AEM ASSIGNMENT

-Ravi Teja C

* **Maven Life Cycle**
  + Phases: **Clean**, **Default**, **Site**.
  + Default phase handles compile, test, package, install.
  + Site phase generates project documentation.
  + Goals are executed sequentially within these phases.
* **What is pom.xml and Why Use It?**
  + **pom.xml**: Configuration file for Maven projects.
  + Defines dependencies, plugins, build settings, and project metadata.
  + Standardizes build processes and manages external libraries automatically.
* **How Dependencies Work?**
  + Dependencies are libraries or modules required by the project.
  + Declared in pom.xml file.
  + Maven automatically downloads and manages them from repositories.
  + Handles transitive dependencies (dependencies of dependencies).
* **Check the Maven Repository**
  + Public repositories: **Maven Central** ([search.maven.org](https://search.maven.org)).
  + Private repositories: Nexus, Artifactory (enterprise repositories).
  + Search by group ID, artifact ID, and version to find dependencies.
* **How Do All Modules Build Using Maven?**
  + Parent pom.xml manages multi-module projects.
  + Child modules inherit common settings from the parent.
  + Maven builds modules in sequence or parallel based on dependencies.
* **Can We Build a Specific Module?**
  + Yes, use -pl <module-name> to specify the module to build.
  + Example: mvn clean install -pl <module-name> -am to build a specific module and its dependencies.
* **Role of ui.apps, ui.content, and ui.frontend Folders**
  + **ui.apps**: Backend code (OSGi components, servlets).
  + **ui.content**: Content (pages, components) to be deployed into the JCR.
  + **ui.frontend**: Frontend assets (HTML, JS, CSS, images) served to users.
* **Why Use Run Modes?**
  + Define configurations for different environments (e.g., dev, prod).
  + Ensures environment-specific settings for AEM.
  + Helps maintain separate configurations for different stages of the project.
* **What is Publish Environment?**
  + The **Publish Environment** serves live, publicly accessible content.
  + Content from **Author Environment** is published here.
  + Typically optimized with a **Dispatcher** for caching and load balancing.
* **Why Use Dispatcher?**
  + A reverse proxy and caching tool for AEM.
  + Improves performance by caching rendered pages.
  + Reduces load on AEM server and optimizes response times.
  + Helps with load balancing across AEM instances.
* **Accessing CRX/DE**
  + **CRX/DE** is a web-based interface to interact with the AEM JCR.
  + Access via http://<AEM-host>:<port>/crx/de.
  + Allows developers to manage content, configurations, and packages in AEM.