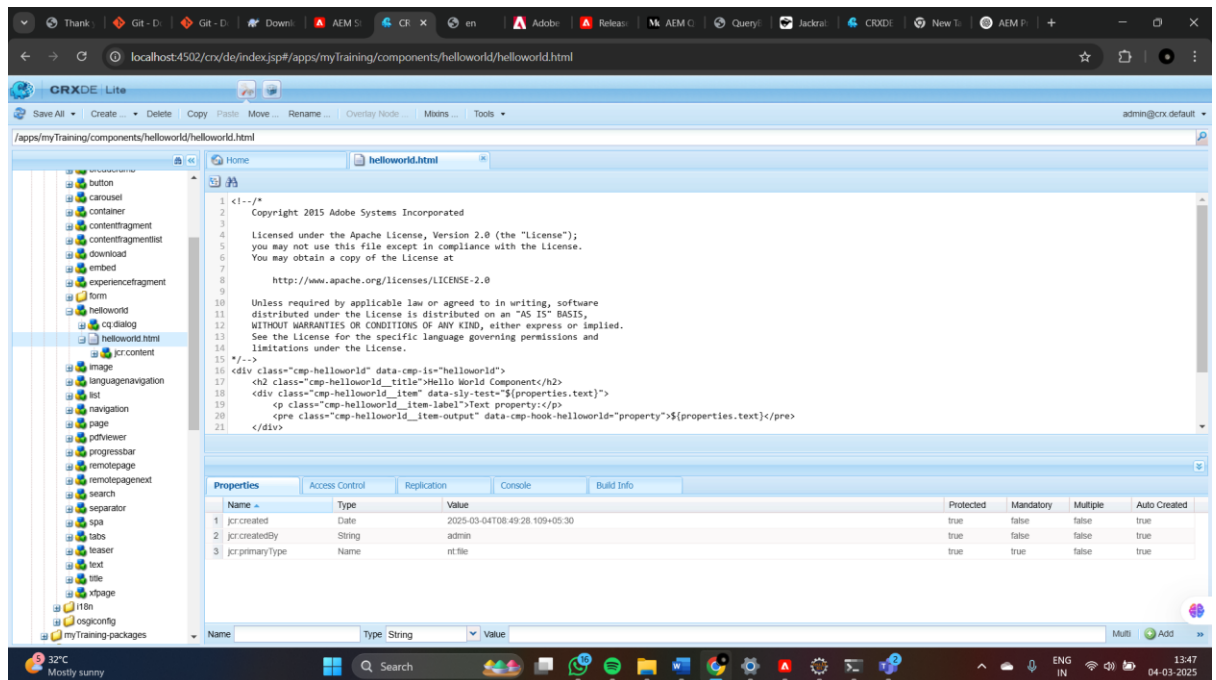


### 3. Explain the role of ui.apps in AEM projects.

It is responsible for the frontend of an AEM project



### 4. How are components structured in the ui.apps folder?

`/apps/myTraining/components/helloworld/helloworld.html`

`/apps/myTraining/components/helloworld/helloworld.html`

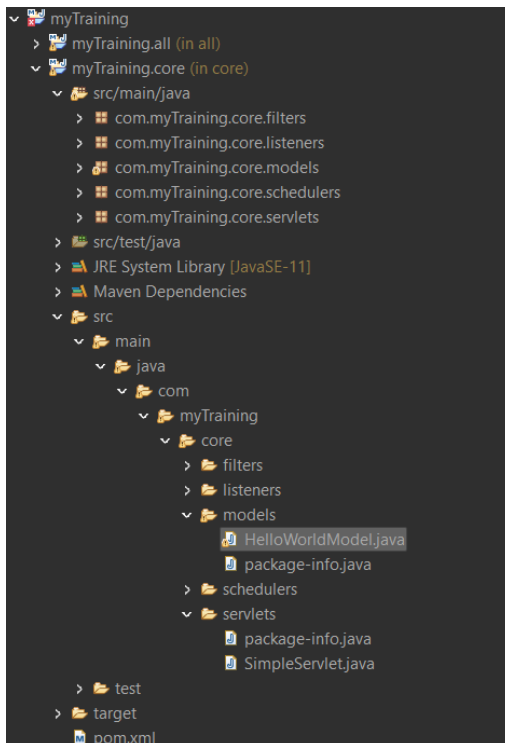
## Hello World Component:

1. Where is the Hello World component located in both core and ui.apps?

### Core (Backend - Java Logic):

Located in:

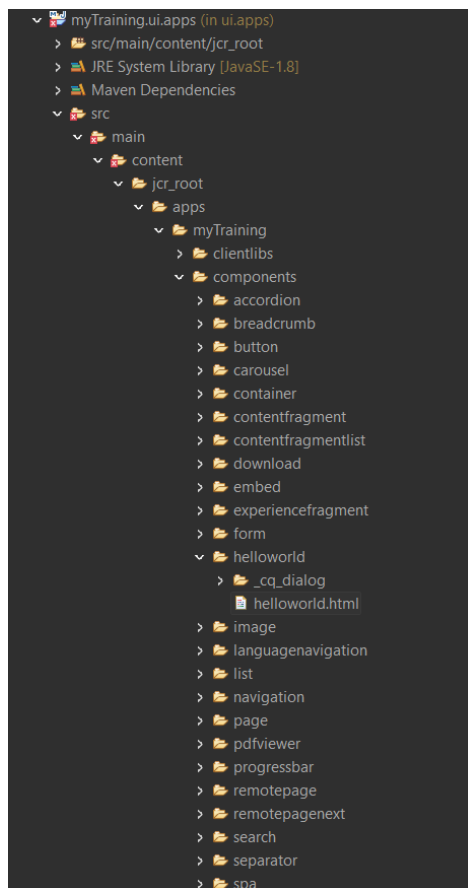
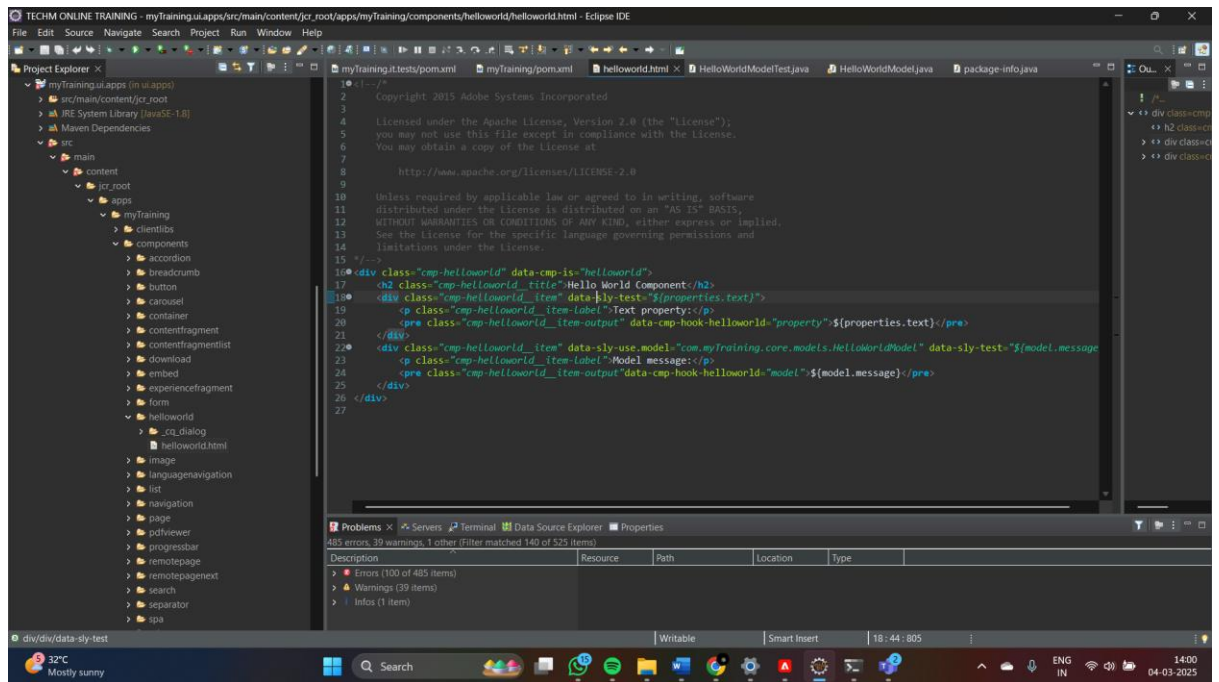
/core/src/main/java/com/project/core/models/HelloWorldModel.java



### UI.Apps (Frontend - HTL, JS, CSS, Dialogs):

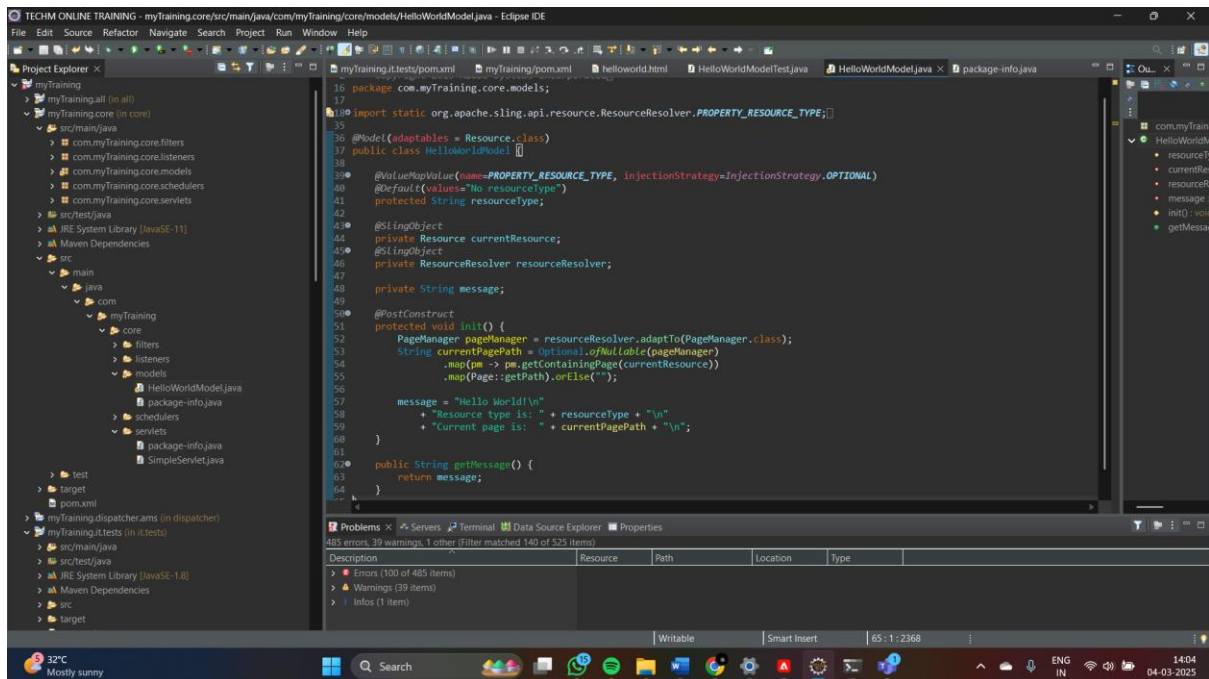
Located in:

/ui.apps/src/main/content/jcr\_root/apps/project/components/helloworld/



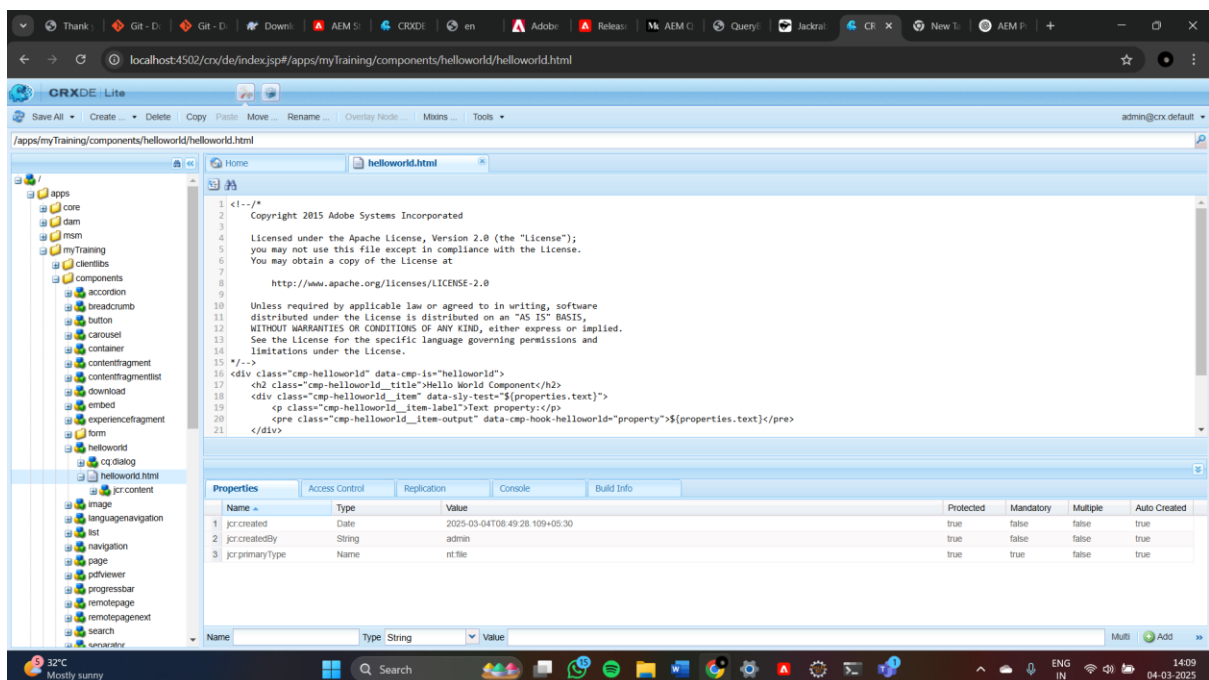
## 2. Explain the Java class (in core) for the Hello World component.

The HelloWorldModel.java is a Sling Model that fetches data for the component.



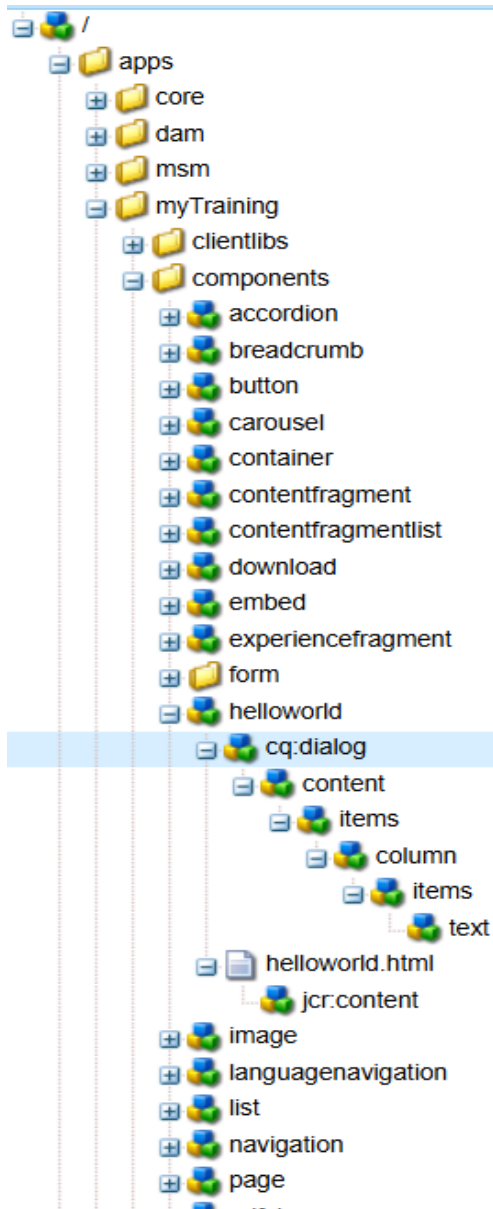
## 3. How does the HTML script work in ui.apps for Hello World?

The HTML file (helloworld.html) displays the data from HelloWorldModel.



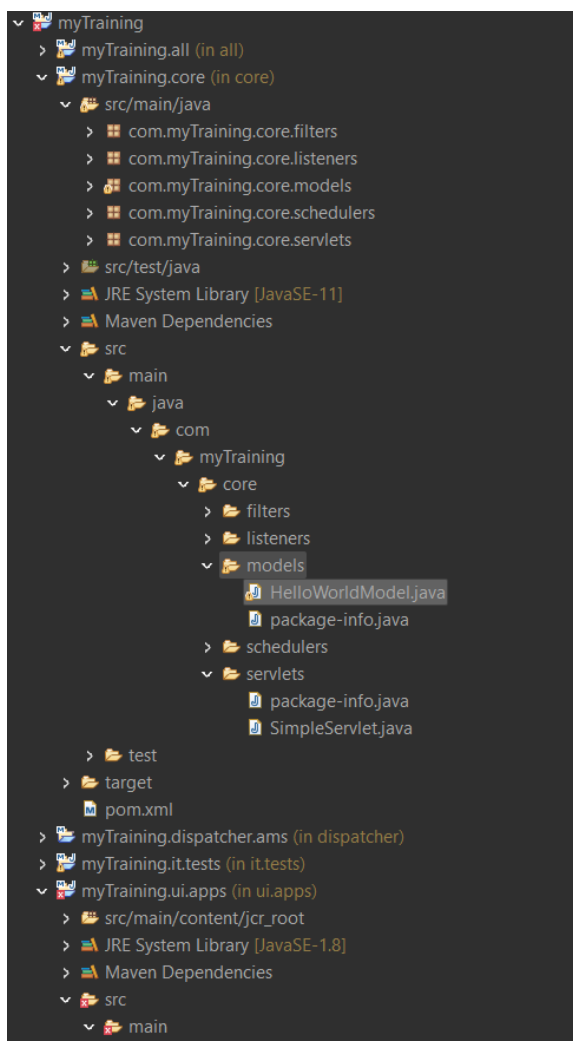
#### 4. How are properties and dialogs defined for this component?

The dialog defines editable fields in AEM's authoring UI.



5. What are the different types of AEM modules (core, ui.apps, ui.content, etc.)?

- core → Java backend (Sling models, services, OSGi components).
- ui.apps → Frontend (components, templates, clientlibs).
- ui.content → Content structure (/content/, pages, templates).
- all → Builds the full project package.



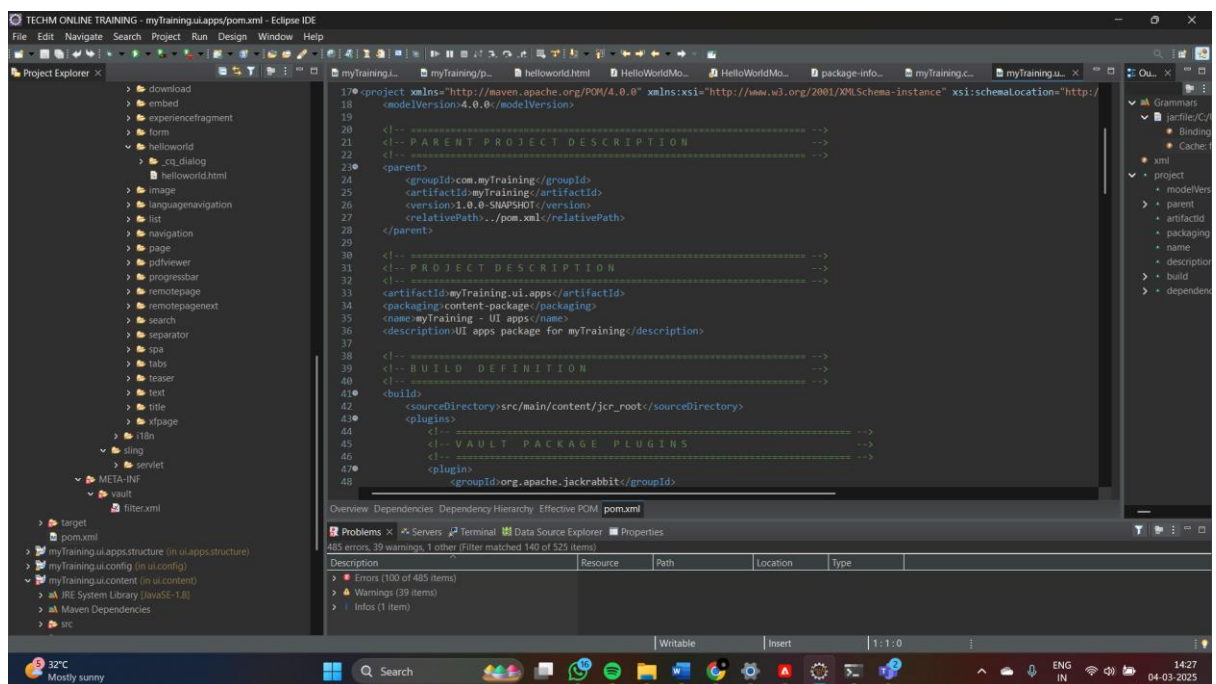
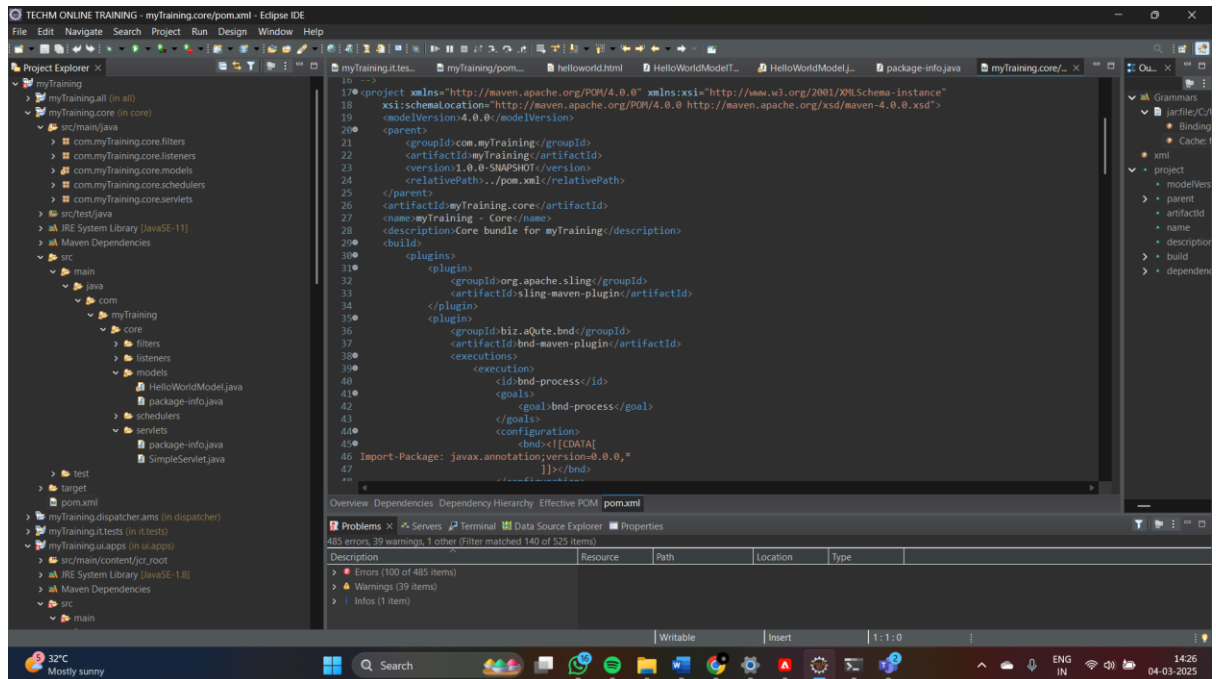
6. How does Maven build these modules?

Maven compiles Java files, processes HTL, and packages the project into a deployable .zip file.



## 7. Explain the build lifecycle of Maven in the context of AEM.

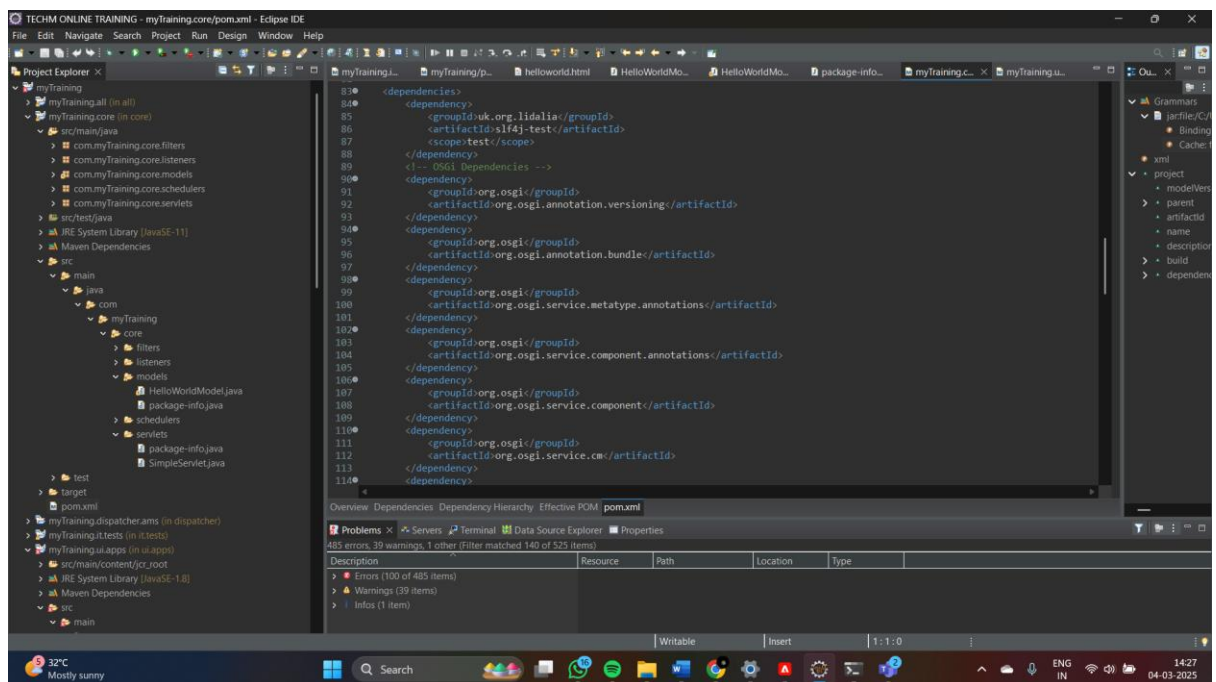
- Compile → Compiles Java code.
- Package → Creates the .jar for core and .zip for ui.apps.
- Install → Installs the package in the local repository.
- Deploy → Deploys the package to AEM.





## 8. How are dependencies managed in pom.xml?

Maven uses the dependencies section to include required libraries.



## 9. Why is Maven used instead of other build tools?

- Standardization → Used widely in Java projects.
- Automation → Handles builds, dependencies, and deployments automatically.

## 10. What advantages does Maven offer for AEM development?

- Manages Dependencies → No manual library setup.
- Handles Packaging → Creates deployable AEM packages.
- Supports Profiles → Different configurations for local and production.

## 11. How does Maven help in managing dependencies and plugins in AEM projects?

Maven helps manage dependencies in AEM by automatically downloading required libraries from repositories like Maven Central and Adobe's public repo. It ensures consistency using the pom.xml file, where dependencies are defined. Plugins automate tasks like compiling Java code, creating OSGi bundles, and deploying content packages.

12. What does mvn clean install do in an AEM project?

- clean → Deletes previous builds.
- install → Builds and installs the package locally

13. How to deploy packages directly to AEM using Maven commands?

```
mvn -PautoInstallPackage clean install
```

14. Explain the purpose of different Maven profiles in AEM (autoInstallPackage, autoInstallBundle)

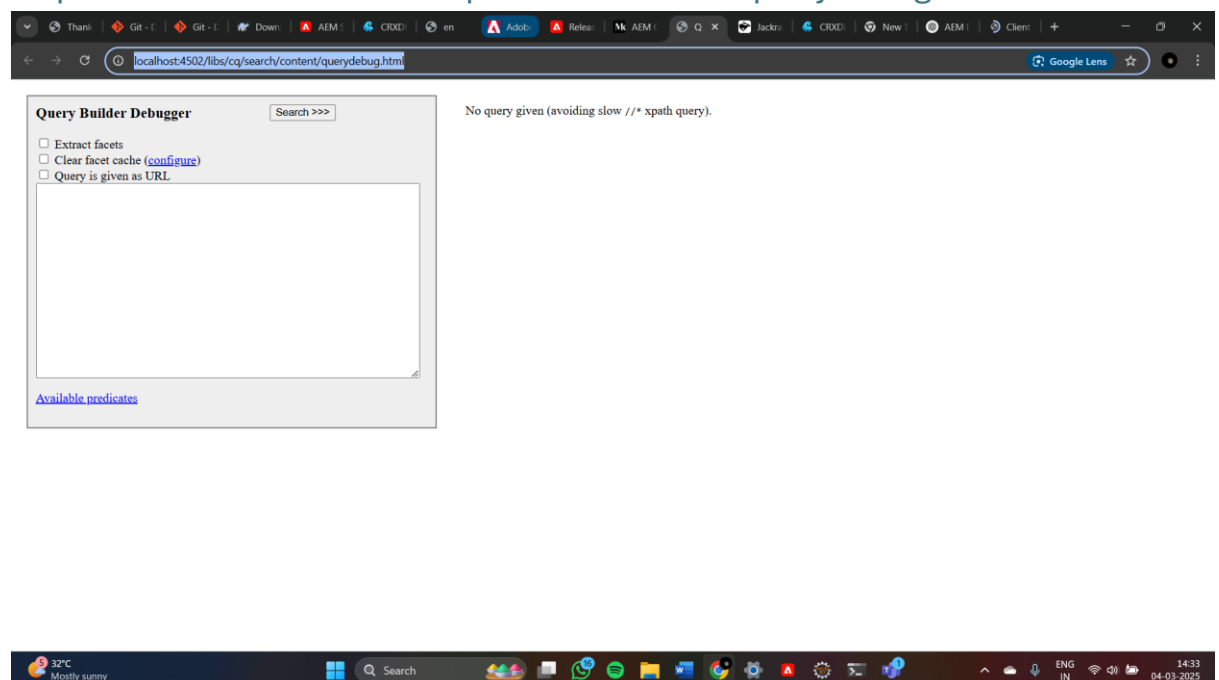
dumplibs helps debug clientlibs (CSS & JS) by showing all libraries loaded on a page.

15. What is the purpose of dumplibs in AEM?

dumplibs helps debug clientlibs (CSS & JS) by showing all libraries loaded on a page.

16. How can you view client libraries using dumplibs?

<http://localhost:4502/libs/cq/search/content/querydebug.html>



17. Explain how client libraries are structured in AEM.

