Maven Lifecycle Maven follows a structured lifecycle for building projects. The primary phases in the Maven lifecycle are:

1. **Validate** - Ensures that the project is correct and that all necessary information is available.
2. **Compile** - Compiles the source code of the project.
3. **Test** - Runs unit tests using a framework like JUnit.
4. **Package** - Packages compiled code into a JAR or WAR file.
5. **Verify** - Runs any checks to validate the package.
6. **Install** - Installs the package into the local repository.
7. **Deploy** - Copies the package to a remote repository for sharing.

What is pom.xml file and why do we use it? pom.xml (Project Object Model) is the fundamental configuration file in a Maven project. It contains:

- Project details (name, version, description, etc.)
- Dependencies for libraries and frameworks
- Build plugins and goals
- Repository locations (local and remote)
- Profiles for different environments

It serves as the blueprint for building and managing the project.

How do dependencies work in Maven? Dependencies are external libraries required by a project. They are defined in pom.xml under the <dependencies> section. When you build the project, Maven automatically downloads required dependencies from repositories and manages versioning and transitive dependencies (dependencies of dependencies).

Example:

```
<dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-core</artifactId>
   <version>5.3.9</version>
</dependency>
```

How to check the Maven repository?

- The **local repository** is located at ~/.m2/repository on your system.
- The **central repository** is https://repo.maven.apache.org/maven2/.
- You can also use custom repositories like Nexus or Artifactory.
- To search for dependencies, visit: Maven Repository.

How are all modules built using Maven? Maven uses a **multi-module** build approach:

1. A parent pom.xml defines common configurations.
2. Modules are sub-projects within the parent project.
3. Running mvn install in the parent directory builds all modules.

Example structure:

```
parent-project/
 ├── pom.xml (Parent POM)
 ├── module1/
|    ├── pom.xml
```

```
 ├── module2/
|    ├── pom.xml
```

Can we build a specific module in Maven? Yes, use:

mvn install -pl module-name -am

- -pl specifies the module
- -am ensures dependencies are also built

Role of ui.apps, ui.content, and ui.frontend folders in AEM

- **ui.apps**: Stores code, components, templates, and configurations.
- **ui.content**: Holds site content and structured content data.
- **ui.frontend**: Contains frontend assets (CSS, JavaScript) for styling and interactivity.

Why are we using run modes in AEM? Run modes allow AEM to adapt configurations for different environments like development, testing, and production.

Example run modes:

- **Author** (for content authors)
- **Publish** (for serving content to end-users)
- **Development**, **Stage**, **Production** (environment-specific settings)

Run mode folders are placed under:

/apps/myproject/config.author/

/apps/myproject/config.publish/

What is the Publish environment in AEM?

- The **Publish** environment is the live instance that serves content to users.
- It contains **published content** from the author instance.
- The dispatcher caches and optimizes content delivery.

Why are we using Dispatcher in AEM? The **Dispatcher** is a caching and load-balancing tool that:

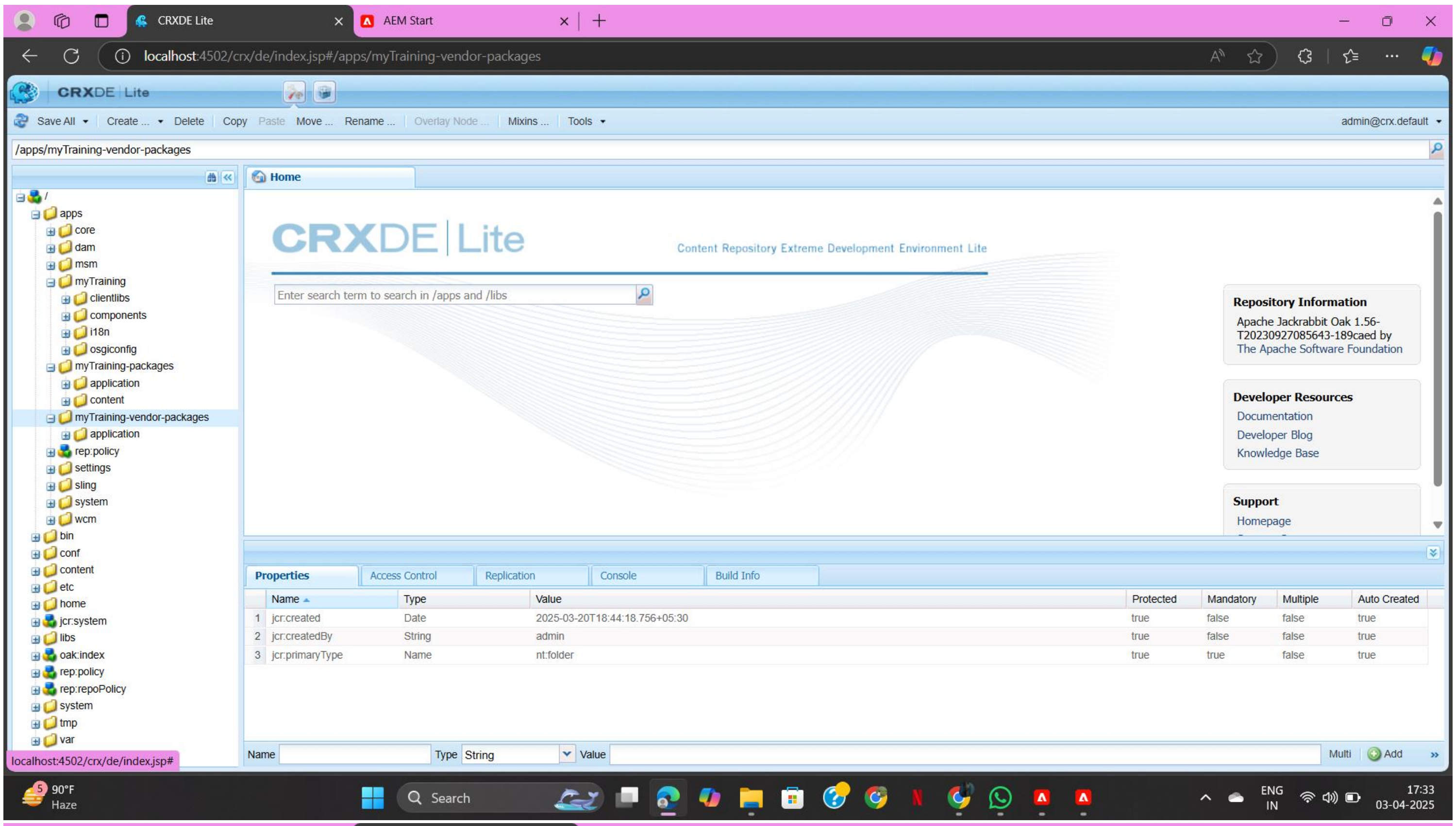
1. Improves performance by caching content.
2. Provides security by filtering requests.
3. Balances load between multiple AEM instances.

Dispatcher configurations are stored in Apache server configuration files.

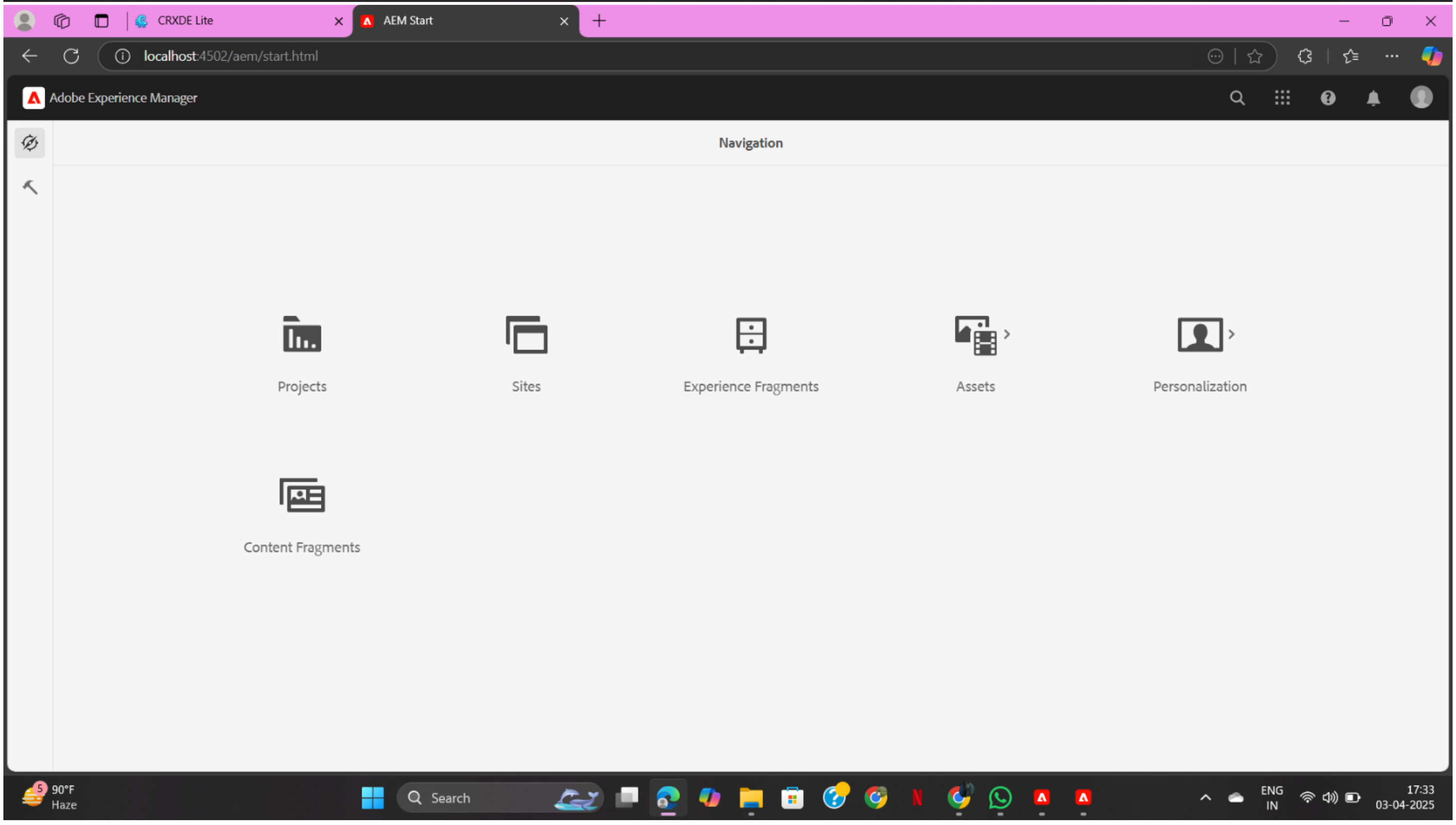
From where can we access crx/de? You can access CRXDE Lite (Content Repository) by navigating to:

http://localhost:4502/crx/de/index.jsp

Here, you can view and manage repository nodes, properties, and configurations.



	Name	Type	Value	Protected	Mandatory	Multiple	Auto Created
1	jcr:created	Date	2025-03-20T18:44:18.756+05:30	true	false	false	true
2	jcr:createdBy	String	admin	true	false	false	true
3	jcr:primaryType	Name	nt:folder	true	true	false	true


CRXDE Lite
AEM Start
localhost:4502/aem/start.html
Adobe Experience Manager
Navigation
Projects
Sites
Experience Fragments
Assets
Personalization
Content Fragments
90°F
Haze
Search
ENG
IN
17:33
03-04-2025