<table>
<tr><td colspan="3"><b>Name of Student: Pushkar Sane</b></td></tr>
<tr><td colspan="2"><b>Roll Number: 45</b></td><td><b>Lab Assignment Number: 8</b></td></tr>
<tr><td colspan="3"><b>Title of Lab Assignment: To use Jenkins to deploy and run test cases for Java/Web applications using Junit/Selenium/TestNG.</b></td></tr>
<tr><td colspan="2"><b>DOP: 28-03-2024</b></td><td><b>DOS: 03-04-2024</b></td></tr>
<tr><td><b>CO Mapped:</b>
<b>CO3</b></td><td><b>PO Mapped:</b>
<b>PO2, PO3, PO5, PSO1, PSO2</b></td><td><b>Signature:</b></td></tr>
</table>

Practical No. 8

Aim: To use Jenkins to deploy and run test cases for Java/Web applications using Junit/Selenium/TestNG.

Introduction:

In modern software development, Continuous Integration (CI) and Continuous Deployment (CD) are crucial practices to ensure the quality and reliability of applications. Jenkins is a widely used automation server that facilitates CI/CD pipelines. Integrating Jenkins with testing frameworks like JUnit, Selenium, and TestNG allows for automated testing and deployment of Java/Web applications, ensuring rapid feedback on code changes and maintaining the application's stability.

Requirements:

- Java Development Kit (JDK)
- Jenkins installed and configured
- Maven installed
- Selenium WebDriver
- JUnit and TestNG frameworks
- A Java/Web application project with test cases

Steps:

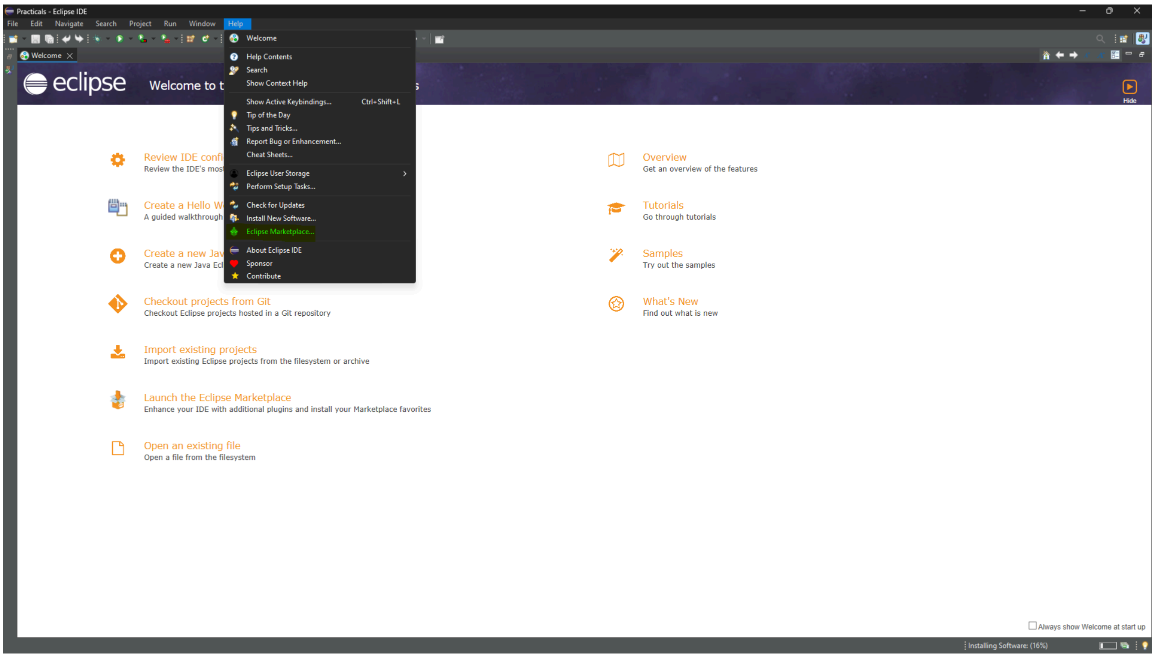
1. Set up Jenkins:
    a. Install Jenkins on your server or local machine.
    b. Configure Jenkins according to your environment requirements.
    c. Install necessary plugins such as Maven Integration, Git, etc.
2. Create a Jenkins Job:
    a. Open Jenkins and create a new Freestyle or Pipeline project.
    b. Configure the project with necessary settings like source code repository (e.g., Git), build triggers, and build environment.
3. Configure Build Steps:
    a. Define build steps to compile the Java/Web application using Maven.
    b. Ensure that necessary dependencies are resolved during the build process.
4. Integrate Testing Frameworks:
    a. Install JUnit and TestNG plugins in Jenkins if not already installed.
    b. Configure the Jenkins job to execute unit tests using JUnit and/or TestNG.

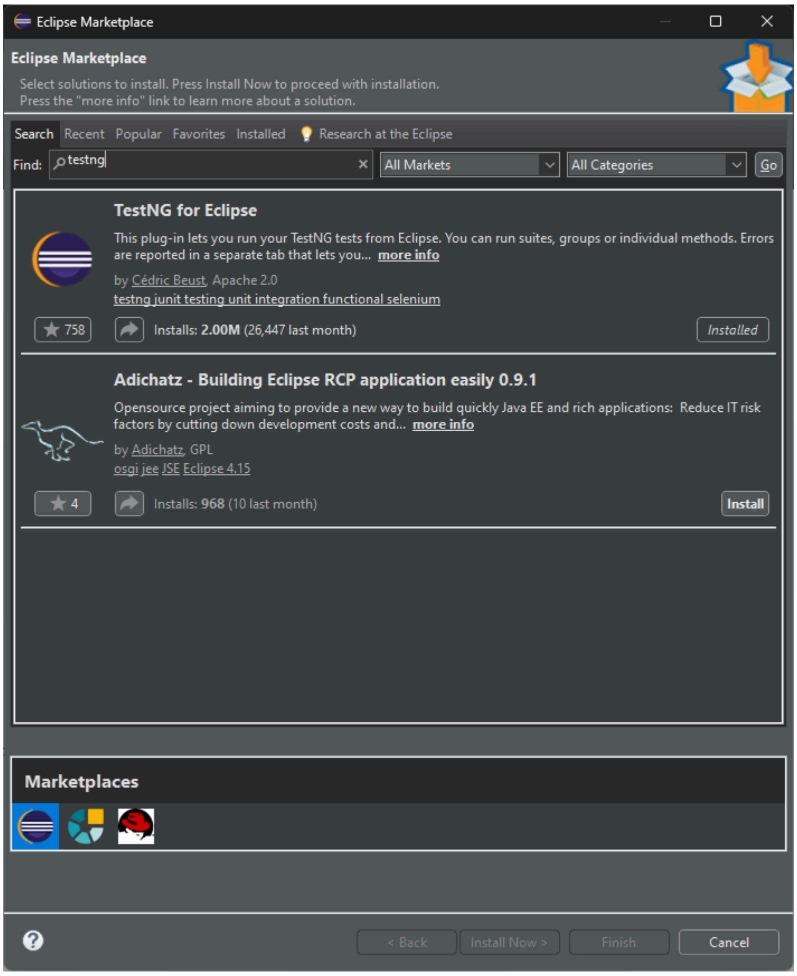
c. Ensure that the test reports are generated in a format compatible with Jenkins (e.g.JUnit XML format).

5. Set up Selenium WebDriver:
    a. Ensure that Selenium WebDriver is available in your project.
    b. Configure Selenium to interact with the web application for automated UI testing.
6. Implement Test Cases:
    a. Develop test cases using JUnit, TestNG, and Selenium WebDriver to validate the functionality of the Java/Web application.
    b. Organize test cases into appropriate test suites.
7. Execute Tests in Jenkins:
    a. Trigger the Jenkins job to run automated tests after the build process.
    b. Monitor the test execution progress within Jenkins.
8. Analyze Test Results:
    a. Review the test results generated by JUnit, TestNG, and Selenium within Jenkins.
    b. Identify any test failures or errors and investigate the root causes.
9. Configure Deployment:
    a. If necessary, configure Jenkins to deploy the application to the target environment after successful testing.
    b. Define deployment steps such as copying artifacts to the deployment server, restarting services, etc.
10. Monitor and Improve:
    a. Continuously monitor the Jenkins pipeline for any issues or failures.
    b. Iterate on the test cases and pipeline configuration to improve efficiency and reliability.

Maven Project with selenium web driver and TestNg

1. Go to the help menu and click on the eclipse marketplace and search for “TestNg” to install the TestNg libraries in eclipse.




Eclipse Marketplace
Search Recent Popular Favorites Installed Research at the Eclipse
All Markets
All Categories
Go
Marketplaces
< Back Install Now > Finish Cancel

Eclipse Marketplace

Select solutions to install. Press Install Now to proceed with installation.
Press the "more info" link to learn more about a solution.

Find: testng

TestNG for Eclipse

This plug-in lets you run your TestNG tests from Eclipse. You can run suites, groups or individual methods. Errors are reported in a separate tab that lets you... more info

by Cédric Beust, Apache 2.0

testng junit testing unit integration functional selenium

758 Installs: 2.00M (26,447 last month) Installed

Adichatz - Building Eclipse RCP application easily 0.9.1

Opensource project aiming to provide a new way to build quickly Java EE and rich applications: Reduce IT risk factors by cutting down development costs and... more info

by Adichatz, GPL

osgi jee JSE Eclipse 4.15

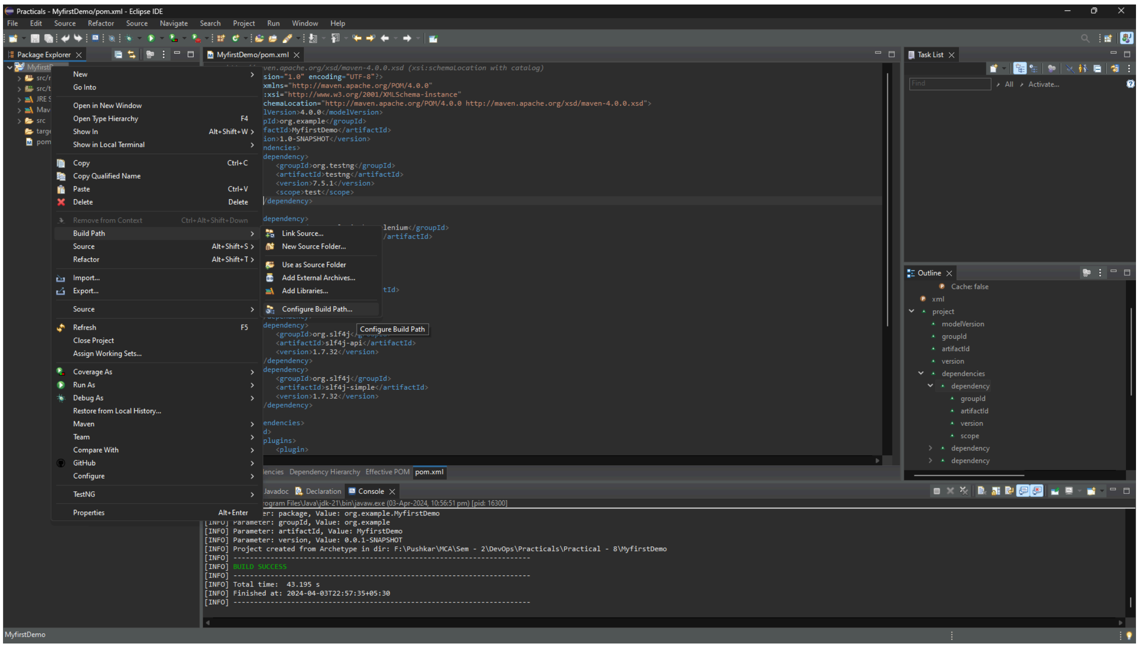
4 Installs: 968 (10 last month) Install

2. Create a maven project and include all the dependencies in the pom.xml file

Pom.xml

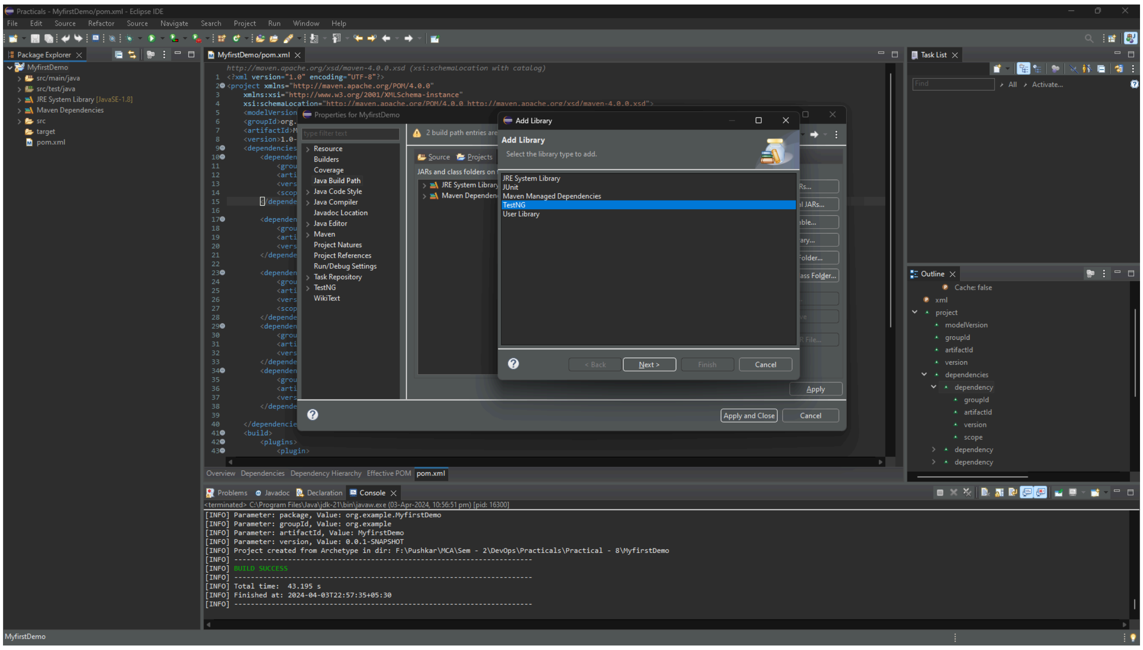
```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>
        <groupId>org.example</groupId>
        <artifactId>MyfirstDemo</artifactId>
        <version>1.0-SNAPSHOT</version>
        <dependencies>
                <dependency>
                        <groupId>org.testng</groupId>
                        <artifactId>testng</artifactId>
                        <version>7.5.1</version>
                        <scope>test</scope>
                </dependency>

                <dependency>
                        <groupId>org.seleniumhq.selenium</groupId>
                        <artifactId>selenium-java</artifactId>
                        <version>4.1.1</version>
                </dependency>

                <dependency>
                        <groupId>junit</groupId>
                        <artifactId>junit</artifactId>
                        <version>4.13.2</version>
                        <scope>test</scope>
                </dependency>
                <dependency>
                        <groupId>org.slf4j</groupId>
                        <artifactId>slf4j-api</artifactId>
                        <version>1.7.32</version>
                </dependency>
                <dependency>
                        <groupId>org.slf4j</groupId>
                        <artifactId>slf4j-simple</artifactId>
                        <version>1.7.32</version>
                </dependency>

        </dependencies>
        <build>
                <plugins>
                        <plugin>
                                <artifactId>maven-compiler-plugin</artifactId>
                                <version>3.8.1</version>
                                <configuration>
                                        <source>1.8</source>
                                        <target>1.8</target>
                                </configuration>
                        </plugin>
                        <plugin>
                                <groupId>org.apache.maven.plugins</groupId>
                                <artifactId>maven-surefire-plugin</artifactId>
                                <version>3.0.0-M5</version>
                                <configuration>
                                        <suiteXmlFiles>


<suiteXmlFile>D:\EclipseW\MyfirstDemo\testng.xml</suiteXmlFile>
                                        </suiteXmlFiles>
                                </configuration>
                        </plugin>
                </plugins>
        </build>
</project>
```

3. Once all the dependencies are added in the pom.xml file right click on the maven project name and select preferences and then go to the libraries menu and click on add library. Then select the TestNG library.



```
ven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation with catalog)
sion="1.0" encoding="UTF-8"?>
xmlns="http://maven.apache.org/POM/4.0.0"
:xsi="http://www.w3.org/2001/XMLSchema-instance"
chemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
lVersion>4.0.0</modelVersion>
pId>org.example</groupId>
factId>MyfirstDemo</artifactId>
ion>1.0-SNAPSHOT</version>
ndencies>
dependency>
    <groupId>org.testng</groupId>
    <artifactId>testng</artifactId>
    <version>7.5.1</version>
    <scope>test</scope>
/dependency>

dependency>
dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>1.7.32</version>
/dependency>
dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-simple</artifactId>
    <version>1.7.32</version>
/dependency>

endencies>
d>
plugins>
    <plugin>
```

```
rogram Files\Java\jdk-21\bin\javaw.exe (03-Apr-2024, 10:56:51 pm) [pid: 16300]
er: package, Value: org.example.MyfirstDemo
[INFO] Parameter: groupId, Value: org.example
[INFO] Parameter: artifactId, Value: MyfirstDemo
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] Project created from Archetype in dir: F:\Pushkar\MCA\Sem - 2\DevOps\Practicals\Practical - 8\MyfirstDemo
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  43.195 s
[INFO] Finished at: 2024-04-03T22:57:35+05:30
[INFO] ------------------------------------------------------------------------
```



```
http://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation with catalog)
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```
<terminated> C:\Program Files\Java\jdk-21\bin\javaw.exe (03-Apr-2024, 10:56:51 pm) [pid: 16300]
[INFO] Parameter: package, Value: org.example.MyfirstDemo
[INFO] Parameter: groupId, Value: org.example
[INFO] Parameter: artifactId, Value: MyfirstDemo
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] Project created from Archetype in dir: F:\Pushkar\MCA\Sem - 2\DevOps\Practicals\Practical - 8\MyfirstDemo
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  43.195 s
[INFO] Finished at: 2024-04-03T22:57:35+05:30
[INFO] ------------------------------------------------------------------------
```

4. Once the libraries are added we can proceed with writing the test cases. Navigate to the src > java > test and create a new package. Once the package is created, create a new test class and write all the test cases which you want to perform.

Sample.java

package Demo;

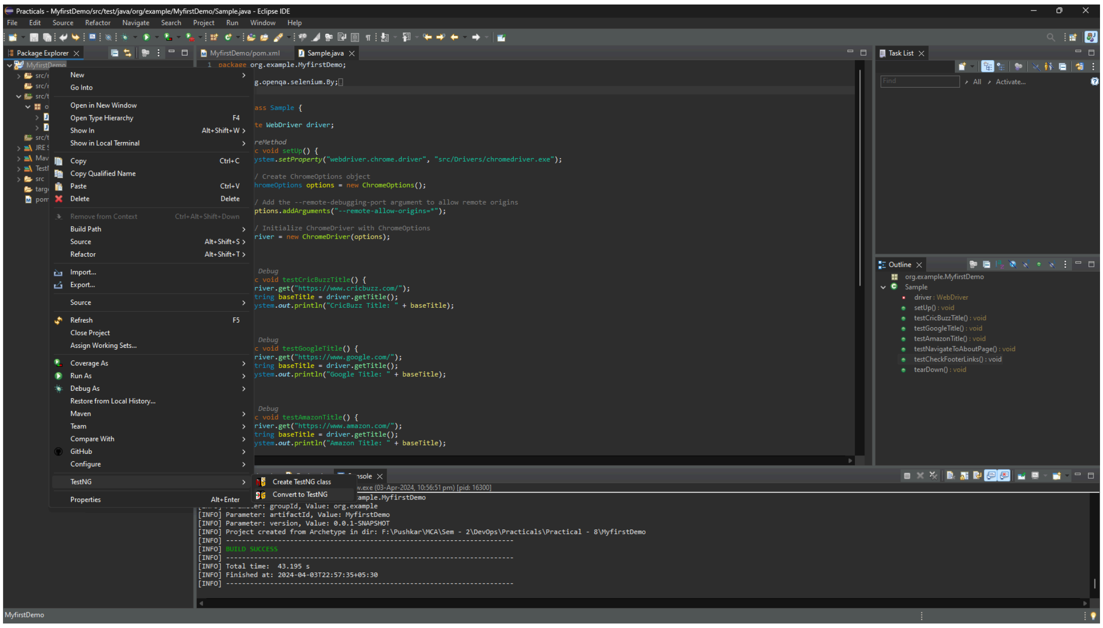
import org.openqa.selenium.By;

```
import org.openqa.selenium.WebElement;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
public class Sample {
	private WebDriver driver;
	@BeforeMethod
	public void setUp() {
		System.setProperty("webdriver.chrome.driver",
"src/Drivers/chromedriver.exe");
		// Create ChromeOptions object
		ChromeOptions options = new ChromeOptions();
		// Add the --remote-debugging-port argument to allow remote origins
		options.addArguments("--remote-allow-origins=*");
		// Initialize ChromeDriver with ChromeOptions
		driver = new ChromeDriver(options);
	}
	@Test
	public void testCricBuzzTitle() {
		driver.get("https://www.cricbuzz.com/");
		String baseTitle = driver.getTitle();
		System.out.println("CricBuzz Title: " + baseTitle);
	}
	@Test
	public void testGoogleTitle() {
		driver.get("https://www.google.com/");
		String baseTitle = driver.getTitle();
		System.out.println("Google Title: " + baseTitle);
	}
	@Test
	public void testAmazonTitle() {
		driver.get("https://www.amazon.com/");
		String baseTitle = driver.getTitle();
```

```
        System.out.println("Amazon Title: " + baseTitle);
    }
    @Test
    public void testNavigateToAboutPage() {
        driver.get("https://www.google.com/");
        WebElement aboutLink = driver.findElement(By.linkText("About"));
        aboutLink.click();
        String pageTitle = driver.getTitle();
        System.out.println("Google About page title: " + pageTitle);
    }
    @Test
    public void testCheckFooterLinks() {
        driver.get("https://www.amazon.com/");
        WebElement footerLink = driver.findElement(By.linkText("Privacy
Notice"));
        footerLink.click();
        String pageTitle = driver.getTitle();
        System.out.println("Amazon Privacy Notice page title: " + pageTitle);
        driver.navigate().back();
        footerLink = driver.findElement(By.linkText("Conditions of Use"));
        footerLink.click();
        pageTitle = driver.getTitle();
        System.out.println("Amazon Conditions of Use page title: " +
pageTitle);
    }
    @AfterMethod
    public void tearDown() {
        if (driver != null) {
            driver.quit();
        }
    }
}
```

5. As we have used the webdriver in our program we have to create a new folder named Drivers in the src directory and add “chromedriver.exe” file in it. Link to download chromeDriver: https://storage.googleapis.com/chrome-for-testing-public/123.0.6312.86/win64/chromedriver-win64.zip
   Make sure you are downloading the correct version of drivers as per chrome installed in your device.

6. Next step is to generate a TestNG.xml file. Right click on your project then look for the TestNg option in the list and then select convert to TestNg.



```
package org.example.MyfirstDemo;

g.openqa.selenium.By;

ass Sample {

te WebDriver driver;

reMethod
c void setUp() {
ystem.setProperty("webdriver.chrome.driver", "src/Drivers/chromedriver.exe");

/ Create ChromeOptions object
hromeOptions options = new ChromeOptions();

/ Add the --remote-debugging-port argument to allow remote origins
ptions.addArguments("--remote-allow-origins=*");

/ Initialize ChromeDriver with ChromeOptions
river = new ChromeDriver(options);


 Debug
c void testCricBuzzTitle() {
river.get("https://www.cricbuzz.com/");
tring baseTitle = driver.getTitle();
ystem.out.println("CricBuzz Title: " + baseTitle);


 Debug
c void testGoogleTitle() {
river.get("https://www.google.com/");
tring baseTitle = driver.getTitle();
ystem.out.println("Google Title: " + baseTitle);


 Debug
c void testAmazonTitle() {
river.get("https://www.amazon.com/");
tring baseTitle = driver.getTitle();
ystem.out.println("Amazon Title: " + baseTitle);
```

```
w.exe (03-Apr-2024, 10:56:51 pm) [pid: 16300]
ample.MyfirstDemo
er: groupId, Value: org.example
[INFO] Parameter: artifactId, Value: MyfirstDemo
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] Project created from Archetype in dir: F:\Pushkar\MCA\Sem - 2\DevOps\Practicals\Practical - 8\MyfirstDemo
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  43.195 s
[INFO] Finished at: 2024-04-03T22:57:35+05:30
[INFO] ------------------------------------------------------------------------
```

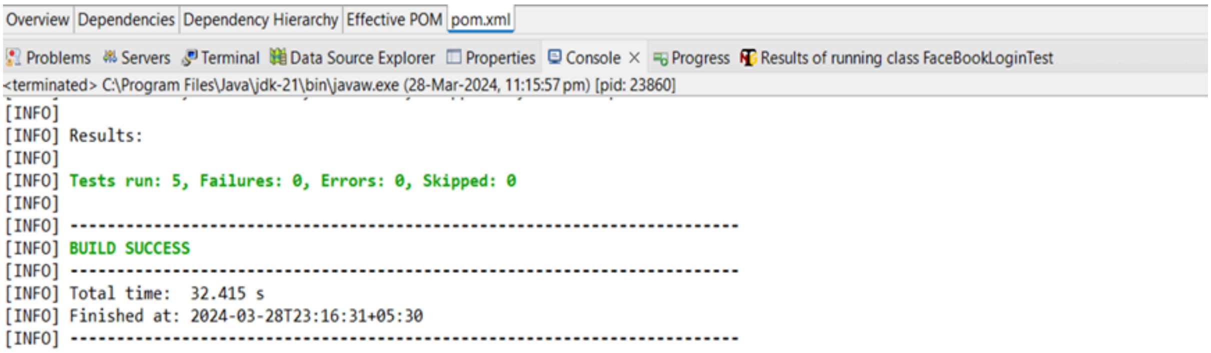
Then click on Next and the Testng.xml file will be generated.

TestNg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
        <test thread-count="5" name="Test">
                <classes>
                        <class name="Demo.Sample" />
                </classes>
        </test> <!-- Test -->
</suite> <!-- Suite -->
```

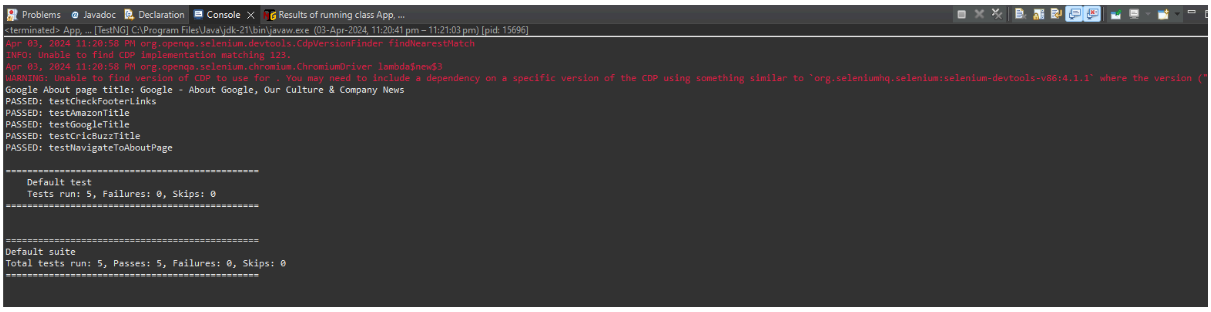
7. Once the TestNg file is generated we can try testing our test cases.

   Right click on project and then select “Maven test”



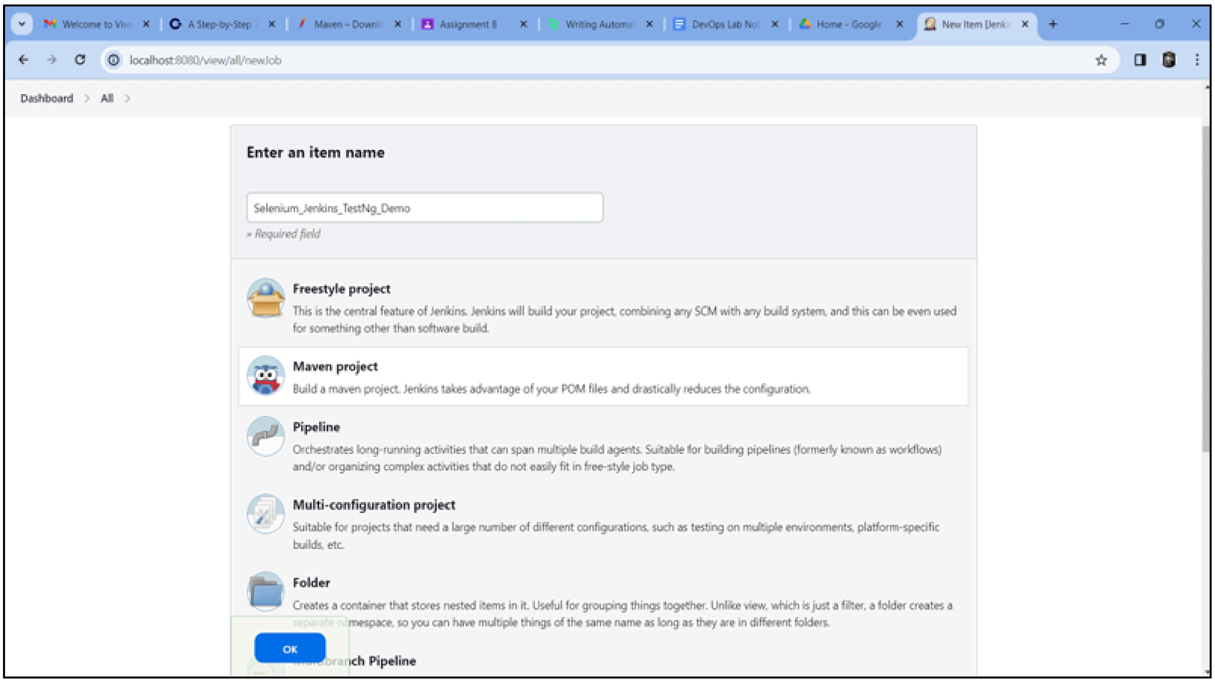
```
<terminated> C:\Program Files\Java\jdk-21\bin\javaw.exe (28-Mar-2024, 11:15:57 pm) [pid: 23860]
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  32.415 s
[INFO] Finished at: 2024-03-28T23:16:31+05:30
[INFO] ------------------------------------------------------------------------
```

If all the test cases are successful then we can move towards Jenkins deployment.



```
<terminated> App, ... [TestNG] C:\Program Files\Java\jdk-21\bin\javaw.exe (03-Apr-2024, 11:20:41 pm – 11:21:03 pm) [pid: 15696]
Apr 03, 2024 11:20:58 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Unable to find CDP implementation matching 123.
Apr 03, 2024 11:20:58 PM org.openqa.selenium.chromium.ChromiumDriver lambda$new$3
WARNING: Unable to find version of CDP to use for . You may need to include a dependency on a specific version of the CDP using something similar to `org.seleniumhq.selenium:selenium-devtools-v86:4.1.1` where the version ("
Google About page title: Google - About Google, Our Culture & Company News
PASSED: testCheckFooterLinks
PASSED: testAmazonTitle
PASSED: testGoogleTitle
PASSED: testCricBuzzTitle
PASSED: testNavigateToAboutPage

===============================================
    Default test
    Tests run: 5, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 5, Passes: 5, Failures: 0, Skips: 0
===============================================
```

8. Open Jenkins (localhost:8080) and create a new maven project.



Enter an item name

Selenium_Jenkins_TestNg_Demo

» 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.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

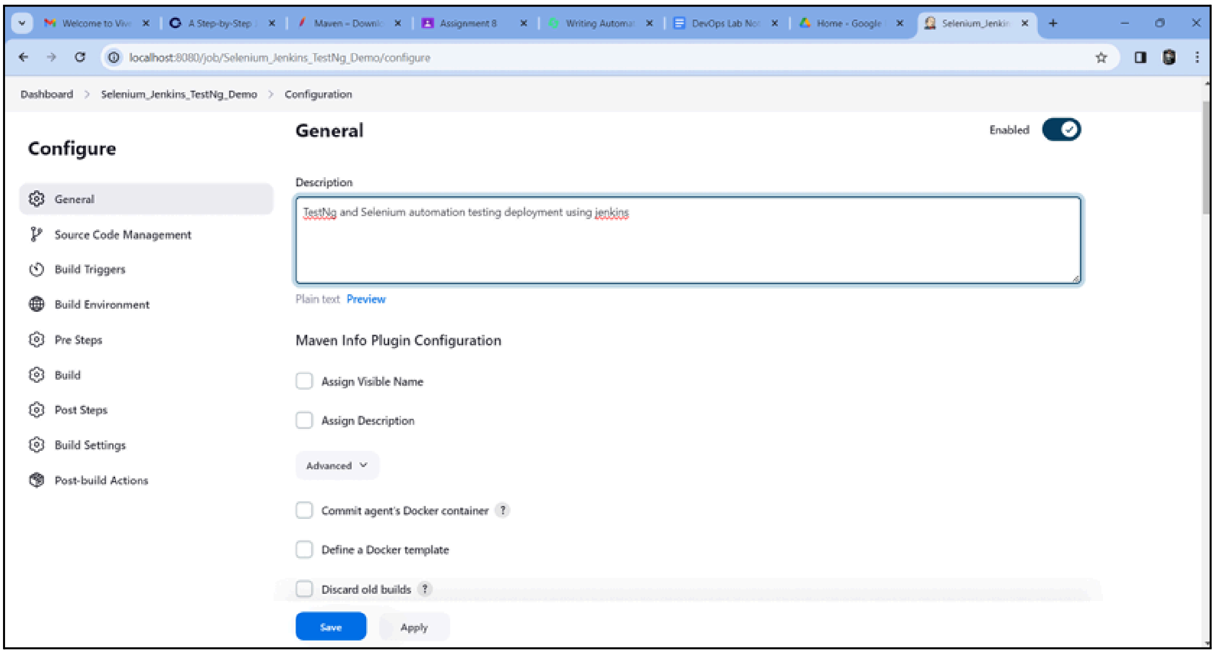
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

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

9. Then configure the maven project as per below.



localhost:8080/job/Selenium_Jenkins_TestNg_Demo/configure

General

Enabled ●

Description

TestNg and Selenium automation testing deployment using jenkins

Maven Info Plugin Configuration

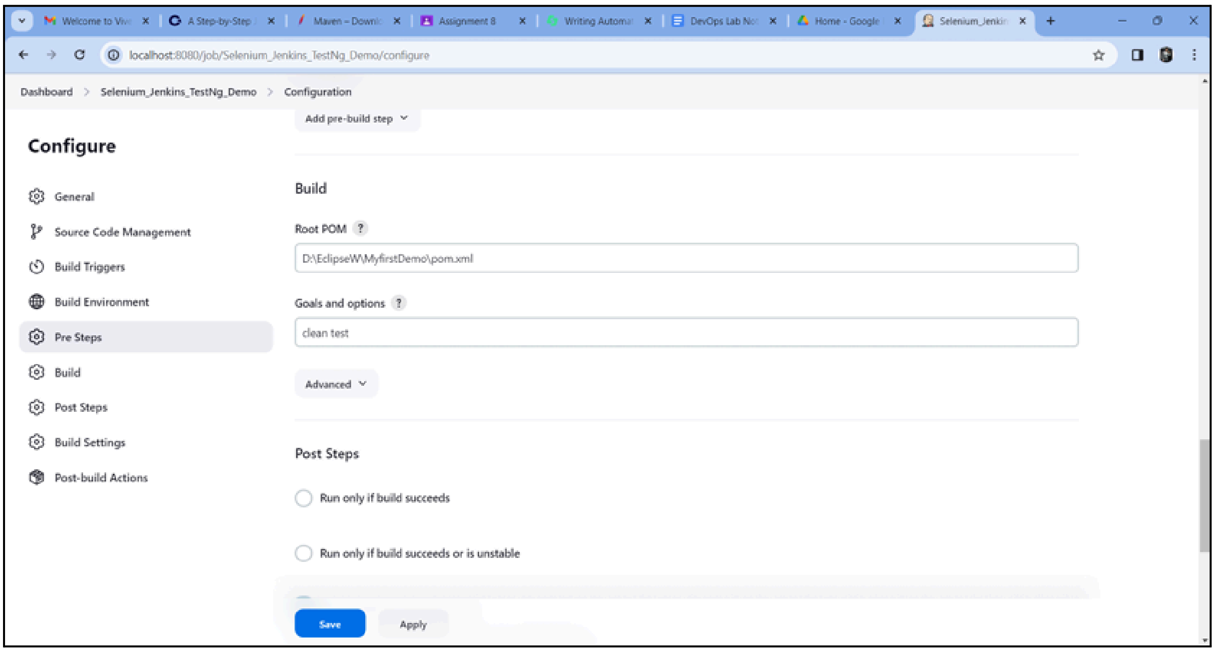
□ Assign Visible Name

□ Assign Description

□ Commit agent's Docker container ?

□ Define a Docker template

□ Discard old builds ?



localhost:8080/job/Selenium_Jenkins_TestNg_Demo/configure

Build

Root POM ?

D:\EclipseW\MyfirstDemo\pom.xml

Goals and options ?

clean test

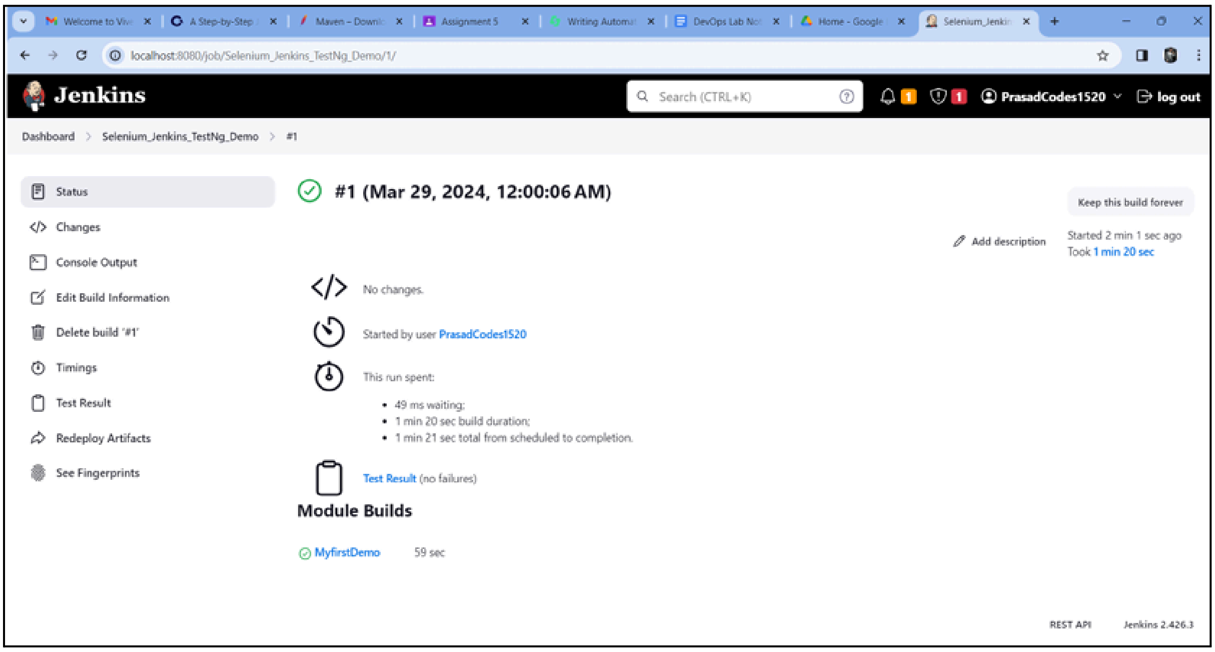
Post Steps

○ Run only if build succeeds

○ Run only if build succeeds or is unstable

10. Once the configuration is done, click on build now to run the test cases.

Output:



localhost:8080/job/Selenium_Jenkins_TestNg_Demo/1/

Dashboard > Selenium_Jenkins_TestNg_Demo > #1

#1 (Mar 29, 2024, 12:00:06 AM)

Keep this build forever

Add description

Started 2 min 1 sec ago
Took 1 min 20 sec

No changes.

Started by user PrasadCodes1520

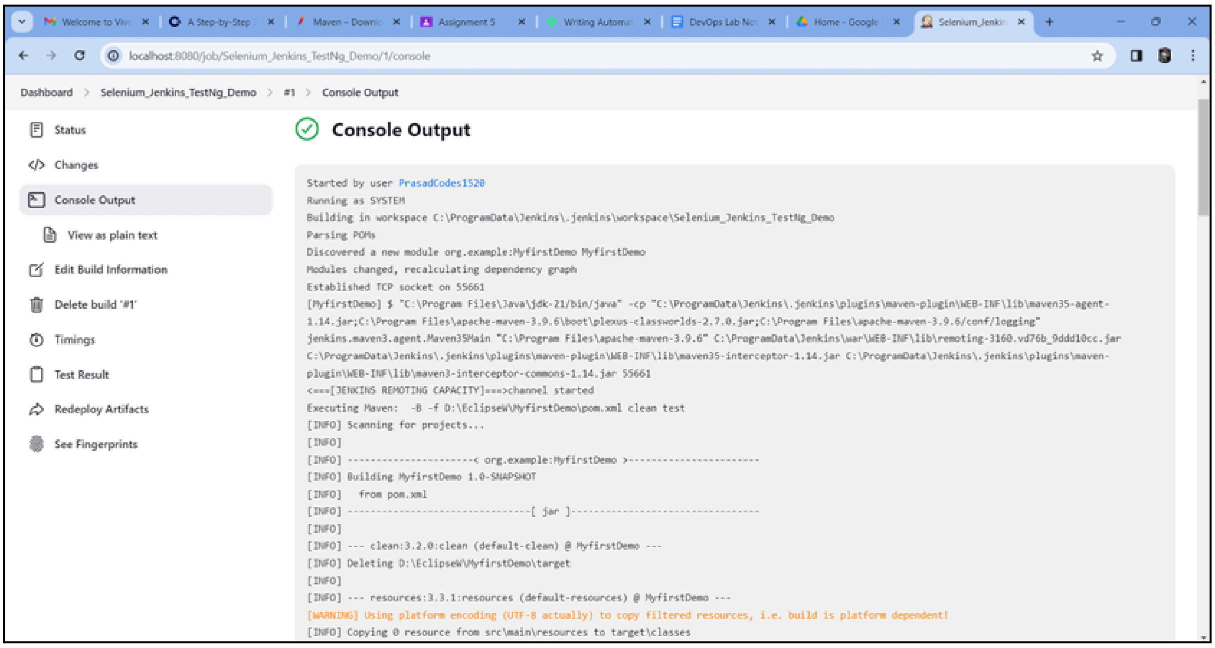
This run spent:

- 49 ms waiting;
- 1 min 20 sec build duration;
- 1 min 21 sec total from scheduled to completion.

Test Result (no failures)

Module Builds

MyfirstDemo 59 sec

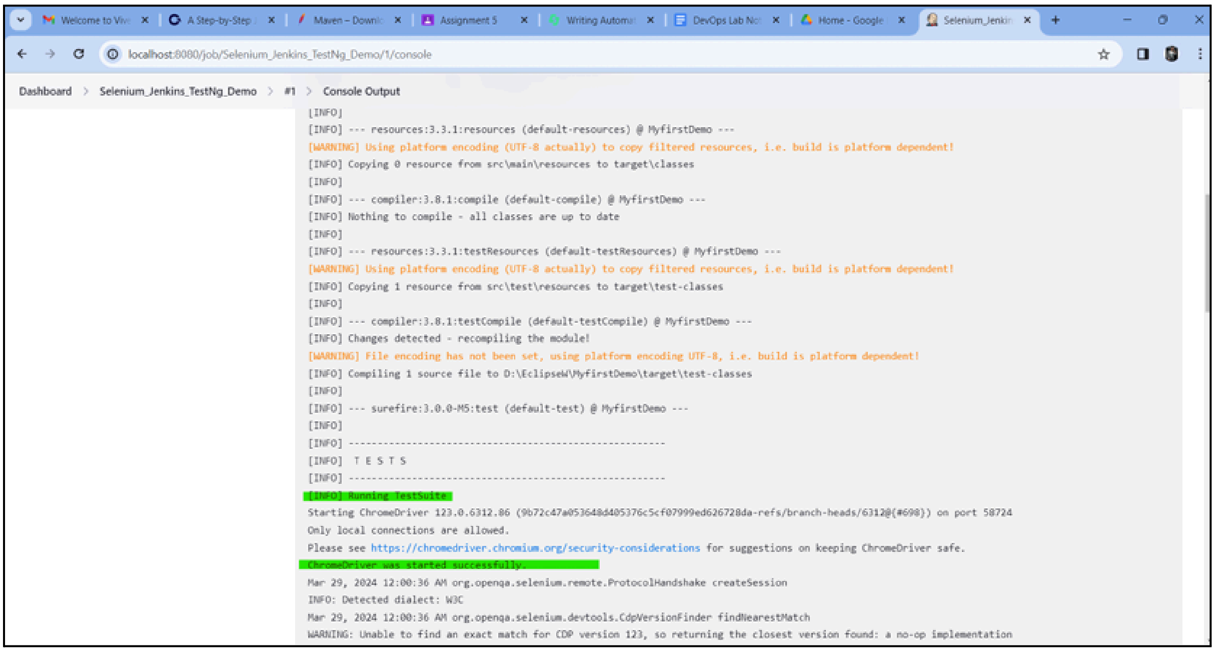


localhost:8080/job/Selenium_Jenkins_TestNg_Demo/1/console

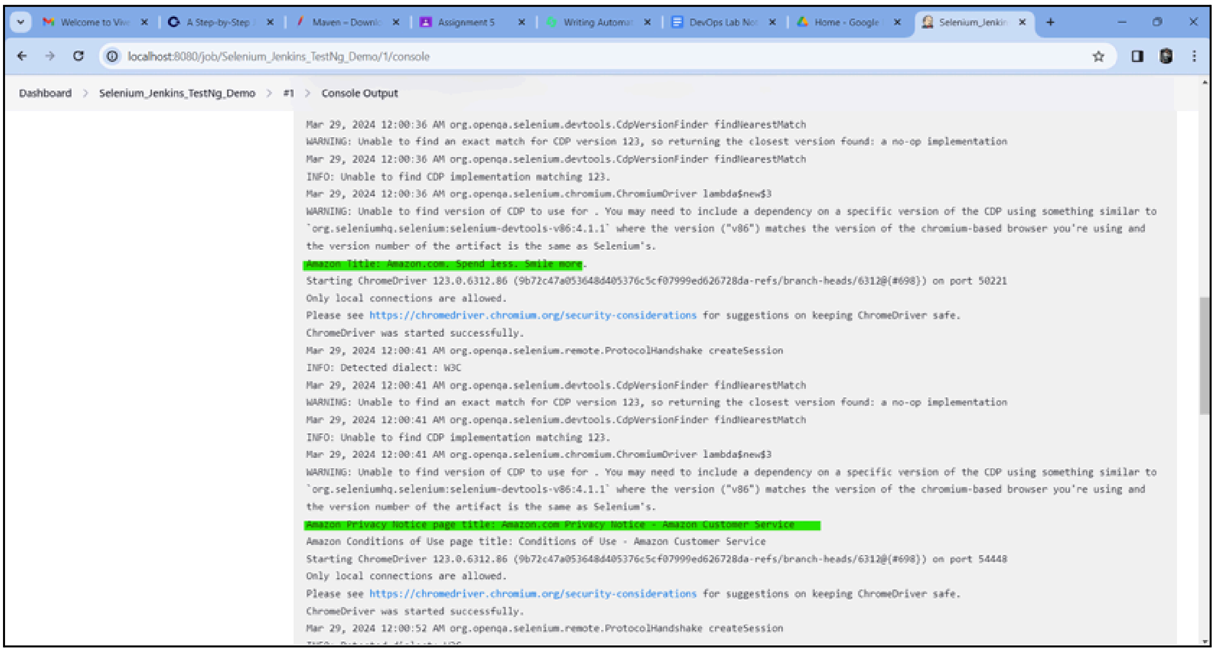
Dashboard > Selenium_Jenkins_TestNg_Demo > #1 > Console Output

Console Output

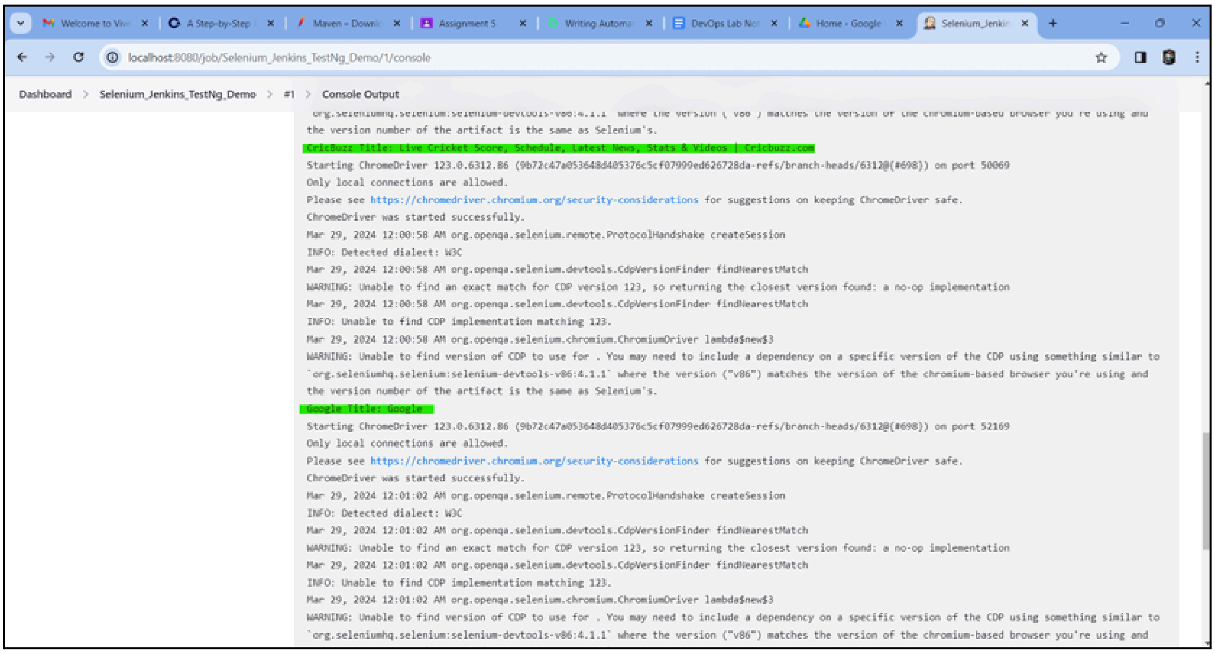
```
Started by user PrasadCodes1520
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\Selenium_Jenkins_TestNg_Demo
Parsing POMs
Discovered a new module org.example:MyfirstDemo MyfirstDemo
Modules changed, recalculating dependency graph
Established TCP socket on 55661
[MyfirstDemo] $ "C:\Program Files\Java\jdk-21/bin/java" -cp "C:\ProgramData\Jenkins\.jenkins\plugins\maven-plugin\WEB-INF\lib\maven35-agent-
1.14.jar;C:\Program Files\apache-maven-3.9.6\boot\plexus-classworlds-2.7.0.jar;C:\Program Files\apache-maven-3.9.6/conf/logging"
jenkins.maven3.agent.Maven35Main "C:\Program Files\apache-maven-3.9.6" C:\ProgramData\Jenkins\war\WEB-INF\lib\remoting-3160.vd76b_9ddd10cc.jar
C:\ProgramData\Jenkins\.jenkins\plugins\maven-plugin\WEB-INF\lib\maven35-interceptor-1.14.jar C:\ProgramData\Jenkins\.jenkins\plugins\maven-
plugin\WEB-INF\lib\maven3-interceptor-commons-1.14.jar 55661
<===[JENKINS REMOTING CAPACITY]===>channel started
Executing Maven:  -B -f D:\EclipseW\MyfirstDemo\pom.xml clean test
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< org.example:MyfirstDemo >-----------------------
[INFO] Building MyfirstDemo 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ MyfirstDemo ---
[INFO] Deleting D:\EclipseW\MyfirstDemo\target
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ MyfirstDemo ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource from src\main\resources to target\classes
```


Welcome to Viv
A Step-by-Step
Maven – Downl
Assignment 5
Writing Automa
DevOps Lab No
Home - Google
Selenium_Jenkin
localhost:8080/job/Selenium_Jenkins_TestNg_Demo/1/console
Dashboard > Selenium_Jenkins_TestNg_Demo > #1 > Console Output

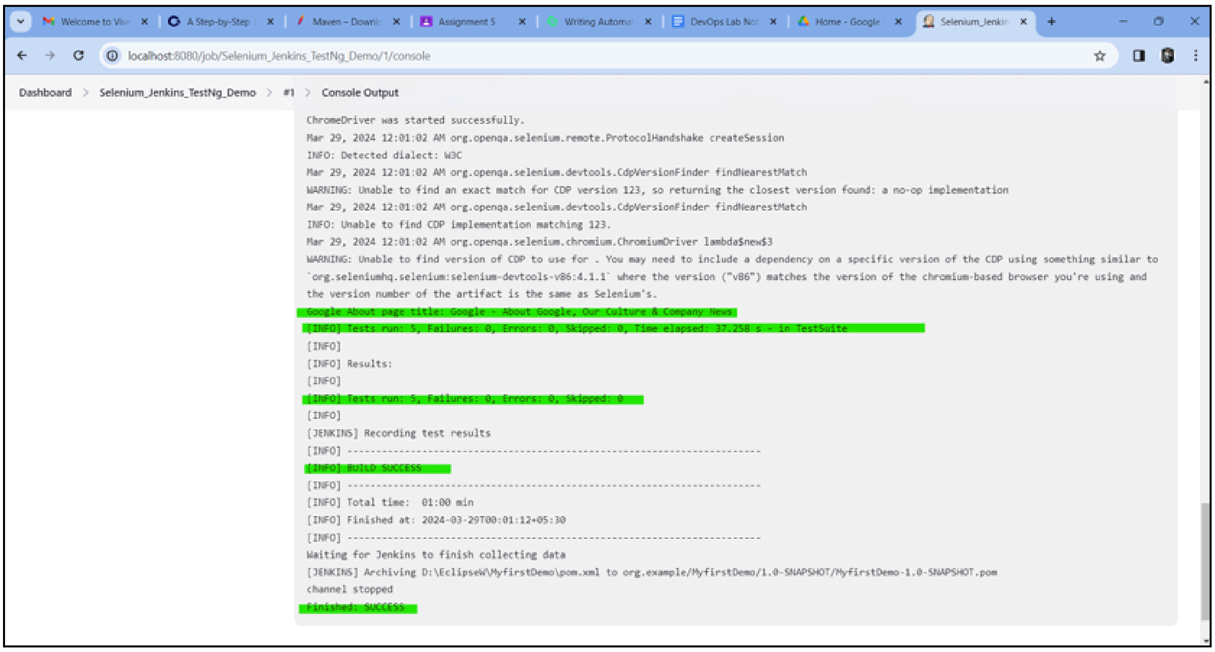
```
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ MyfirstDemo ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource from src\main\resources to target\classes
[INFO]
[INFO] --- compiler:3.8.1:compile (default-compile) @ MyfirstDemo ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ MyfirstDemo ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 1 resource from src\test\resources to target\test-classes
[INFO]
[INFO] --- compiler:3.8.1:testCompile (default-testCompile) @ MyfirstDemo ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 1 source file to D:\EclipseW\MyfirstDemo\target\test-classes
[INFO]
[INFO] --- surefire:3.0.0-M5:test (default-test) @ MyfirstDemo ---
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running TestSuite
Starting ChromeDriver 123.0.6312.86 (9b72c47a053648d405376c5cf07999ed626728da-refs/branch-heads/6312@{#698}) on port 58724
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Mar 29, 2024 12:00:36 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Mar 29, 2024 12:00:36 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 123, so returning the closest version found: a no-op implementation
```


Welcome to Viv
A Step-by-Step
Maven – Downl
Assignment 5
Writing Automa
DevOps Lab No
Home - Google
Selenium_Jenkin
localhost:8080/job/Selenium_Jenkins_TestNg_Demo/1/console
Dashboard > Selenium_Jenkins_TestNg_Demo > #1 > Console Output

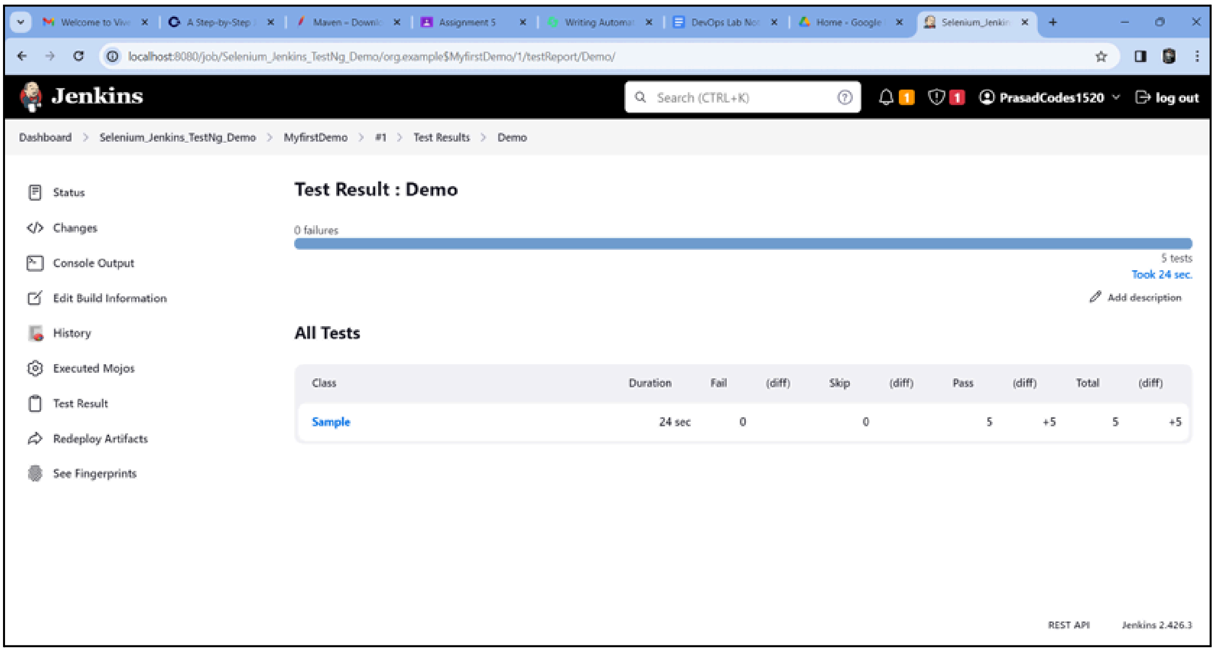
```
Mar 29, 2024 12:00:36 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 123, so returning the closest version found: a no-op implementation
Mar 29, 2024 12:00:36 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Unable to find CDP implementation matching 123.
Mar 29, 2024 12:00:36 AM org.openqa.selenium.chromium.ChromiumDriver lambda$new$3
WARNING: Unable to find version of CDP to use for . You may need to include a dependency on a specific version of the CDP using something similar to
`org.seleniumhq.selenium:selenium-devtools-v86:4.1.1` where the version ("v86") matches the version of the chromium-based browser you're using and
the version number of the artifact is the same as Selenium's.
Amazon Title: Amazon.com. Spend less. Smile more.
Starting ChromeDriver 123.0.6312.86 (9b72c47a053648d405376c5cf07999ed626728da-refs/branch-heads/6312@{#698}) on port 50221
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Mar 29, 2024 12:00:41 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Mar 29, 2024 12:00:41 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 123, so returning the closest version found: a no-op implementation
Mar 29, 2024 12:00:41 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Unable to find CDP implementation matching 123.
Mar 29, 2024 12:00:41 AM org.openqa.selenium.chromium.ChromiumDriver lambda$new$3
WARNING: Unable to find version of CDP to use for . You may need to include a dependency on a specific version of the CDP using something similar to
`org.seleniumhq.selenium:selenium-devtools-v86:4.1.1` where the version ("v86") matches the version of the chromium-based browser you're using and
the version number of the artifact is the same as Selenium's.
Amazon Privacy Notice page title: Amazon.com Privacy Notice - Amazon Customer Service
Amazon Conditions of Use page title: Conditions of Use - Amazon Customer Service
Starting ChromeDriver 123.0.6312.86 (9b72c47a053648d405376c5cf07999ed626728da-refs/branch-heads/6312@{#698}) on port 54448
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Mar 29, 2024 12:00:52 AM org.openqa.selenium.remote.ProtocolHandshake createSession
```


Welcome to Viv
A Step-by-Step
Maven – Downl
Assignment 5
Writing Automa
DevOps Lab No
Home - Google
Selenium_Jenkin
localhost:8080/job/Selenium_Jenkins_TestNg_Demo/1/console
Dashboard > Selenium_Jenkins_TestNg_Demo > #1 > Console Output

```
`org.seleniumhq.selenium:selenium-devtools-v86:4.1.1` where the version ("v86") matches the version of the chromium-based browser you're using and
the version number of the artifact is the same as Selenium's.
CricBuzz Title: Live Cricket Score, Schedule, Latest News, Stats & Videos | Cricbuzz.com
Starting ChromeDriver 123.0.6312.86 (9b72c47a053648d405376c5cf07999ed626728da-refs/branch-heads/6312@{#698}) on port 50069
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Mar 29, 2024 12:00:58 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Mar 29, 2024 12:00:58 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 123, so returning the closest version found: a no-op implementation
Mar 29, 2024 12:00:58 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Unable to find CDP implementation matching 123.
Mar 29, 2024 12:00:58 AM org.openqa.selenium.chromium.ChromiumDriver lambda$new$3
WARNING: Unable to find version of CDP to use for . You may need to include a dependency on a specific version of the CDP using something similar to
`org.seleniumhq.selenium:selenium-devtools-v86:4.1.1` where the version ("v86") matches the version of the chromium-based browser you're using and
the version number of the artifact is the same as Selenium's.
Google Title: Google
Starting ChromeDriver 123.0.6312.86 (9b72c47a053648d405376c5cf07999ed626728da-refs/branch-heads/6312@{#698}) on port 52169
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Mar 29, 2024 12:01:02 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Mar 29, 2024 12:01:02 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 123, so returning the closest version found: a no-op implementation
Mar 29, 2024 12:01:02 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Unable to find CDP implementation matching 123.
Mar 29, 2024 12:01:02 AM org.openqa.selenium.chromium.ChromiumDriver lambda$new$3
WARNING: Unable to find version of CDP to use for . You may need to include a dependency on a specific version of the CDP using something similar to
`org.seleniumhq.selenium:selenium-devtools-v86:4.1.1` where the version ("v86") matches the version of the chromium-based browser you're using and
```


Welcome to Viv
A Step-by-Step
Maven – Downl
Assignment 5
Writing Automa
DevOps Lab No
Home - Google
Selenium_Jenkin
localhost:8080/job/Selenium_Jenkins_TestNg_Demo/1/console
Dashboard > Selenium_Jenkins_TestNg_Demo > #1 > Console Output

```
ChromeDriver was started successfully.
Mar 29, 2024 12:01:02 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Mar 29, 2024 12:01:02 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 123, so returning the closest version found: a no-op implementation
Mar 29, 2024 12:01:02 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Unable to find CDP implementation matching 123.
Mar 29, 2024 12:01:02 AM org.openqa.selenium.chromium.ChromiumDriver lambda$new$3
WARNING: Unable to find version of CDP to use for . You may need to include a dependency on a specific version of the CDP using something similar to
`org.seleniumhq.selenium:selenium-devtools-v86:4.1.1` where the version ("v86") matches the version of the chromium-based browser you're using and
the version number of the artifact is the same as Selenium's.
Google About page title: Google - About Google, Our Culture & Company News
[INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 37.258 s - in TestSuite
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[JENKINS] Recording test results
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  01:00 min
[INFO] Finished at: 2024-03-29T00:01:12+05:30
[INFO] ------------------------------------------------------------------------
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving D:\EclipseW\MyfirstDemo\pom.xml to org.example/MyfirstDemo/1.0-SNAPSHOT/MyfirstDemo-1.0-SNAPSHOT.pom
channel stopped
Finished: SUCCESS
```



localhost:8080/job/Selenium_Jenkins_TestNg_Demo/org.example$MyfirstDemo/1/testReport/Demo/

Dashboard > Selenium_Jenkins_TestNg_Demo > MyfirstDemo > #1 > Test Results > Demo

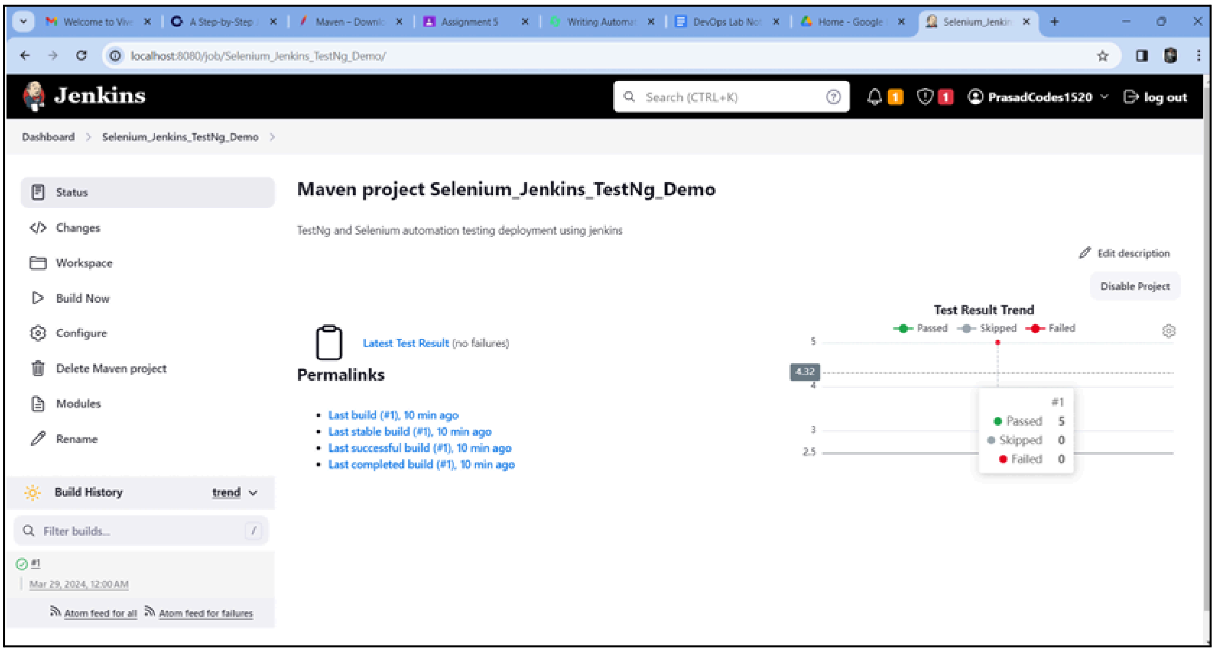
Test Result : Demo

0 failures

5 tests
Took 24 sec.

All Tests

Class	Duration	Fail	(diff)	Skip	(diff)	Pass	(diff)	Total	(diff)
Sample	24 sec	0		0		5	+5	5	+5



localhost:8080/job/Selenium_Jenkins_TestNg_Demo/

Dashboard > Selenium_Jenkins_TestNg_Demo >

Maven project Selenium_Jenkins_TestNg_Demo

TestNg and Selenium automation testing deployment using jenkins

Latest Test Result (no failures)

Permalinks

- Last build (#1), 10 min ago
- Last stable build (#1), 10 min ago
- Last successful build (#1), 10 min ago
- Last completed build (#1), 10 min ago

#1
Mar 29, 2024, 12:00 AM

Conclusion:

This practical demonstrates the integration of Jenkins with JUnit, TestNG, and Selenium for automated deployment and testing of Java/Web applications. By following these steps, developers can establish a robust CI/CD pipeline that ensures the quality and reliability of their software products while accelerating the development process.