**Maven**

**02/07/24**

**What is maven?**

Maven is a powerful build tool that simplifies the management of Java projects.

Apache group then developed Maven which can build multiple projects together, publish projects information, deploy projects, share JARs across several projects, and help in the collaboration of teams.

It generates the Artifacts.

The artifacts are:

1. .jar (for building the same similarities of a class)
2. .war (for web applications [jar + list of objects])
3. .ear (for big enterprises[jar + war] )

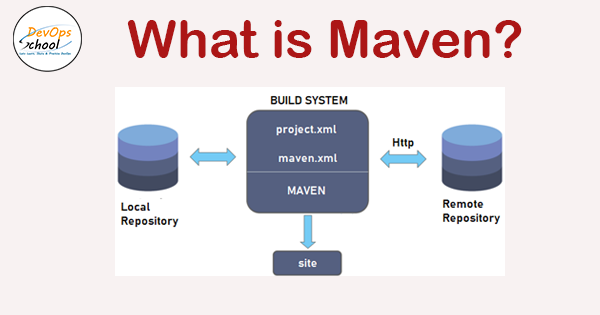
Maven had two repositories

1. **Local Repository**: Maven local repository is a folder location on your machine.

Maven local repository keeps your project's all dependencies.

When you run a Maven build, then Maven automatically downloads all the dependency jars into the local repository.

1. **Remote Repository:**
2. JFrog
3. Maven



**POM** stands for Project Object Model. It is the fundamental unit of work in Maven. It is an XML file that resides in the project's base directory as pom.xml.

The POM contains information about the project and various configuration detail used by Maven to build the project.

* All POM files require the **project** element and three mandatory fields: **groupId, artifactId, and version**.
  1. **GroupId:** This is an Id of the project's group. This is generally unique to an organization or a project.

For example, a banking group com.company.bank has all bank-related projects.

* 1. **ArtifactId:** This is an Id of the project. This is generally the name of the project. For example, consumer banking. Along with the groupId, the artifactId defines the artifact's location within the repository.
  2. **Version:** This is the version of the project. Along with the groupId, It is used within an artifact's repository to separate versions from each other.

For example,

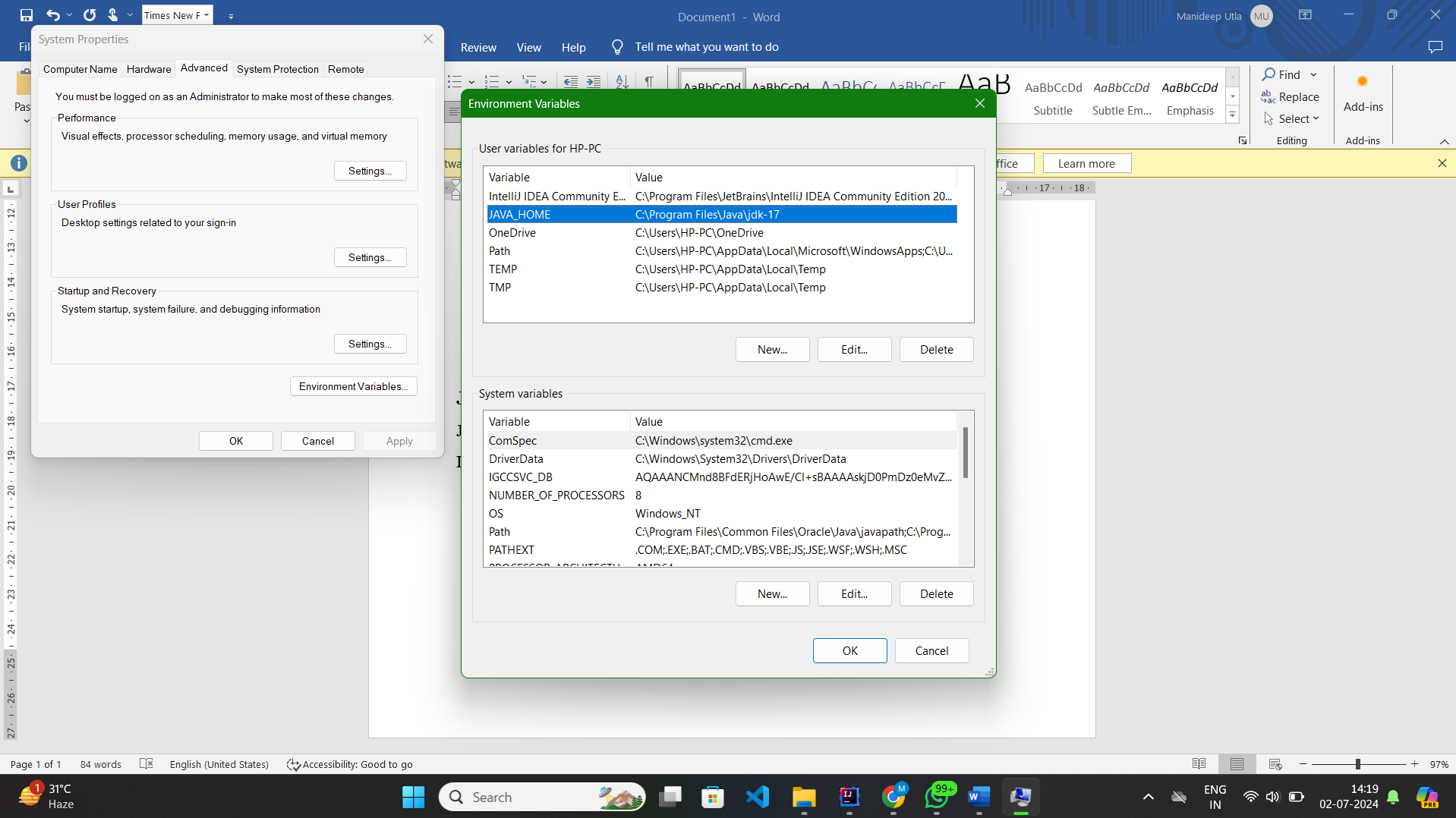
com.company.bank:consumer-banking:1.0

org.company.bank:consumer-banking:1.1.

**JAVA HOME:**

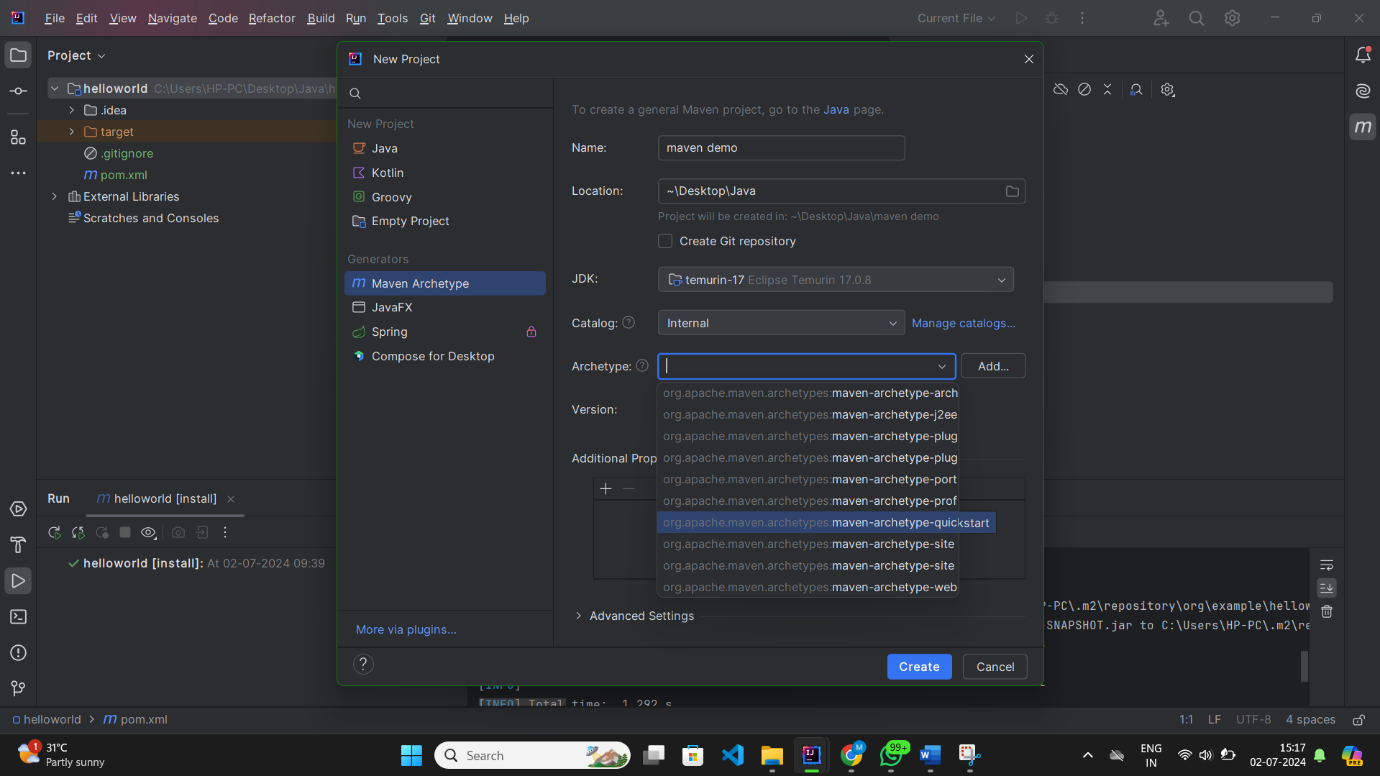
JAVA\_HOME is used to run the External softwares which depends on Java.

It acts as a bridge between two softwares depends on eachother.

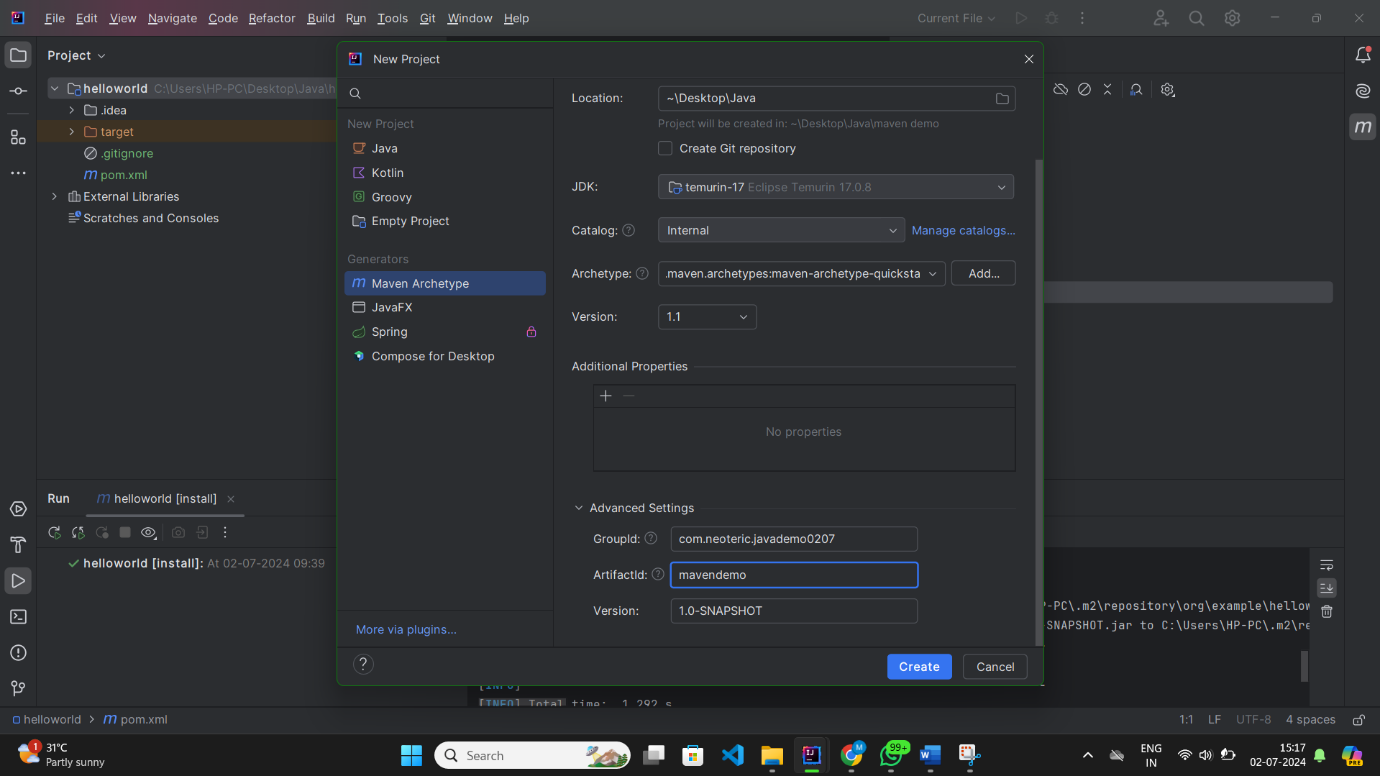


**Fig:** It show the java home path.

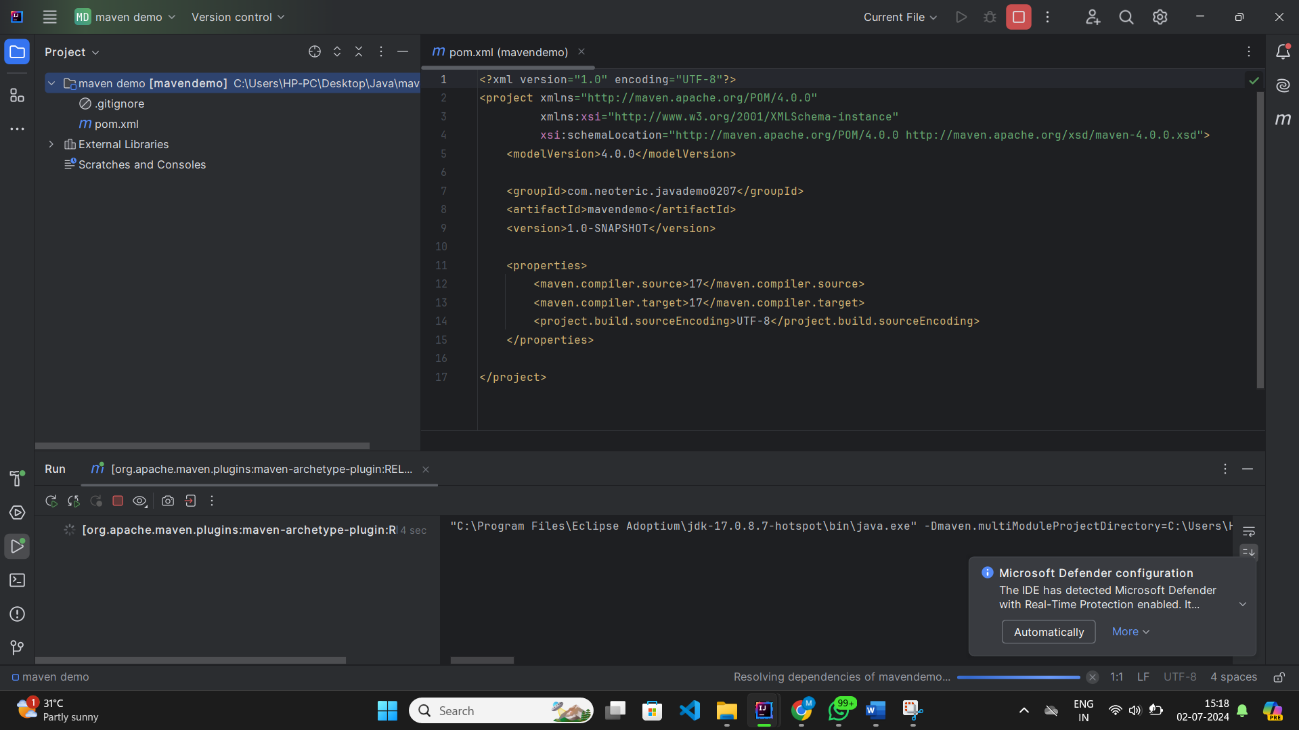
**Excute on intellij:**

****

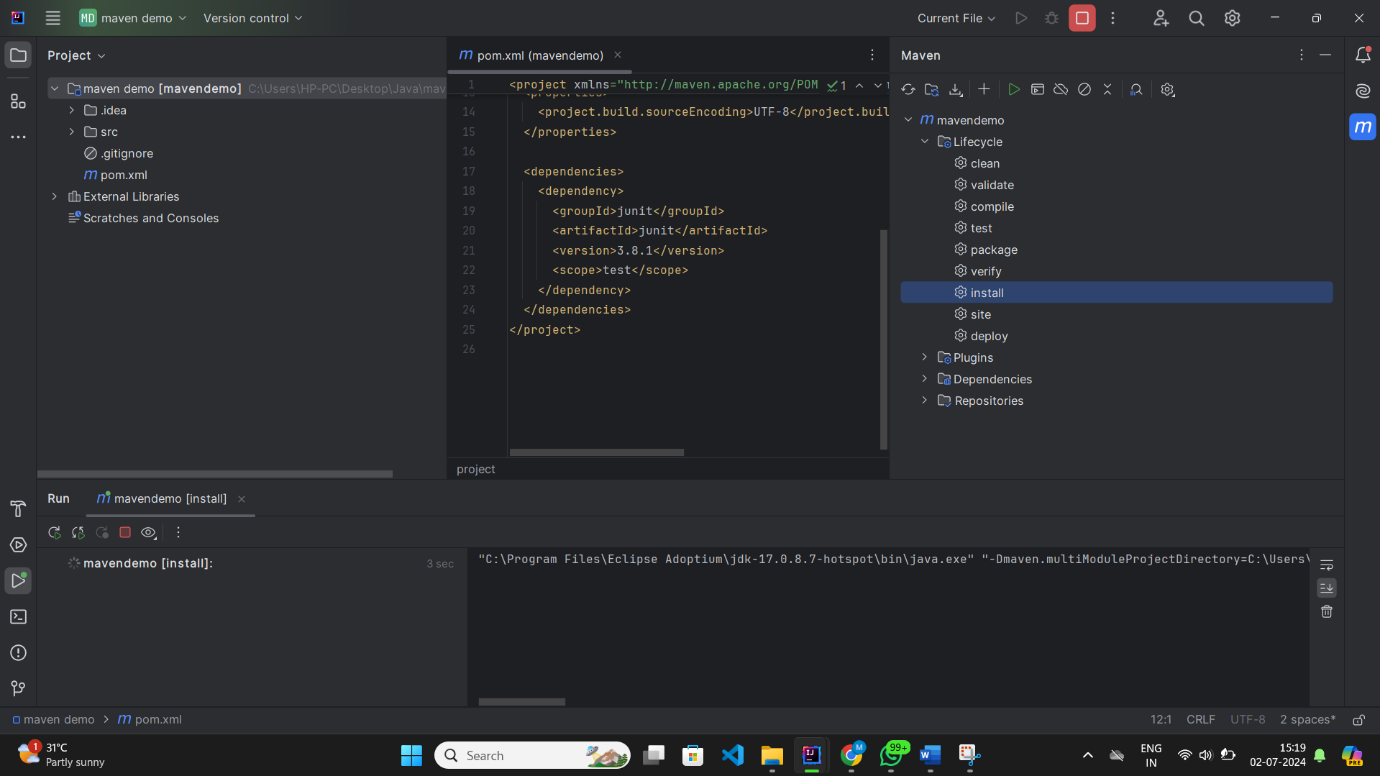
**Fig:** It is shows the creating maven file using maven archetype (predefined – quickstart archetype).



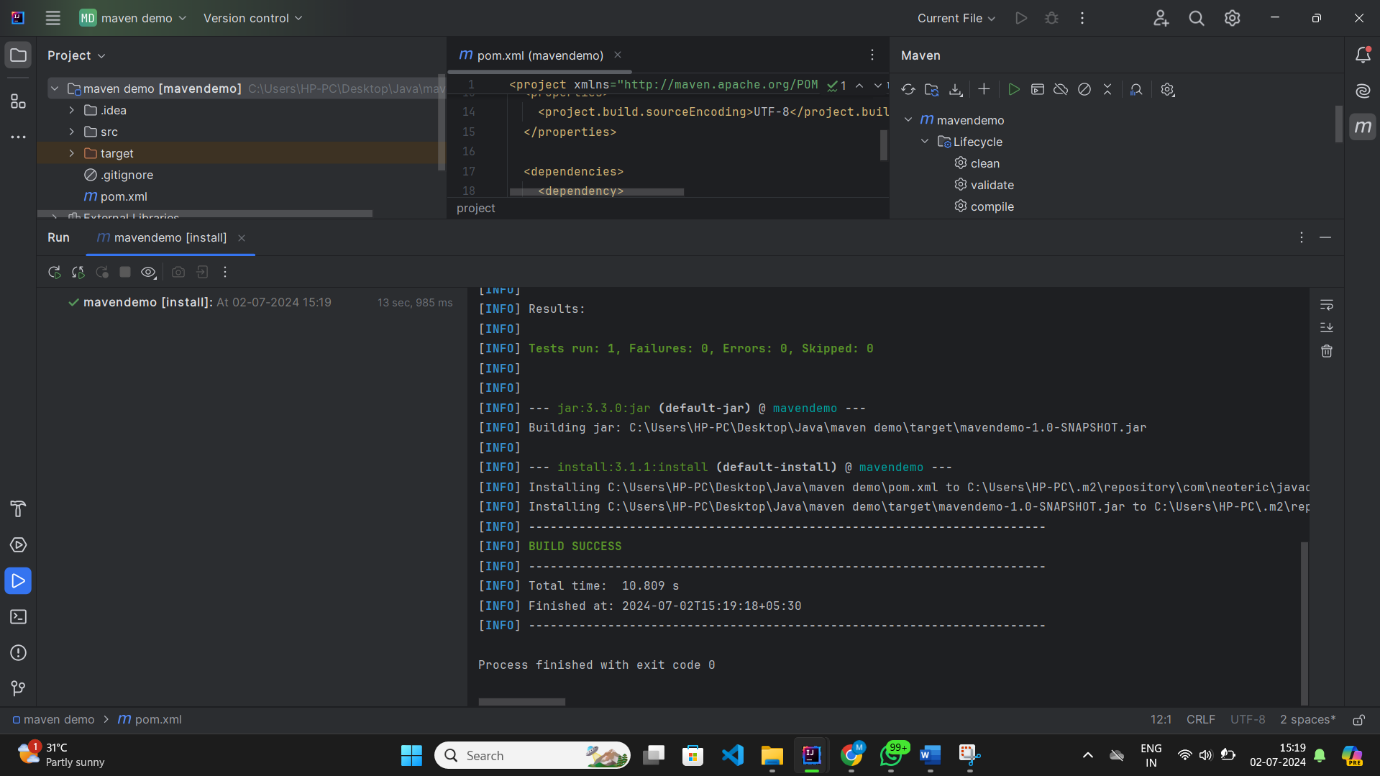
**Fig:** It Shows the Group Id, Artifactid, and version.



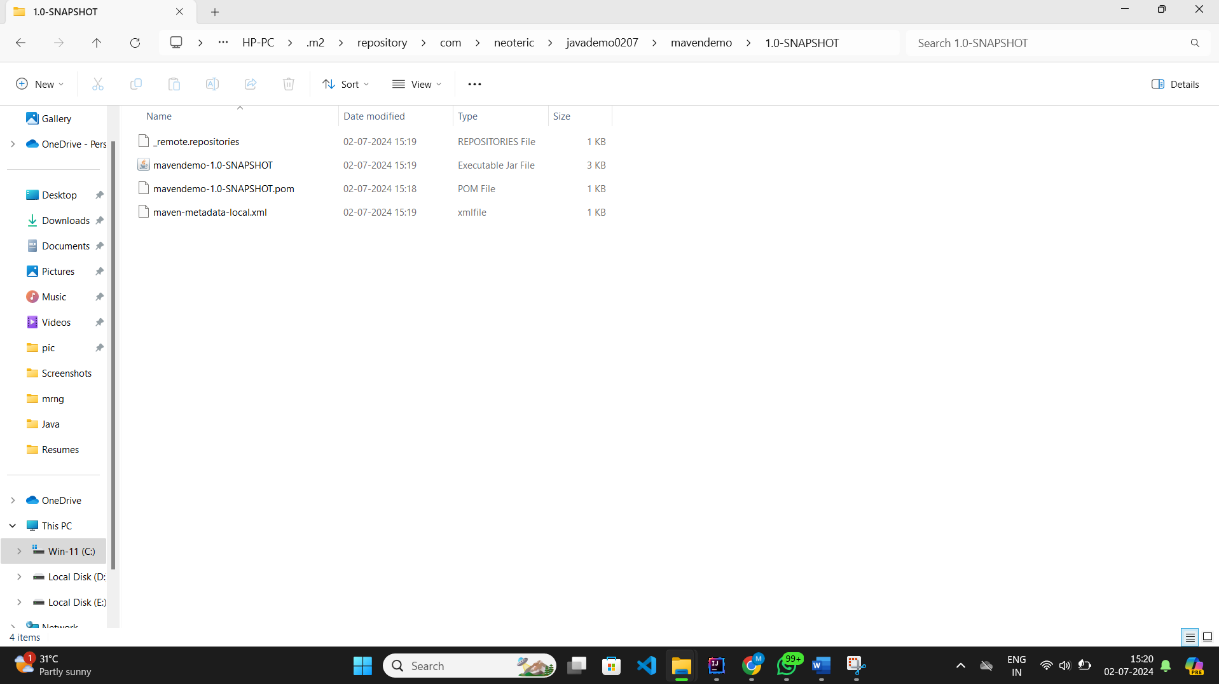
**Fig:** It shows the pom.xml file on the screen in that we observe groupid, artifactid, version.



**Fig:** It shows the maven lifecycle and I install the mavenfile.



**Fig:** It shows the installing and file address



**Fig:** It shows the file location which I created using intellij

**Thank you…….**