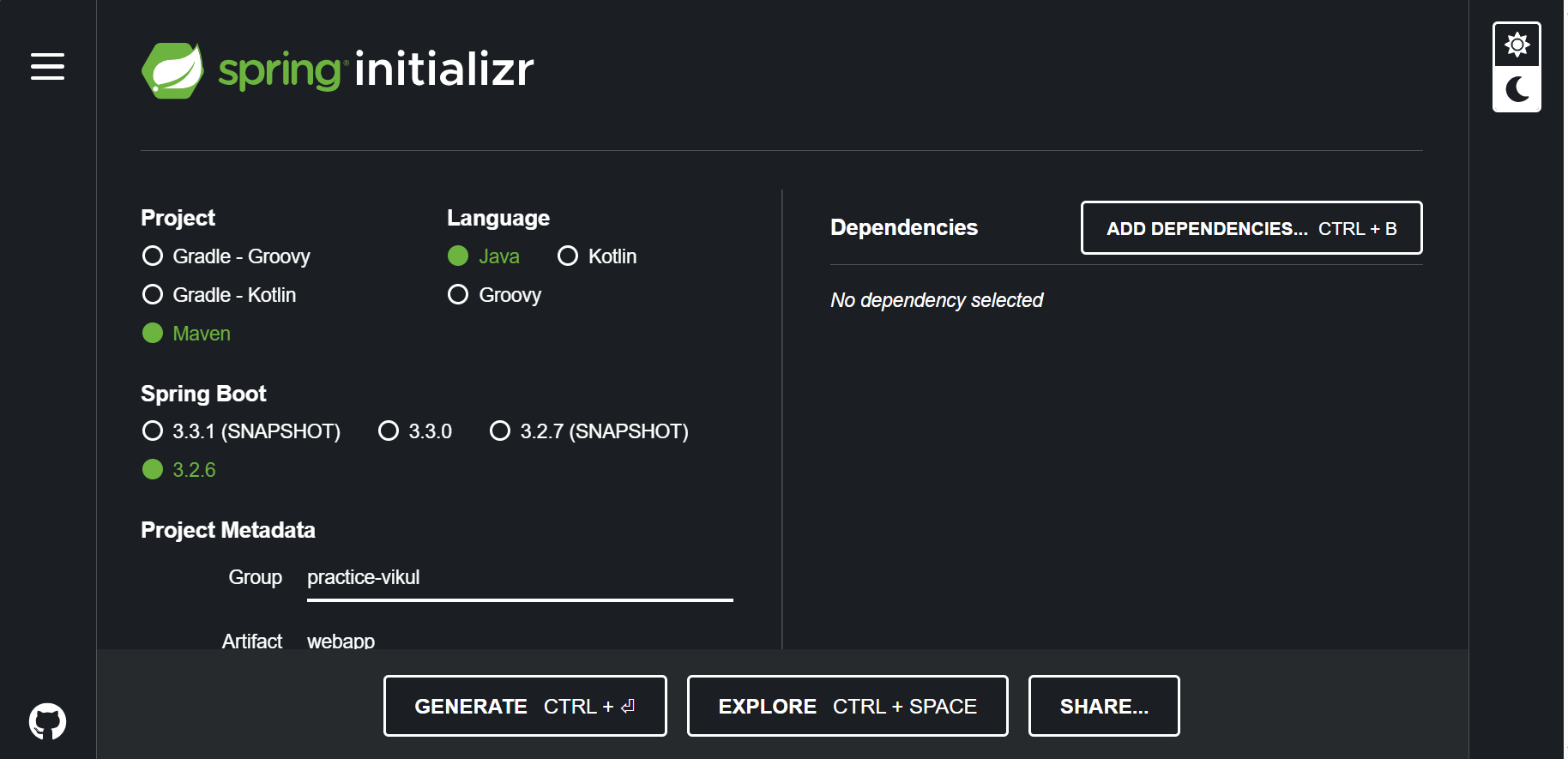
[Spring Initializr](https://start.spring.io/):

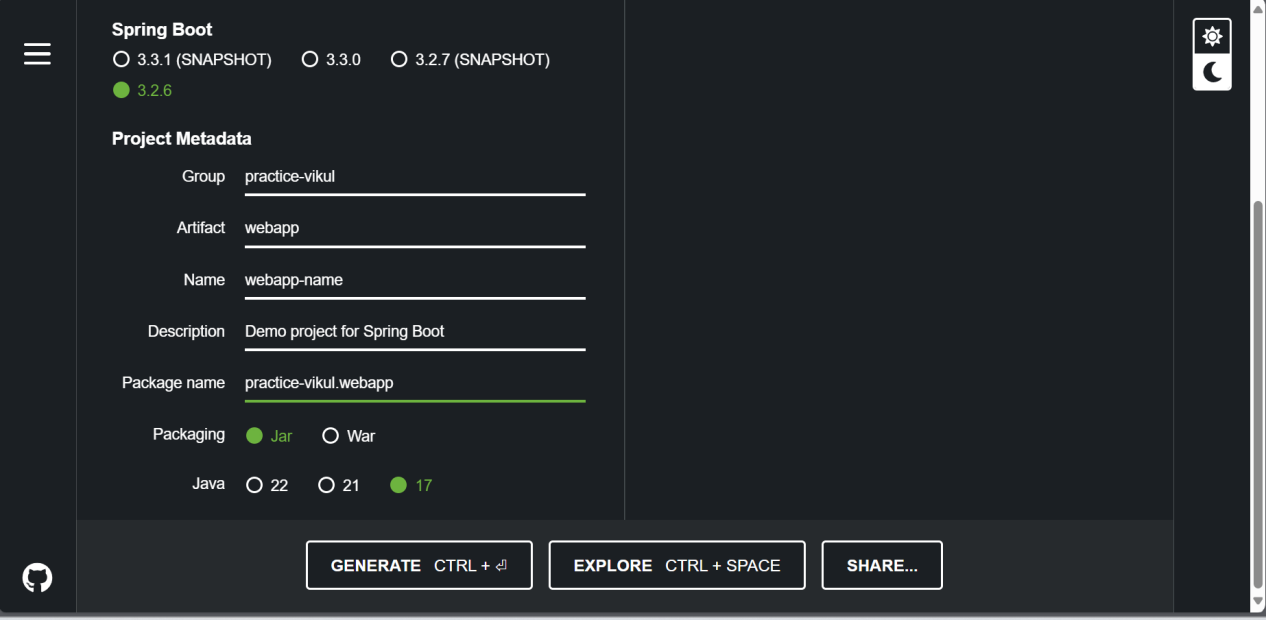
**Spring boot framework:**

- Spring boot framework : Where you can build your maven projects and add dependencies without any external implication.

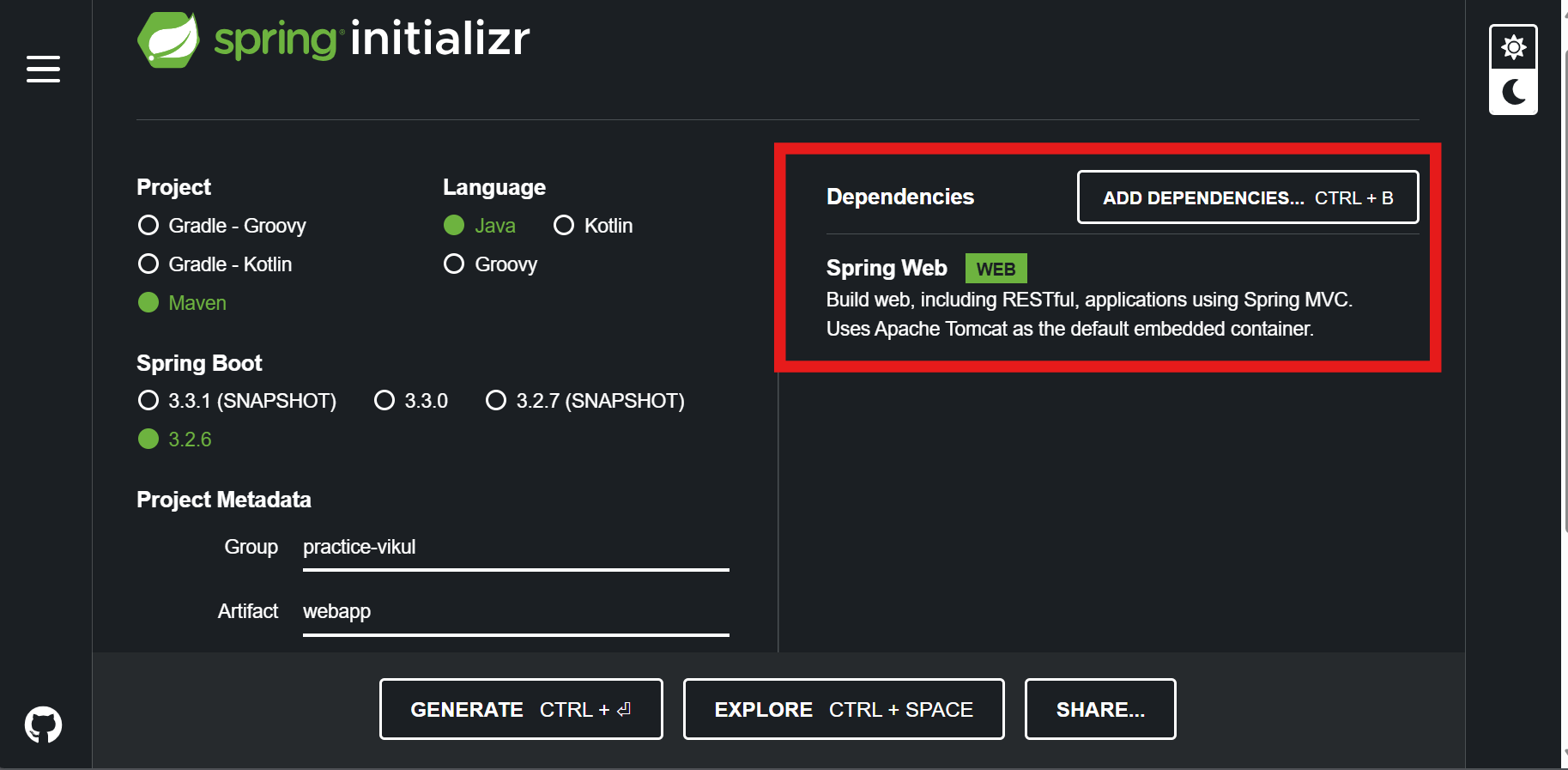
- Its specially use for run application in very fast

[https://start.spring.io](https://start.spring.io/)

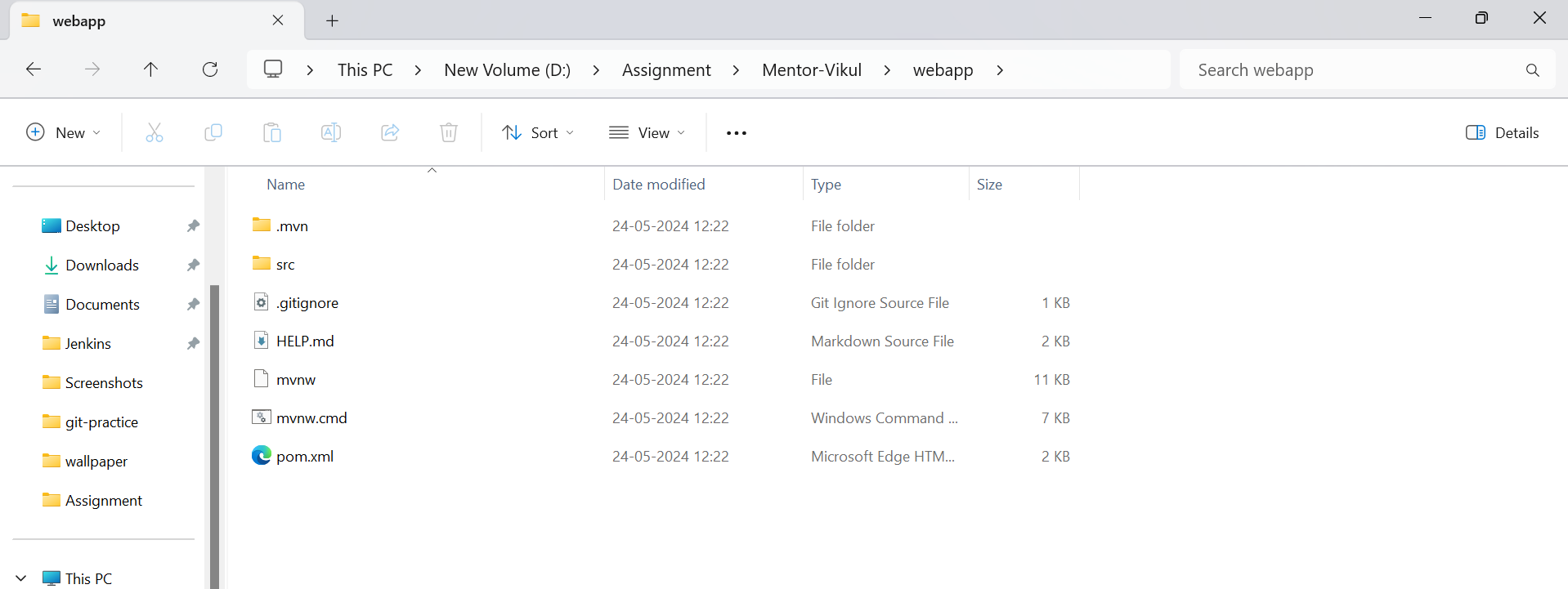




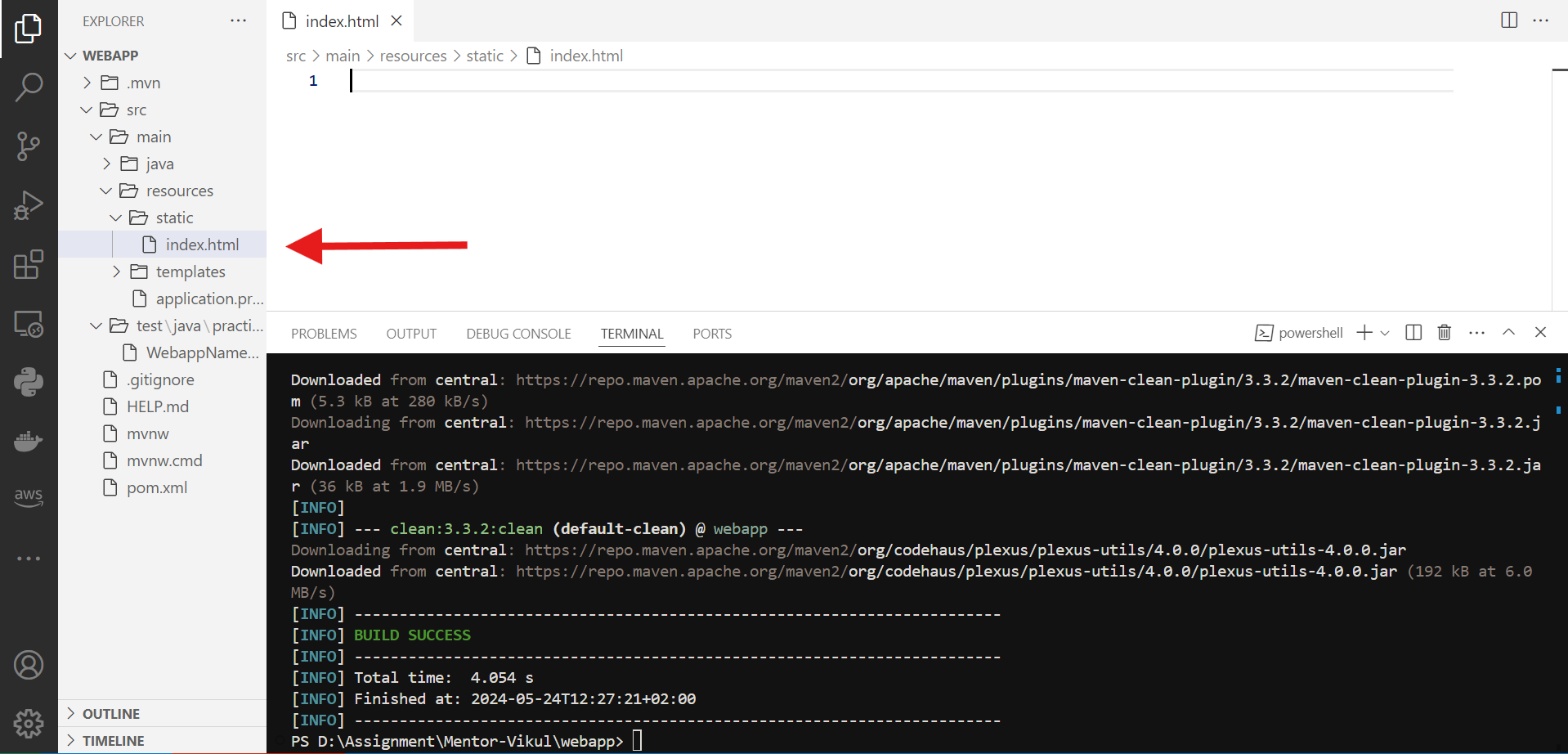
Add Spring web dependency, which is help to run we app framework(Apache-tomcat)



**“GENERATE the file and unzip it:::::**  
*there thr sample project ready with spring boot web dependencies”*

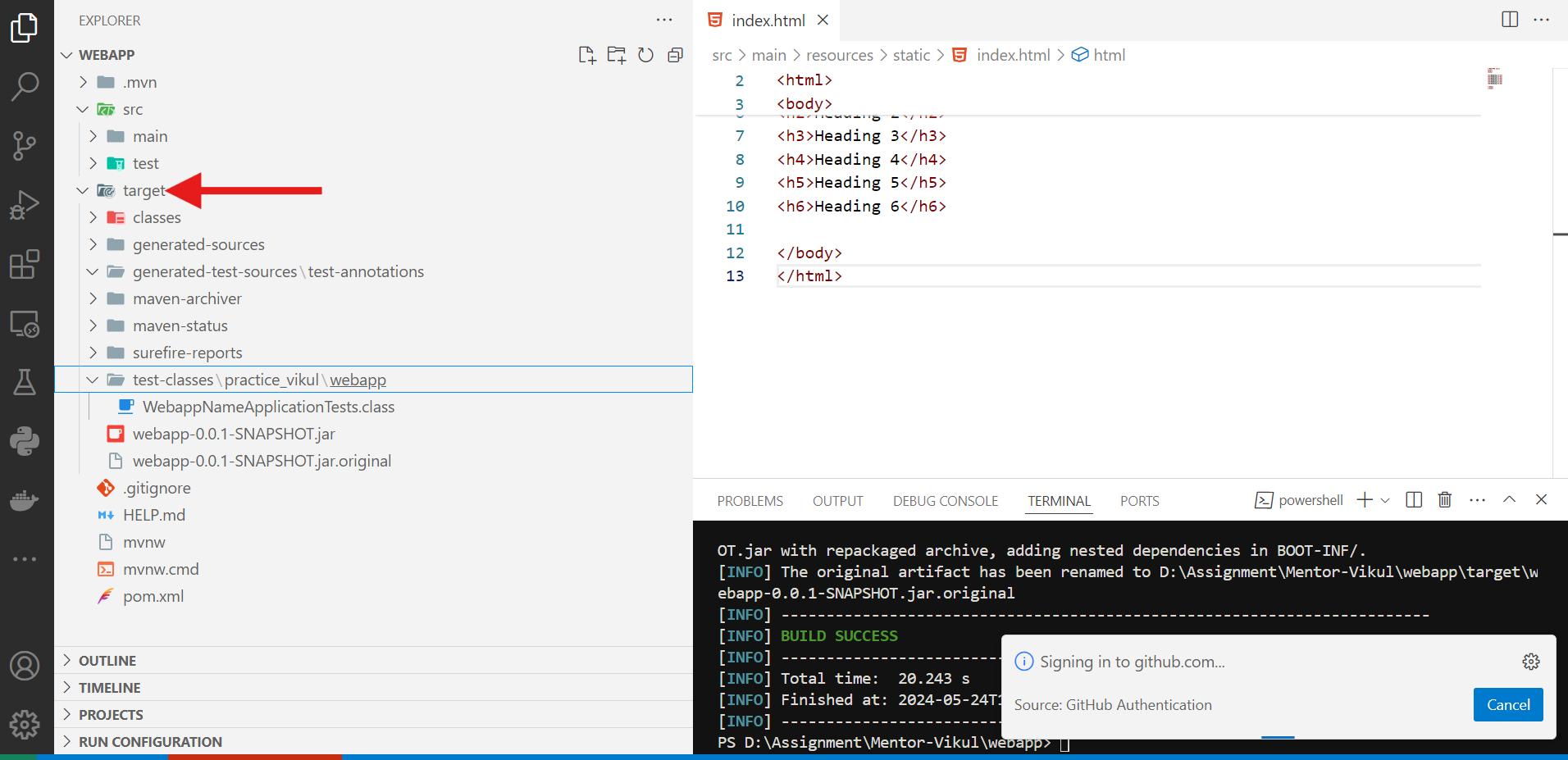


**In “static” create index.html**: For Write sample code for deploying in web



Run : **mvn package**

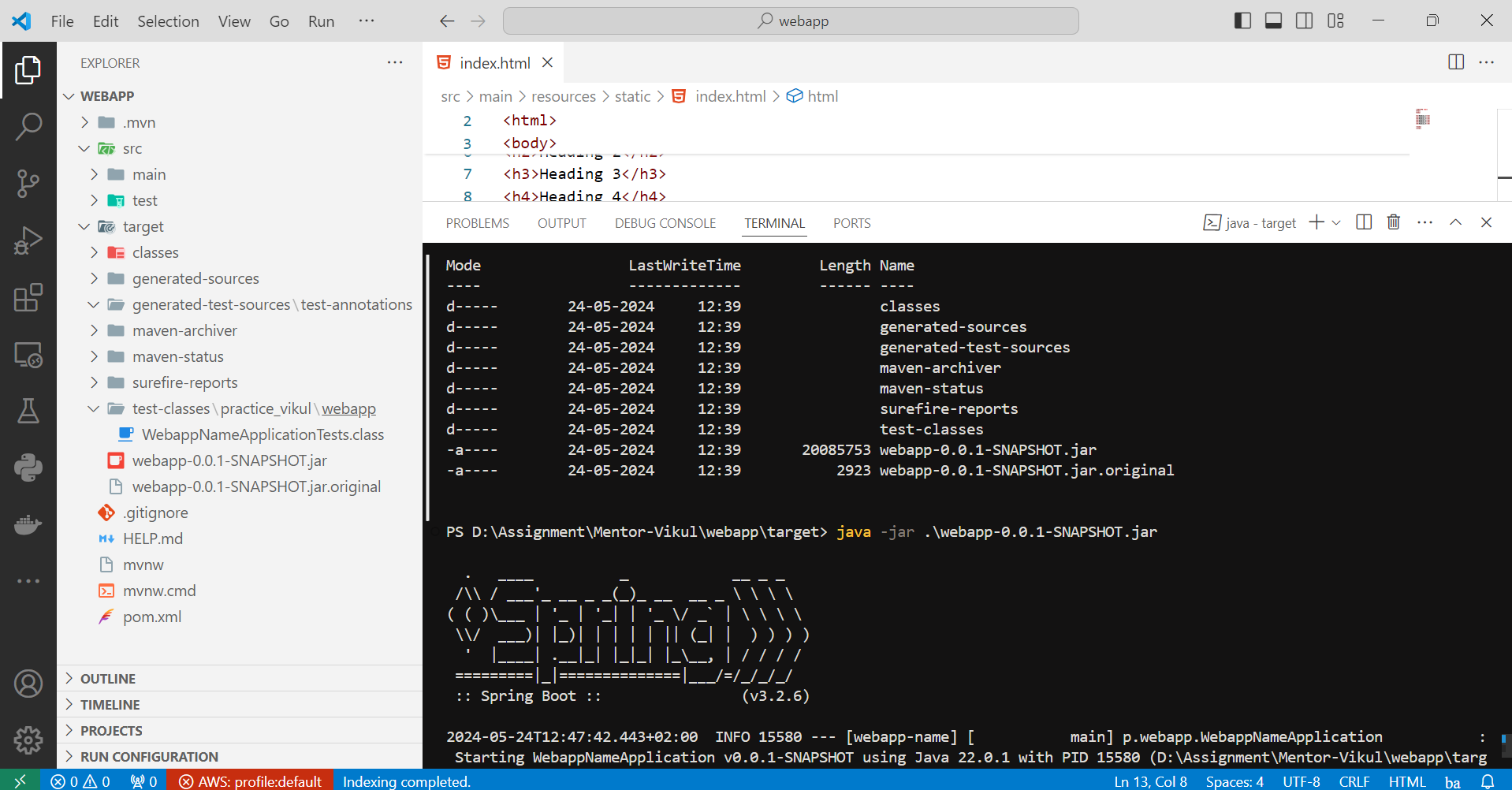
**With this it wull package the code and create target file**



How to run app:

Goto target folder where .jar file available

***java -jar <snapshot.jatr>***



MAVEN FILE SRUCTURE

: Maven uses a standard directory structure.

Main have main source code and test have after testing code that’s how it will seprate from each other

SRC - **MAIN**

Briefly, **src** holds your project's source code and resources, with

**src/main** containing the core code **(src/main/java**) and non-code resources (**src/main/resources**) for your application.

SRC - **TEST**

In Maven, the **src/test** directory is dedicated to test code. It usually mirrors the structure of src/main but holds unit tests for your main code. This separation keeps your main application code clean and separates testing concerns.

TARGET:

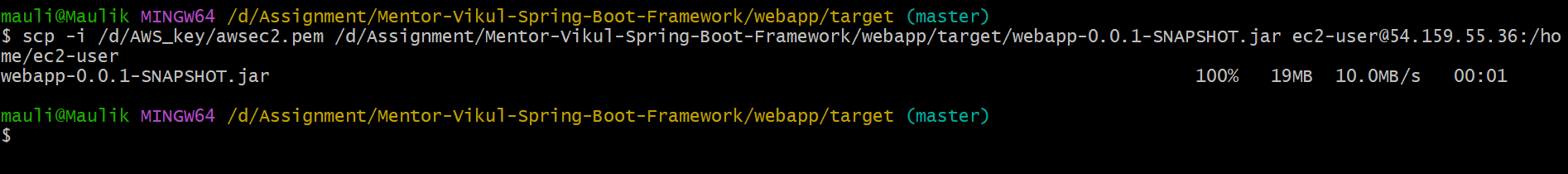
In Maven, the target directory is where the build process stores compiled class files, packaged applications (JARs, WARs, etc.), and other temporary files generated during the build. It's generally not part of your source code and isn't included when you share your project with others.

How to deploy on EC2?

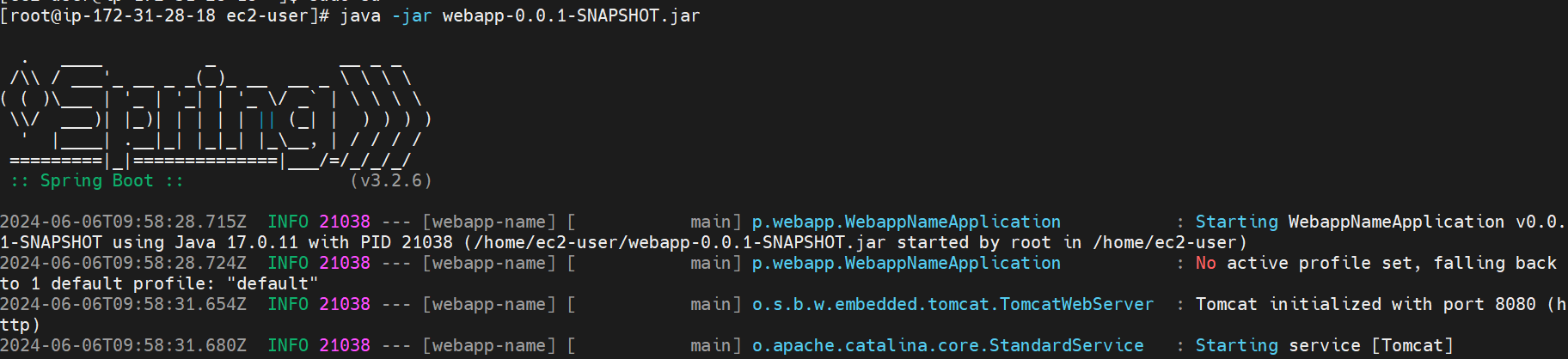
Steps:  
Install java on Ec2(Check pom.xml or install java which is selected while generate spring boot framework== same install on ec2)

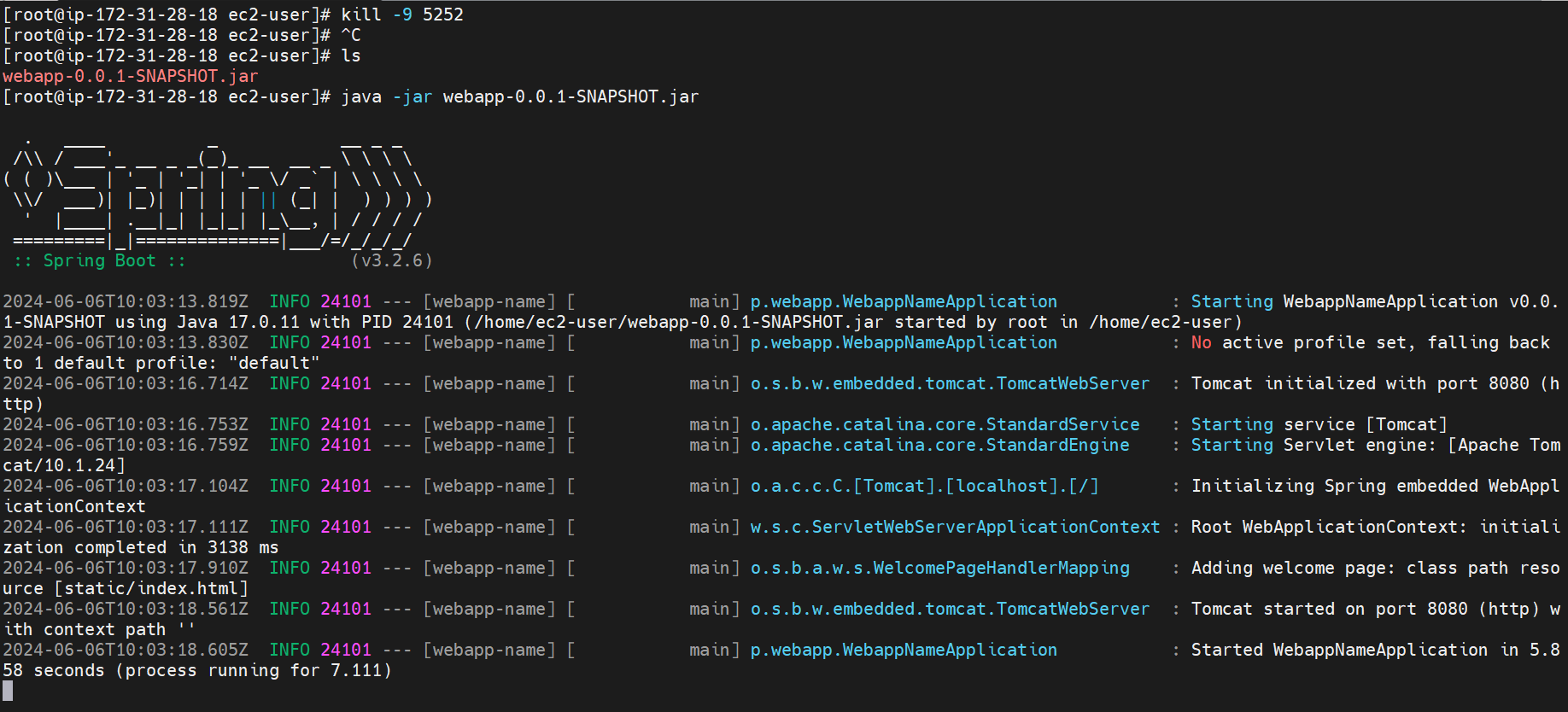
Copy this file from local to ec2 with scp command

Scp -I <Path of secret key> <path-of.jar file(Source)> <username@publicip:path(/home/ec2-user) (Destination)>



Check file on ec2- Run the app from ec2 with java -jar .



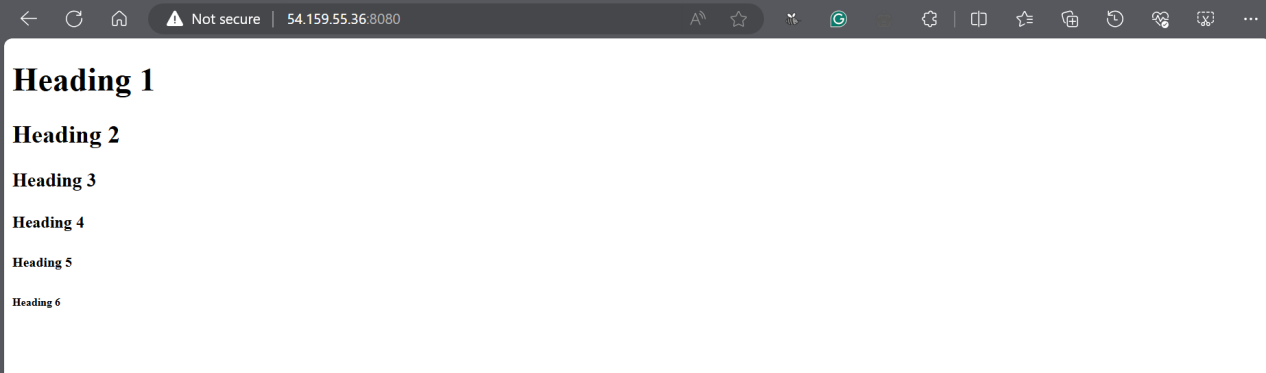


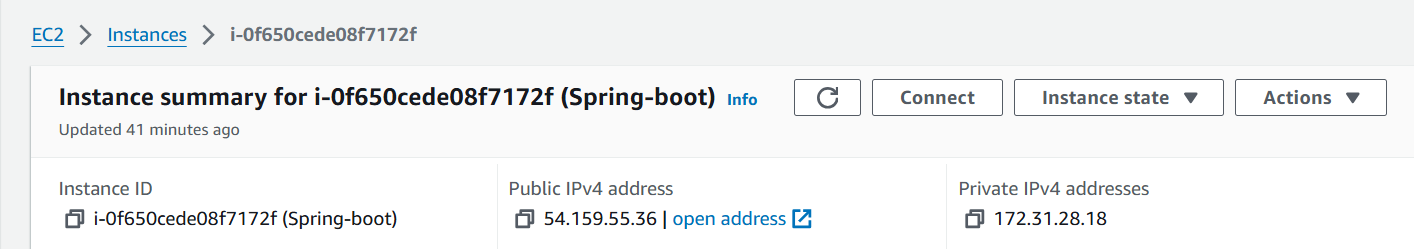
Kill -9 5252(5252=PID(Process id)), which is killed process with pid id== how to find pid id?  
[root@ip-172-31-28-18 ec2-user]# ps aux | grep java

root 5252 0.5 14.1 2496612 138200 pts/0 Sl+ 09:29 0:09 java -jar /home/ec2-user/webapp-0.0.1-SNAPSHOT.jar

root 22988 0.0 0.0 119424 924 pts/2 S+ 10:00 0:00 grep --color=auto java

The command ps aux | grep java lists all running processes and filters the output to show only those processes that include the word "java". This helps identify running Java processes

b



SG

