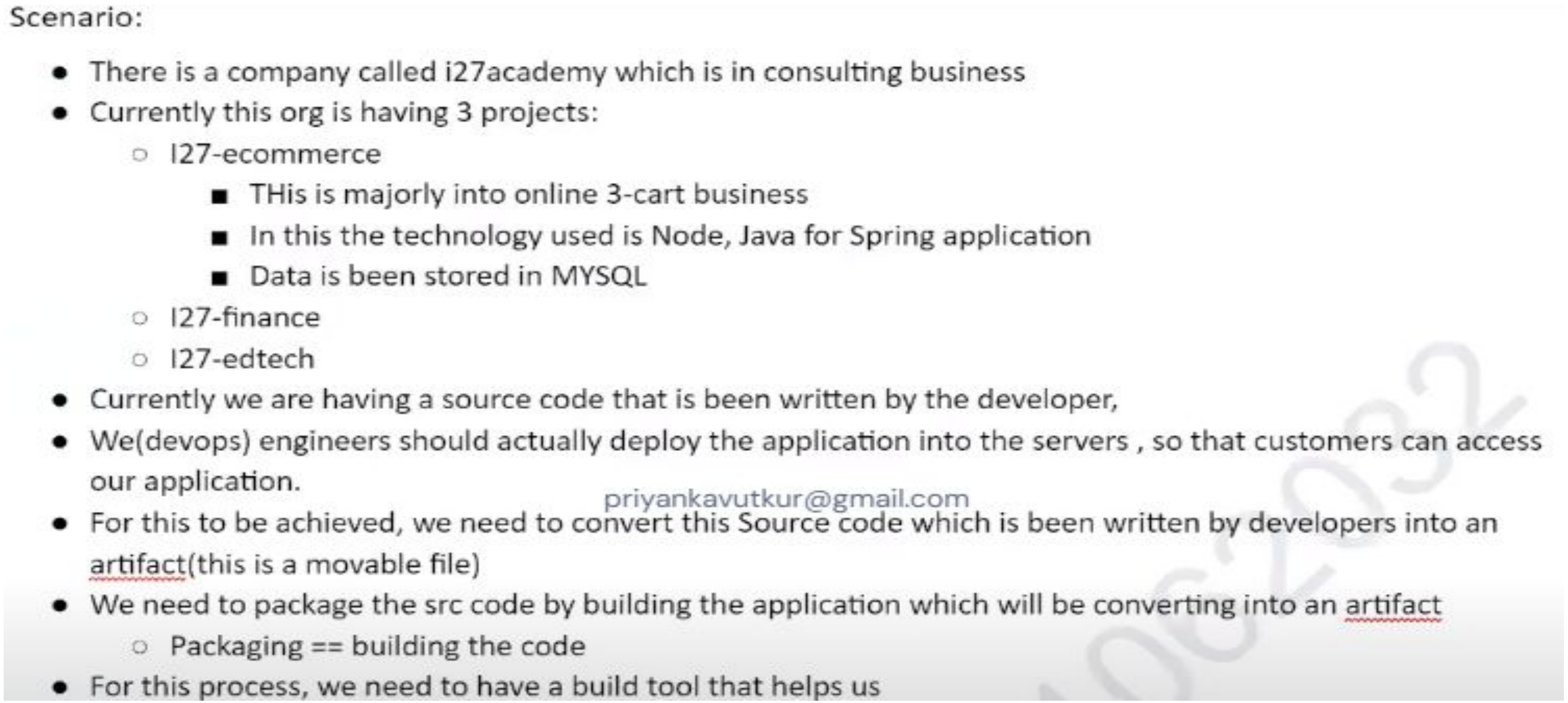
Devops

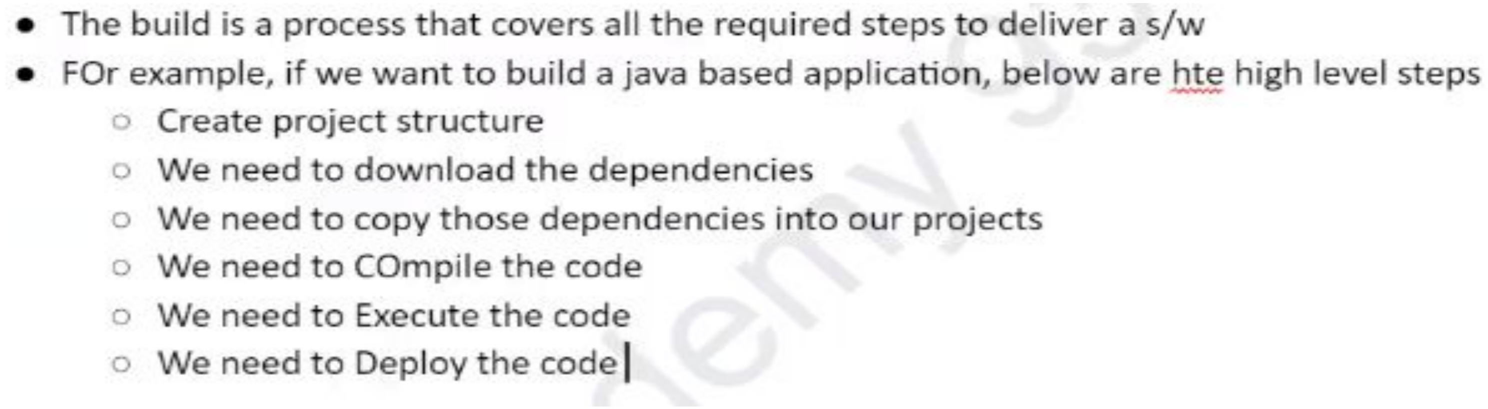
Build tool — Maven



Scenario:

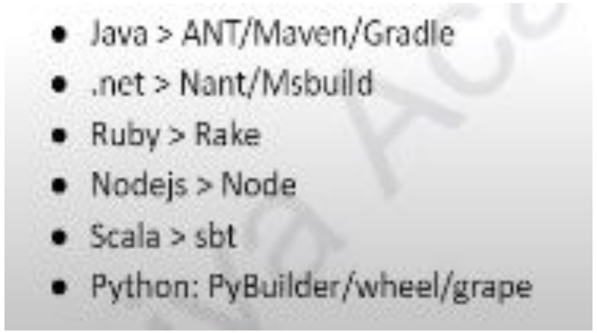
- There is a company called i27academy which is in consulting business
- Currently this org is having 3 projects:
  - I27-ecommerce
    - THis is majorly into online 3-cart business
    - In this the technology used is Node, Java for Spring application
    - Data is been stored in MYSQL
  - I27-finance
  - I27-edtech
- Currently we are having a source code that is been written by the developer,
- We(devops) engineers should actually deploy the application into the servers , so that customers can access our application.
- For this to be achieved, we need to convert this Source code which is been written by developers into an artifact(this is a movable file)
- We need to package the src code by building the application which will be converting into an artifact
  - Packaging == building the code
- For this process, we need to have a build tool that helps us

What is build



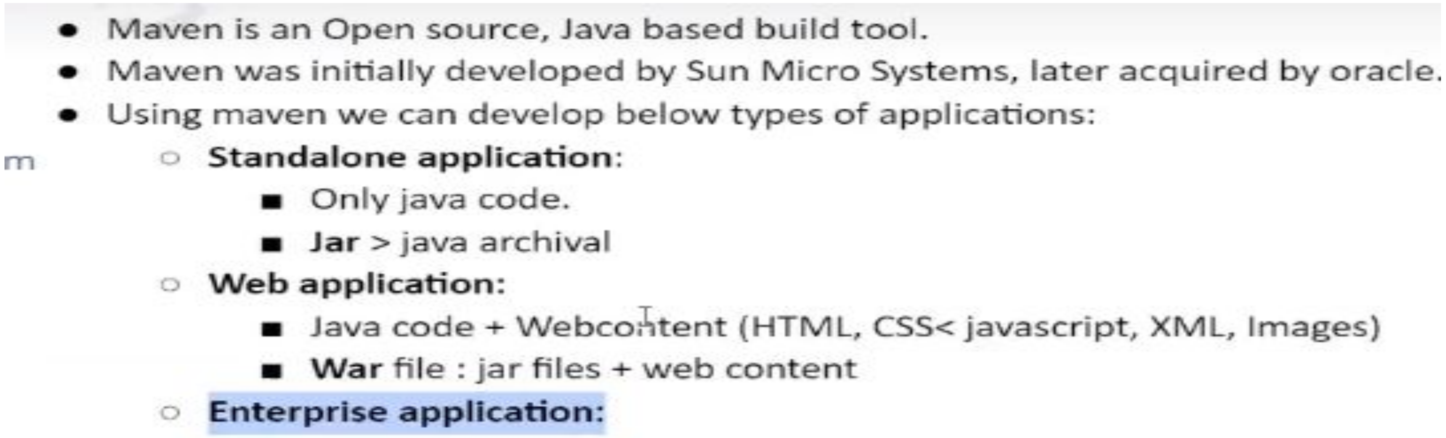
- The build is a process that covers all the required steps to deliver a s/w
- FOr example, if we want to build a java based application, below are hte high level steps
  - Create project structure
  - We need to download the dependencies
  - We need to copy those dependencies into our projects
  - We need to COmpile the code
  - We need to Execute the code
  - We need to Deploy the code

Various build tools

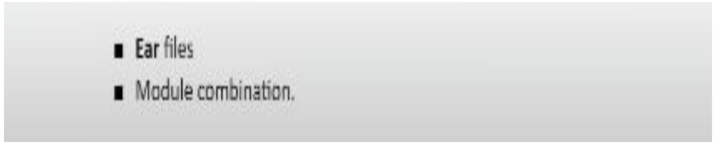


- Java > ANT/Maven/Gradle
- .net > Nant/Msbuild
- Ruby > Rake
- Nodejs > Node
- Scala > sbt
- Python: PyBuilder/wheel/grape

About maven



- Maven is an Open source, Java based build tool.
- Maven was initially developed by Sun Micro Systems, later acquired by oracle.
- Using maven we can develop below types of applications:
  - **Standalone application**:
    - Only java code.
    - **Jar** > java archival
  - **Web application:**
    - Java code + Webcontent (HTML, CSS< javascript, XML, Images)
    - **War** file : jar files + web content
  - **Enterprise application:**

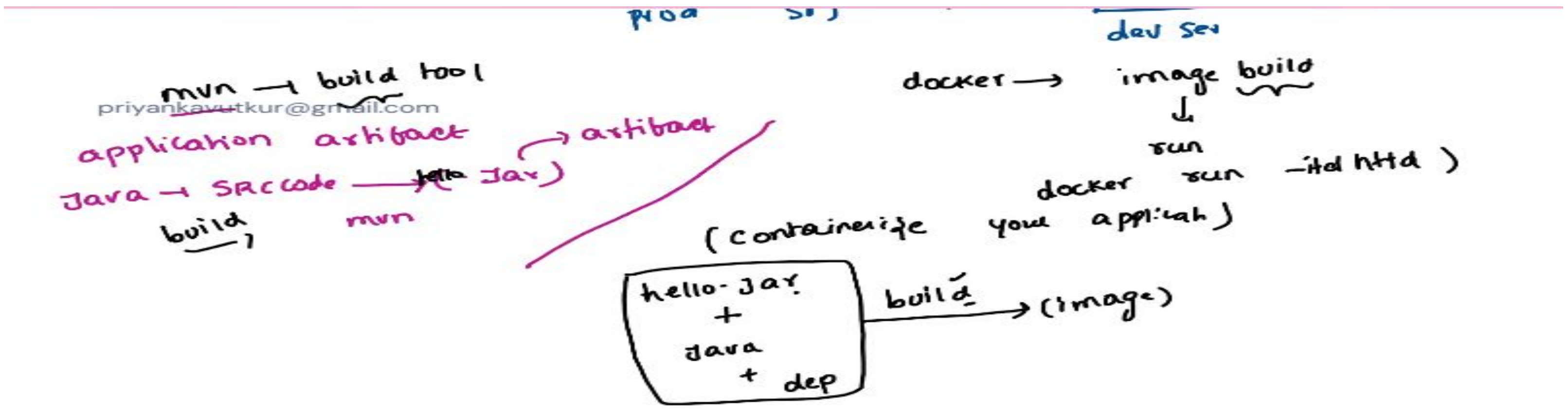


    - Ear files
    - Module combination.

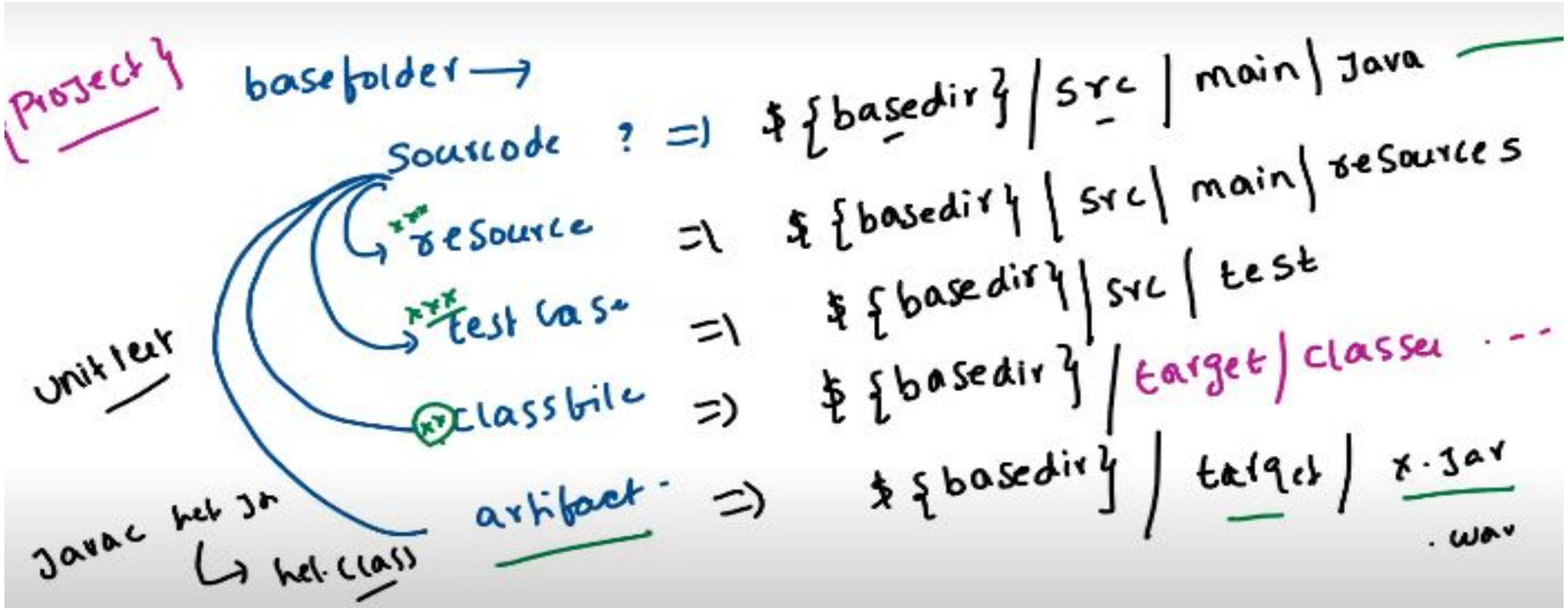
.jar and .war are the packages

→mvn generates build artifacts(.war and .jar files for java based application

→similarly docker also generates build artifacts(image) it acts as containerized for our application.it takes .jar which is generated by maven tool and based on their dependencies it stores theinfo as image.

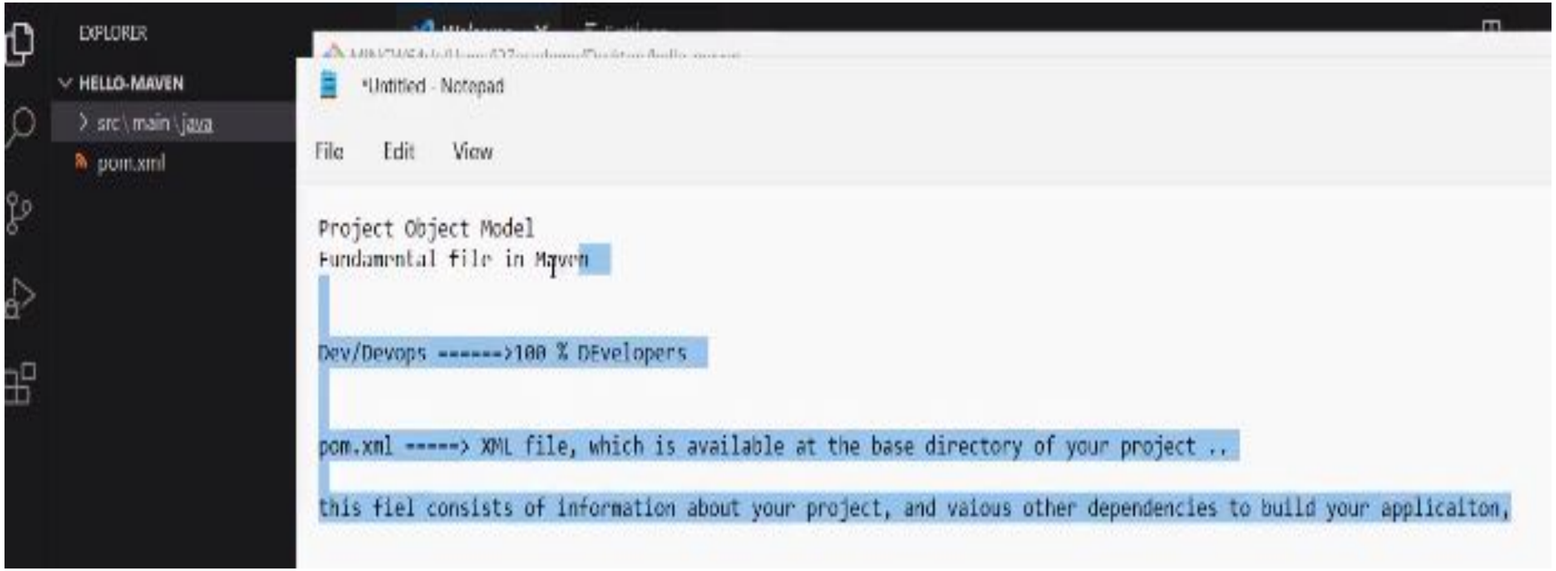


Project (java based application)


{Project}
basefolder →
Sourcode ? =) ${basedir}/src/main/Java
resource =) ${basedir}/src/main/resources
test case =) ${basedir}/src/test
classfile =) ${basedir}/target/classes ...
artifact =) ${basedir}/target/x.jar
.war
unittest
Javac hel.Jn
hel.class

Pom.xml file

Pom.xml file is created under base directory as below



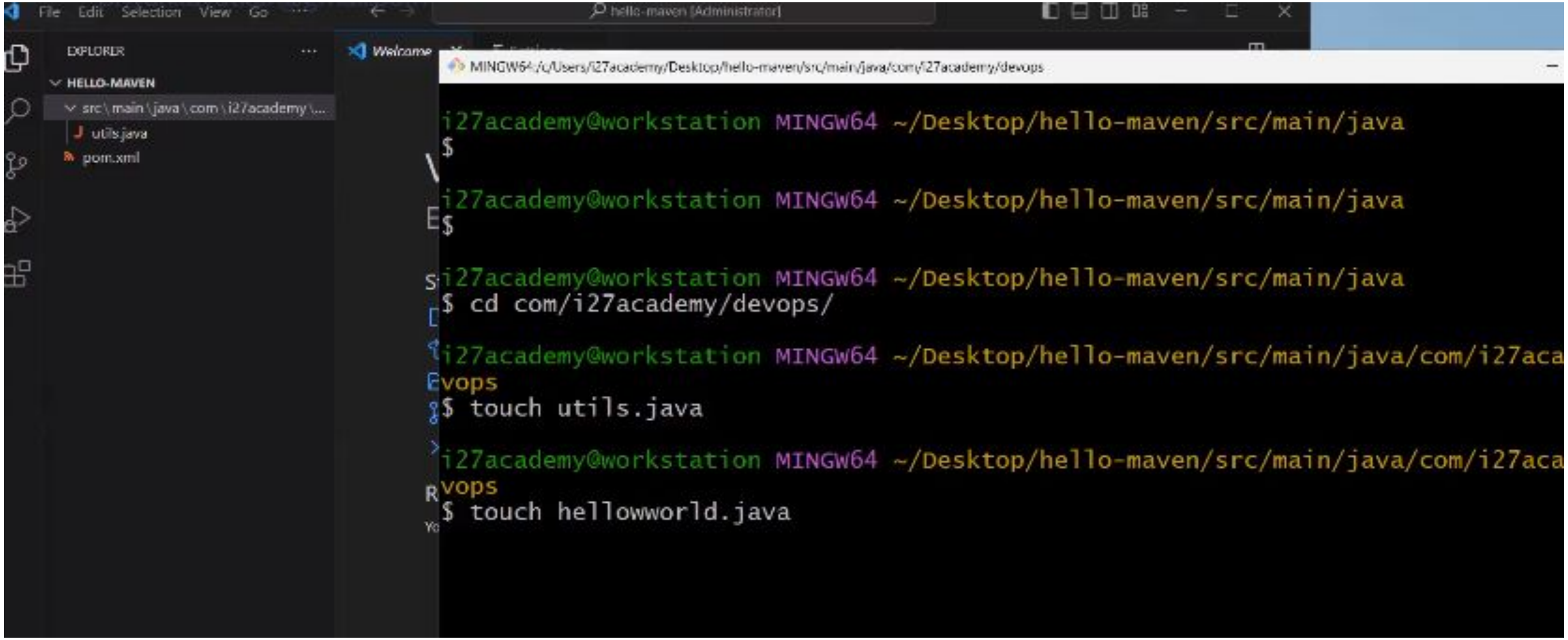
```
Project Object Model
Fundamental file in Maven


Dev/Devops ------>100 % DEvelopers


pom.xml -----> XML file, which is available at the base directory of your project ..

this fiel consists of information about your project, and vaious other dependencies to build your applicalton,
```

Below folder create 2 java files as below


File Edit Selection View Go
hello-maven [Administrator]
EXPLORER
HELLO-MAVEN
src \ main \ java \ com \ i27academy \ ...
utils.java
pom.xml
Welcome
MINGW64:/c/Users/i27academy/Desktop/hello-maven/src/main/java/com/i27academy/devops

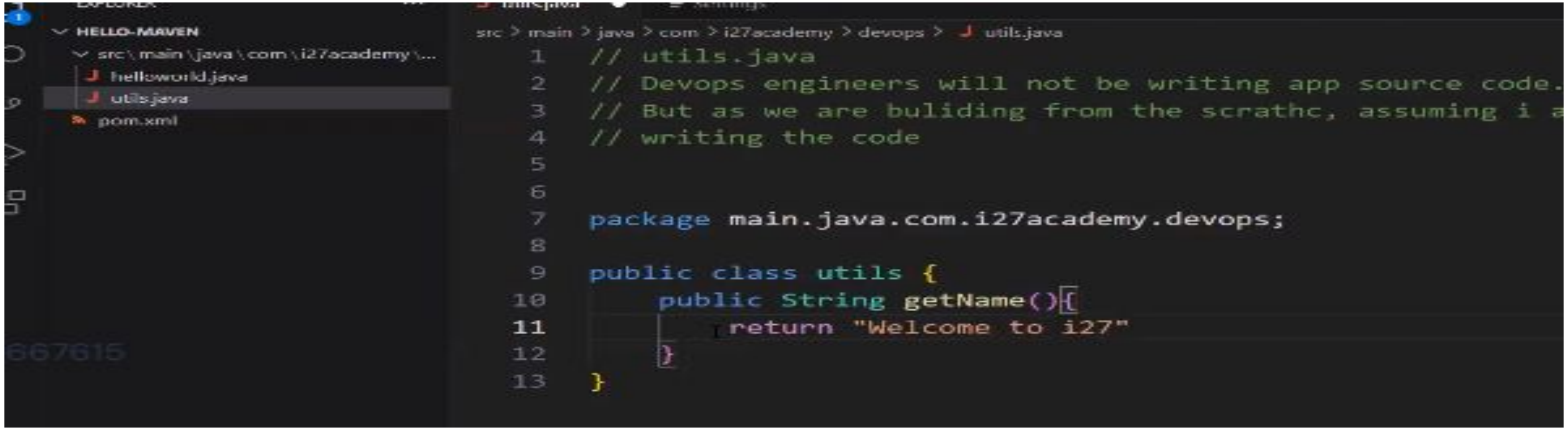
```
i27academy@workstation MINGW64 ~/Desktop/hello-maven/src/main/java
$

i27academy@workstation MINGW64 ~/Desktop/hello-maven/src/main/java
$

i27academy@workstation MINGW64 ~/Desktop/hello-maven/src/main/java
$ cd com/i27academy/devops/

i27academy@workstation MINGW64 ~/Desktop/hello-maven/src/main/java/com/i27aca
vops
$ touch utils.java

i27academy@workstation MINGW64 ~/Desktop/hello-maven/src/main/java/com/i27aca
vops
$ touch hellowworld.java
```


EXPLORER

HELLO-MAVEN

src \ main \ java \ com \ i27academy \ ...

helloworld.java

utils.java

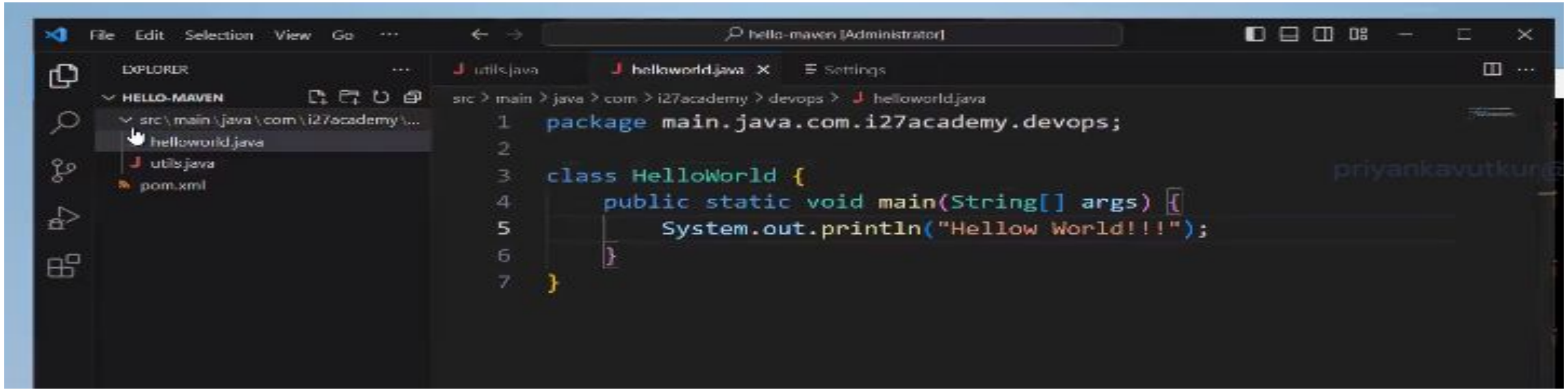
pom.xml

src > main > java > com > i27academy > devops > utils.java

```
// utils.java
// Devops engineers will not be writing app source code.
// But as we are buliding from the scrathc, assuming i a
// writing the code


package main.java.com.i27academy.devops;

public class utils {
    public String getName(){
        return "Welcome to i27"
    }
}
```


File Edit Selection View Go ···

hello-maven [Administrator]

EXPLORER

HELLO-MAVEN

src \ main \ java \ com \ i27academy \ ...

helloworld.java

utils.java

pom.xml

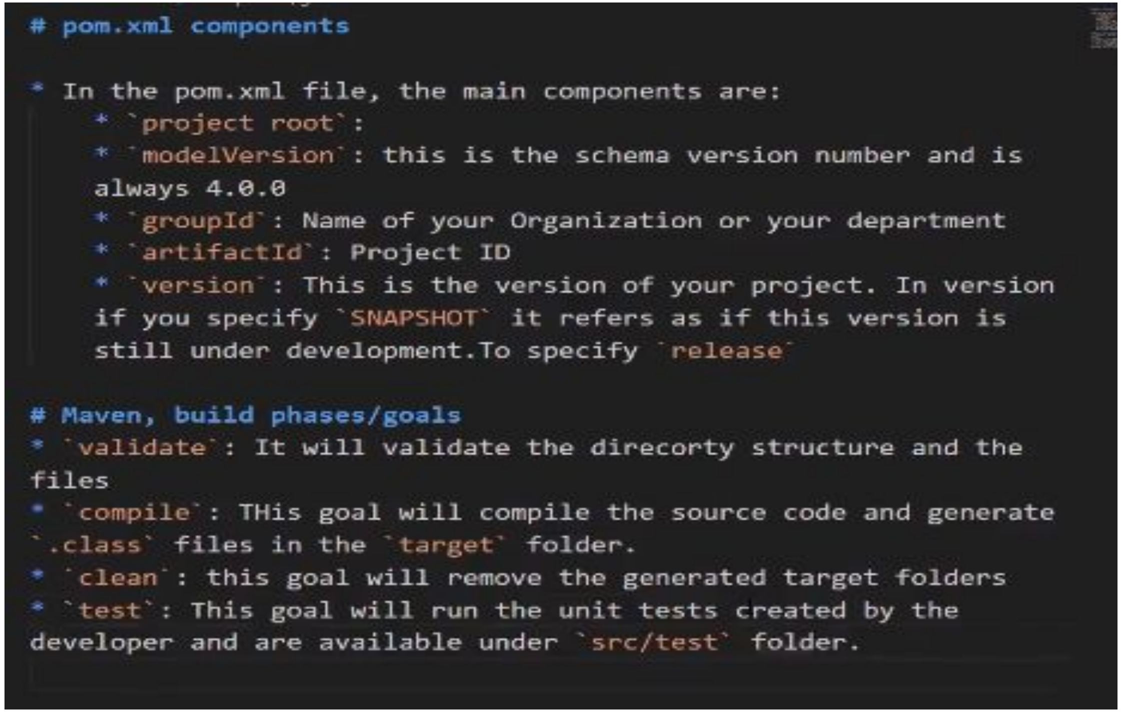
utils.java helloworld.java Settings

src > main > java > com > i27academy > devops > helloworld.java

```
package main.java.com.i27academy.devops;

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

Readme file —info in pom.xml file



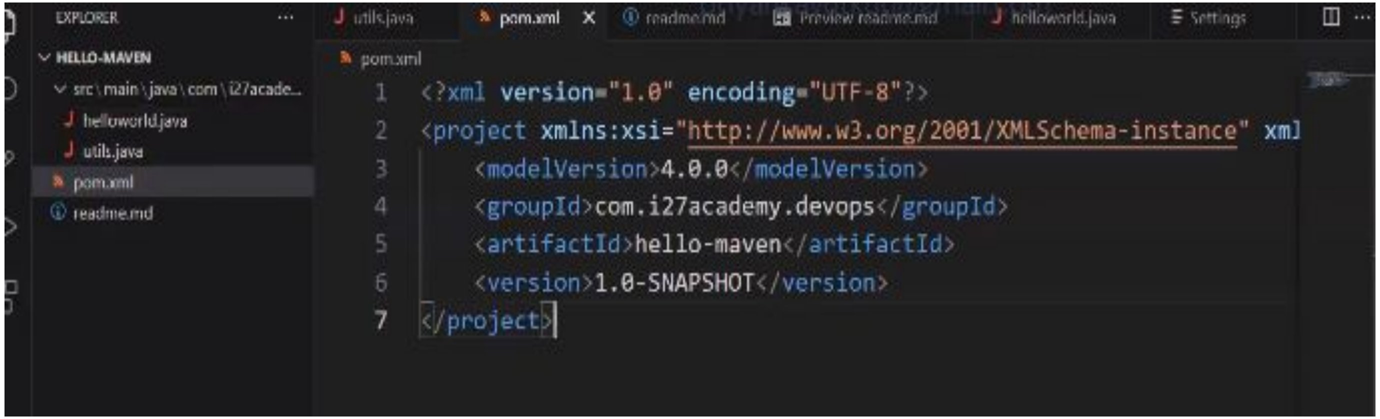
```
# pom.xml components

* In the pom.xml file, the main components are:
    * `project root`:
    * `modelVersion`: this is the schema version number and is
    always 4.0.0
    * `groupId`: Name of your Organization or your department
    * `artifactId`: Project ID
    * `version`: This is the version of your project. In version
    if you specify `SNAPSHOT` it refers as if this version is
    still under development.To specify `release`

# Maven, build phases/goals
* `validate`: It will validate the direcorty structure and the
files
* `compile`: THis goal will compile the source code and generate
`.class` files in the `target` folder.
* `clean`: this goal will remove the generated target folders
* `test`: This goal will run the unit tests created by the
developer and are available under `src/test` folder.
```

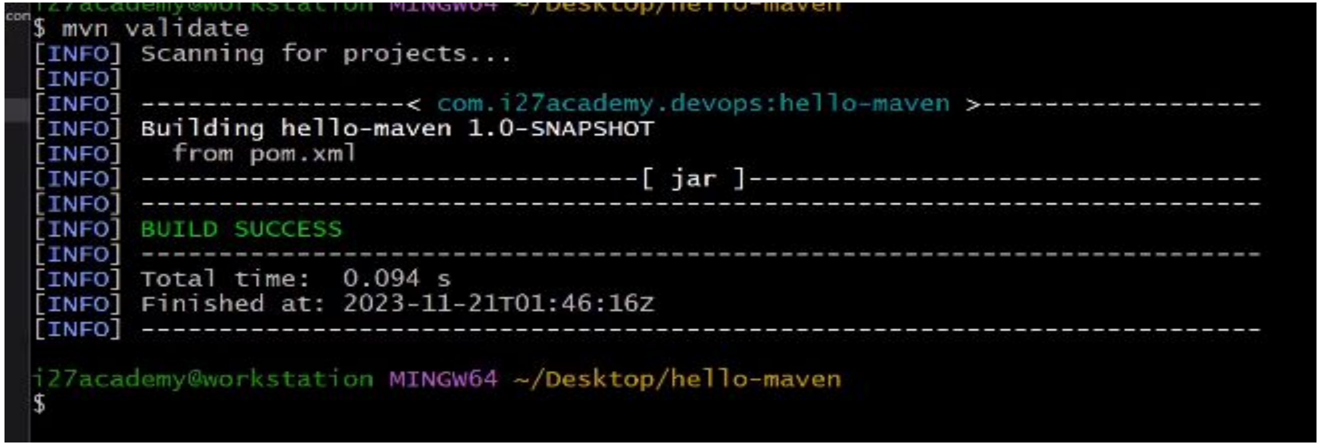
[spring-projects/spring-petclinic: A sample Spring-based application (github.com)](https://github.com/spring-projects/spring-petclinic)

This is java based application—refer this for pom.xml file



```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xml
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.i27academy.devops</groupId>
    <artifactId>hello-maven</artifactId>
    <version>1.0-SNAPSHOT</version>
</project>
```

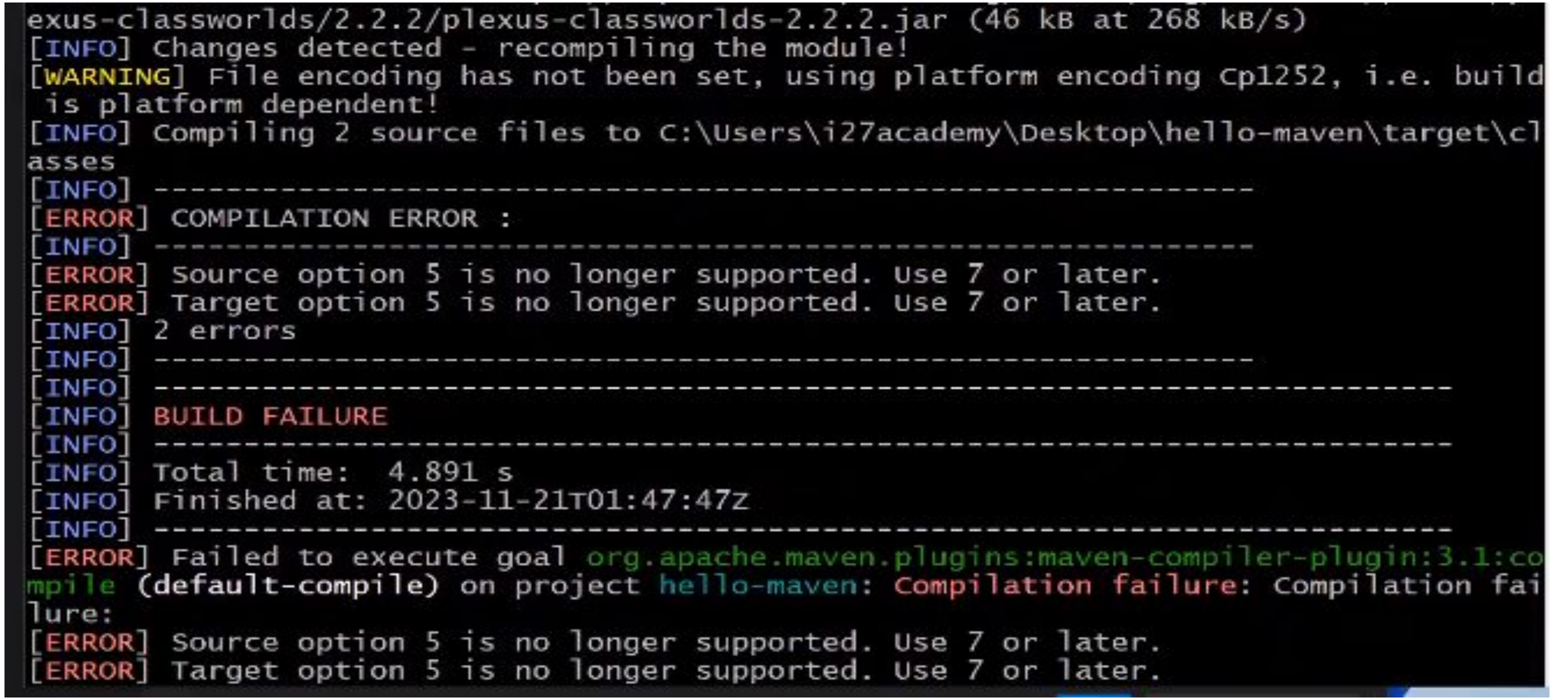
Mvn commands must execute where pom.xml file is available



```
$ mvn validate
[INFO] Scanning for projects...
[INFO]
[INFO] -----------------< com.i27academy.devops:hello-maven >------------------
[INFO] Building hello-maven 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.094 s
[INFO] Finished at: 2023-11-21T01:46:16Z
[INFO] ------------------------------------------------------------------------

i27academy@workstation MINGW64 ~/Desktop/hello-maven
$
```

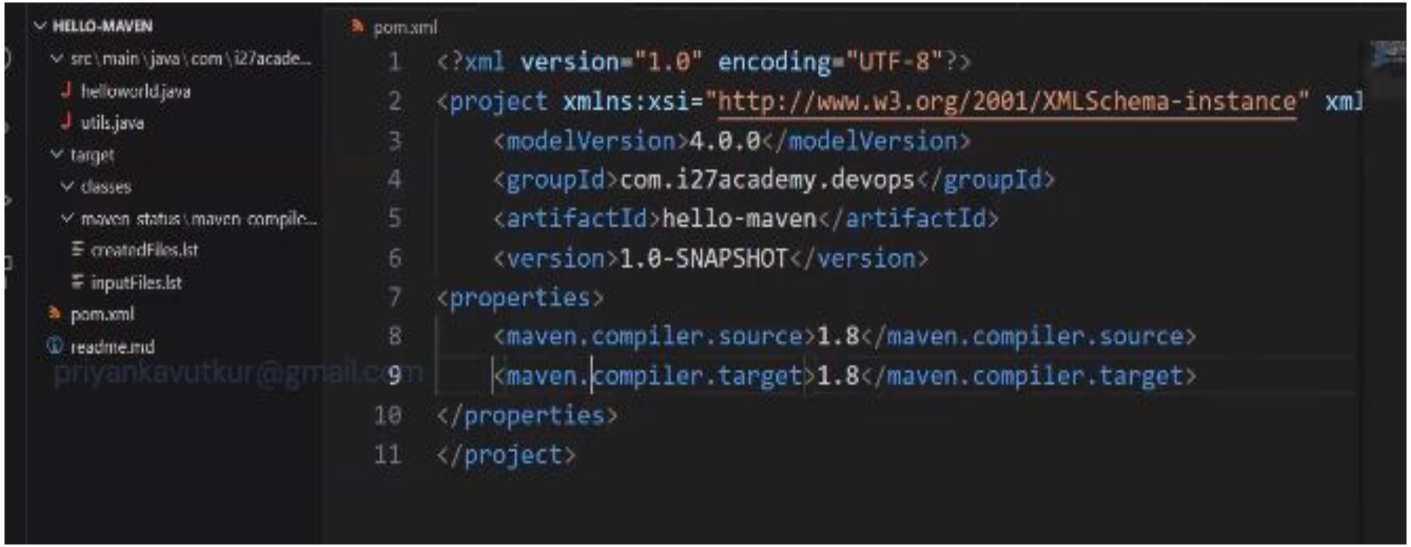
Now give mvn compile then there may be a chance to get below error



```
exus-classworlds/2.2.2/plexus-classworlds-2.2.2.jar (46 kB at 268 kB/s)
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build
 is platform dependent!
[INFO] Compiling 2 source files to C:\Users\i27academy\Desktop\hello-maven\target\cl
asses
[INFO] -------------------------------------------------------------
[ERROR] COMPILATION ERROR :
[INFO] -------------------------------------------------------------
[ERROR] Source option 5 is no longer supported. Use 7 or later.
[ERROR] Target option 5 is no longer supported. Use 7 or later.
[INFO] 2 errors
[INFO] -------------------------------------------------------------
[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  4.891 s
[INFO] Finished at: 2023-11-21T01:47:47Z
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-compiler-plugin:3.1:co
mpile (default-compile) on project hello-maven: Compilation failure: Compilation fai
lure:
[ERROR] Source option 5 is no longer supported. Use 7 or later.
[ERROR] Target option 5 is no longer supported. Use 7 or later.
```

Now to avoid the compilation error modify the pom.xml file by adding properties

Even though i got error …taget folder got created

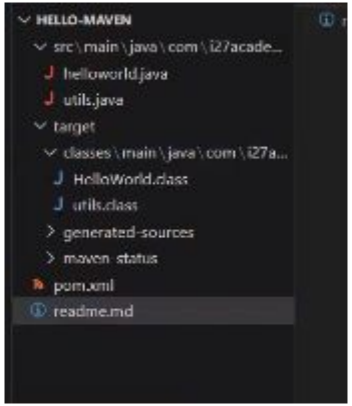


```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xml
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.i27academy.devops</groupId>
    <artifactId>hello-maven</artifactId>
    <version>1.0-SNAPSHOT</version>
<properties>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
</properties>
</project>
```

To remove that target folder and the clean target folder again then run below command

mvn clean – after cleaning the target folder and give mvn compile

Then i got build success and new target folder got created



```
∨ HELLO-MAVEN
  ∨ src\main\java\com\i27acade...
      helloworld.java
      utils.java
  ∨ target
    ∨ classes\main\java\com\i27a...
        HelloWorld.class
        utils.class
    > generated-sources
    > maven-status
    pom.xml
    readme.md
```

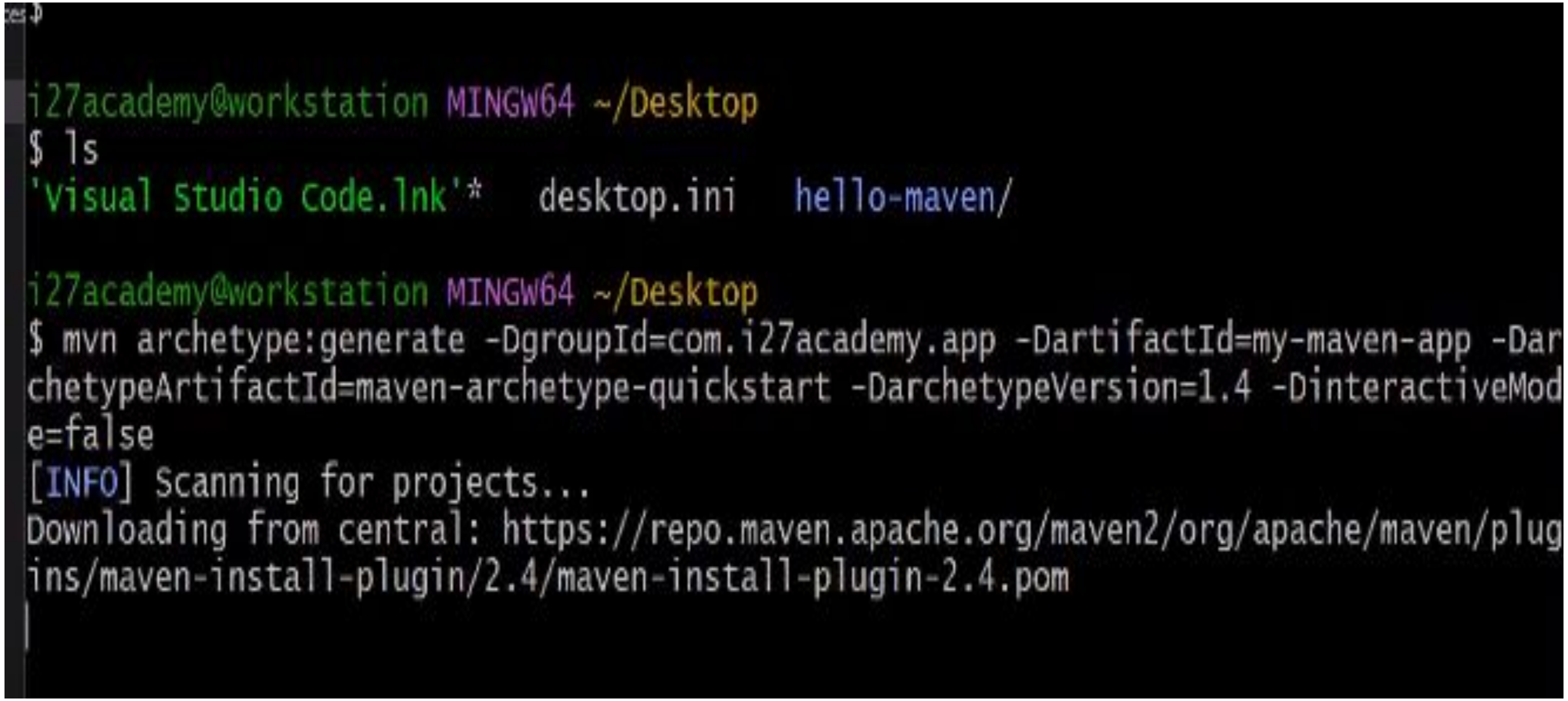
For the first time i have created mvn … so i dont know the structure of pom.xml file so i need to ask mvn for that i have to go through the below doc

[Maven – Maven in 5 Minutes (apache.org)]()

```
mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=my-app
-DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 -DinteractiveMode=false
```

```
Change the company name as academy and change the artifact id to my-maven or something as per our
requirement.
```

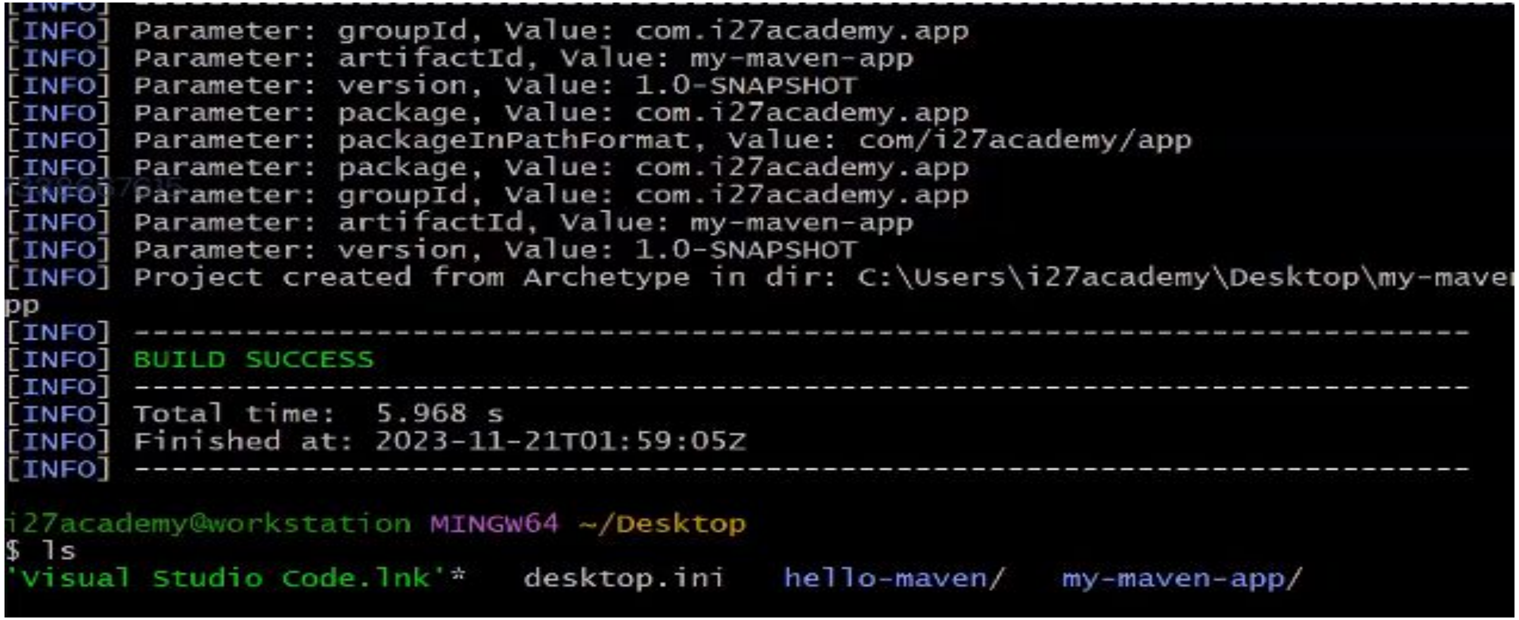
```
Now run that command on bash
```



```
i27academy@workstation MINGW64 ~/Desktop
$ ls
'Visual Studio Code.lnk'*   desktop.ini   hello-maven/

i27academy@workstation MINGW64 ~/Desktop
$ mvn archetype:generate -DgroupId=com.i27academy.app -DartifactId=my-maven-app -Dar
chetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 -DinteractiveMod
e=false
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plug
ins/maven-install-plugin/2.4/maven-install-plugin-2.4.pom
```

I got the output as below

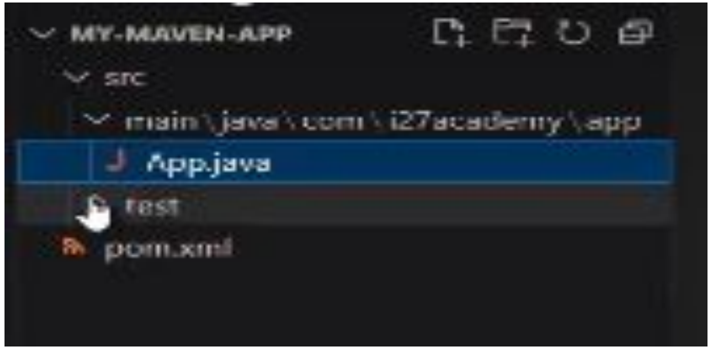


```
[INFO] Parameter: groupId, Value: com.i27academy.app
[INFO] Parameter: artifactId, Value: my-maven-app
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: com.i27academy.app
[INFO] Parameter: packageInPathFormat, Value: com/i27academy/app
[INFO] Parameter: package, Value: com.i27academy.app
[INFO] Parameter: groupId, Value: com.i27academy.app
[INFO] Parameter: artifactId, Value: my-maven-app
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Project created from Archetype in dir: C:\Users\i27academy\Desktop\my-maven
pp
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  5.968 s
[INFO] Finished at: 2023-11-21T01:59:05Z
[INFO] ------------------------------------------------------------------------

i27academy@workstation MINGW64 ~/Desktop
$ ls
'Visual Studio Code.lnk'*   desktop.ini   hello-maven/   my-maven-app/
```

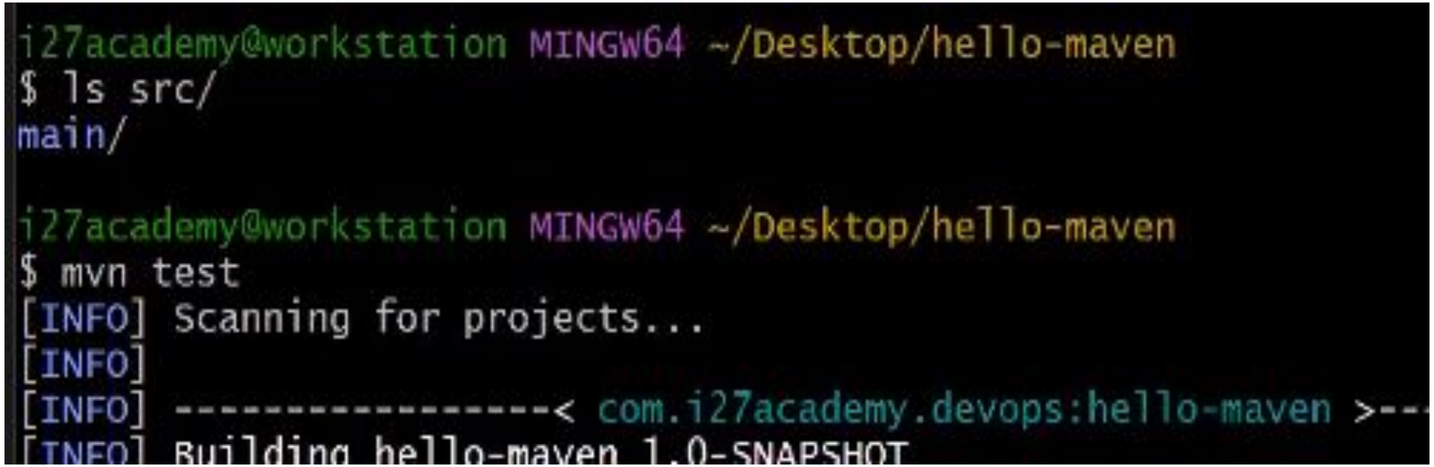
Open maven folder and verify the pom.xml file ..this file also contains dependencies as well

In maven folder along with pom.xml file it also contains main folder, in that src code,test files are there.



mvn test –validate+compile

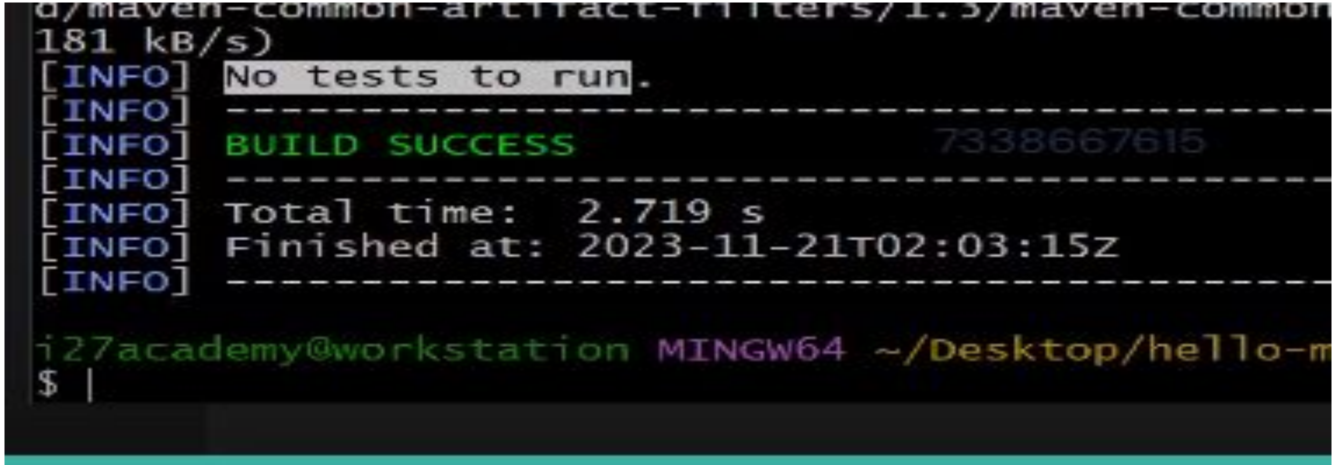
This is one scenario and we have creates one folder hello maven in that test folder is not there so do the below steps



```
i27academy@workstation MINGW64 ~/Desktop/hello-maven
$ ls src/
main/

i27academy@workstation MINGW64 ~/Desktop/hello-maven
$ mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------< com.i27academy.devops:hello-maven >---
[INFO] Building hello-maven 1.0-SNAPSHOT
```

It displays like no tests to run as below



```
d/maven-common-artifact-filters/1.5/maven-common
181 kB/s)
[INFO] No tests to run.
[INFO] ----------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ----------------------------------------------------
[INFO] Total time:  2.719 s
[INFO] Finished at: 2023-11-21T02:03:15Z
[INFO] ----------------------------------------------------

i27academy@workstation MINGW64 ~/Desktop/hello-m
$ |
```

packages

→now let us assume some scenario

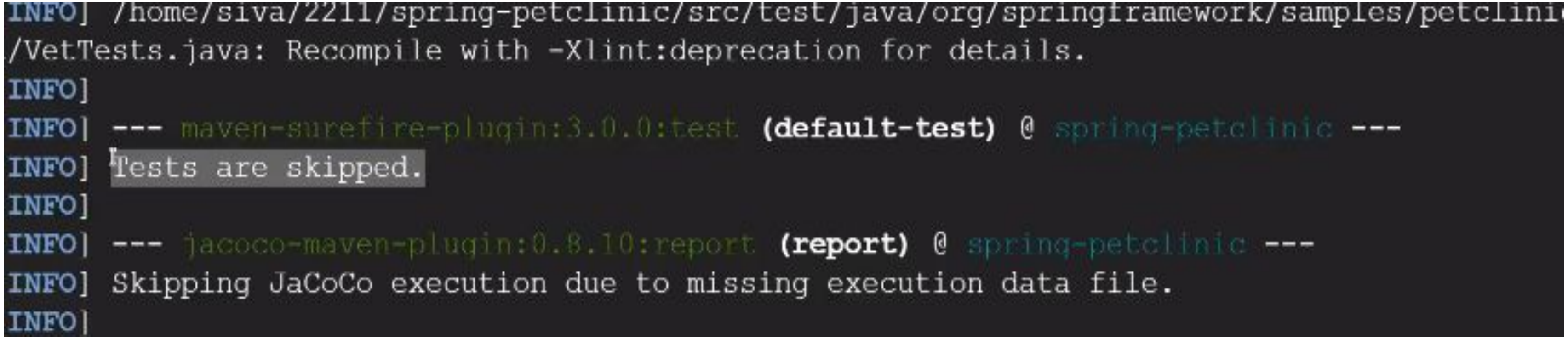
→take one java application git hub project it has lot of commits in src code from developer…assume one commit and that commit have some error so directly go to that commit by clicking right option and and create one branch(nonworking) in that commit.

→now go to bash and clone that github url and see commits using log command git checkout that commit path to point main to that branch

Now do mvn clean and mvn package(validate+compile+test)

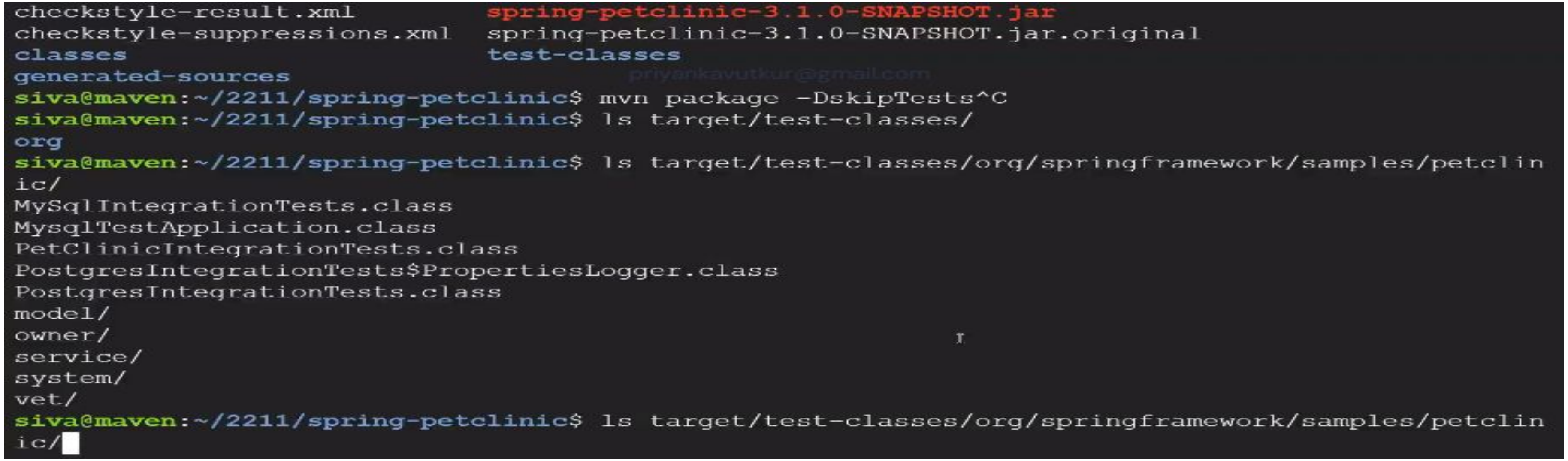
→but i got an error in test step…before i validate this it is working but when i clone the new changes then only it got fails so to skip the test steps to get package step we have a command as below

mvn package -DskipTests



```
INFO] /home/siva/2211/spring-petclinic/src/test/java/org/springframework/samples/petclini
/VetTests.java: Recompile with -Xlint:deprecation for details.
INFO]
INFO] --- maven-surefire-plugin:3.0.0:test (default-test) @ spring-petclinic ---
INFO] Tests are skipped.
INFO]
INFO] --- jacoco-maven-plugin:0.8.10:report (report) @ spring-petclinic ---
INFO] Skipping JaCoCo execution due to missing execution data file.
INFO]
```

We have testclasses even though we skip the tests becausewithin testfolder,java files got compiled and those files will be stored as testclass file in target folder



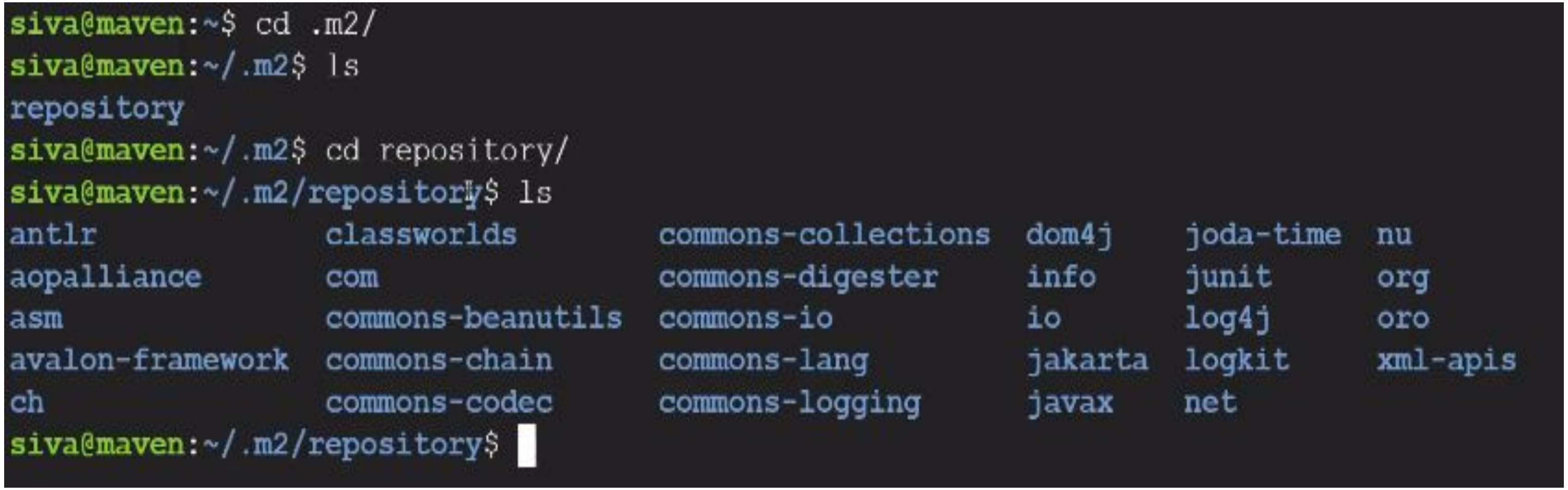
```
checkstyle-result.xml          spring-petclinic-3.1.0-SNAPSHOT.jar
checkstyle-suppressions.xml    spring-petclinic-3.1.0-SNAPSHOT.jar.original
classes                        test-classes
generated-sources
siva@maven:~/2211/spring-petclinic$ mvn package -DskipTests^C
siva@maven:~/2211/spring-petclinic$ ls target/test-classes/
org
siva@maven:~/2211/spring-petclinic$ ls target/test-classes/org/springframework/samples/petclin
ic/
MySqlIntegrationTests.class
MysqlTestApplication.class
PetClinicIntegrationTests.class
PostgresIntegrationTests$PropertiesLogger.class
PostgresIntegrationTests.class
model/
owner/
service/
system/
vet/
siva@maven:~/2211/spring-petclinic$ ls target/test-classes/org/springframework/samples/petclin
ic/
```

We didnot do tests so what is need to compile those files so to avoid that we hava another command as below

mvn package -Dmaven.test.skip=true

Then it removes test cases file also and finally generated packages(war and jar files)

All the downloaded files while packaging command run will be there in m2 folder as below



```
siva@maven:~$ cd .m2/
siva@maven:~/.m2$ ls
repository
siva@maven:~/.m2$ cd repository/
siva@maven:~/.m2/repository$ ls
antlr              classworlds        commons-collections  dom4j    joda-time  nu
aopalliance        com                commons-digester     info     junit      org
asm                commons-beanutils  commons-io           io       log4j      oro
avalon-framework   commons-chain      commons-lang         jakarta  logkit     xml-apis
ch                 commons-codec      commons-logging      javax    net
siva@maven:~/.m2/repository$ 
```

→within home directory of mvn we have bin , boot,lib and conf folders

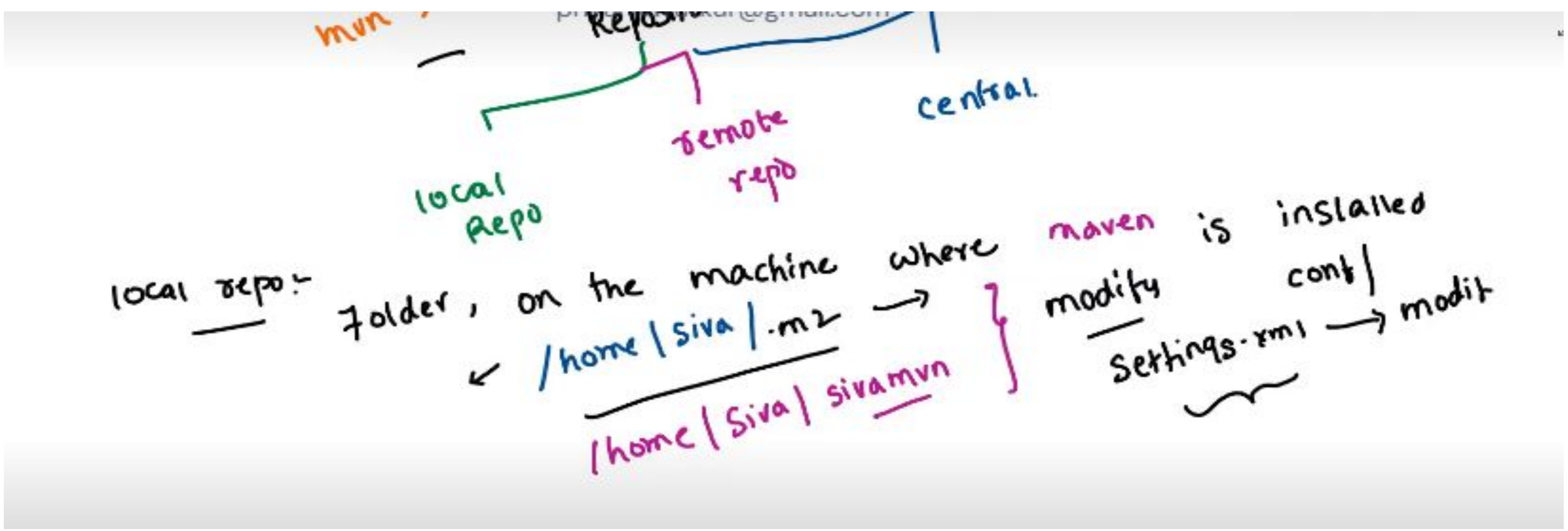
→bin folder we have binary files,boot folder we have running files,with in lib folder we have jar and war files and finally conf folder we have settings.xml files which is used for modifying mvn related objects

→we have 3 types of repositories

1.local.remote and centralized repositories

Local repository

Where we installed maven we got .m2 folder and we can change the name as well by through settings.xml



local repo:- folder, on the machine where maven is installed

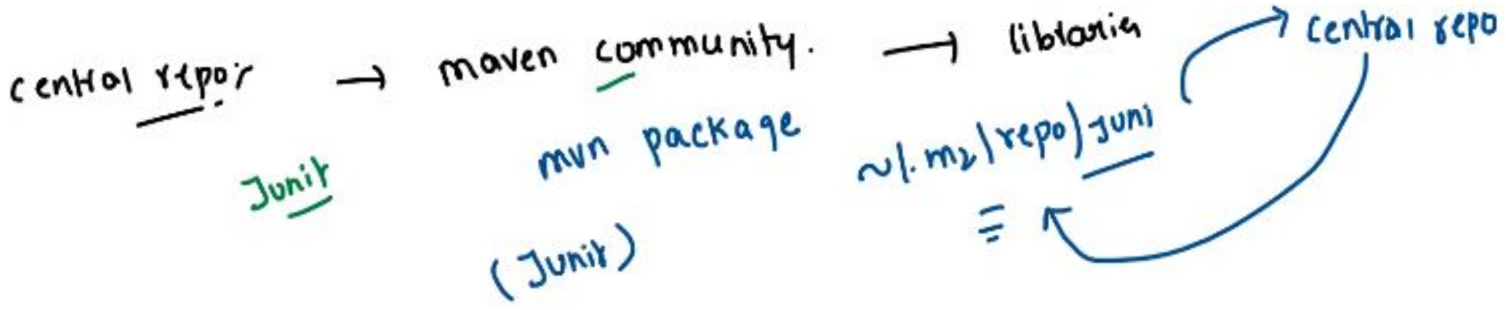
/home/siva/.m2 → } modify conf/

/home/siva/sivamvn

settings.xml → modif

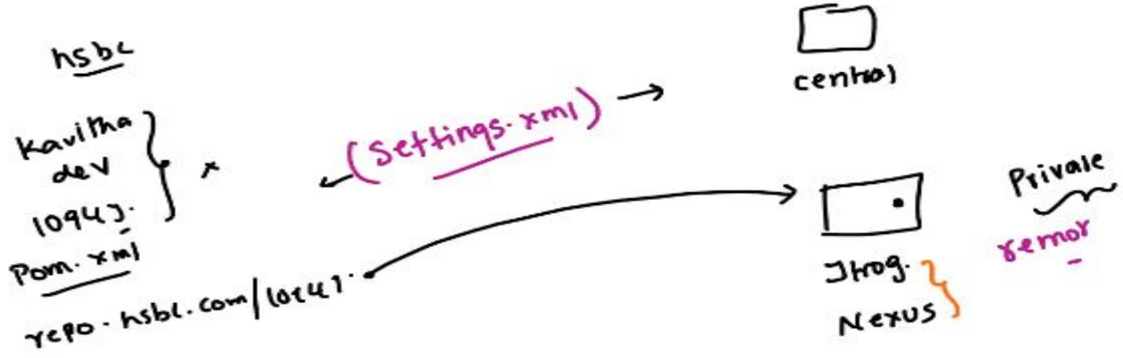
Centralised repo

If the local repo any dependencies which is not available in local then it takes it from centralised repo



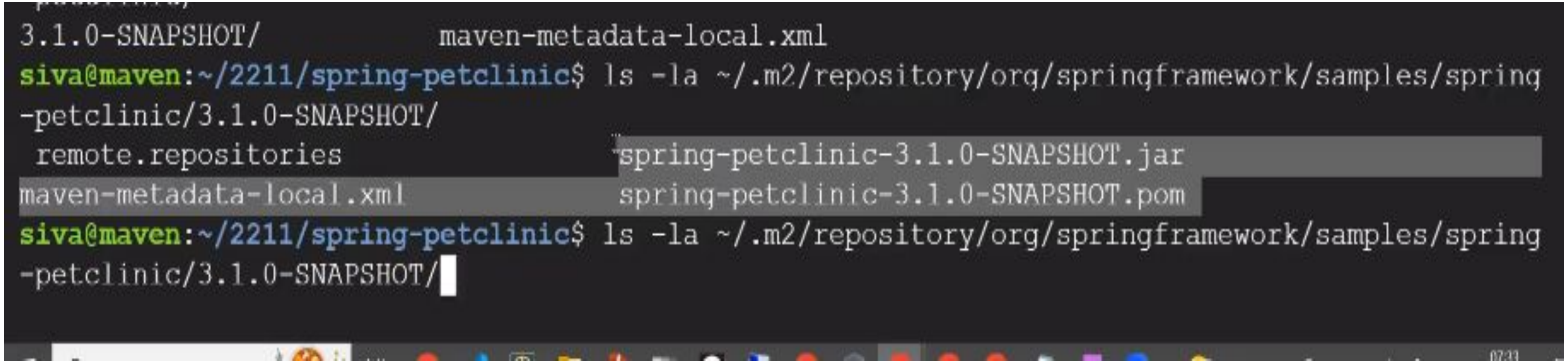
Remote repo

Normally any organisation will not support centralised repo they created their own repo which is called remote repo ..here nexus and jfrog …in this we have lot of dependencies as like centralised …settings.xml will decide from where we can get dependencies if not available in local .



mvn package —will create jar/war folder within target folder

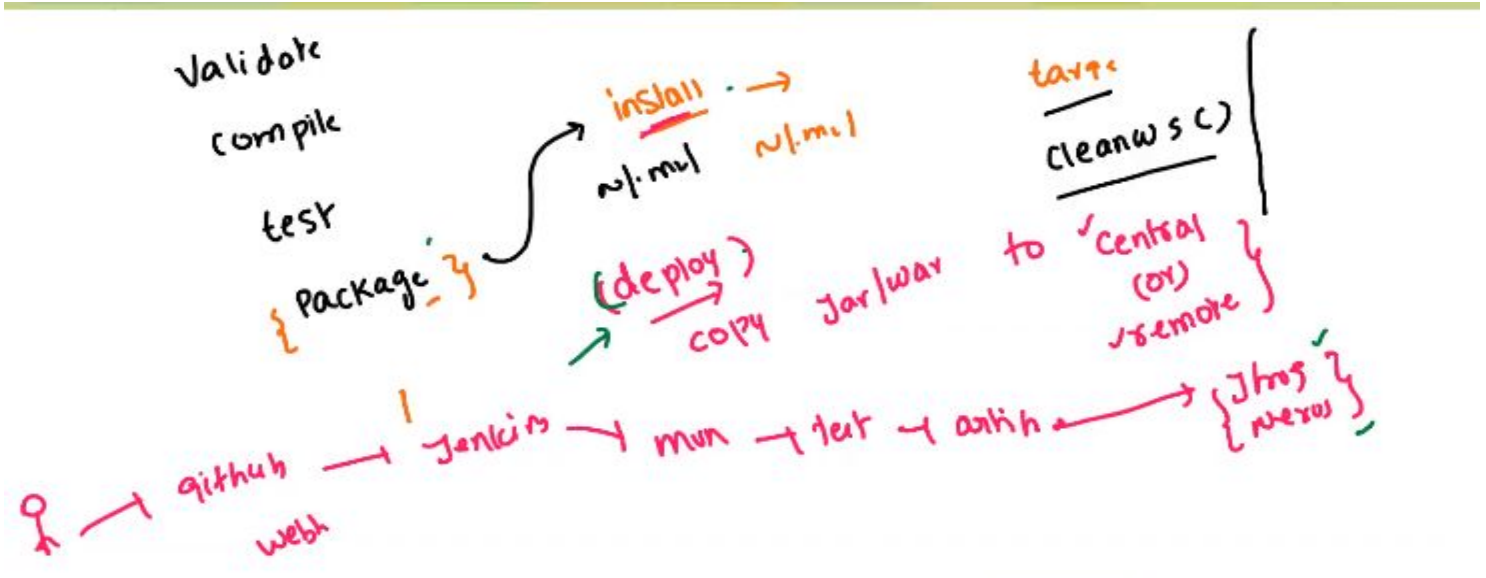
mvn install –it installs above 2 files and pom file within m2 /repo directory i.e., local directory



```
3.1.0-SNAPSHOT/           maven-metadata-local.xml
siva@maven:~/2211/spring-petclinic$ ls -la ~/.m2/repository/org/springframework/samples/spring
-petclinic/3.1.0-SNAPSHOT/
 remote.repositories                    spring-petclinic-3.1.0-SNAPSHOT.jar
maven-metadata-local.xml                spring-petclinic-3.1.0-SNAPSHOT.pom
siva@maven:~/2211/spring-petclinic$ ls -la ~/.m2/repository/org/springframework/samples/spring
-petclinic/3.1.0-SNAPSHOT/
```

Final step deploy

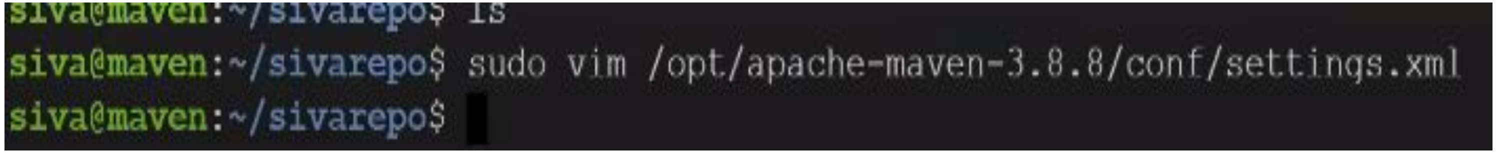
All these process can be done by modifying settings.xml file



7338667615

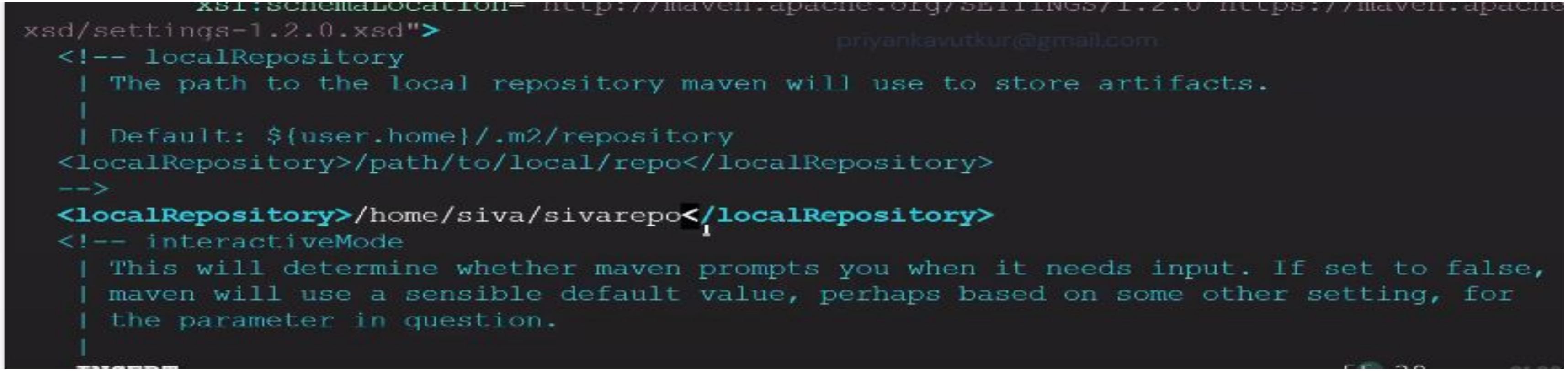
Modifying folder name from m2 to mypath

Open below path



```
siva@maven:~/sivarepo$ ls
siva@maven:~/sivarepo$ sudo vim /opt/apache-maven-3.8.8/conf/settings.xml
siva@maven:~/sivarepo$ 
```

Modify as below



```
          xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.2.0 https://maven.apache
xsd/settings-1.2.0.xsd">
  <!-- localRepository
   | The path to the local repository maven will use to store artifacts.
   |
   | Default: ${user.home}/.m2/repository
  <localRepository>/path/to/local/repo</localRepository>
  -->
  <localRepository>/home/siva/sivarepo</localRepository>
  <!-- interactiveMode
   | This will determine whether maven prompts you when it needs input. If set to false,
   | maven will use a sensible default value, perhaps based on some other setting, for
   | the parameter in question.
   |
```