Experiment no 7

Title:

Demonstrate Maven Build Life Cycle

Objective:

To understand and implement the Maven Build Life Cycle and observe how different build phases work in a Java project.

Theory (Simple Words):

Maven is a **build tool** used in Java projects. It helps us **compile**, **test**, **package**, and **install** Java programs easily.

Maven uses a file called **pom.xml** to manage:

- Project info
- Dependencies (external JARs)
- Plugins (for extra features)

Maven has **lifecycles** (like steps) such as:

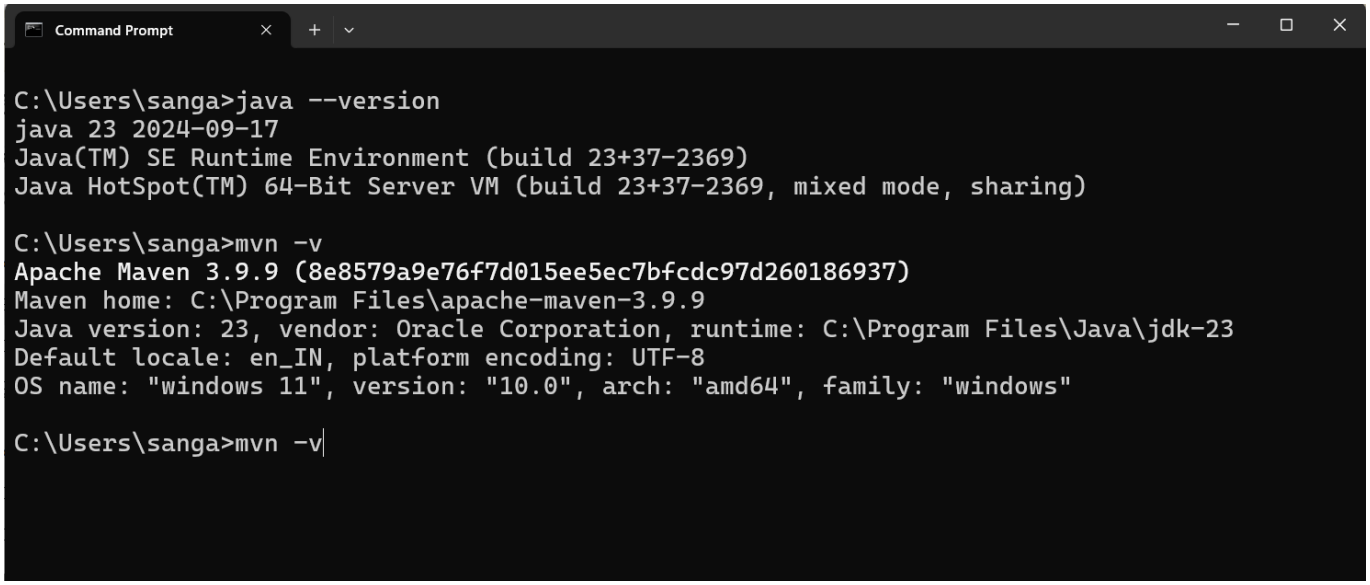
- clean – remove old files
- compile – compile Java code
- test – run test cases
- package – create JAR file
- install – put JAR in local repo
- deploy – send JAR to remote repo

Materials Needed:

- Windows PC
- Java JDK (8 or above)
- Maven
- Any code editor (like VS Code or IntelliJ)

Experiment Steps

Step 1: Install Java & Maven



Step 2: Create Maven Project (Manually)

Name of folder is : MavenDemo

1.Create a folder:

mkdir MavenDemo

cd MavenDemo

2. Create folders manually:

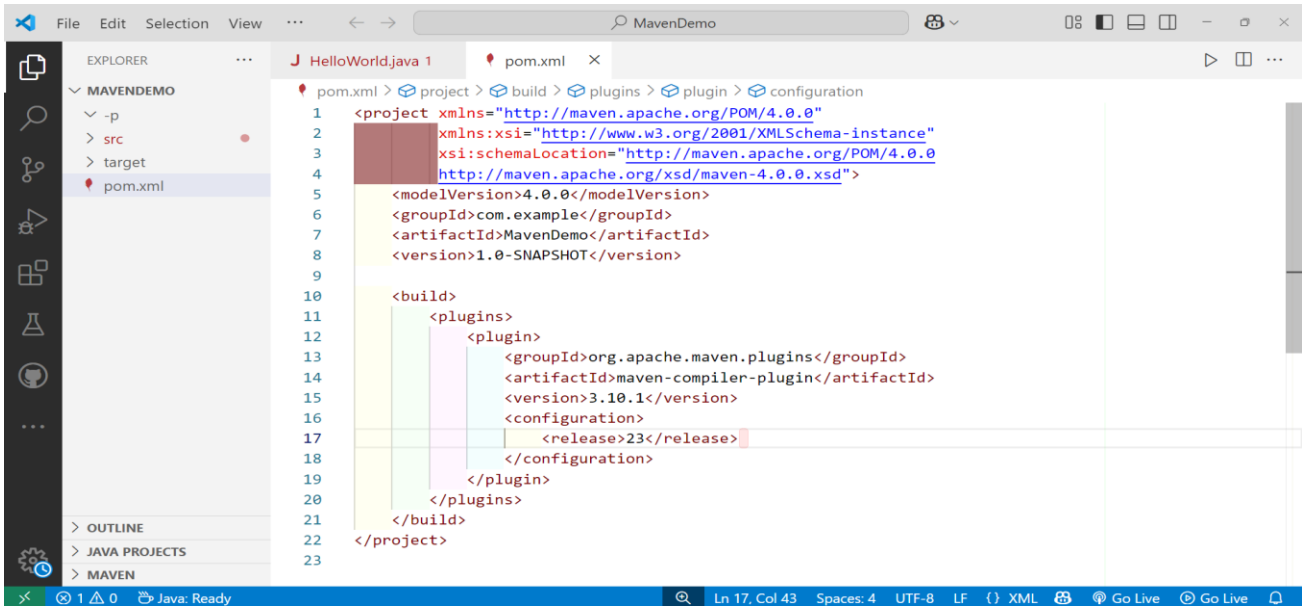
mkdir -p src\main\java

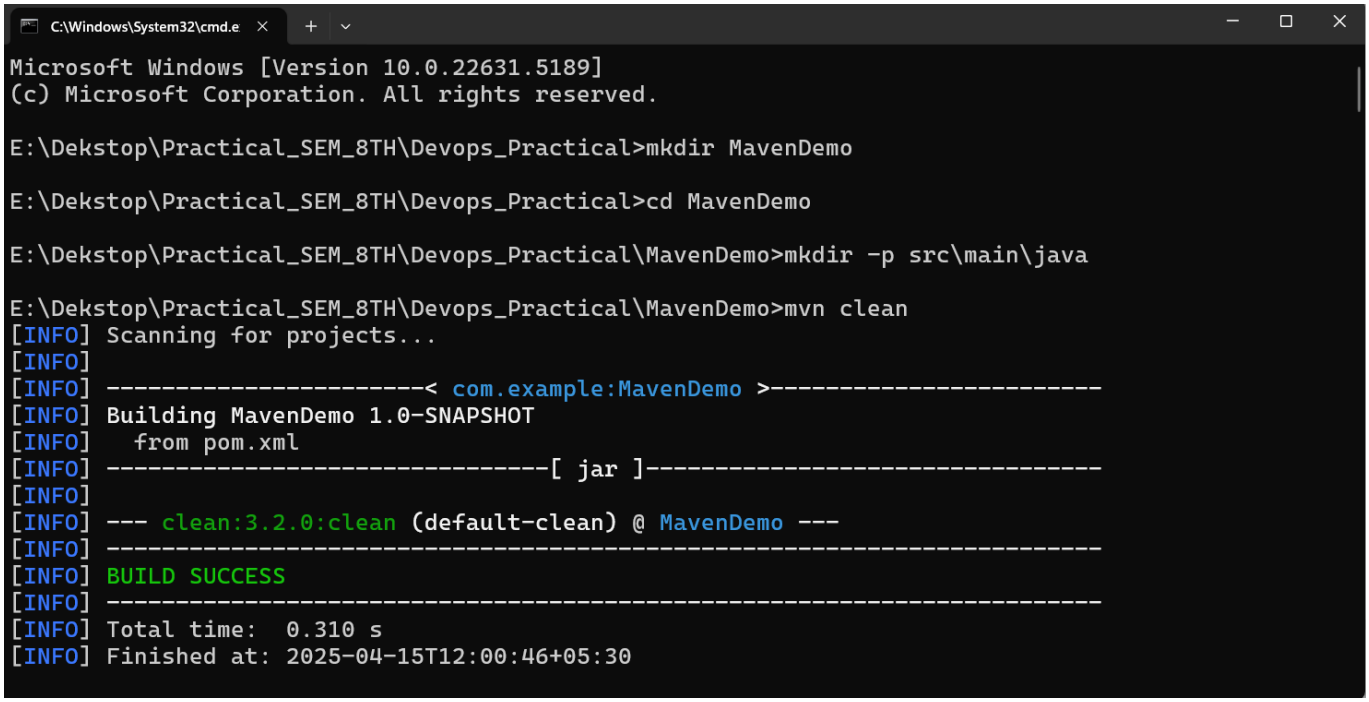
3. Create Java file inside src\main\java\HelloWorld.java

CopyEdit

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, Maven!");
    }
}
```

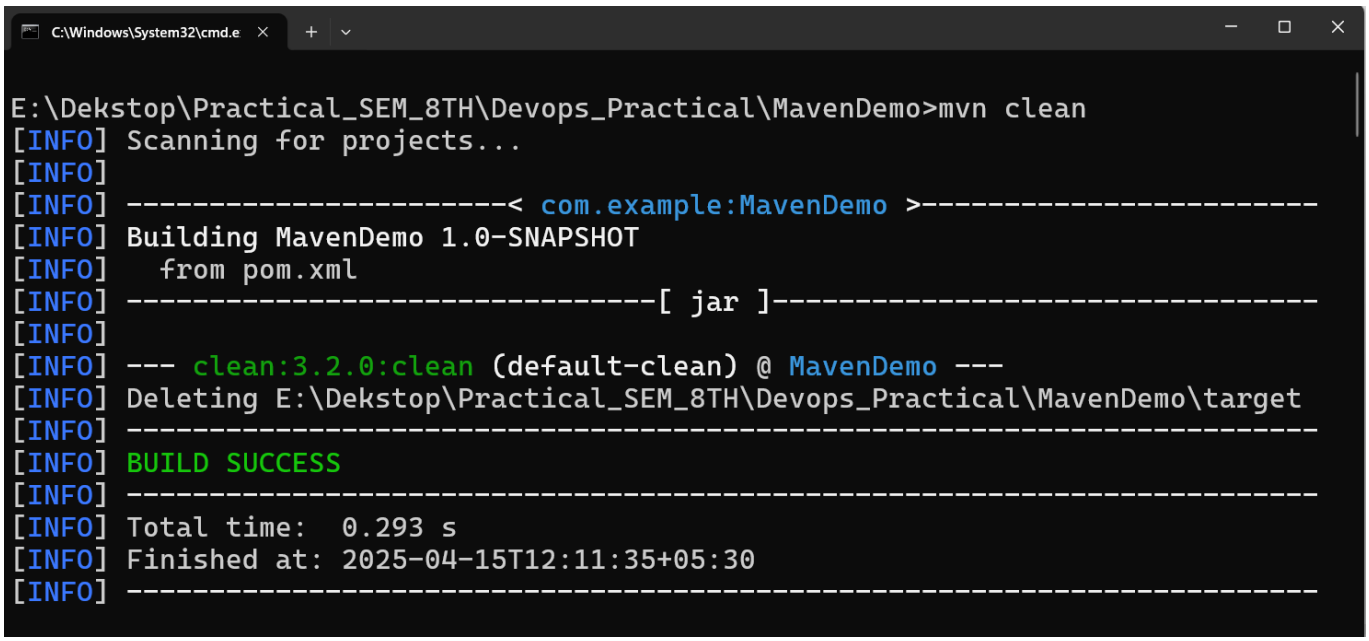
4. Create pom.xml in MavenDemo folder





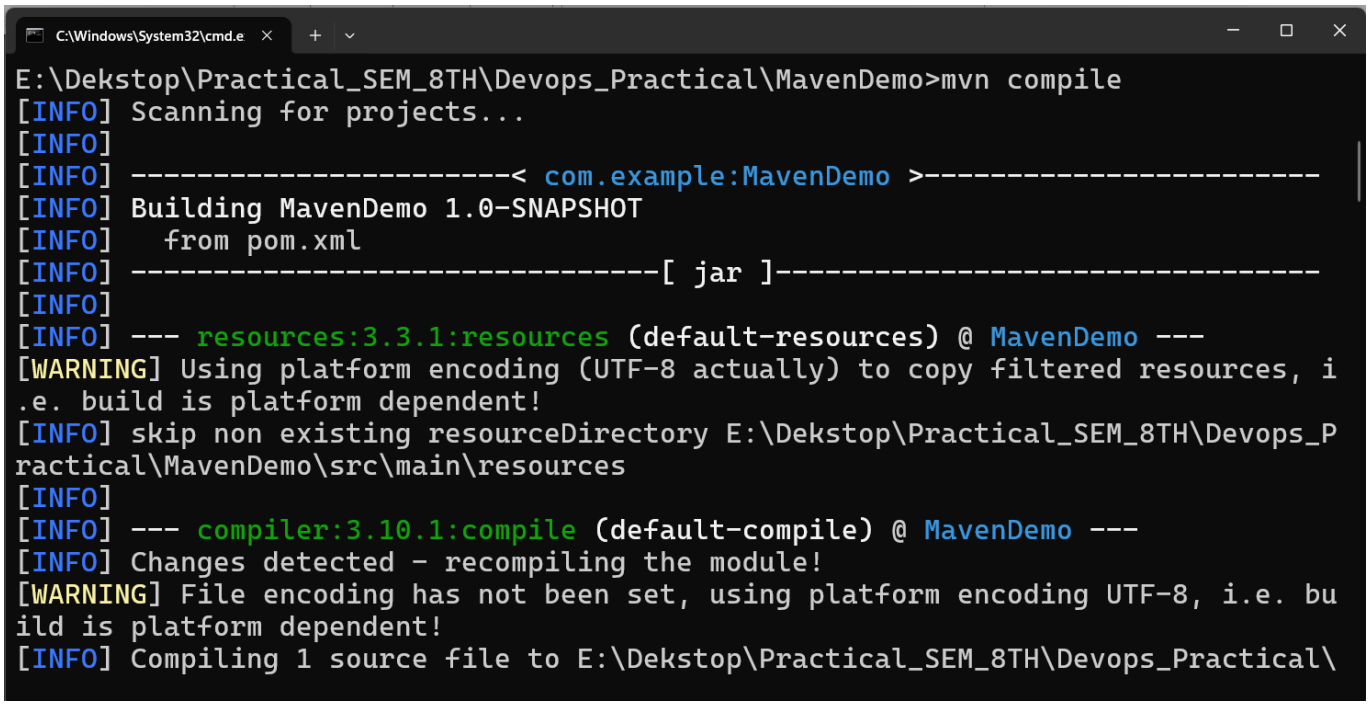
Step 3: Run Maven Build Phases

1. Clean (Deletes target folder if exists) : mvn clean

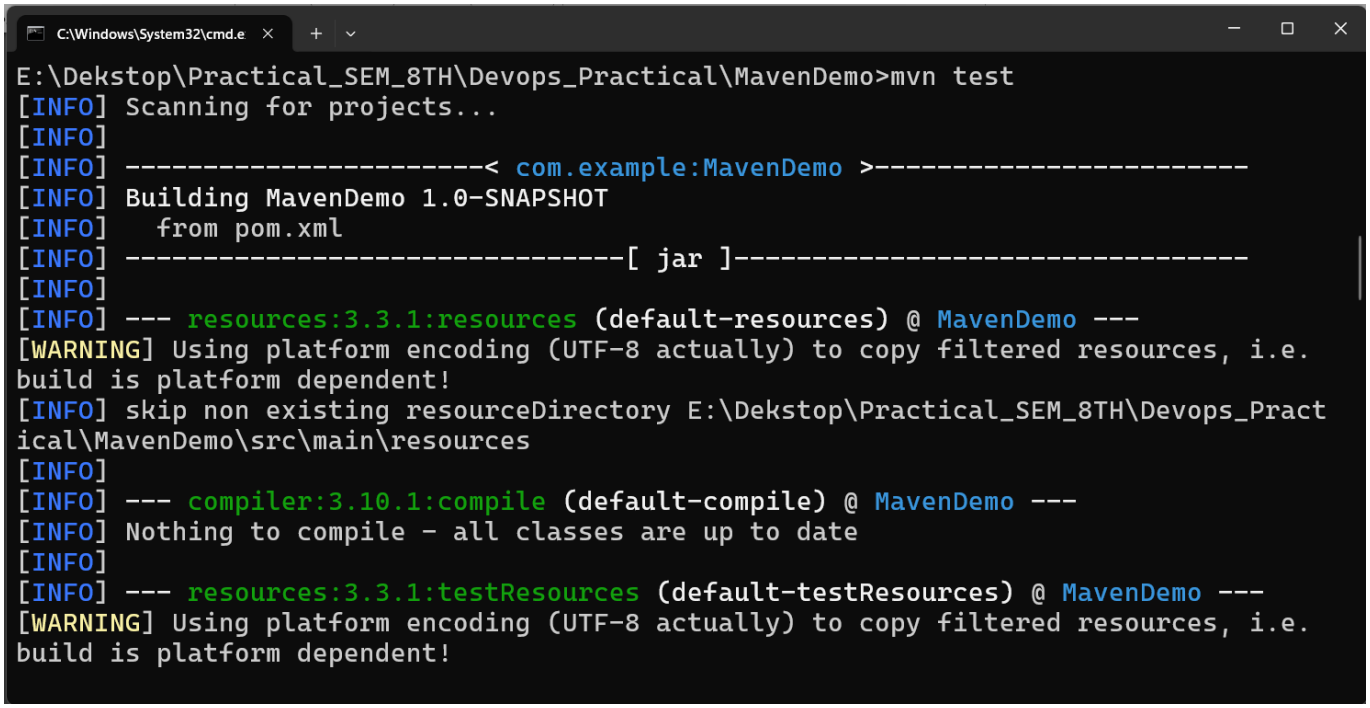


2. Compile (Compiles your Java file into .class file) : mvn compile

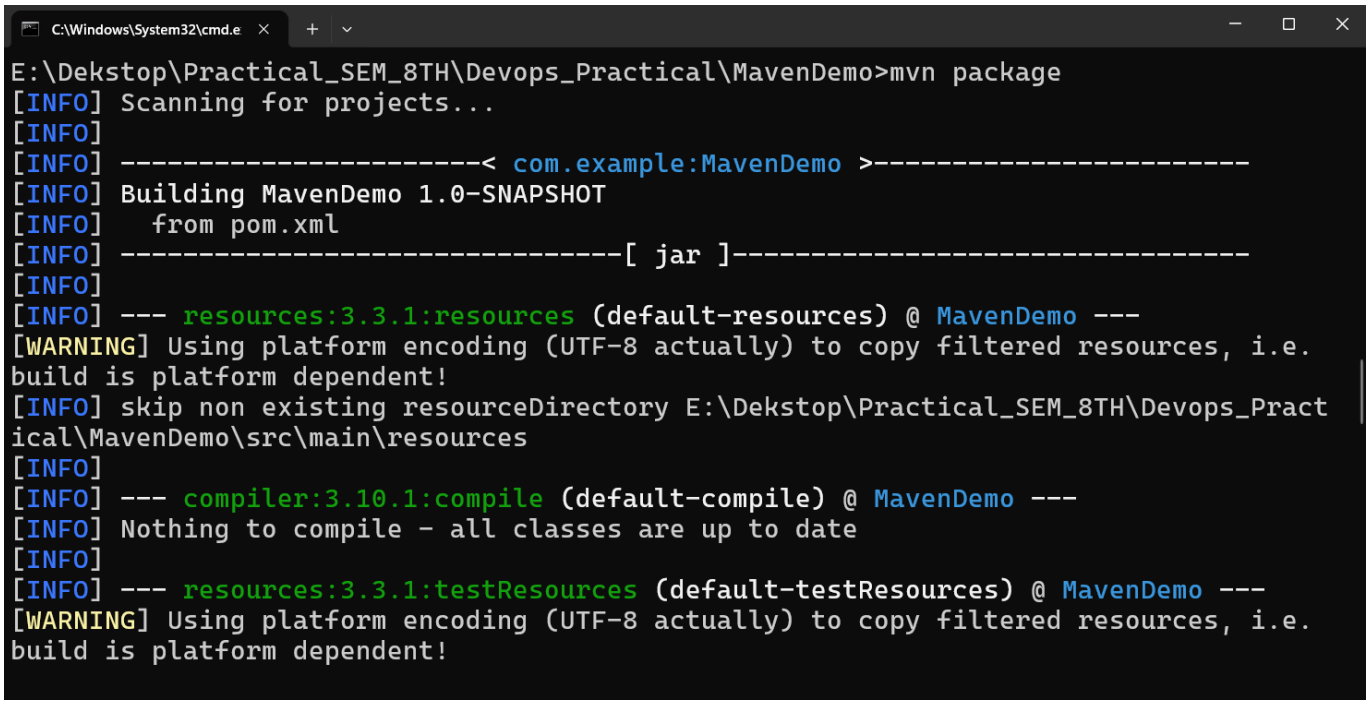
-



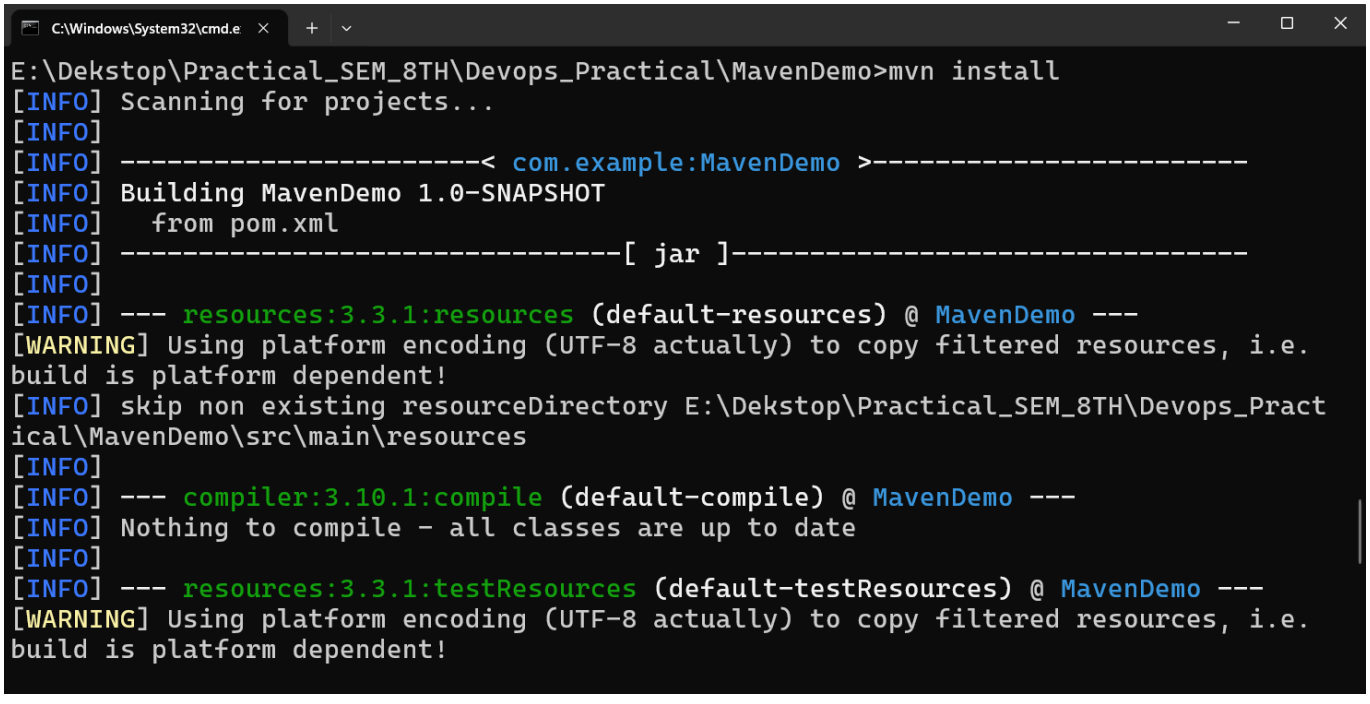
3. Test (Even if there are no test cases, it will run test phase) : mvn test

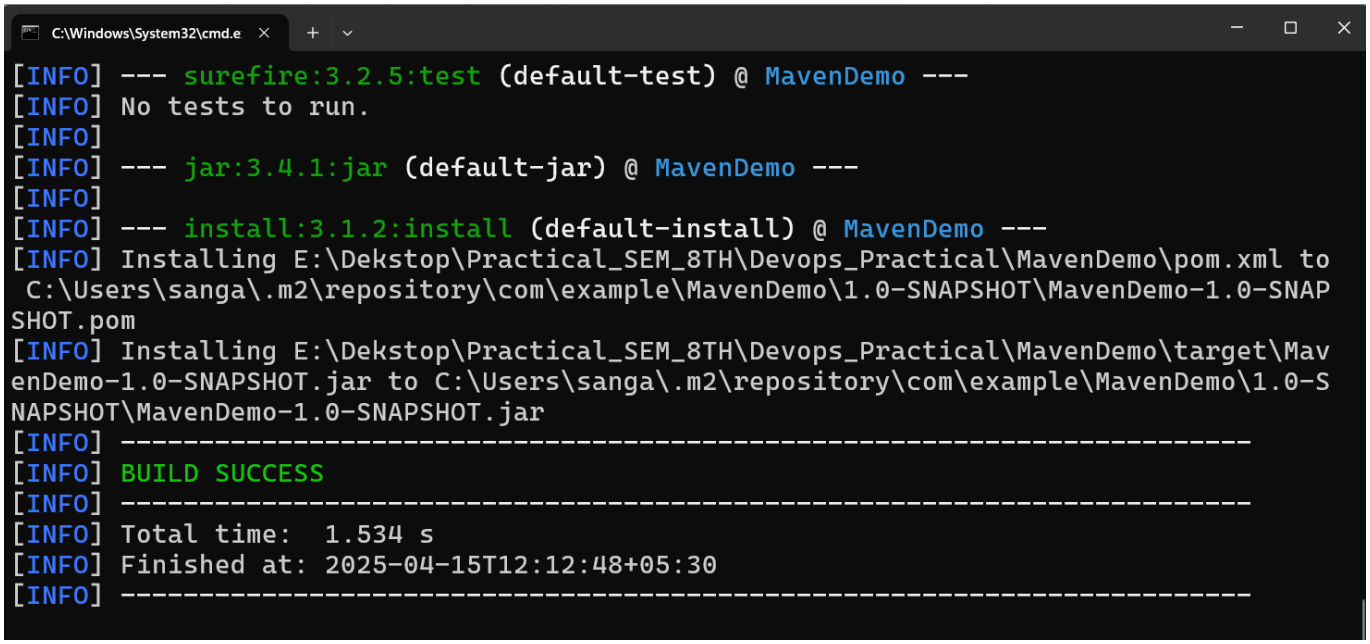


4. Package (Packages into .jar file) : mvn package



5. Install (Installs the .jar in local Maven repository) : mvn install

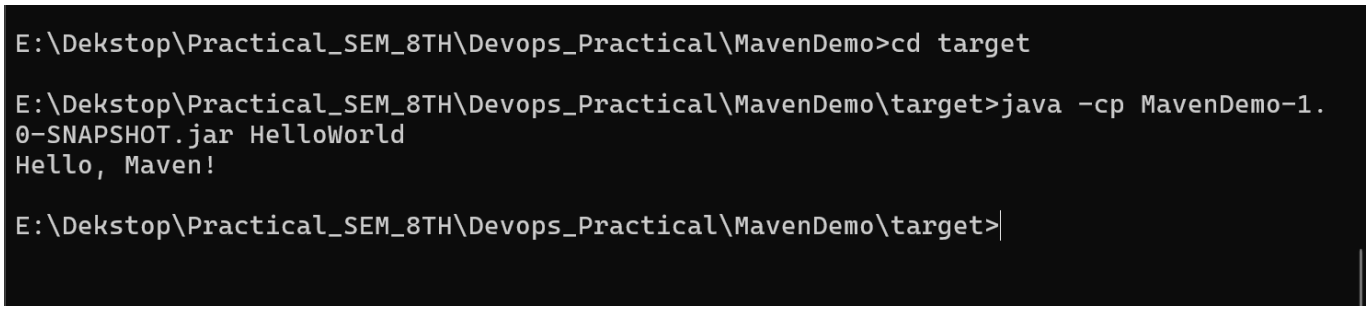




Step 4: Run the .jar File

cd target

java -cp MavenDemo-1.0-SNAPSHOT.jar HelloWorld



Conclusion:

This experiment helped us learn how Maven works, including:

- Creating a Maven project manually
- Understanding pom.xml
- Using build lifecycle commands like clean, compile, package, etc.
- Generating and running a .jar file

Maven simplifies Java project management by automating builds, handling dependencies, and standardizing structure.