**Lab Experiment-8**

**SUBJECT** – Build and Release Management

**NAME** –Devesh yadav

**ROLL NO** – R171218038

**Aim: Build Profile using Maven**

A Build profile is a set of configuration values, which can be used to set or override default values of Maven build. Using a build profile, you can customize build for different environments such as Production v/s Development environments.

Profiles are specified in pom.xml file using its activeProfiles/profiles elements and are triggered in variety of ways. Profiles modify the POM at build time, and are used to give parameters different target environments (for example, the path of the database server in the development, testing, and production environments).

**Profile Activation**

A Maven Build Profile can be activated in various ways.

* Explicitly using command console input.
* Through maven settings.
* Based on environment variables (User/System variables).
* OS Settings (for example, Windows family).
* Present/missing files.

**Explicit Profile Activation**

In the following example, we will attach maven-antrun-plugin:run goal to test the phase. This will allow us to echo text messages for different profiles. We will be using pom.xml to define different profiles and will activate profile at command console using maven command.

Assume, we've created the following pom.xml in C:\MVN\project folder.

<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>com.companyname.projectgroup</groupId>

<artifactId>project</artifactId>

<version>1.0</version>

<profiles>

<profile>

<id>test</id>

<build>

<plugins>

<plugin>

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

<artifactId>maven-antrun-plugin</artifactId>

<version>1.1</version>

<executions>

<execution>

<phase>test</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<tasks>

<echo>Using env.test.properties</echo>

<copy file="src/main/resources/env.test.properties"

tofile="${project.build.outputDirectory}

/env.properties"/>

</tasks>

</configuration>

</execution>

</executions>

</plugin>

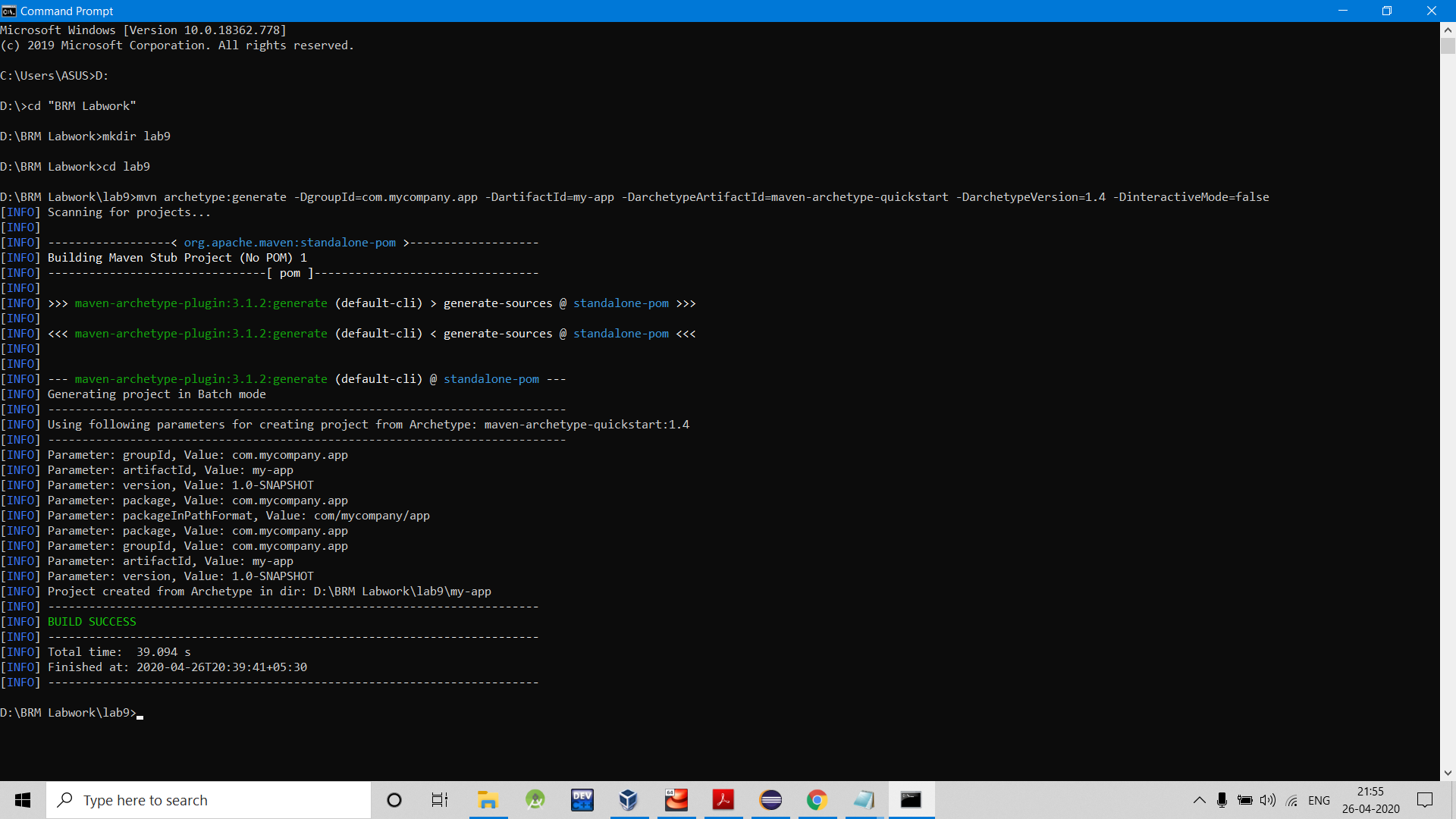
</plugins>

</build>

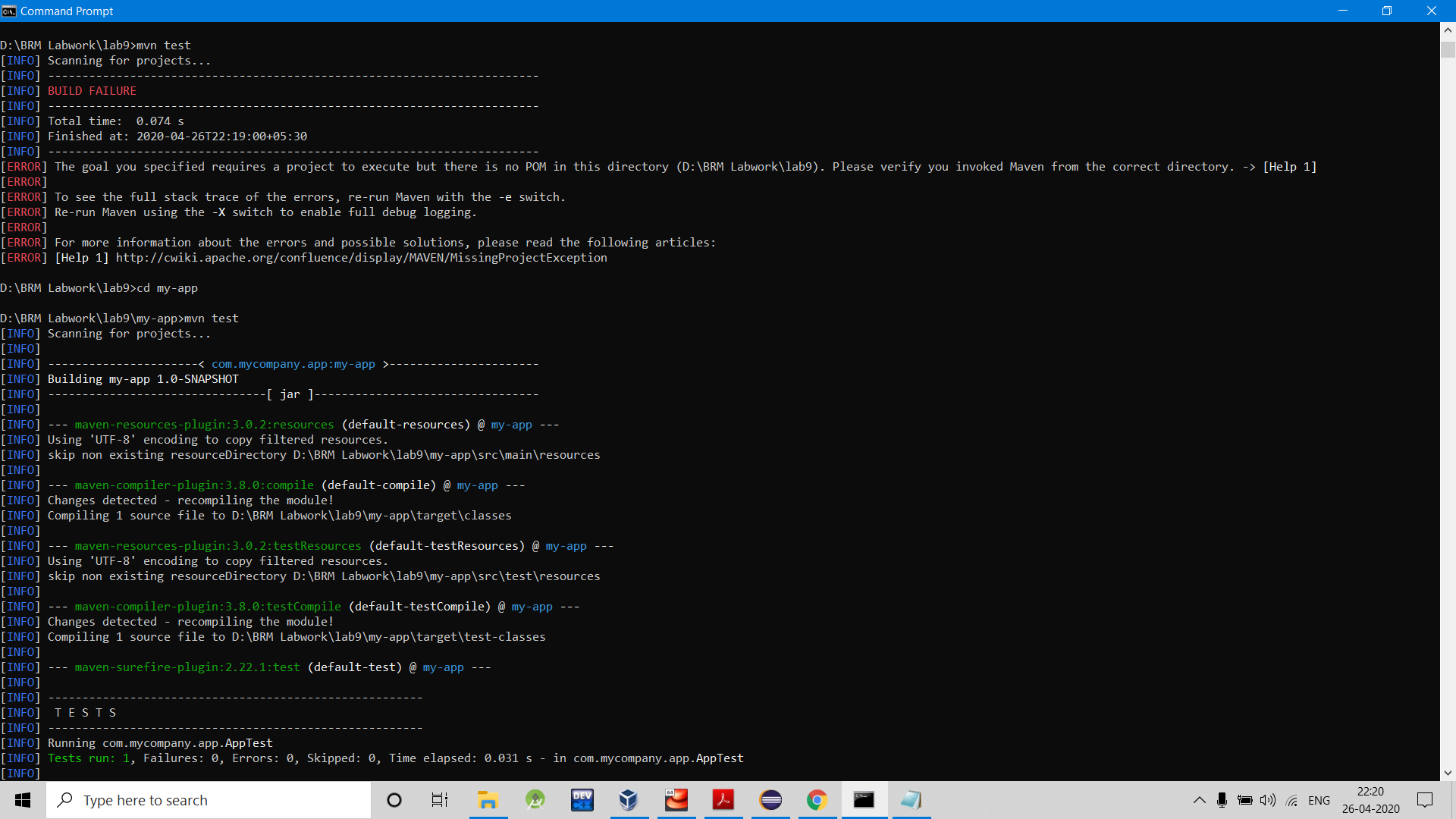
</profile>

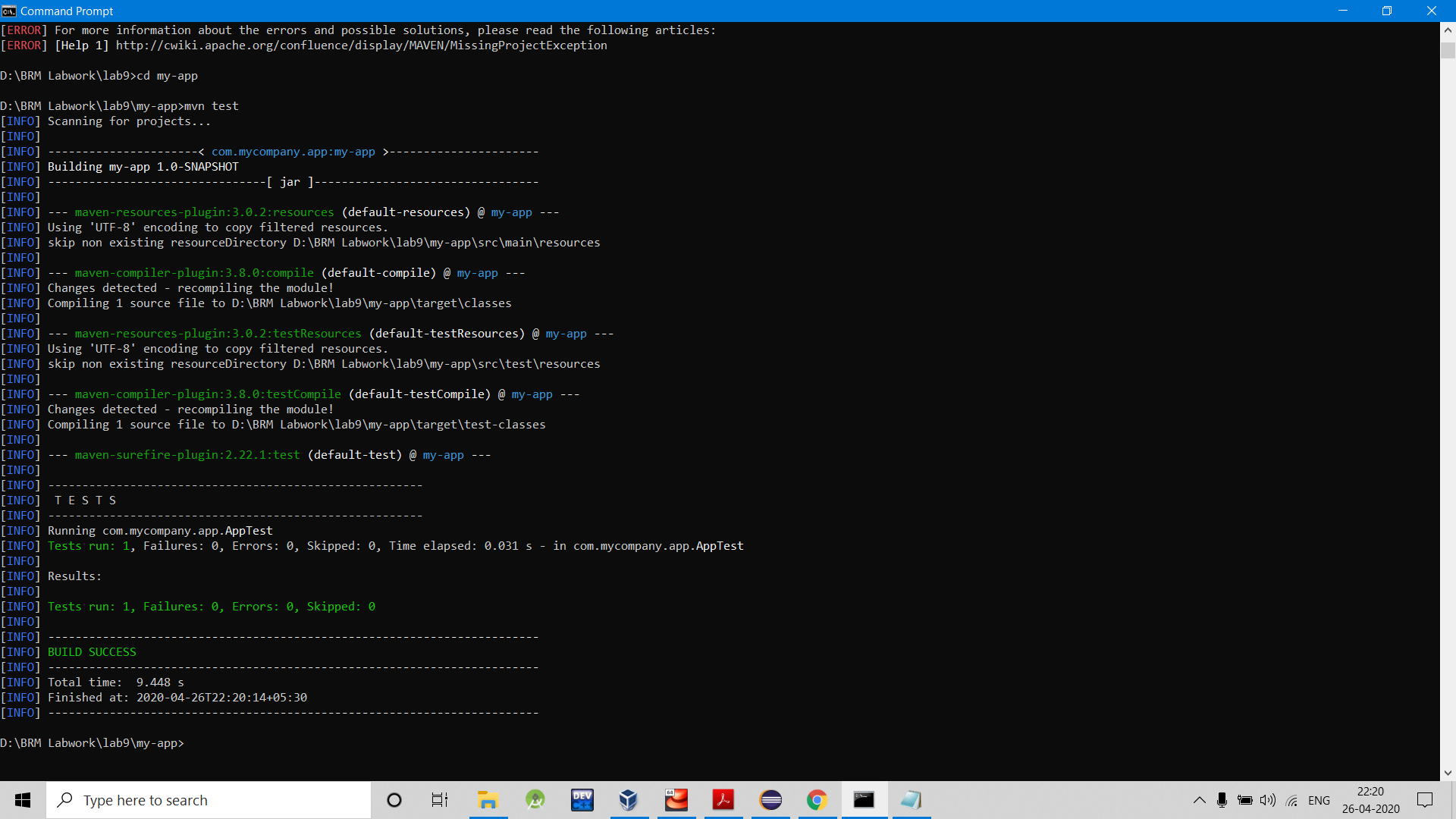
</profiles>

</project>



You always have to test inside the my-app folder which you created otherwise there will be a Build failure.

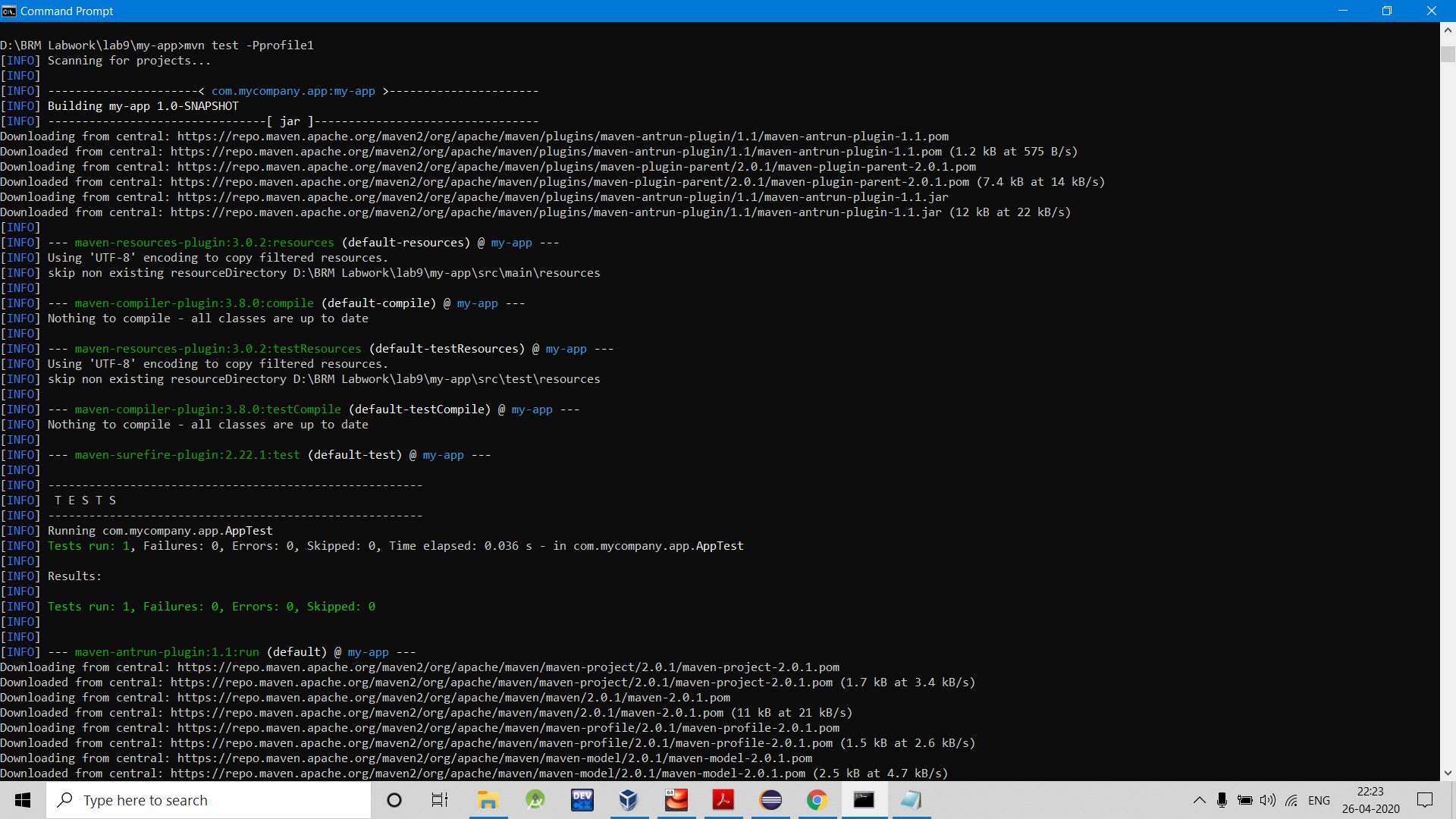


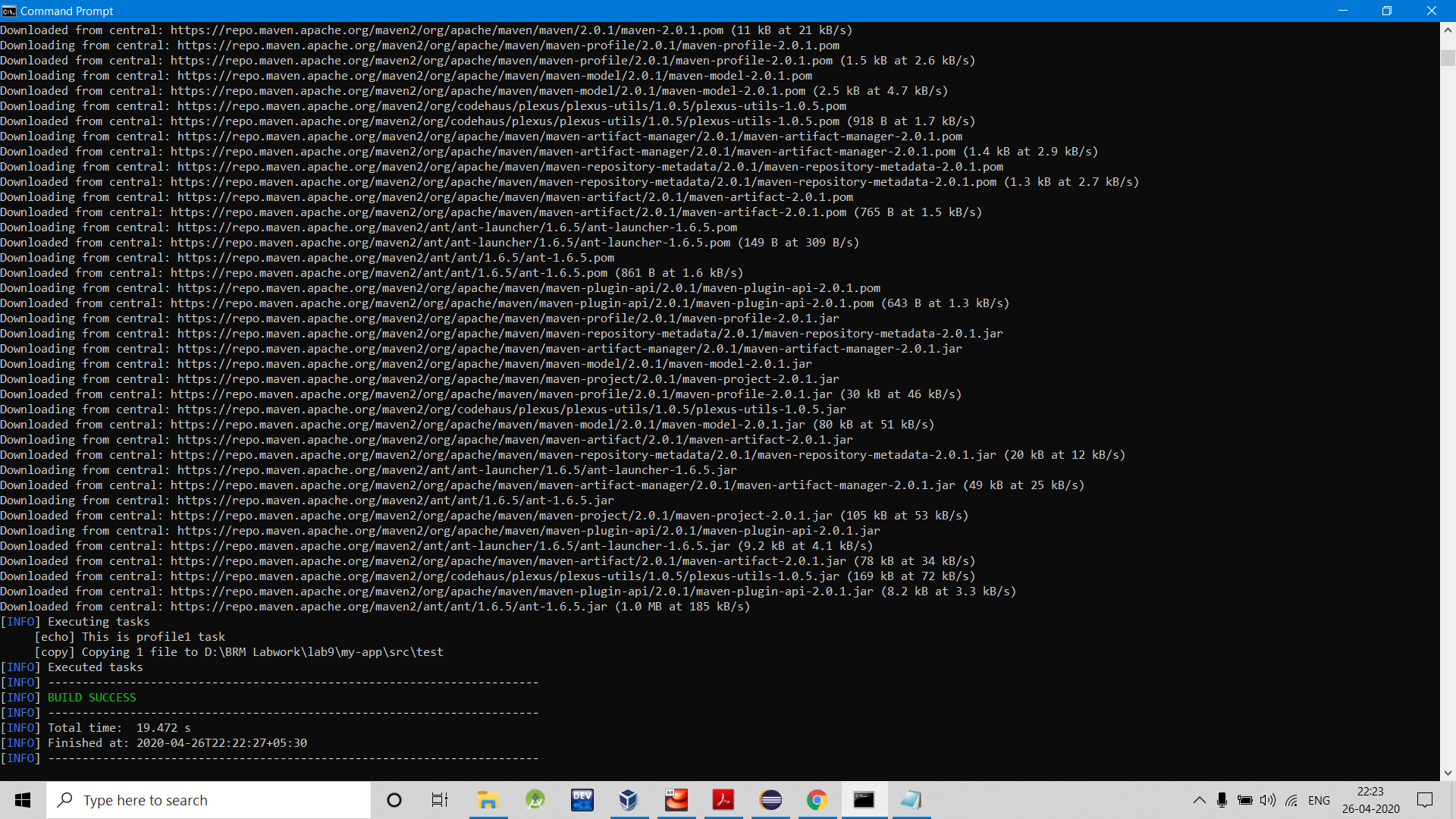


Now open the command console, go to the folder containing pom.xml and execute the following **mvn** command. Pass the profile name as argument using -P option.

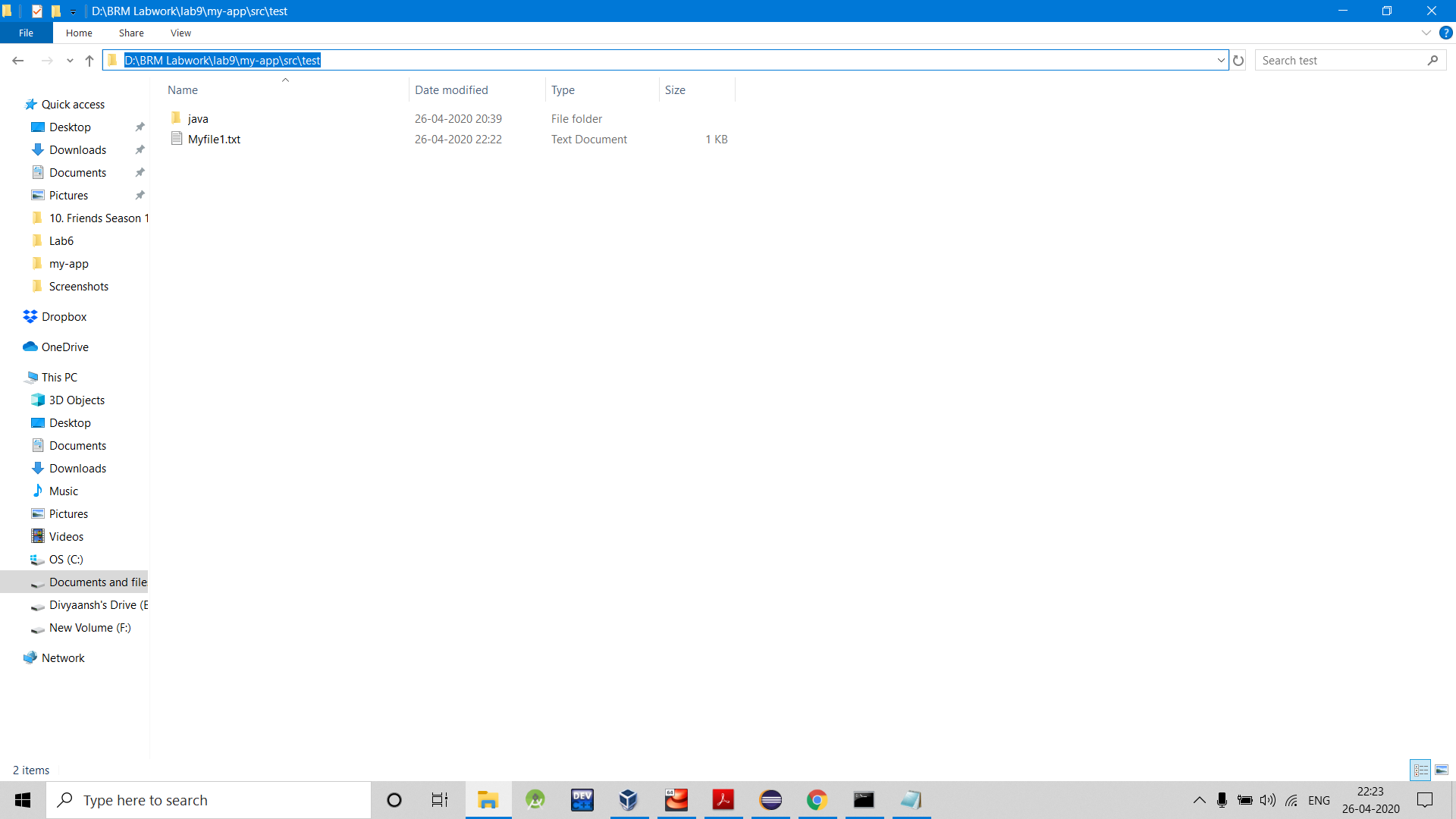
C:\MVN\project>mvn test -Ptest

Maven will start processing and displaying the result of test build profile.

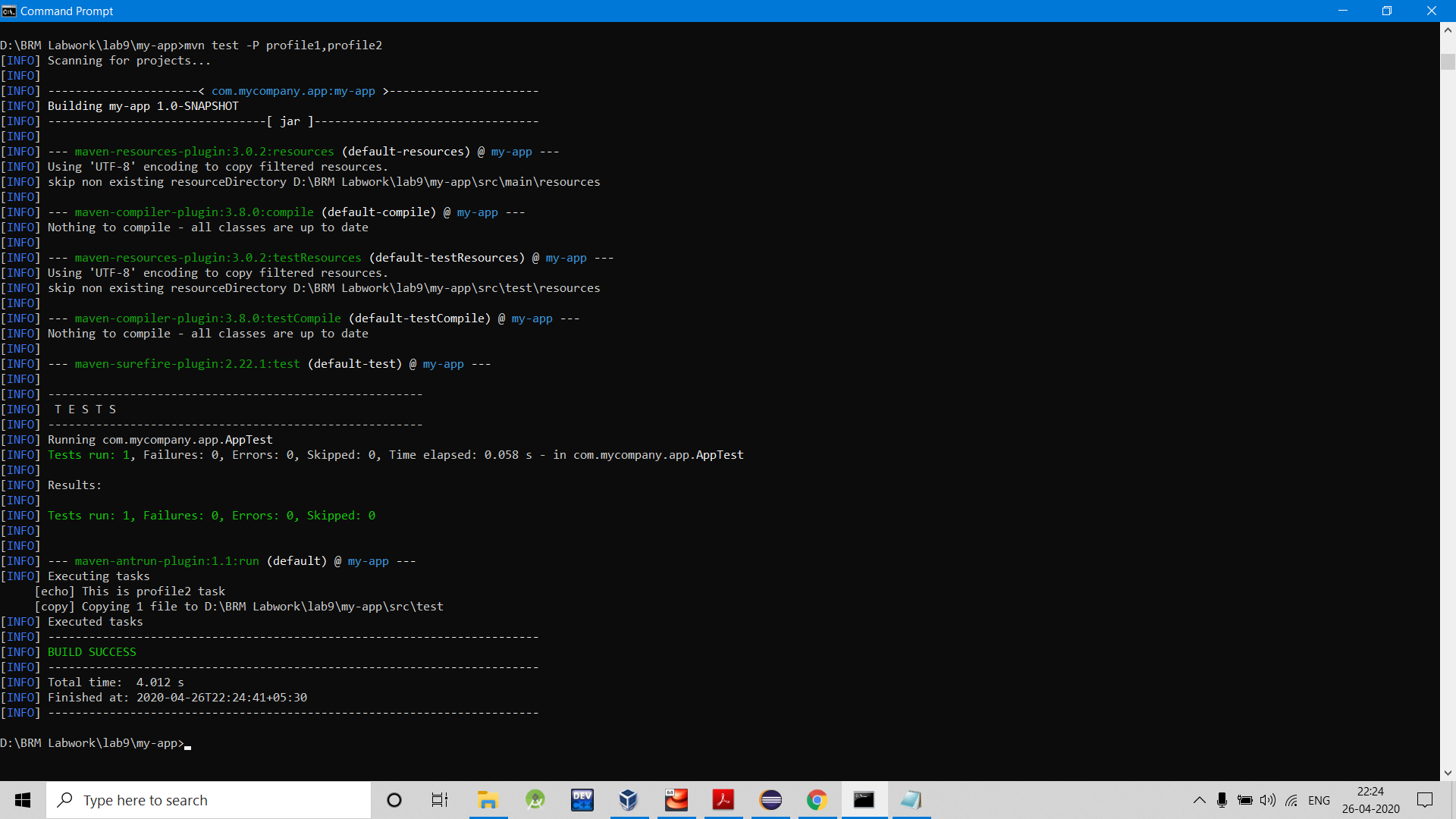




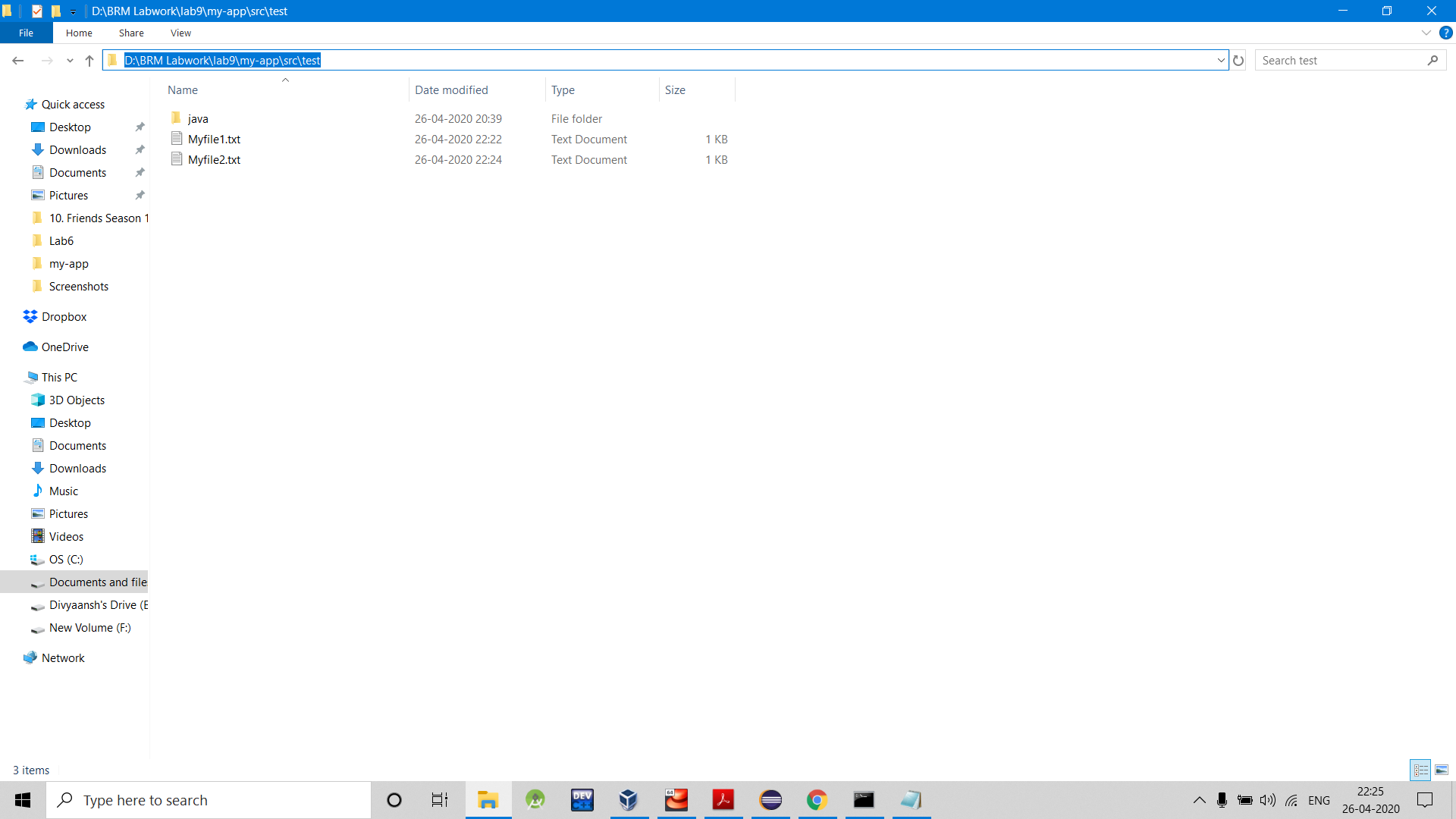
After the completion of Build you will find a new text file created in the folder specified in profile.



Multiple profile testing in a single command but since we are running test goal profile which is last will overwrite the work of all the previous profiles.



But we have made a profile1 previously so we will see two files made by profile1 and profile2.



Running test goal by changing activation to default in pom.xml file

