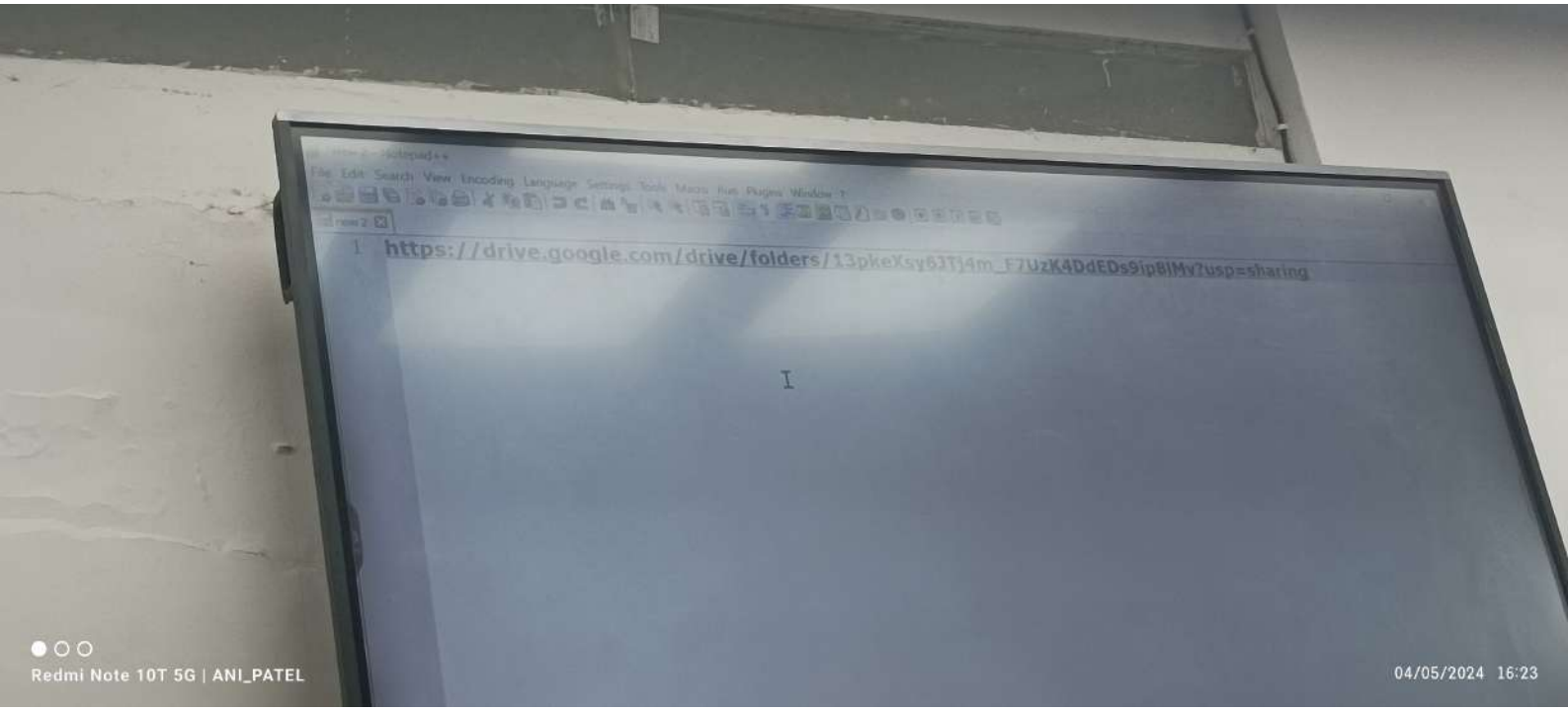
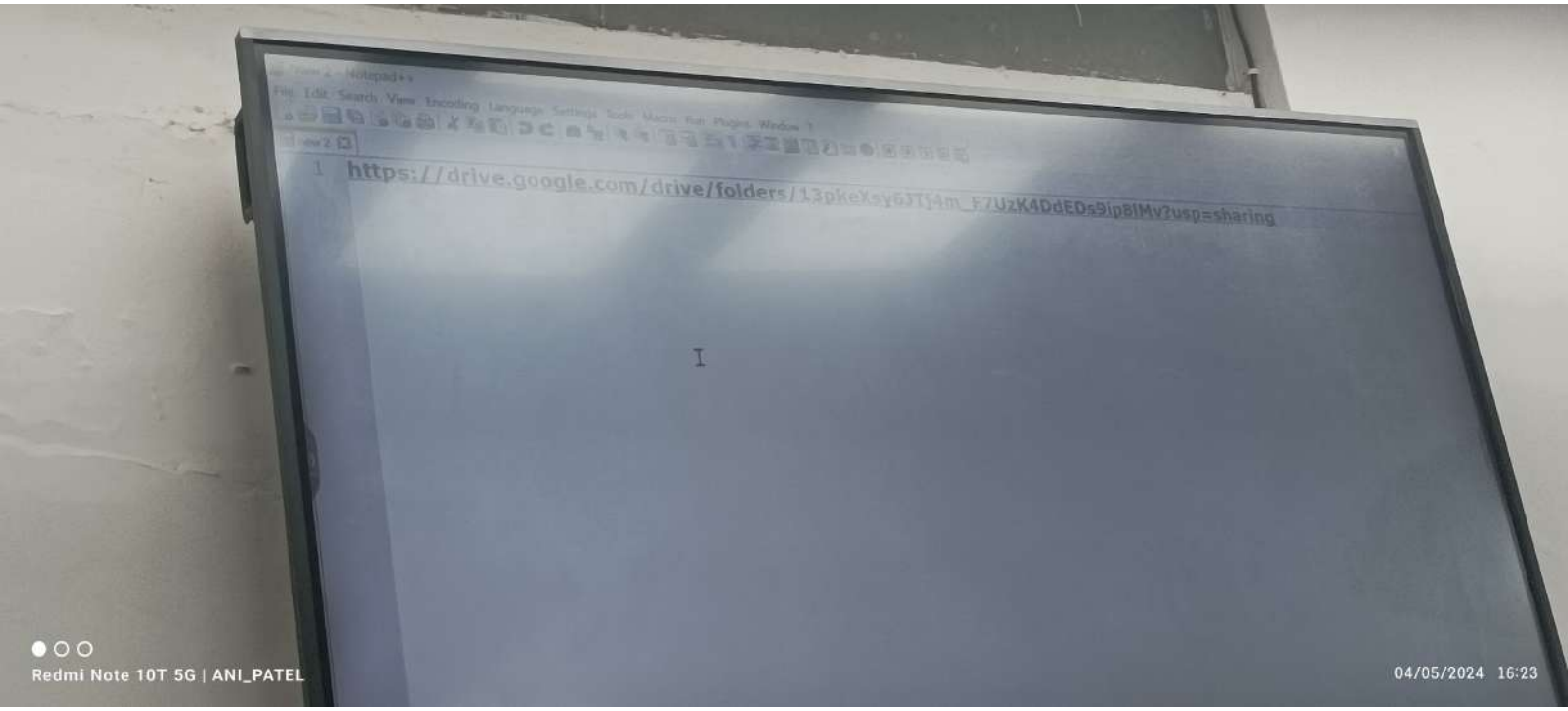
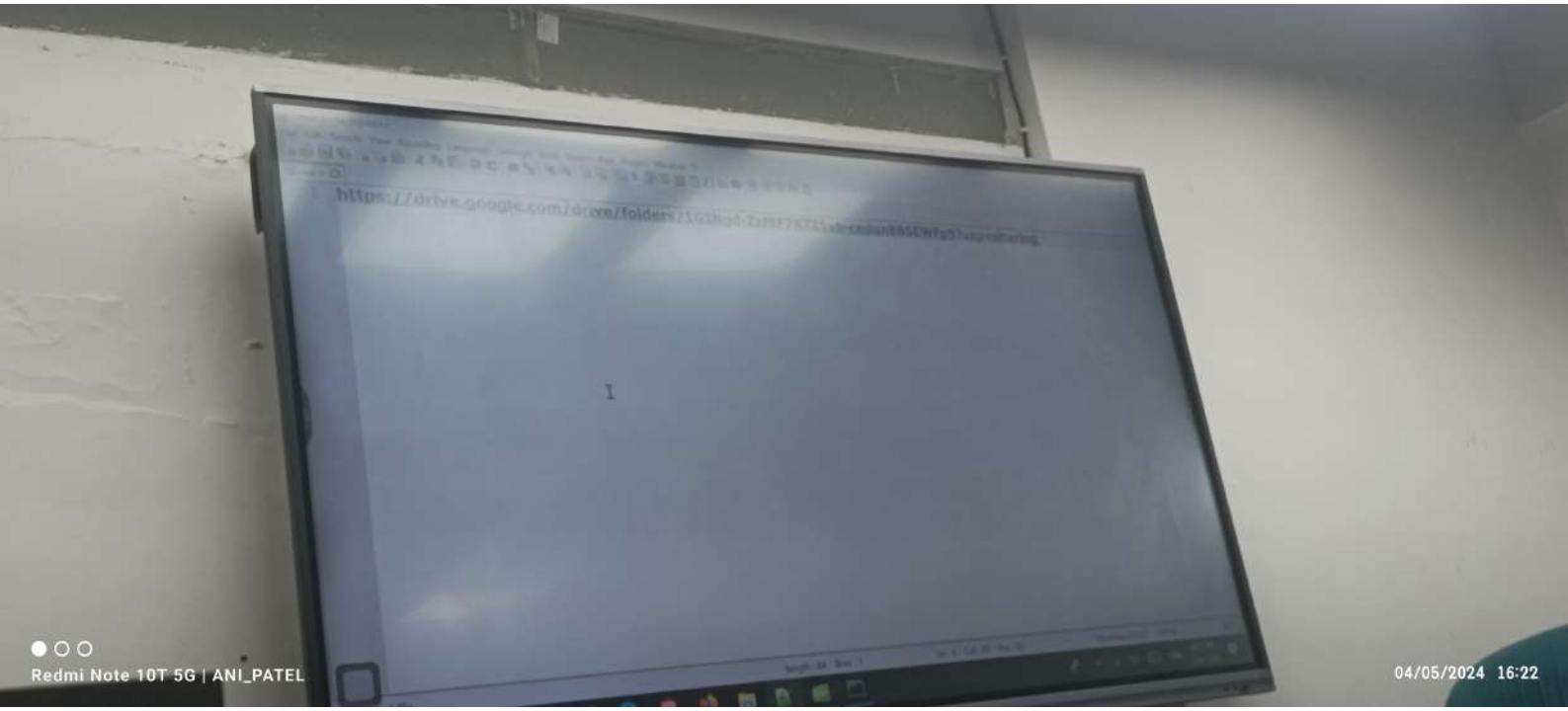
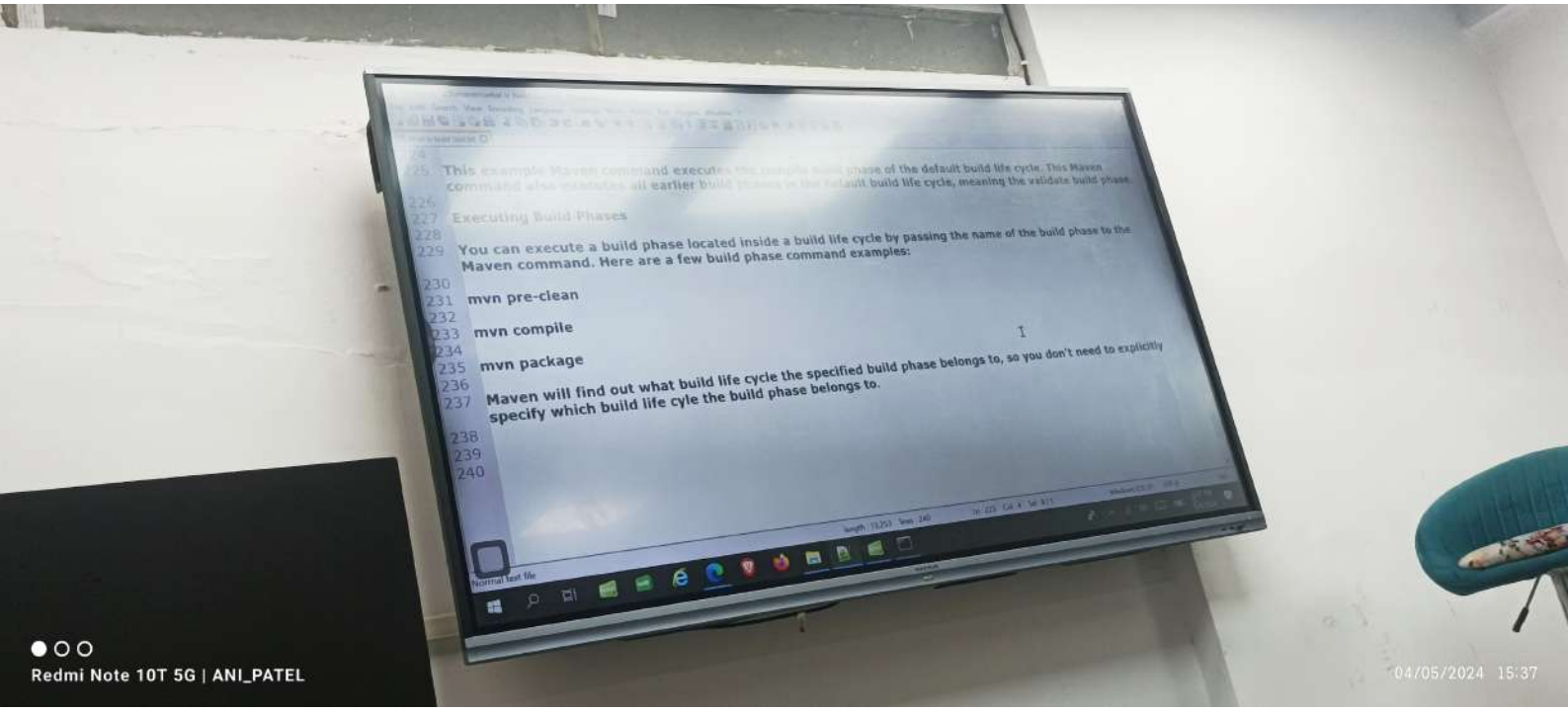
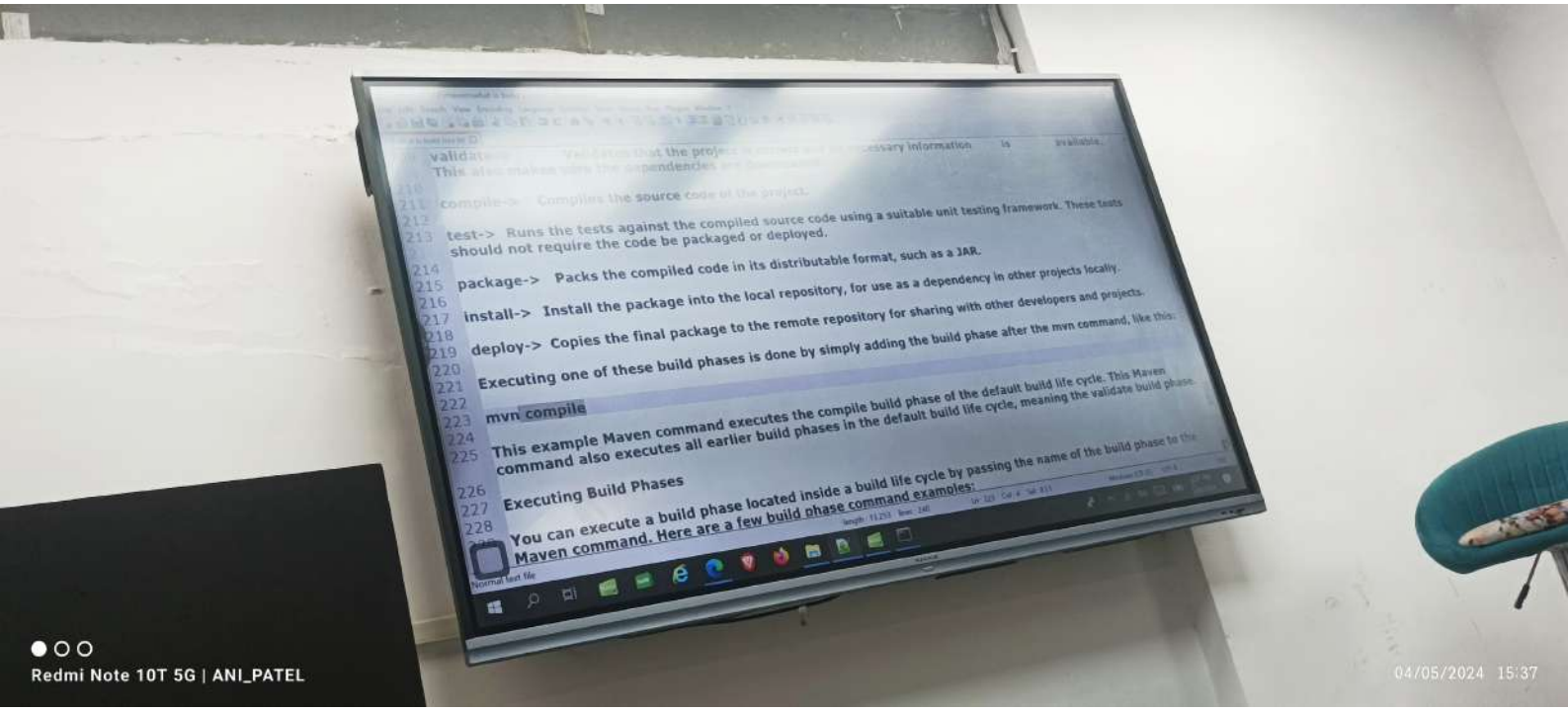
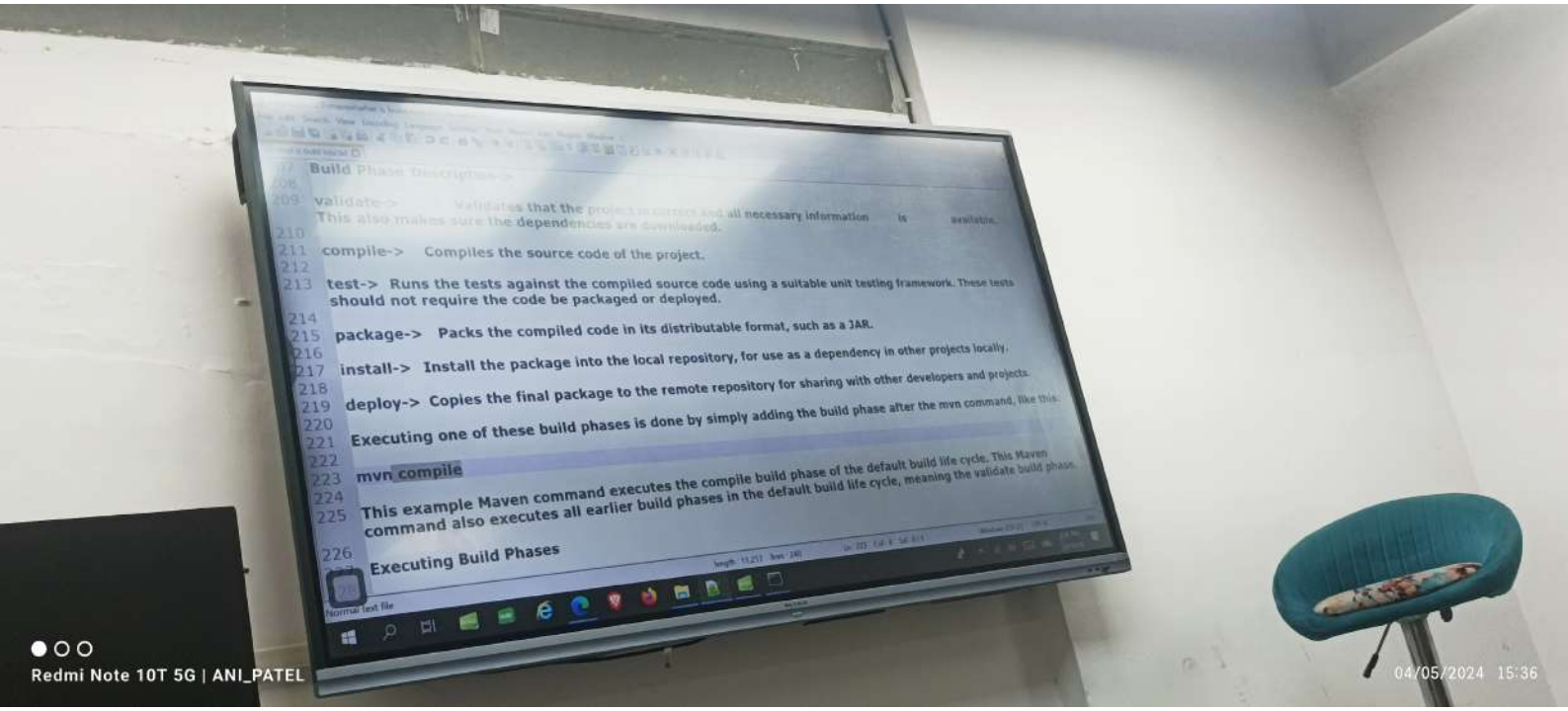

https://drive.google.com/drive/folders/13pkeXsy63Tj4m_F7UzK4DdEDs9ip8jMv?usp=sharing
Redmi Note 10T 5G | ANI_PATEL
04/05/2024 16:23

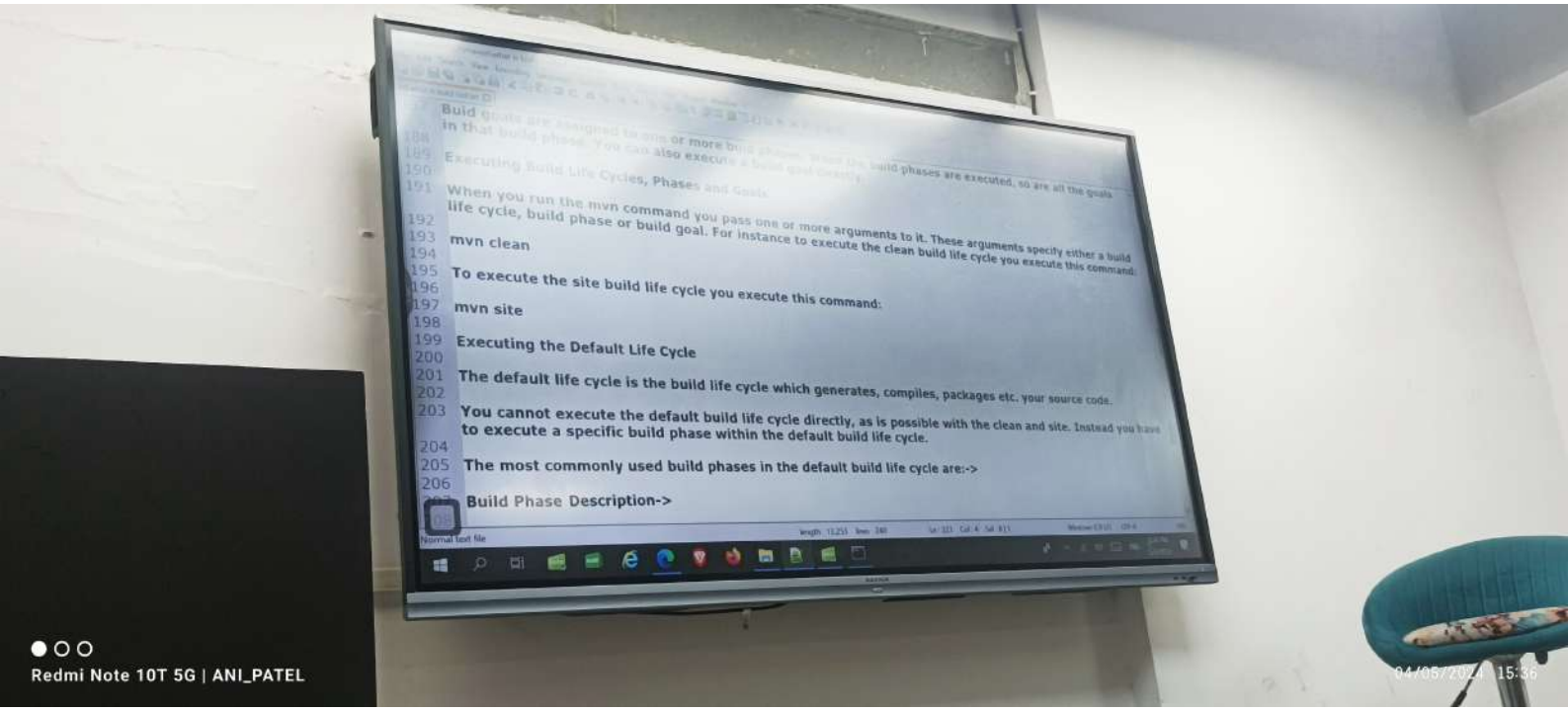

1 https://drive.google.com/drive/folders/13pkeXsy6JTj4m_F7UzK4DdEDs9ip8IMv?usp=sharing
Redmi Note 10T 5G | ANI_PATEL
04/05/2024 16:23

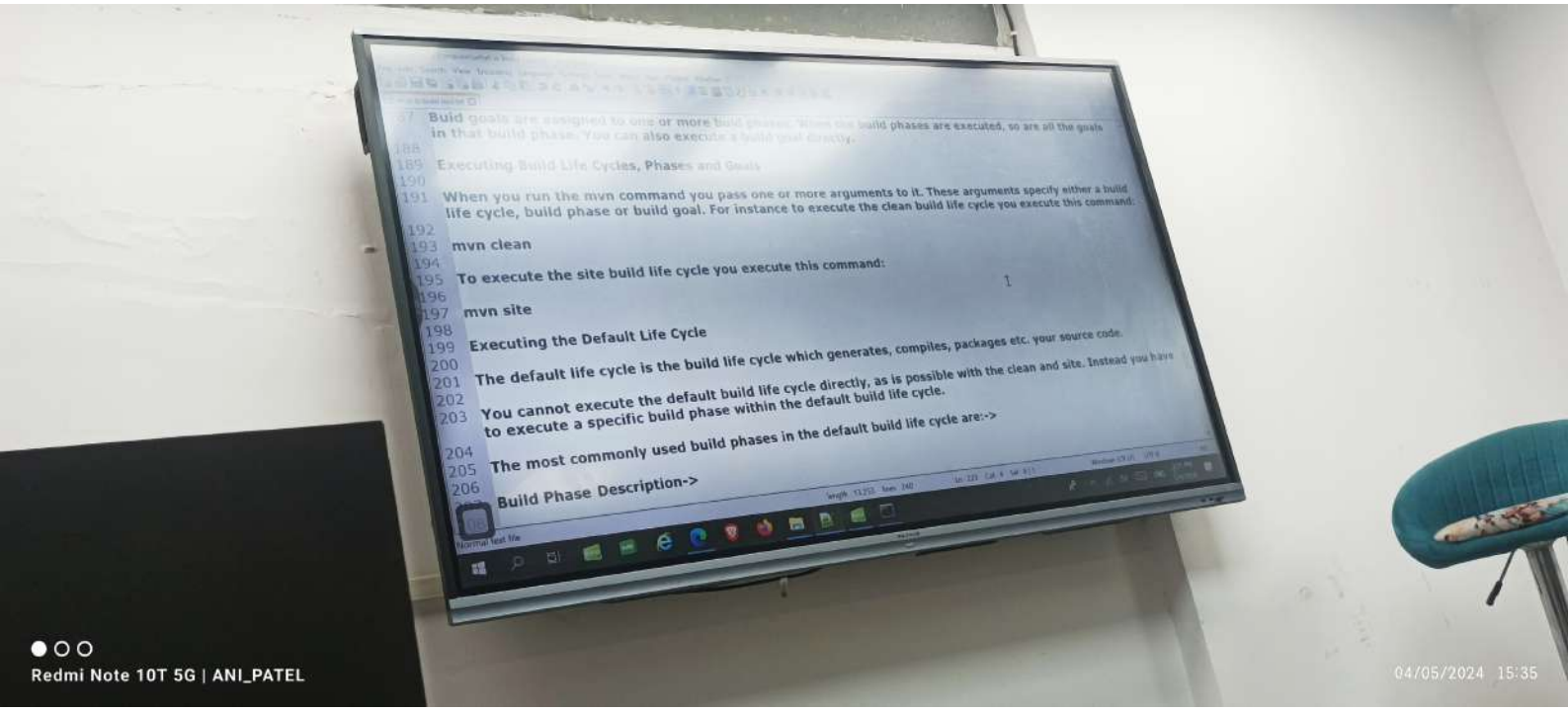

https://drive.google.com/drive/folders/
Redmi Note 10T 5G | ANI_PATEL
04/05/2024 16:22

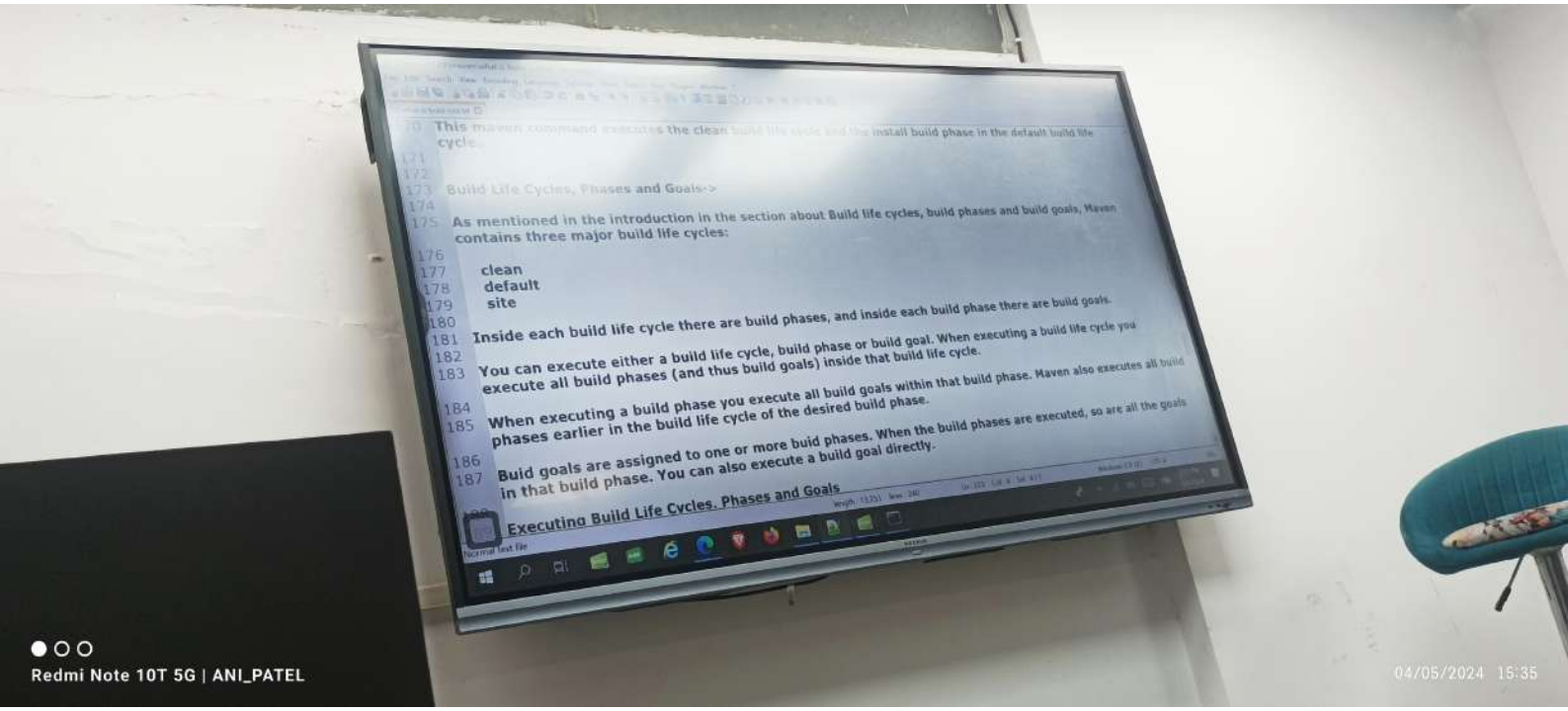

This example Maven command executes the compile build phase of the default build life cycle. This Maven command also executes all earlier build phases in the default build life cycle, meaning the validate build phase.
Executing Build Phases
You can execute a build phase located inside a build life cycle by passing the name of the build phase to the Maven command. Here are a few build phase command examples:
mvn pre-clean
mvn compile
mvn package
Maven will find out what build life cycle the specified build phase belongs to, so you don't need to explicitly specify which build life cyle the build phase belongs to.
Normal text file
Redmi Note 10T 5G | ANI_PATEL
04/05/2024 15:37

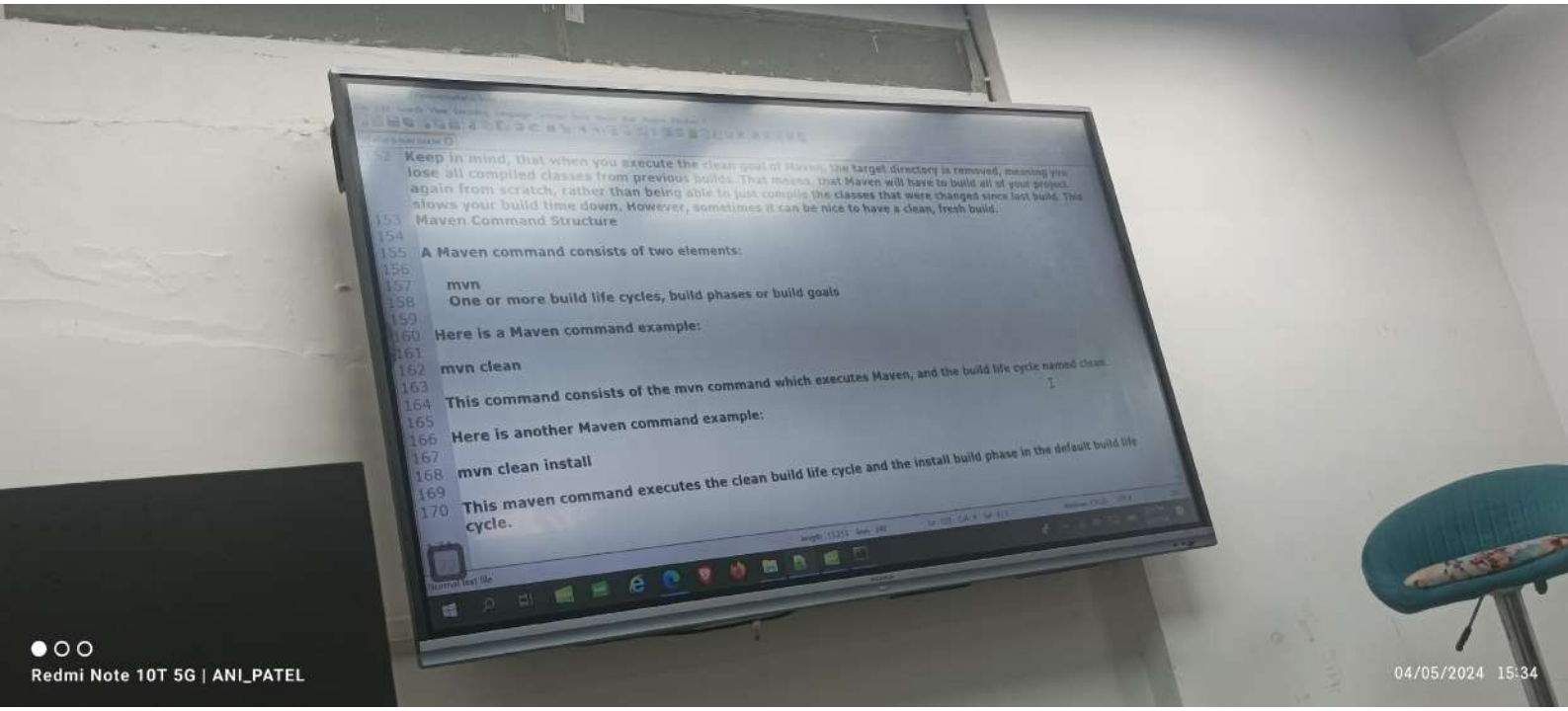

test-> Runs the tests against the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed.
package-> Packs the compiled code in its distributable format, such as a JAR.
install-> Install the package into the local repository, for use as a dependency in other projects locally.
deploy-> Copies the final package to the remote repository for sharing with other developers and projects.
Executing one of these build phases is done by simply adding the build phase after the mvn command, like this:
mvn compile
This example Maven command executes the compile build phase of the default build life cycle. This Maven command also executes all earlier build phases in the default build life cycle, meaning the validate build phase.
Executing Build Phases
You can execute a build phase located inside a build life cycle by passing the name of the build phase to the Maven command. Here are a few build phase command examples:
Redmi Note 10T 5G | ANI_PATEL
04/05/2024 15:37

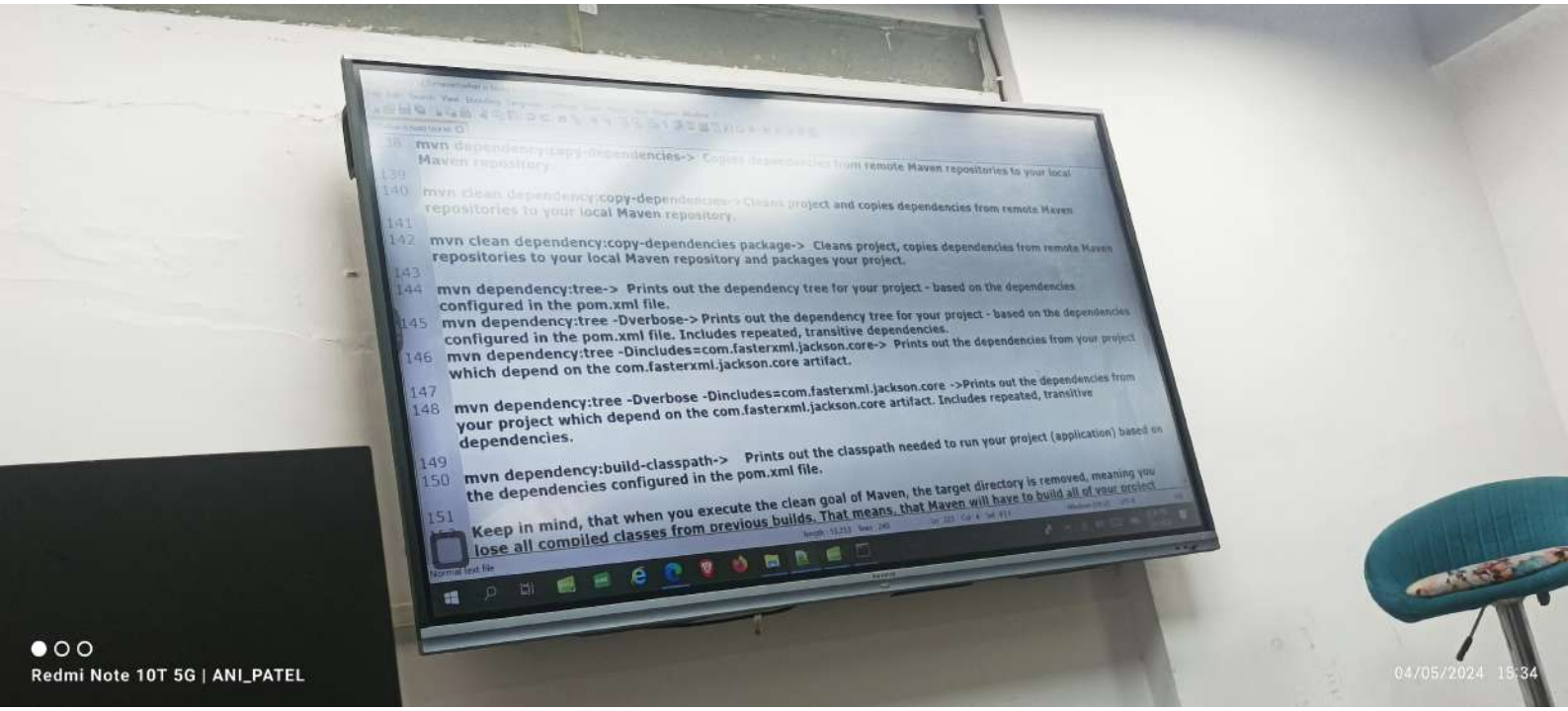

Build Phase Description->
validate-> Validates that the project is correct and all necessary information is available.
This also makes sure the dependencies are downloaded.
compile-> Compiles the source code of the project.
test-> Runs the tests against the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed.
package-> Packs the compiled code in its distributable format, such as a JAR.
install-> Install the package into the local repository, for use as a dependency in other projects locally.
deploy-> Copies the final package to the remote repository for sharing with other developers and projects.
Executing one of these build phases is done by simply adding the build phase after the mvn command, like this.
mvn compile
This example Maven command executes the compile build phase of the default build life cycle. This Maven command also executes all earlier build phases in the default build life cycle, meaning the validate build phase.
Executing Build Phases
Normal text file
Redmi Note 10T 5G | ANI_PATEL
04/05/2024 15:36

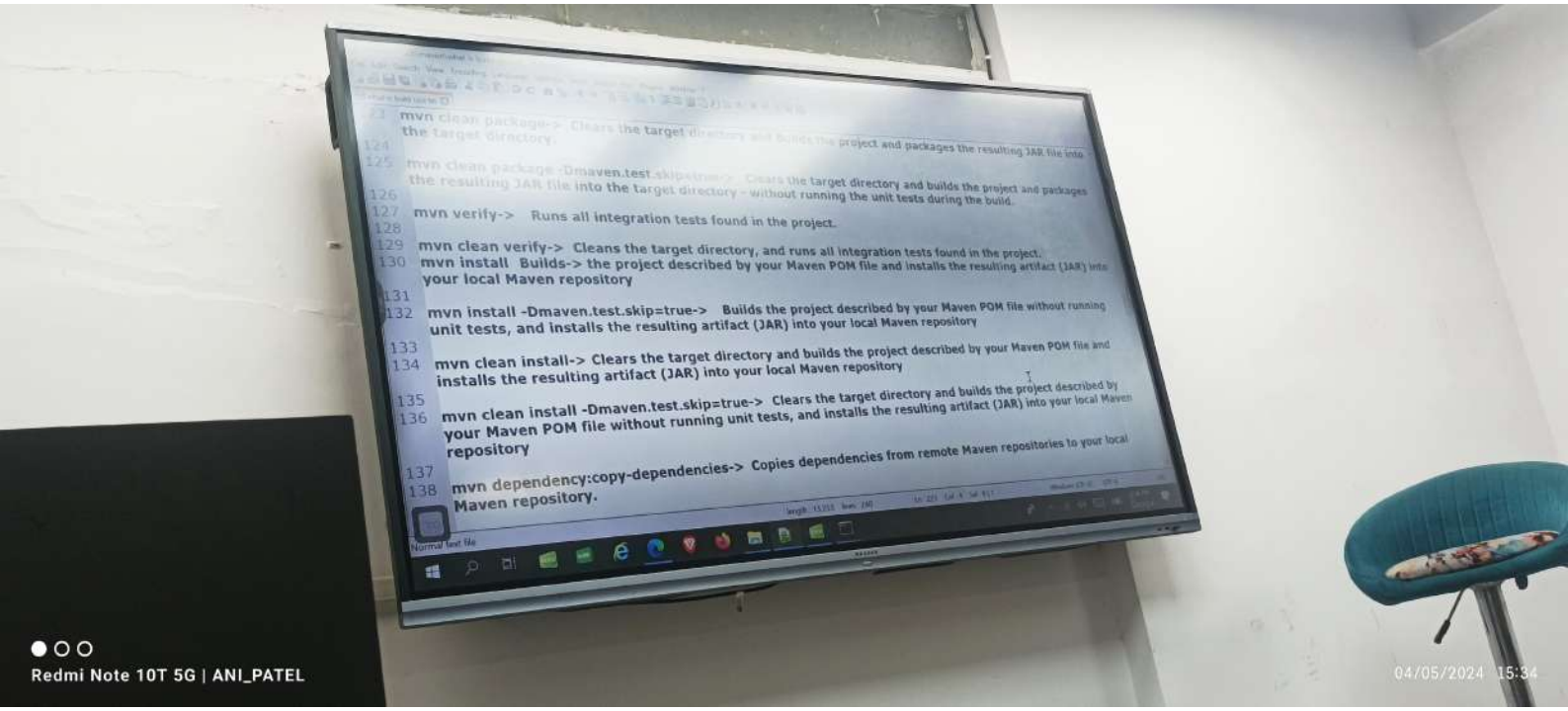

in that build phase. You can also execute a build goal
Executing Build Life Cycles, Phases and Goals
When you run the mvn command you pass one or more arguments to it. These arguments specify either a build life cycle, build phase or build goal. For instance to execute the clean build life cycle you execute this command:
mvn clean
To execute the site build life cycle you execute this command:
mvn site
Executing the Default Life Cycle
The default life cycle is the build life cycle which generates, compiles, packages etc. your source code.
You cannot execute the default build life cycle directly, as is possible with the clean and site. Instead you have to execute a specific build phase within the default build life cycle.
The most commonly used build phases in the default build life cycle are:->
Build Phase Description->
Redmi Note 10T 5G | ANI_PATEL
04/05/2024 15:36

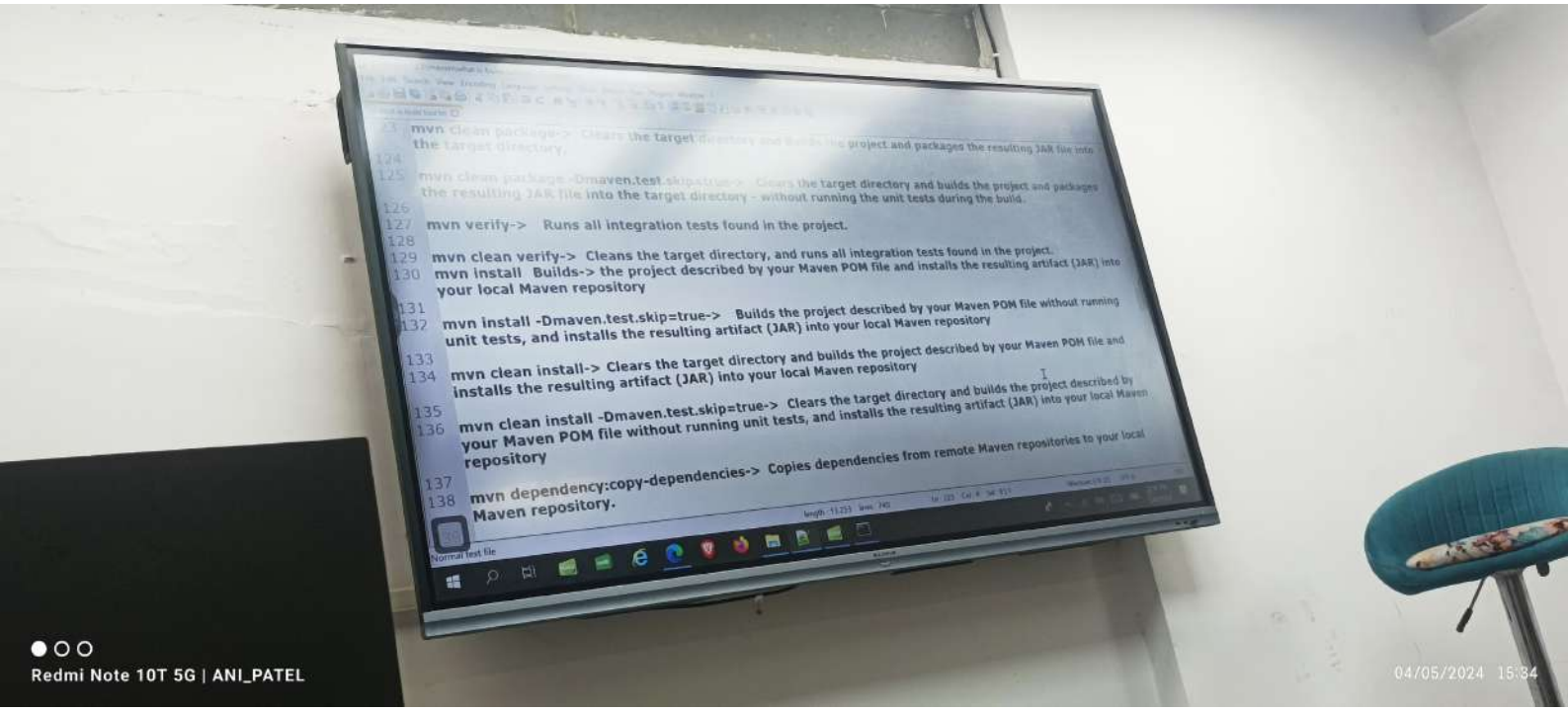

Buid goals are assigned to one or more build phases. When the build phases are executed, so are all the goals in that build phase. You can also execute a build goal directly.
Executing Build Life Cycles, Phases and Goals
When you run the mvn command you pass one or more arguments to it. These arguments specify either a build life cycle, build phase or build goal. For instance to execute the clean build life cycle you execute this command:
mvn clean
To execute the site build life cycle you execute this command:
mvn site
Executing the Default Life Cycle
The default life cycle is the build life cycle which generates, compiles, packages etc. your source code.
You cannot execute the default build life cycle directly, as is possible with the clean and site. Instead you have to execute a specific build phase within the default build life cycle.
The most commonly used build phases in the default build life cycle are:->
Build Phase Description->
Redmi Note 10T 5G | ANI_PATEL
04/05/2024 15:35

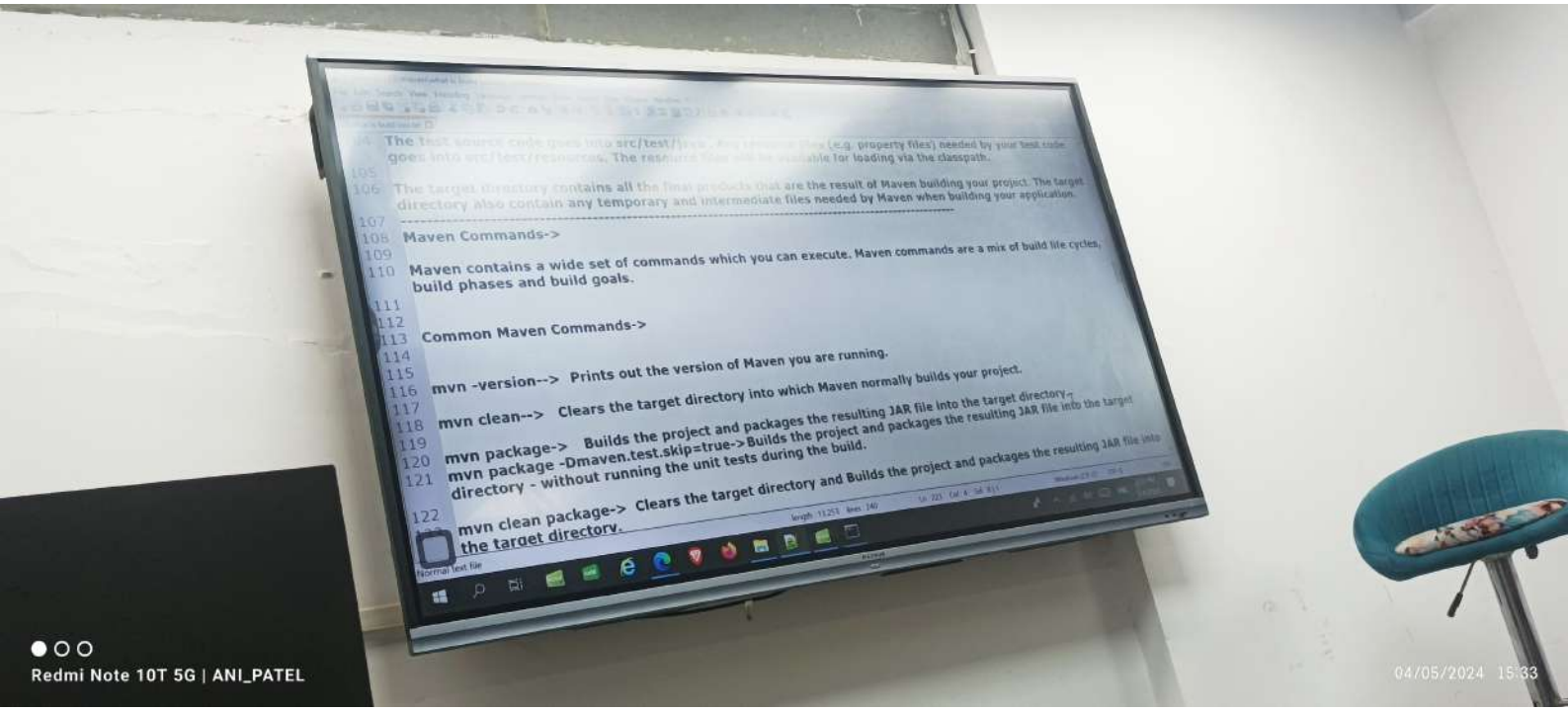

This maven command executes the clean build life cycle and the install build phase in the default build life cycle.
Build Life Cycles, Phases and Goals->
As mentioned in the introduction in the section about Build life cycles, build phases and build goals, Maven contains three major build life cycles:
clean
default
site
Inside each build life cycle there are build phases, and inside each build phase there are build goals.
You can execute either a build life cycle, build phase or build goal. When executing a build life cycle you execute all build phases (and thus build goals) inside that build life cycle.
When executing a build phase you execute all build goals within that build phase. Maven also executes all build phases earlier in the build life cycle of the desired build phase.
Buid goals are assigned to one or more buid phases. When the build phases are executed, so are all the goals in that build phase. You can also execute a build goal directly.
Executing Build Life Cycles, Phases and Goals
Redmi Note 10T 5G | ANI_PATEL
04/05/2024 15:35

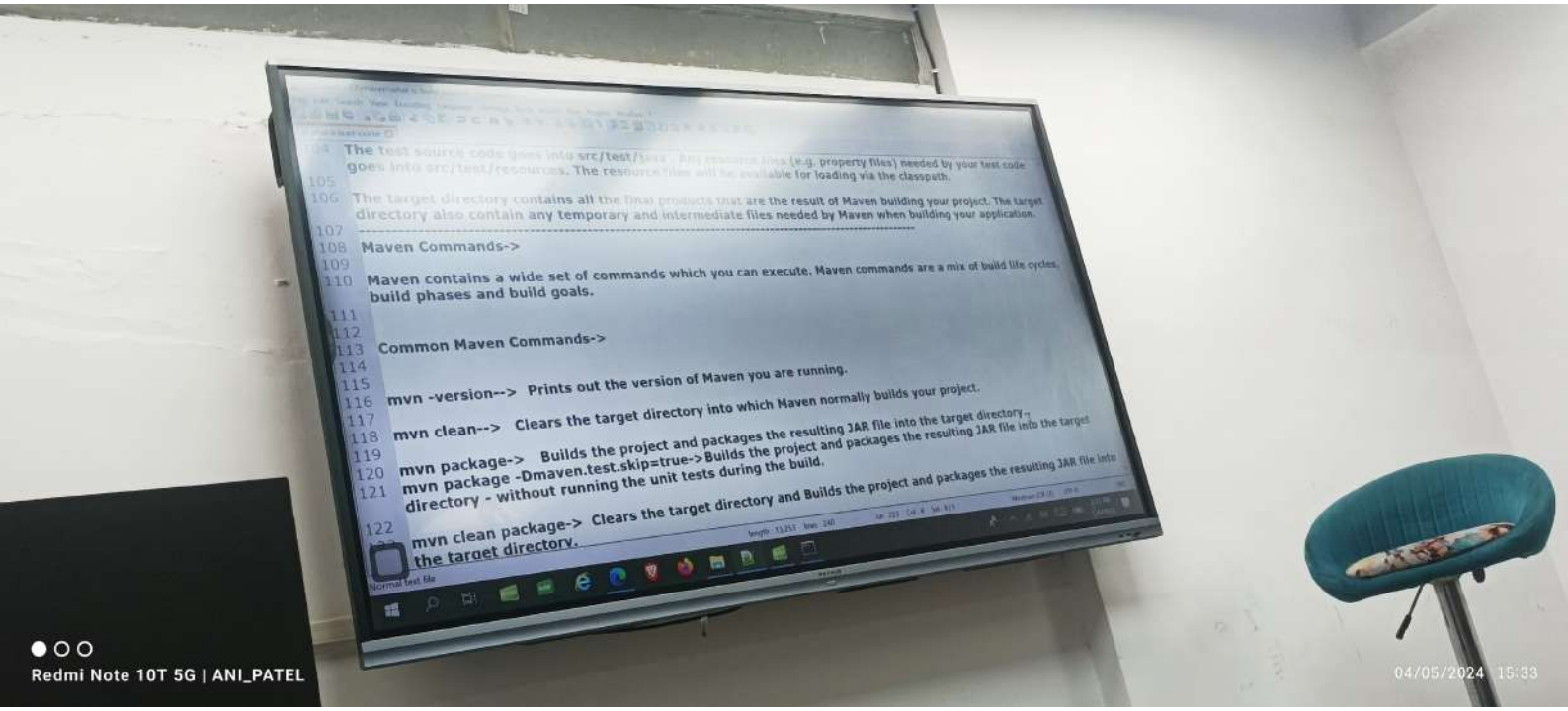


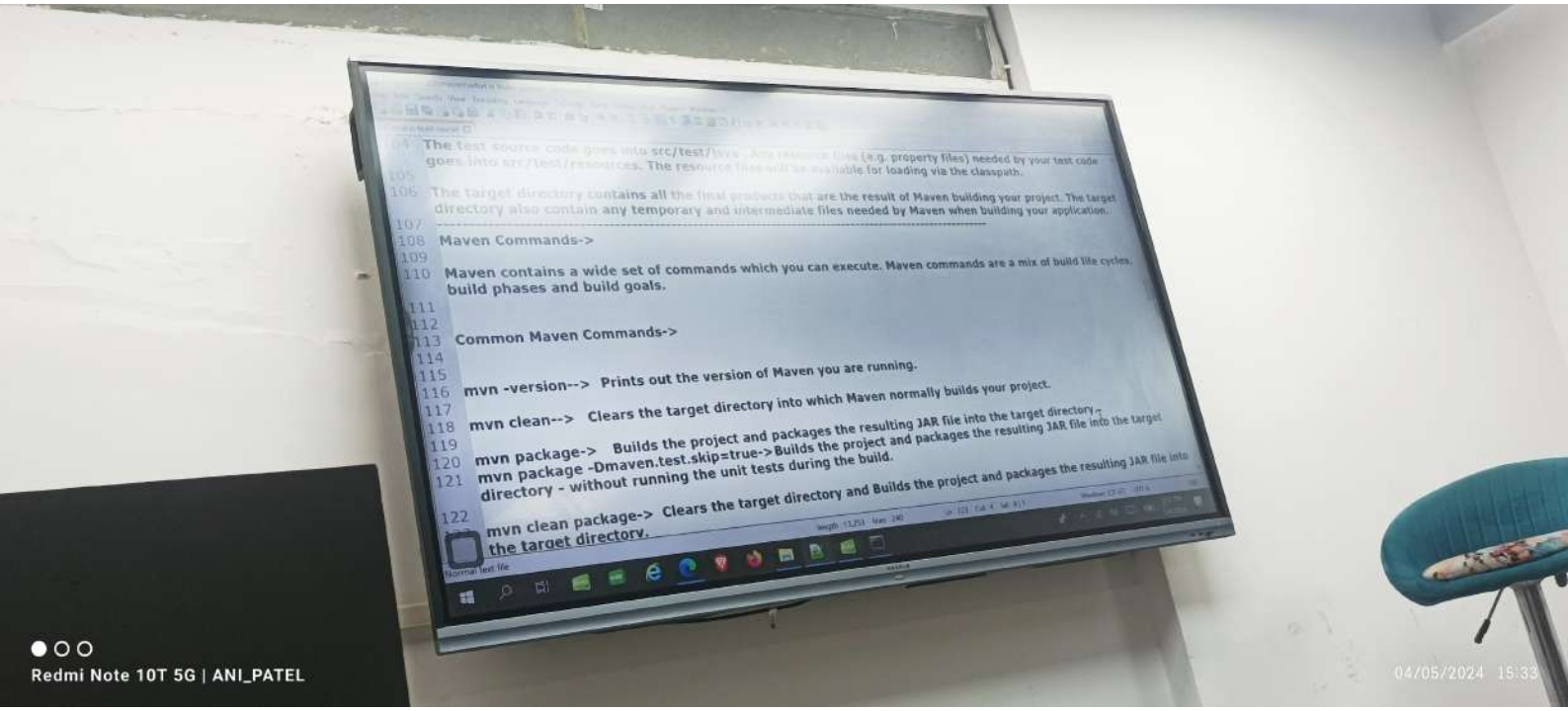


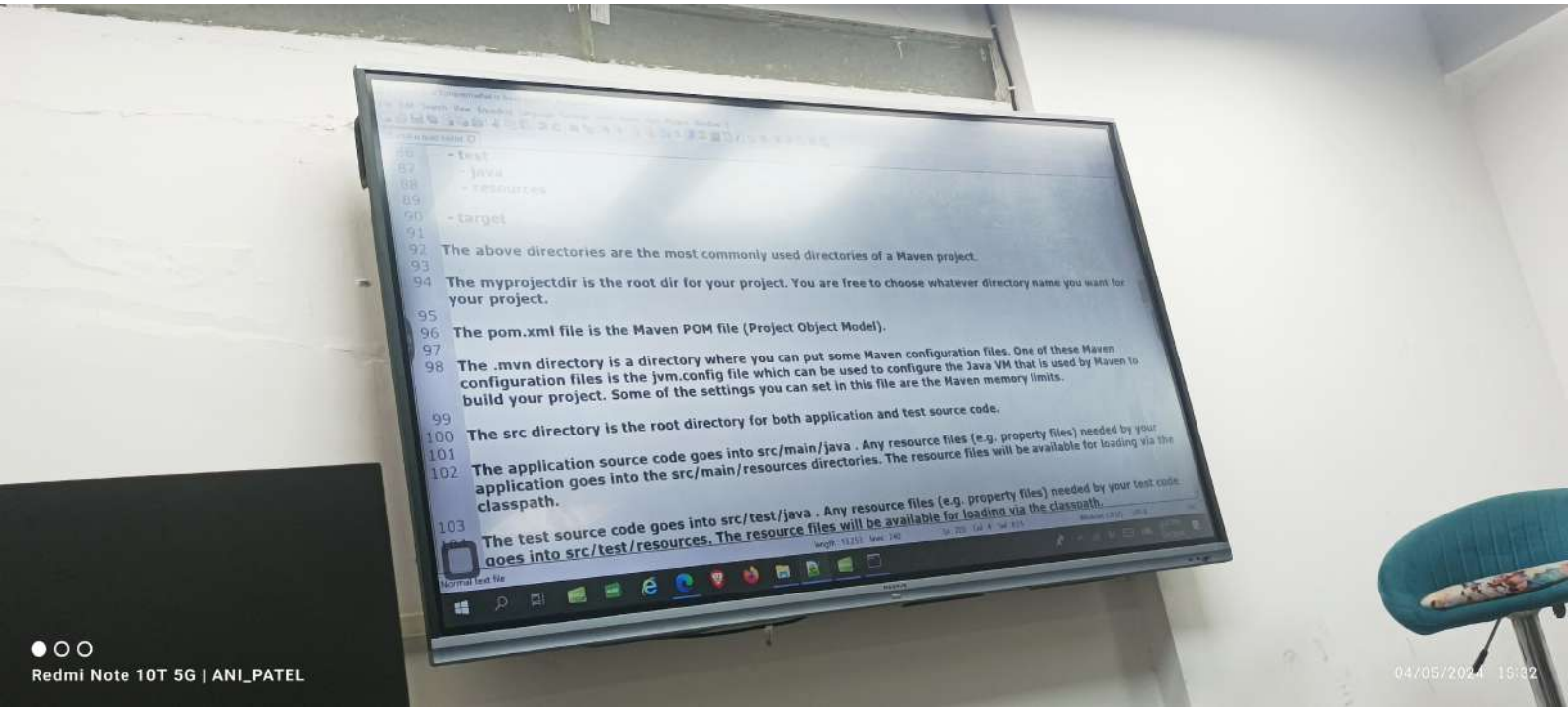


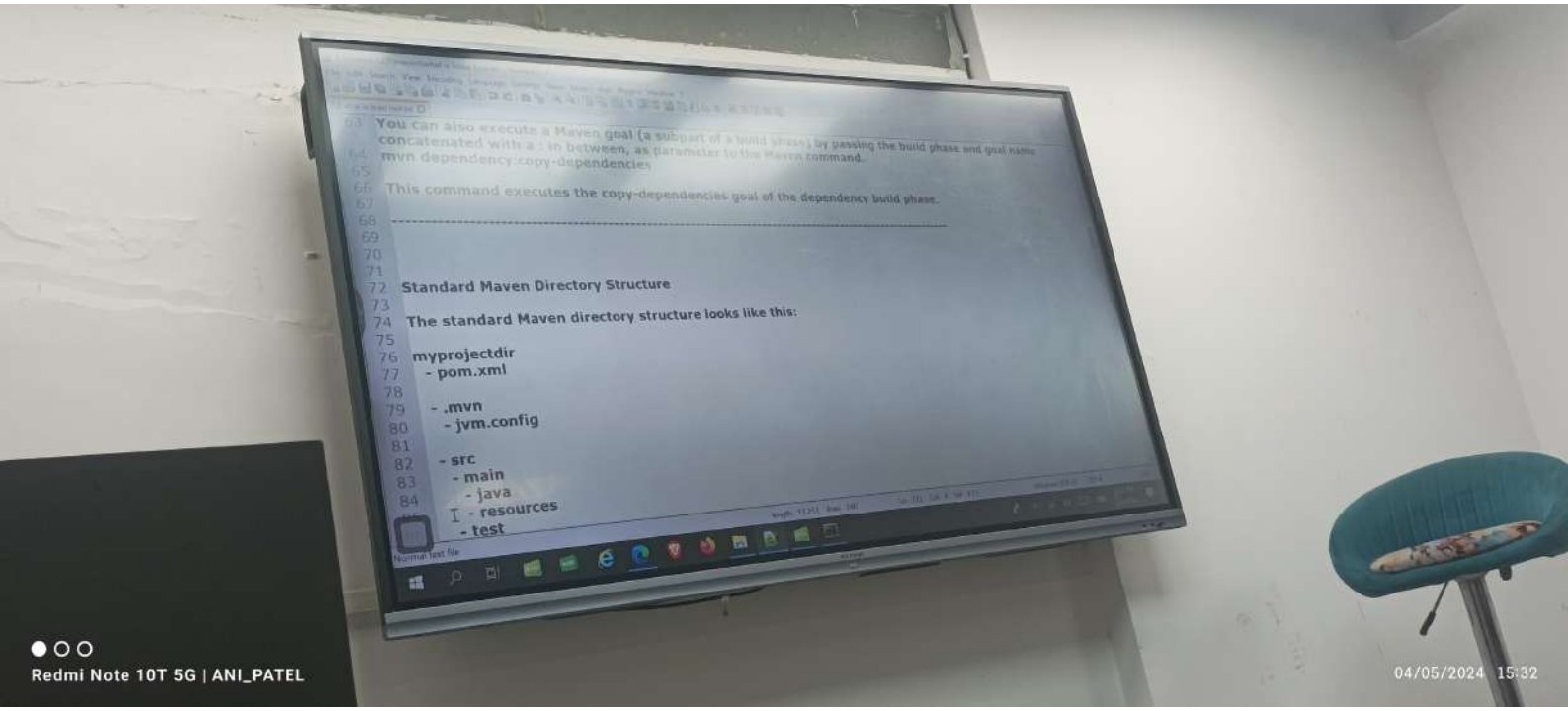


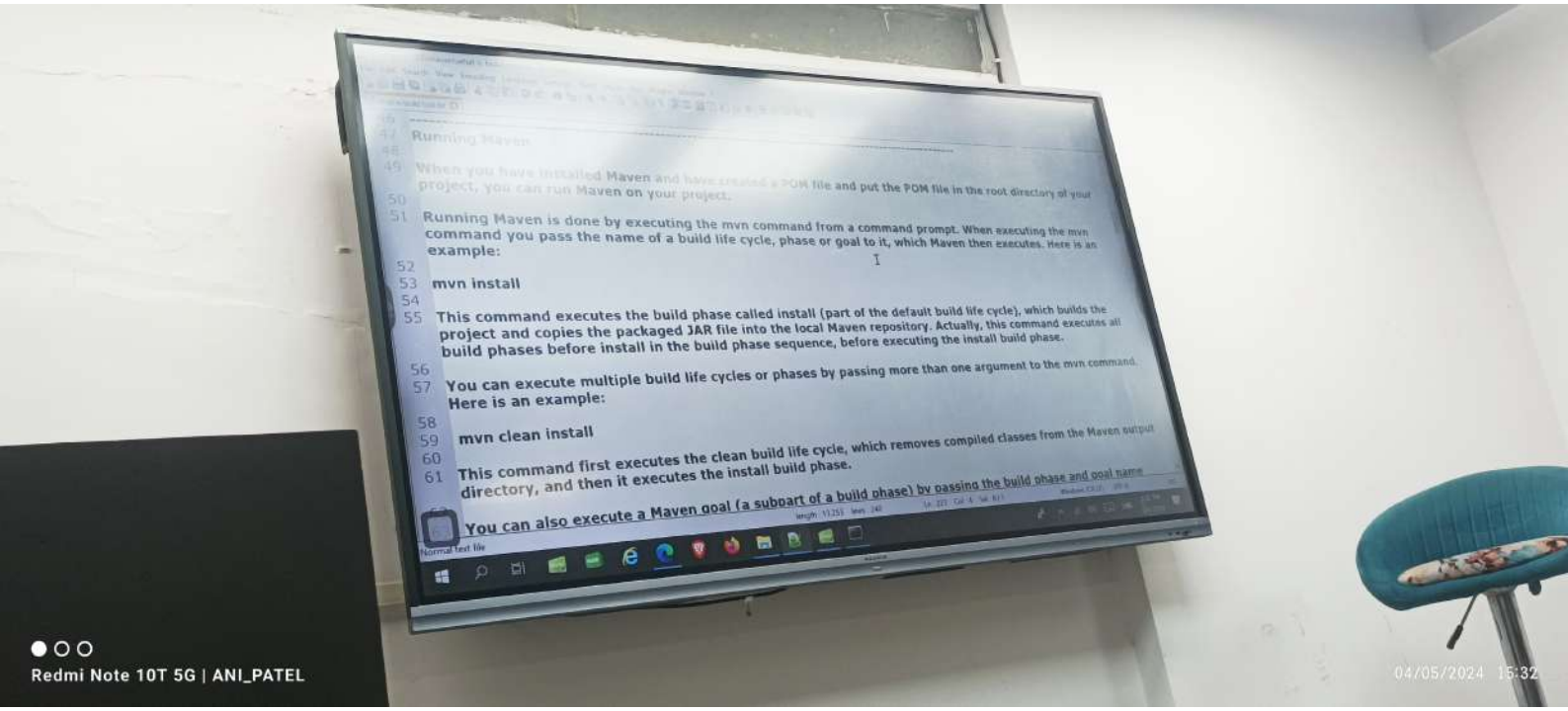

The test source code goes into src/test/. Any resource files (e.g. property files) needed by your test code goes into src/test/resources. The resource files will be available for loading via the classpath.
The target directory contains all the final products that are the result of Maven building your project. The target directory also contain any temporary and intermediate files needed by Maven when building your application.
Maven Commands->
Maven contains a wide set of commands which you can execute. Maven commands are a mix of build life cycles, build phases and build goals.
Common Maven Commands->
mvn -version--> Prints out the version of Maven you are running.
mvn clean--> Clears the target directory into which Maven normally builds your project.
mvn package-> Builds the project and packages the resulting JAR file into the target directory.
mvn package -Dmaven.test.skip=true->Builds the project and packages the resulting JAR file into the target directory - without running the unit tests during the build.
mvn clean package-> Clears the target directory and Builds the project and packages the resulting JAR file into the target directory.
Redmi Note 10T 5G | ANI_PATEL
04/05/2024 15:33

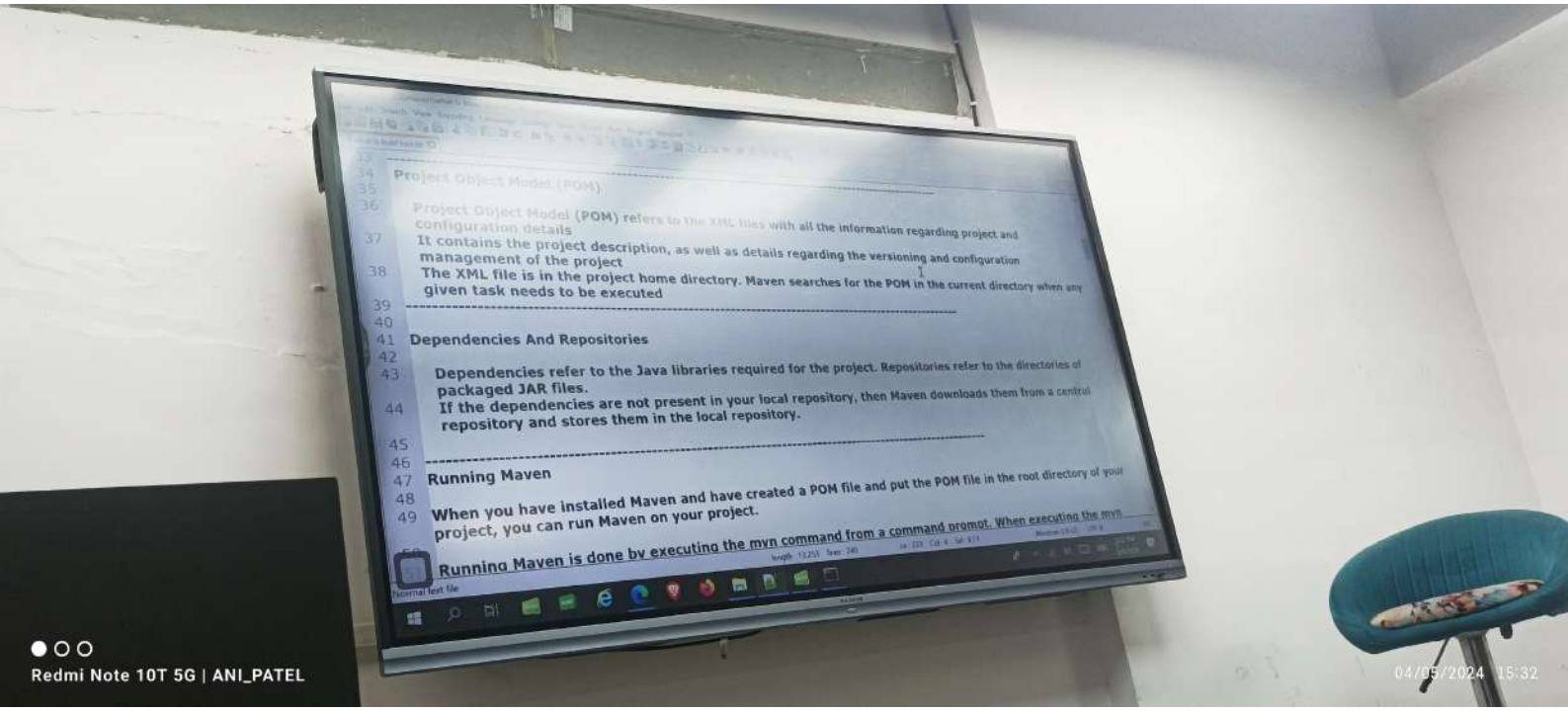

The test source code goes into src/test/java. Any resource files (e.g. property files) needed by your test code goes into src/test/resources. The resource files will be available for loading via the classpath.
The target directory contains all the final products that are the result of Maven building your project. The target directory also contain any temporary and intermediate files needed by Maven when building your application.
Maven Commands->
Maven contains a wide set of commands which you can execute. Maven commands are a mix of build life cycles, build phases and build goals.
Common Maven Commands->
mvn -version--> Prints out the version of Maven you are running.
mvn clean--> Clears the target directory into which Maven normally builds your project.
mvn package-> Builds the project and packages the resulting JAR file into the target directory.
mvn package -Dmaven.test.skip=true-> Builds the project and packages the resulting JAR file into the target directory - without running the unit tests during the build.
mvn clean package-> Clears the target directory and Builds the project and packages the resulting JAR file into the target directory.
Redmi Note 10T 5G | ANI_PATEL
04/05/2024 15:33


The test source code goes into src/test/java. Any resource files (e.g. property files) needed by your test code goes into src/test/resources. The resource files will be available for loading via the classpath.
The target directory contains all the final products that are the result of Maven building your project. The target directory also contain any temporary and intermediate files needed by Maven when building your application.
Maven Commands->
Maven contains a wide set of commands which you can execute. Maven commands are a mix of build life cycles, build phases and build goals.
Common Maven Commands->
mvn -version--> Prints out the version of Maven you are running.
mvn clean--> Clears the target directory into which Maven normally builds your project.
mvn package-> Builds the project and packages the resulting JAR file into the target directory.
mvn package -Dmaven.test.skip=true->Builds the project and packages the resulting JAR file into the target directory - without running the unit tests during the build.
mvn clean package-> Clears the target directory and Builds the project and packages the resulting JAR file into the target directory.
Redmi Note 10T 5G | ANI_PATEL
04/05/2024 15:33


- test
- java
- resources
- target
The above directories are the most commonly used directories of a Maven project.
The myprojectdir is the root dir for your project. You are free to choose whatever directory name you want for your project.
The pom.xml file is the Maven POM file (Project Object Model).
The .mvn directory is a directory where you can put some Maven configuration files. One of these Maven configuration files is the jvm.config file which can be used to configure the Java VM that is used by Maven to build your project. Some of the settings you can set in this file are the Maven memory limits.
The src directory is the root directory for both application and test source code.
The application source code goes into src/main/java . Any resource files (e.g. property files) needed by your application goes into the src/main/resources directories. The resource files will be available for loading via the classpath.
The test source code goes into src/test/java . Any resource files (e.g. property files) needed by your test code goes into src/test/resources. The resource files will be available for loading via the classpath.
Normal text file
Redmi Note 10T 5G | ANI_PATEL
04/05/2024 15:32


You can also execute a Maven goal (a subpart of a build phase) by passing the build phase and goal name concatenated with a : in between, as parameter to the Maven command.
mvn dependency:copy-dependencies
This command executes the copy-dependencies goal of the dependency build phase.
Standard Maven Directory Structure
The standard Maven directory structure looks like this:
myprojectdir
- pom.xml
- .mvn
- jvm.config
- src
- main
- java
- resources
- test
Redmi Note 10T 5G | ANI_PATEL
04/05/2024 15:32


Running Maven
When you have installed Maven and have created a POM file and put the POM file in the root directory of your project, you can run Maven on your project.
Running Maven is done by executing the mvn command from a command prompt. When executing the mvn command you pass the name of a build life cycle, phase or goal to it, which Maven then executes. Here is an example:
mvn install
This command executes the build phase called install (part of the default build life cycle), which builds the project and copies the packaged JAR file into the local Maven repository. Actually, this command executes all build phases before install in the build phase sequence, before executing the install build phase.
You can execute multiple build life cycles or phases by passing more than one argument to the mvn command. Here is an example:
mvn clean install
This command first executes the clean build life cycle, which removes compiled classes from the Maven output directory, and then it executes the install build phase.
You can also execute a Maven goal (a subpart of a build phase) by passing the build phase and goal name
Normal text file
Redmi Note 10T 5G | ANI_PATEL
04/05/2024 15:32


Project Object Model (POM)
Project Object Model (POM) refers to the XML files with all the information regarding project and configuration details
It contains the project description, as well as details regarding the versioning and configuration management of the project
The XML file is in the project home directory. Maven searches for the POM in the current directory when any given task needs to be executed
Dependencies And Repositories
Dependencies refer to the Java libraries required for the project. Repositories refer to the directories of packaged JAR files.
If the dependencies are not present in your local repository, then Maven downloads them from a central repository and stores them in the local repository.
Running Maven
When you have installed Maven and have created a POM file and put the POM file in the root directory of your project, you can run Maven on your project.
Running Maven is done by executing the mvn command from a command prompt. When executing the mvn
Redmi Note 10T 5G | ANI_PATEL
04/09/2024 15:32