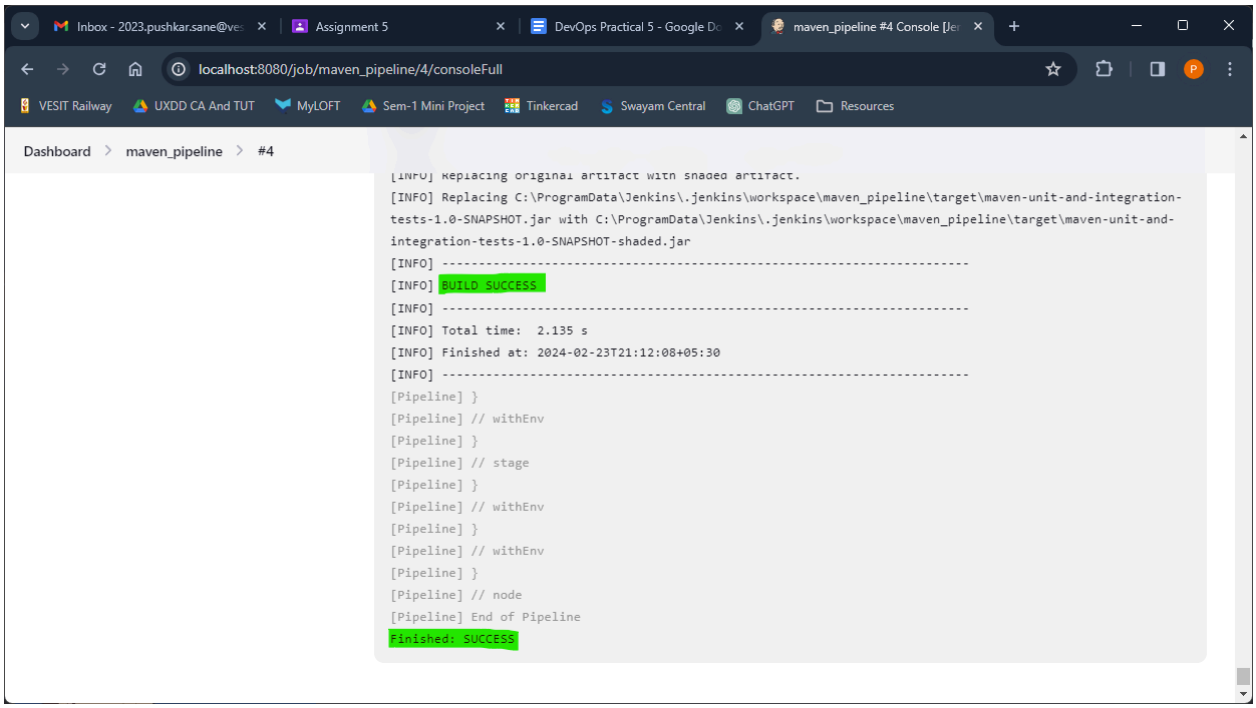
Integration Test Stage:





Creating Build Stage:





Conclusion:

Successfully implemented the Jenkins pipeline using Jenkins declarative pipeline.