Name – Kamal Agrahari Subject – DevOps Lab

ID -VU4F2223028 TE IT A | Batch B

**Experiment No. 7**

**Aim** - : Execute a simple Maven project and integrate it with Jenkins using GitHub.

**Theory** –

Maven is a build automation tool primarily used for Java projects, hosted by the Apache Software Foundation. It simplifies the build process and provides a uniform system for managing project dependencies and configurations through the Project Object Model (POM), defined in the pom.xml file.

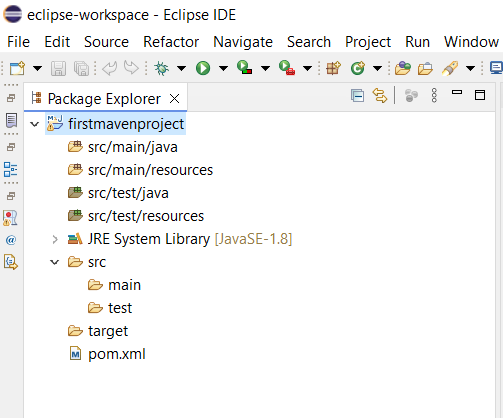
**Tools and Technologies Used**

* **Eclipse Neon**
* **Maven** - 3.5.3
* **JDK** - 1.8

**Steps**

1. **Set Up Maven Project in Eclipse:**

* Install Eclipse IDE from [Eclipse Downloads](https://www.eclipse.org/).
* Open Eclipse and create a new Maven project (File -> New -> Maven Project).
* Select options for a simple project and use the default workspace location.
* Enter GroupId and ArtifactId.



1. **Configure pom.xml:**

* Add the necessary dependencies (e.g., JUnit) and configure the Maven compiler plugin:

<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.yogiraj.mavenproject</groupId>

<artifactId>firstmavenproject</artifactId>

<version>0.0.1-SNAPSHOT</version>

<!-- https://mvnrepository.com/artifact/junit/junit -->

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.1</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.6.0</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

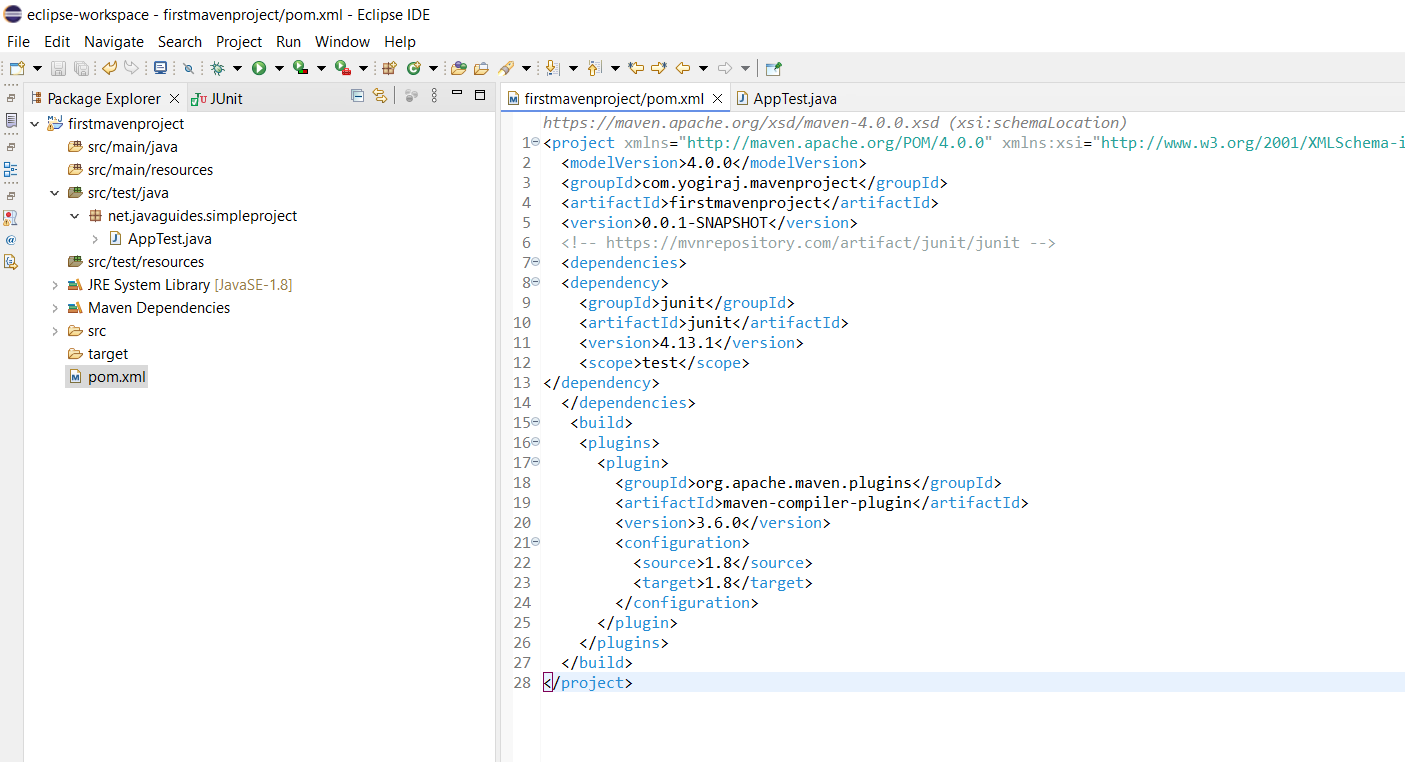
</configuration>

</plugin>

</plugins>

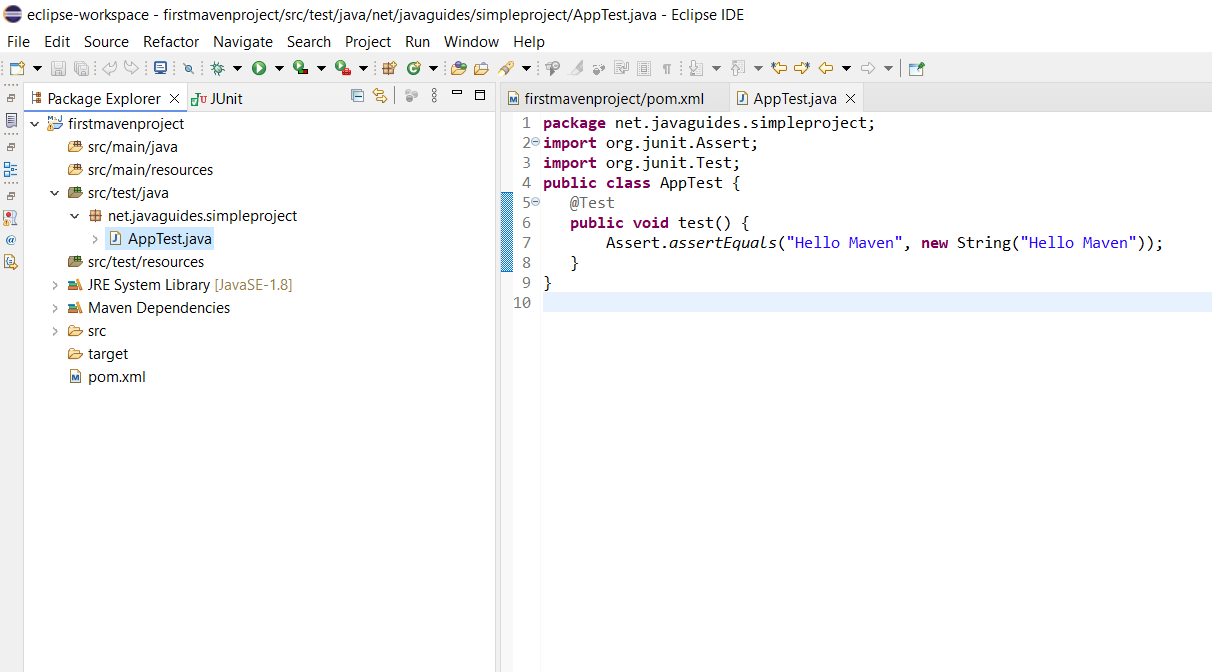
</build>

</project>



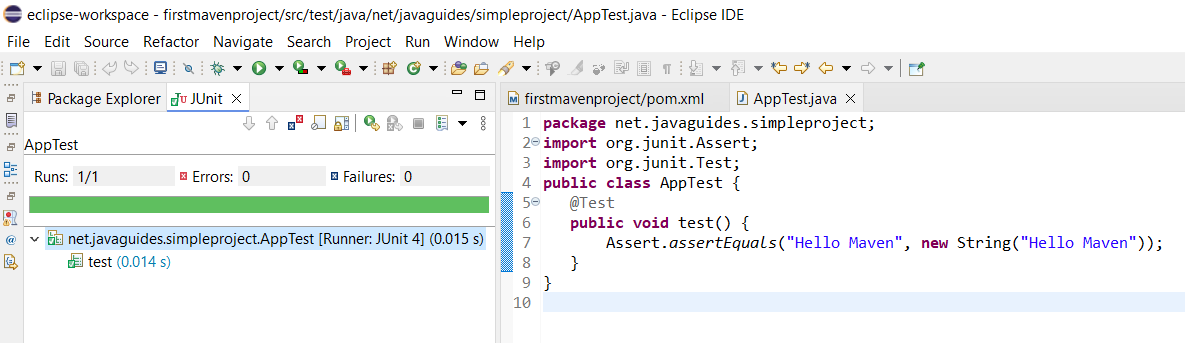
1. **Create a JUnit Test:**

* Create a package net.javaguides.simpleproject under src/test/java.
* Add AppTest.java and write a simple test:



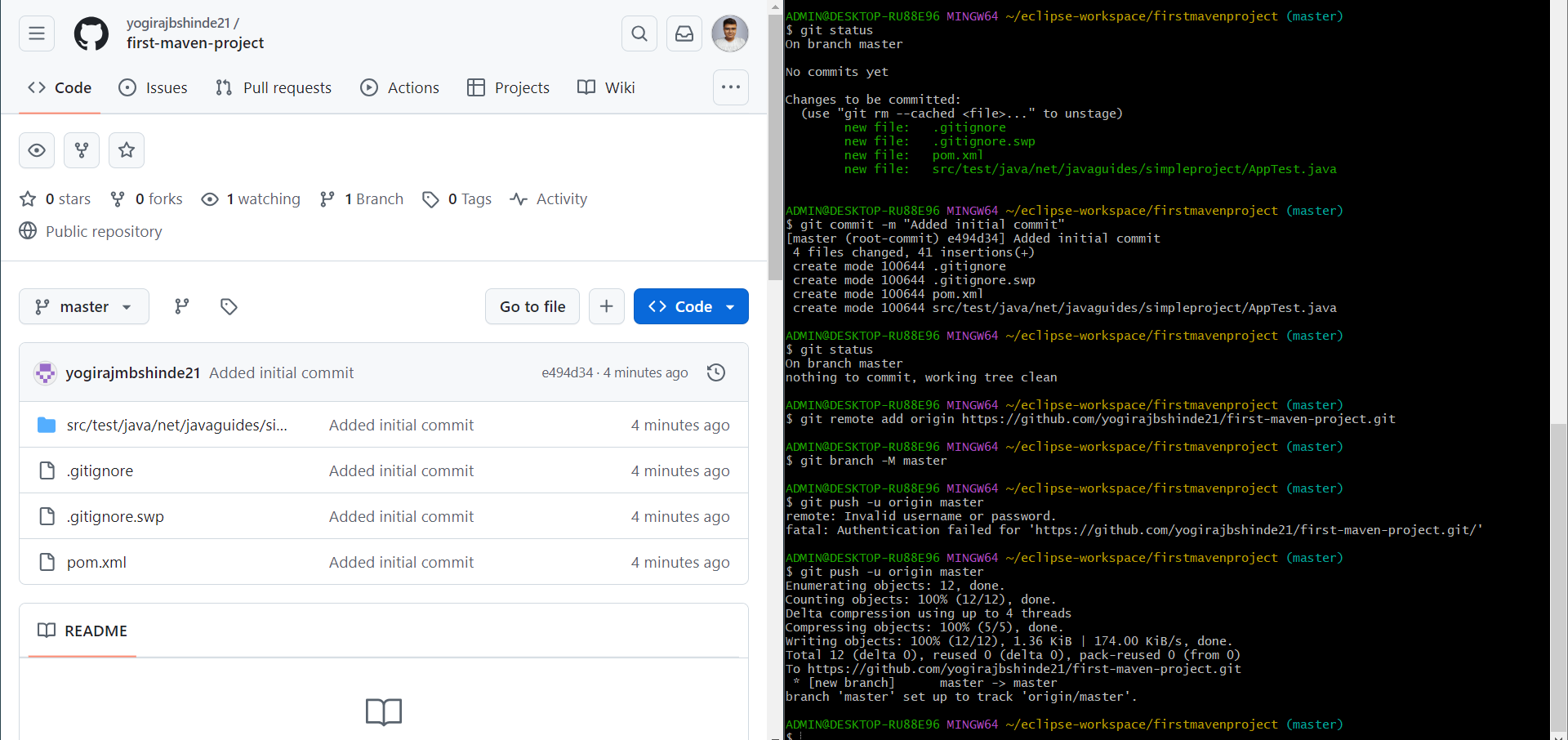
1. **Run the Maven Project:**

* Right-click on AppTest.java and select Run as -> JUnit Test.



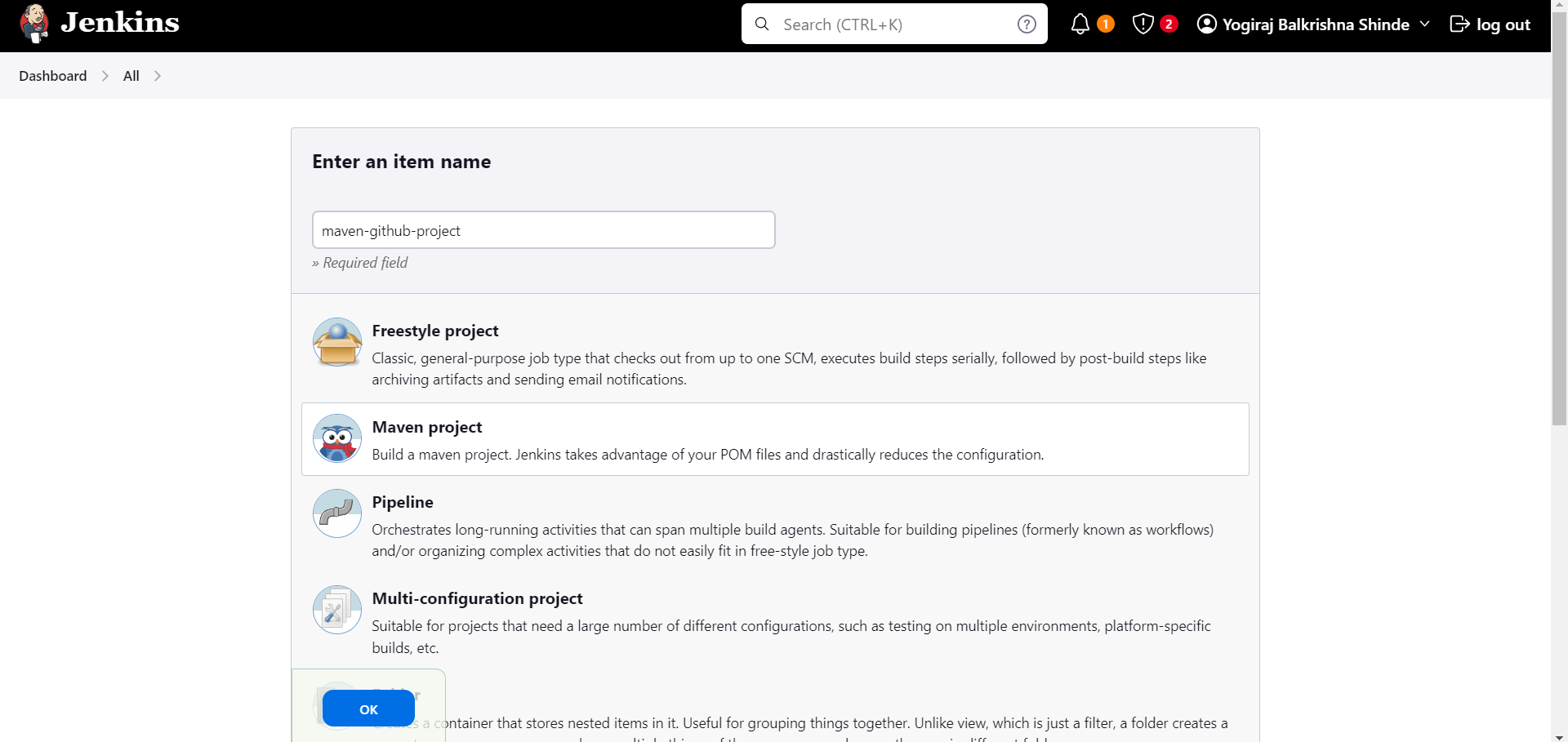
1. **Push the Maven Project to GitHub:**

* Commit and push the project to your GitHub repository.

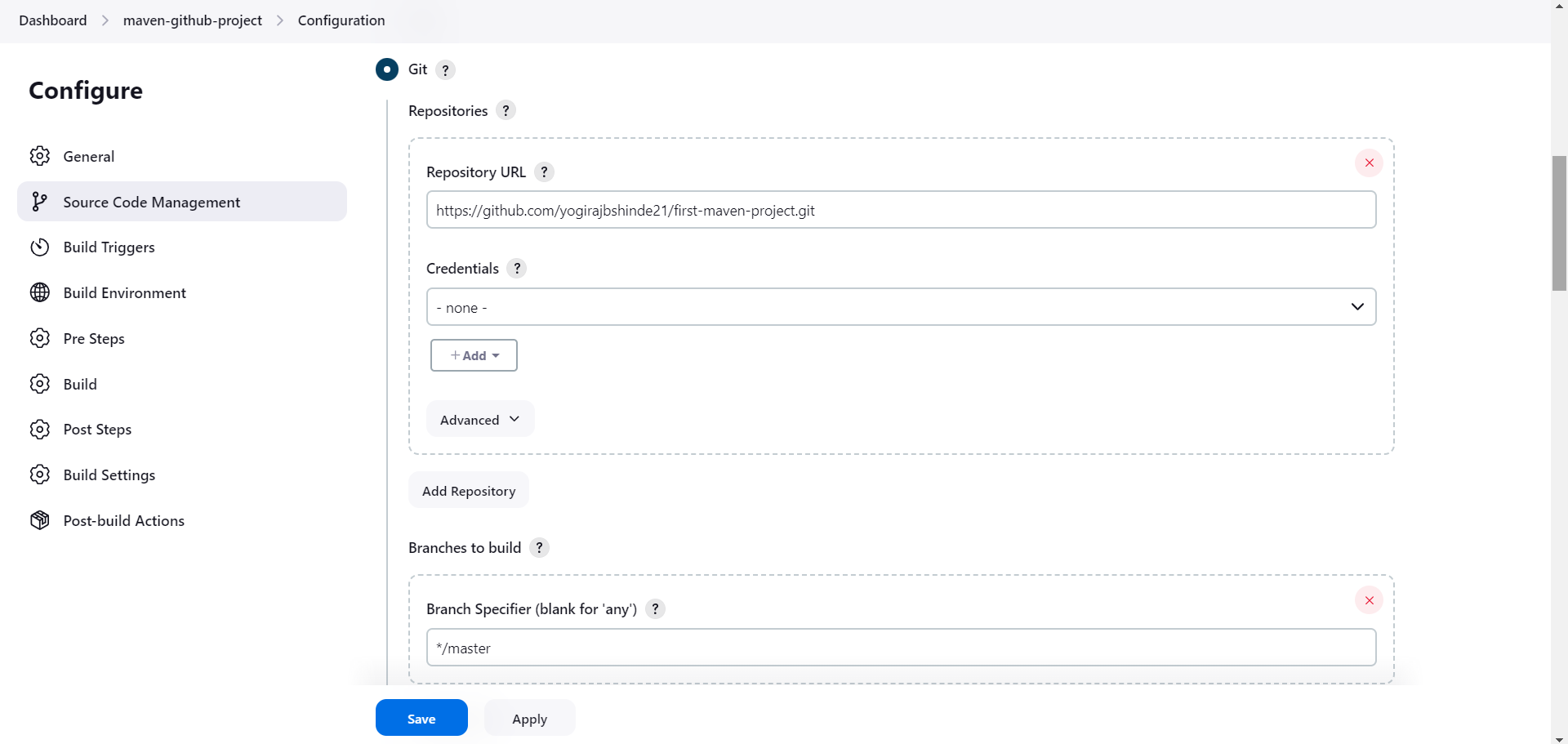
 

1. **Integrate with Jenkins:**

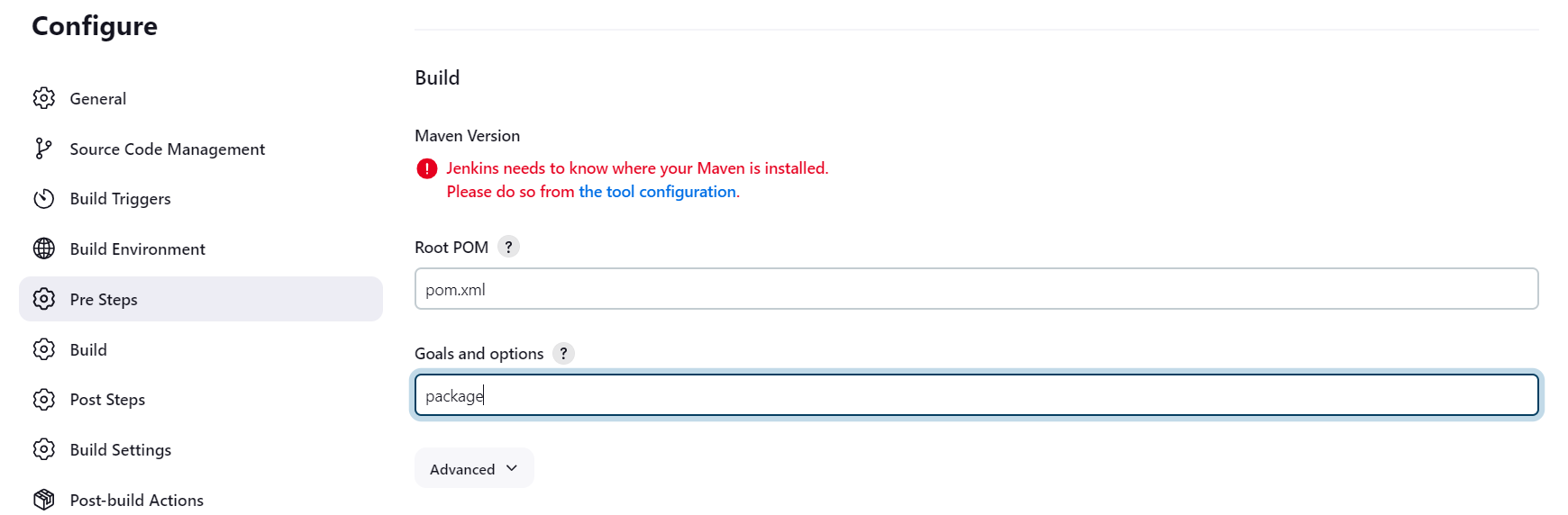
* Create a new job in Jenkins and select **Maven Project**.



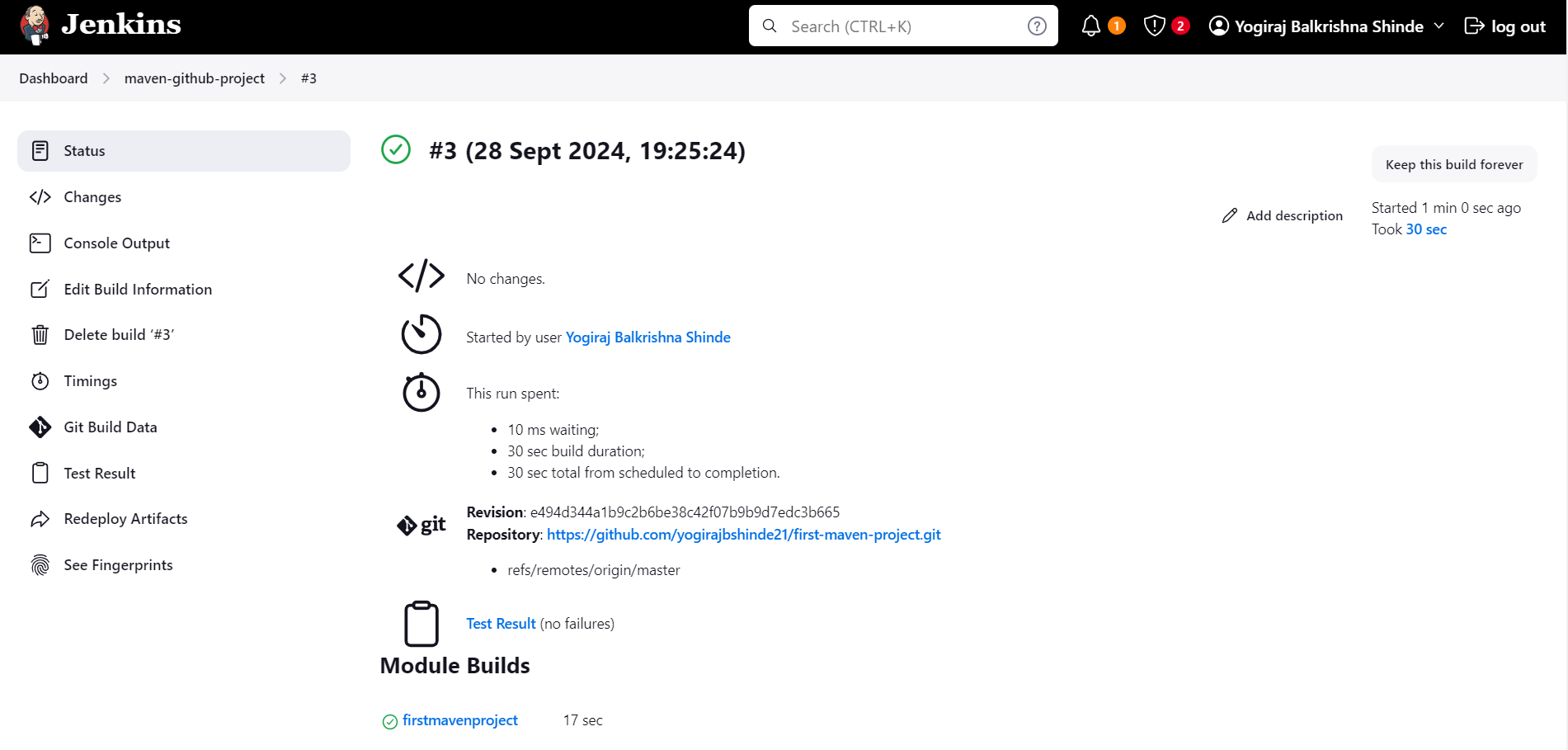
* Configure GitHub repository details in the Source Code Management section.

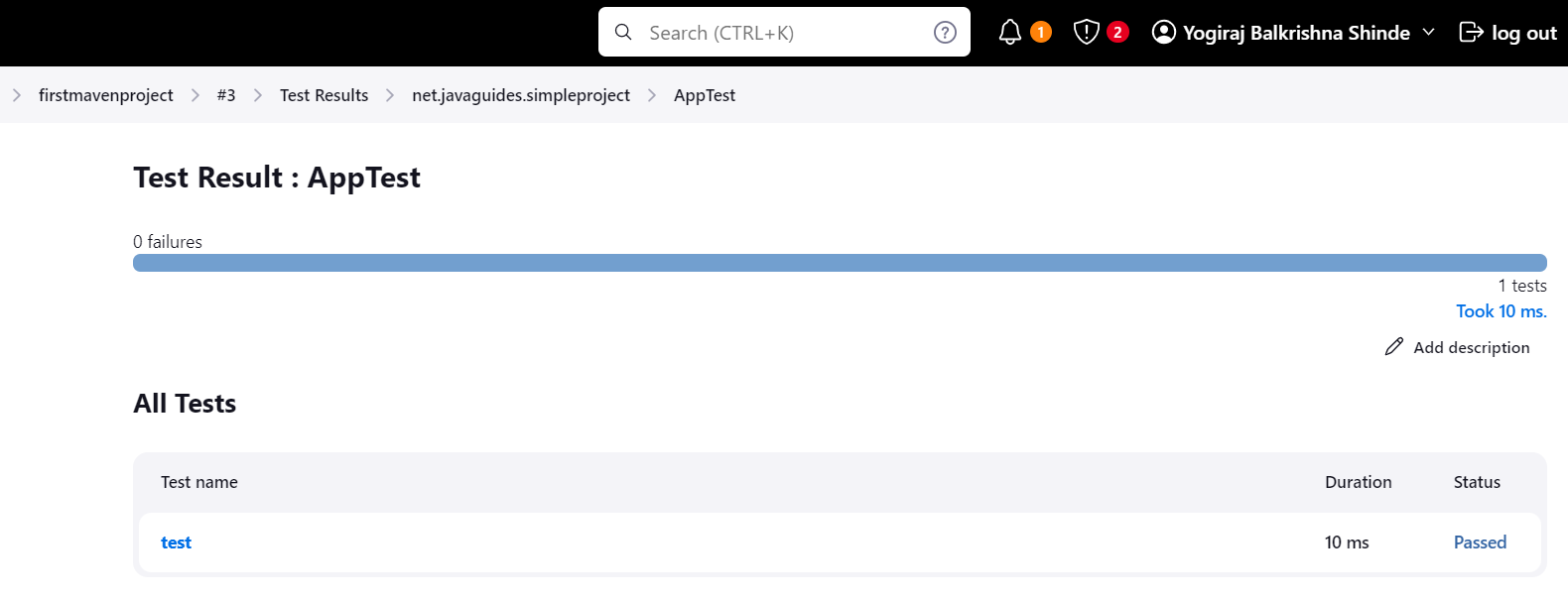


* Set build triggers and goals in the build section (e.g., clean install).



* Build the project.





**Conclusion:**

Successfully executed a simple Maven project in Eclipse, pushed it to GitHub, and integrated it with Jenkins for automated builds.