**What is Maven**

Maven is a powerful build automation tool and project management tool that provides developers a complete build lifecycle framework.A build tool is a tool that automates everything related to building the software project.

**Maven Build Lifecycle Phases**

**validate :**validate the project is correct and all necessary information is available

**compile :**compile the entire source code of the project

**test** :test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed ,it runs all unit test cases and gives us a result

**package** : takes the compiled code and package it in its distributable format or deployment format, such as a JAR. we can use this phase to generate a deployment format files

**verify:** run any checks on results of integration tests to ensure quality criteria are met,

**install:** install the package into the local repository, for use as a dependency in other projects locally (creating a package into a local repositary(it creates jar files for 3 parties dependecnies so we can use depenedcies in any projects)

**deploy:** done in the build environment, copies the final package to the remote repository(like servers) for sharing with other developers and projects.

**Maven Build lifecycle commands**

we have to write all command in that project where our pom.xml present (C:\Users\kinshuk\_garg\CreveProject\cravejava-app>)

validate :**mvn validate**

compile **:mvn compile**

test **: mvn test**

package **: mvn package**

verify **: mvn verify**

install **: mvn install**

deploy **: mvn deploy**

**Maven Project Setup and Execution Observations**

**1. JDK Installation**

**Observation:** Maven requires the Java Development Kit (JDK) to be installed.

**Reason:** Maven is a Java-based tool and relies on Java runtime to execute its tasks.

**2. Downloading Maven from Apache Maven Website**

**Observation:** Maven should be downloaded from the official Apache Maven website.

**Reason:** Official sources ensure the authenticity and reliability of the software.

**3. Setting M2\_HOME Environment Variable**

**Observation:** It's necessary to set the M2\_HOME environment variable to point to the Maven installation directory.

**Reason:** This allows system-wide access to Maven commands.

**4. Verifying Maven Installation**

**Observation:** To confirm Maven is installed correctly, run mvn --version in the command prompt.

**Reason:** Checking the version ensures that Maven is accessible and functional.

**5. Creating a Maven Project using Command Prompt**

**Observation:** Maven projects can be generated using the mvn archetype:generate command.

**Steps:**

mvn archetype:generate -DgroupId=in.cravejava.app -DartifactId=cravejava-app -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.0 -DinteractiveMode=false

**Explanation:** This command initializes a new Maven project with specified parameters.

**6. Building and Packaging the Project**

**Observation:** Maven can build and package the project using the mvn package command.

**Steps:**

cd cravejava-app

mvn package

**Explanation:** This command compiles the project, downloads dependencies, and generates a JAR file in the target directory.

**7. Running the Maven Project**

**Observation:** Executing the Maven-built application is done using the java command.

**Steps:**

cd target

java -cp cravejava-app-1.0-SNAPSHOT.jar in.cravejava.app.App

**Explanation:** This command runs the main application class from the generated JAR file.