**Maven Assignment**

**1.Maven Lifecycle:**Below is a representation of the default Maven lifecycle and its 8 steps: Validate, Compile, Test, Package, Integration test, Verify, Install and Deploy.

The default Maven lifecycle consists of 8 major steps or phases for compiling, testing, building and installing a given Java project as specified below:

1. **Validate:** This step validates if the project structure is correct. For example – It checks if all the dependencies have been downloaded and are available in the local repository.
2. **Compile:** It compiles the source code, converts the .java files to .class and stores the classes in target/classes folder.
3. **Test:** It runs unit tests for the project.
4. **Package:** This step packages the compiled code in distributable format like JAR or WAR.
5. **Integration test:** It runs the integration tests for the project.
6. **Verify:** This step runs checks to verify that the project is valid and meets the quality standards.
7. **Install:** This step installs the packaged code to the local Maven repository.
8. **Deploy:** It copies the packaged code to the remote repository for sharing it with other developers.

**2. Maven Commands:**

mvn --version : This command **prints the maven version and then continue with the build**.

mvn -- compile: **Compiles source code of the project**. mvn test-compile: Compiles the test source code. mvn test: Runs tests for the project.

mvn --test : **The Maven surefire plugin provides a “test” parameter that we can use to specify test classes or methods we want to execute**.

mvn -- install: To install Maven on Windows, we **head over to the Apache Maven site to download the latest version and select the Maven zip file**, for example, apache-maven-3.8. 4-bin. zip.

mvn --clean : The Maven Clean Plugin, as the name implies, **attempts to clean the files and directories generated by Maven during its build**. While there are plugins that generate additional files, the Clean Plugin assumes that these files are generated inside the target directory.

3. WHAT I SWEB SERVICES ?

A web service (WS) is either: a service offered by an electronic device to another electronic device, communicating with each other via the Internet, or. a server running on a computer device, listening for requests at a particular port over a network, serving web documents (HTML, JSON, XML, images).

4) what is rest controler

RestController is **a Spring annotation that is used to build REST API in a declarative way**. RestController annotation is applied to a class to mark it as a request handler, and Spring will do the building and provide the RESTful web service at runtime.

5. we services and its methods: get , post , put delete ? describe its

The primary or most-commonly-used HTTP verbs (or methods, as they are properly called) are POST, GET, PUT, PATCH, and DELETE. These correspond to **create, read, update, and delete (or CRUD) operations**, respectively. There are a number of other verbs, too, but are utilized less frequently.

6. Git life cycle

General workflow is as follows −

* You clone the Git repository as a working copy.
* You modify the working copy by adding/editing files.
* If necessary, you also update the working copy by taking other developer's changes.
* You review the changes before commit.
* You commit changes. If everything is fine, then you push the changes to the repository.
* After committing, if you realize something is wrong, then you correct the last commit and push the changes to the repository.
* 7. git add: takes a modified file in your working directory and places the modified version in a staging area.
* git commit takes everything from the staging area and makes a permanent snapshot of the current state of your repository that is associated with a unique identifier.
* Git pull : Pulling is the automated version of git fetch. It downloads a branch from a remote repository, then immediately merges it into the current branch. This is the Git equivalent of svn update.
* Git push: Pushing is the opposite of fetching (with a few caveats). It lets you move a local branch to another repository, which serves as a convenient way to publish contributions. This is like svn commit, but it sends a series of commits instead of a single changeset.

8) centrilized v/s distrubuted version conrol

The main difference between centralized and distributed version control is that, in centralized version control, the versions are saved in the remote repository, while in distributed version control, versions can be saved in the remote repository as well as in local repositories of the local machines.