

## Lesson 01 Demo 02

### Using Maven CLI to Create Java Web Project

**Objective:** To create a Java Web Project using Maven CLI and work with the Maven commands to configure and execute the project

**Tools required:** Visual Studio Code and Maven

**Prerequisites:** None

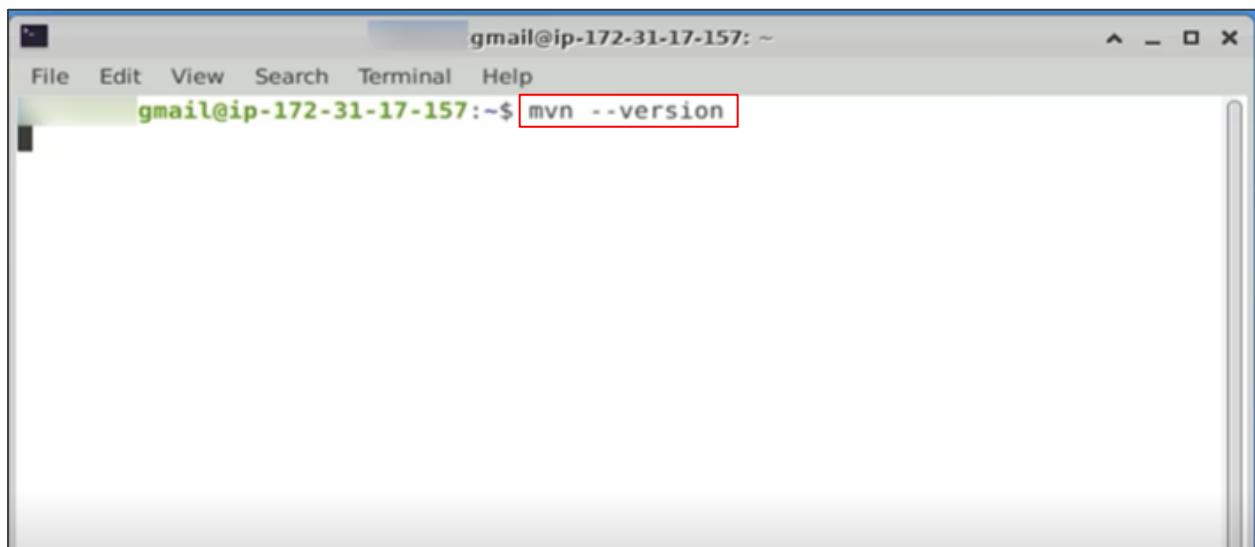
Steps to be followed:

1. Run the mvn package command
2. Open the CMS project
3. Add the dependency

#### Step 1: Run the mvn package command

1.1 Open the terminal and check if Maven is installed using the following command:

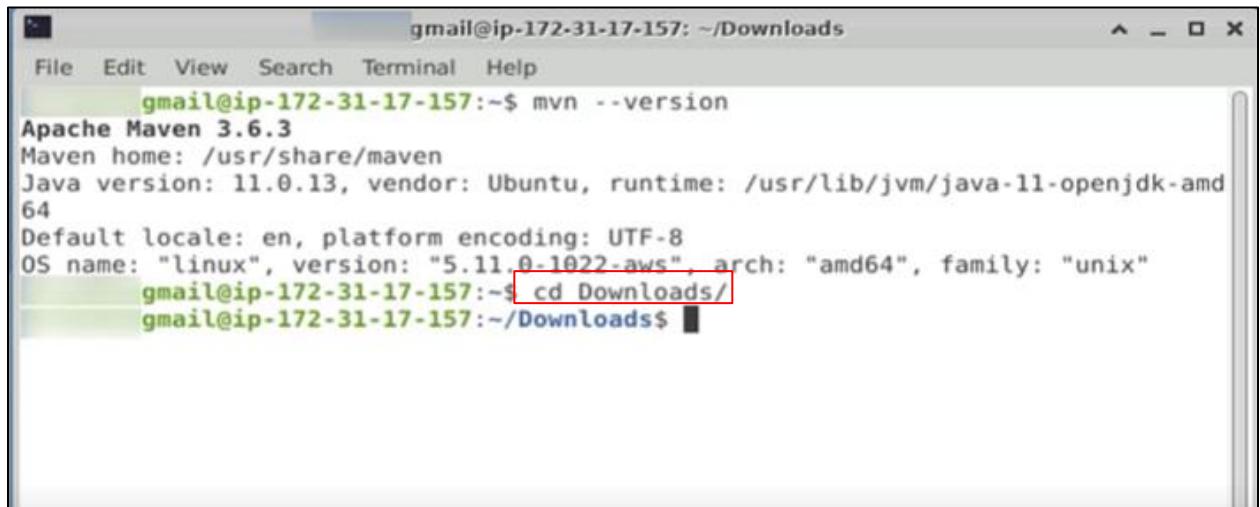
`mvn --version`



A screenshot of a terminal window titled "gmail@ip-172-31-17-157: ~". The window has a standard OS X-style title bar with icons for close, minimize, and maximize. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". Below the menu bar, the prompt "gmail@ip-172-31-17-157:~\$" is visible. A red rectangular box highlights the command "mvn --version" which is being typed into the terminal. The rest of the window is blank, showing a light gray background.

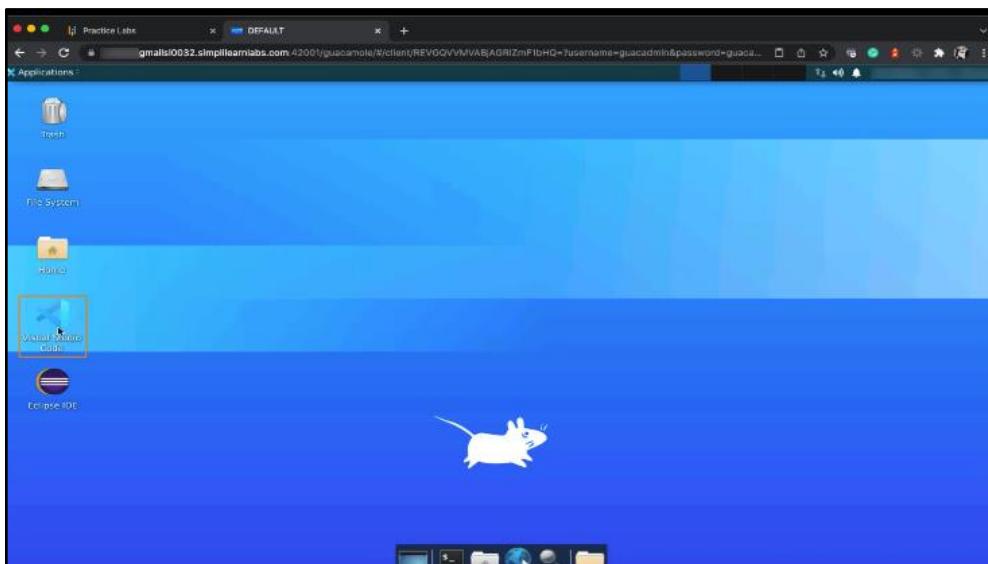
1.2 Use the cd command to navigate to the **Downloads** directory:

```
cd Downloads/
```



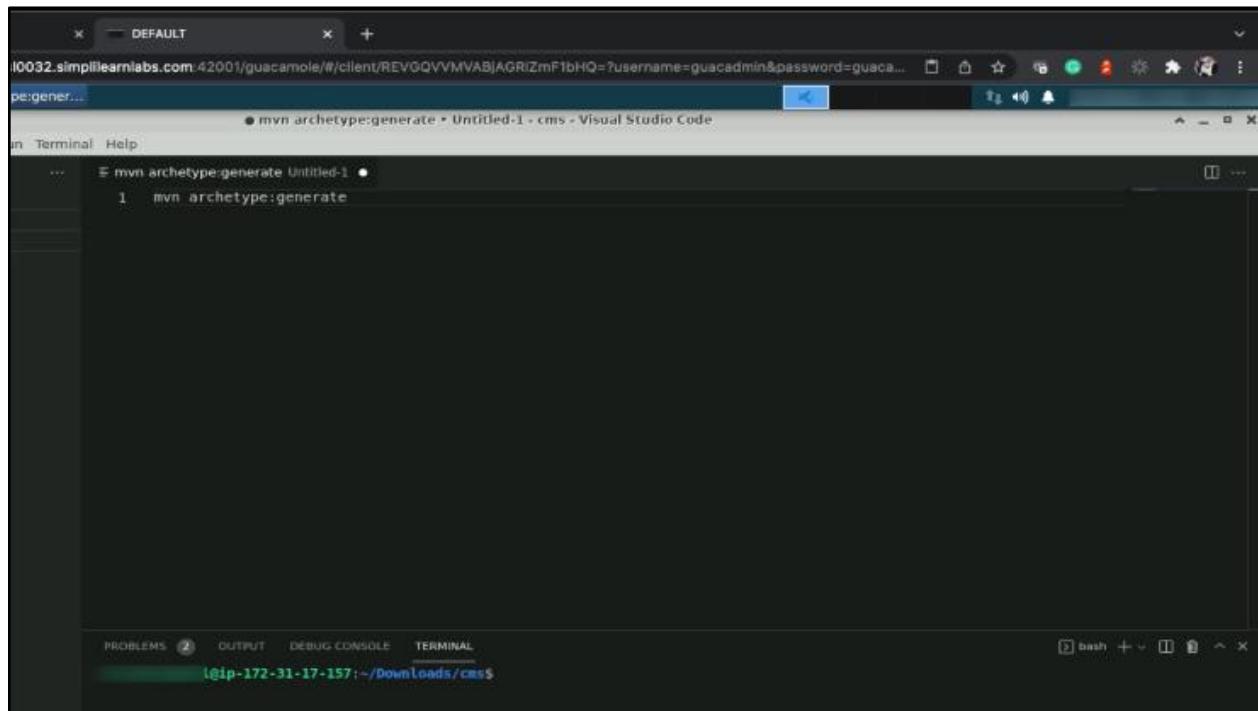
```
gmail@ip-172-31-17-157: ~/Downloads
File Edit View Search Terminal Help
gmail@ip-172-31-17-157:~$ mvn --version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 11.0.13, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "5.11.0-1022-aws", arch: "amd64", family: "unix"
gmail@ip-172-31-17-157:~$ cd Downloads/
gmail@ip-172-31-17-157:~/Downloads$
```

1.3 To execute the different commands, open the Visual Studio Code



1.4 Use the following command to create a project:

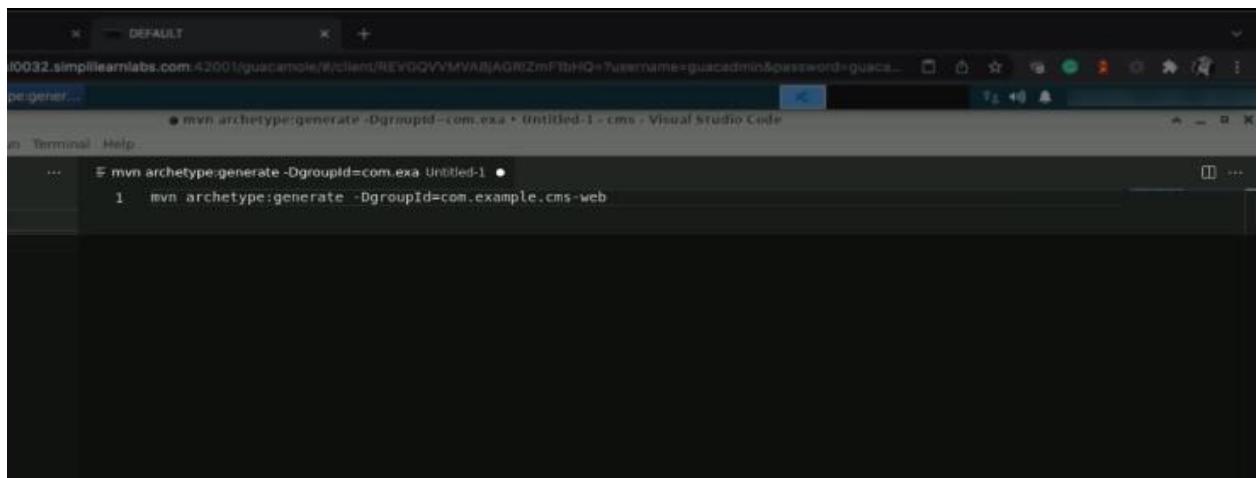
```
mvn archetype:generate
```



The screenshot shows a Visual Studio Code interface with a terminal window open. The title bar indicates the current tab is 'mvn archetype:generate Untitled-1 - cms - Visual Studio Code'. The terminal window displays the command 'mvn archetype:generate' followed by a line number '1'. The status bar at the bottom shows the path '@ip-172-31-17-157:~/Downloads/cms\$'.

1.5 The project must have the group ID, which can be the company's domain name in reverse order. Add group ID for the project:

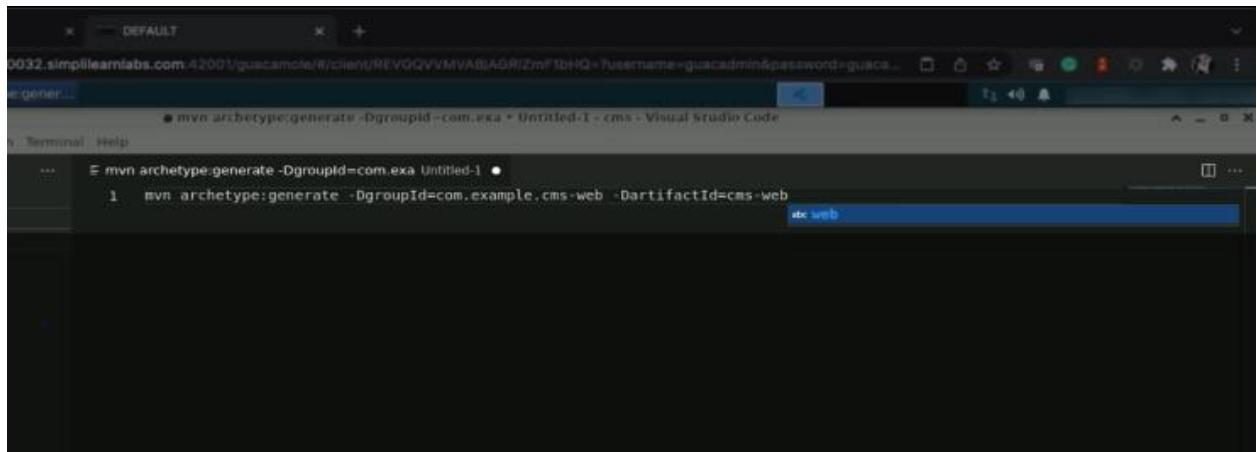
```
-DgroupId=com.example.cms-web
```



The screenshot shows a Visual Studio Code interface with a terminal window open. The title bar indicates the current tab is 'mvn archetype:generate -DgroupId=com.exa Untitled-1 - cms - Visual Studio Code'. The terminal window displays the command 'mvn archetype:generate -DgroupId=com.example.cms-web' followed by a line number '1'. The status bar at the bottom shows the path '@ip-172-31-17-157:~/Downloads/cms\$'.

1.6 Add an artifactId for the project, which is the application name for the project

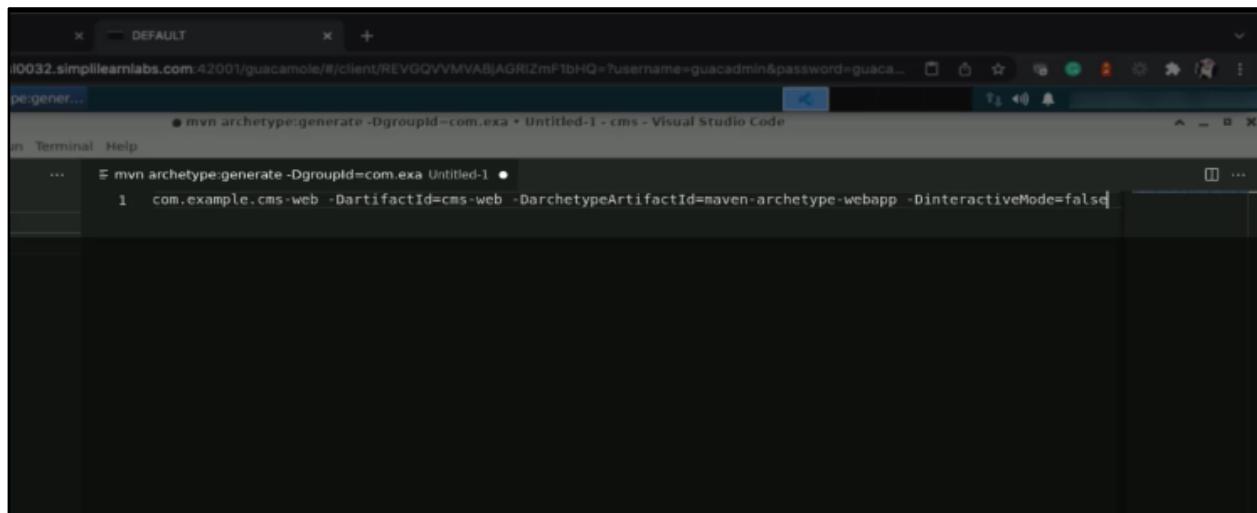
**-DartifactId= cms-web**



```
mvn archetype:generate -DgroupId=com.exa Untitled-1 -DartifactId=cms-web
```

1.7 Add archetype and interactive mode:

**-DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false**

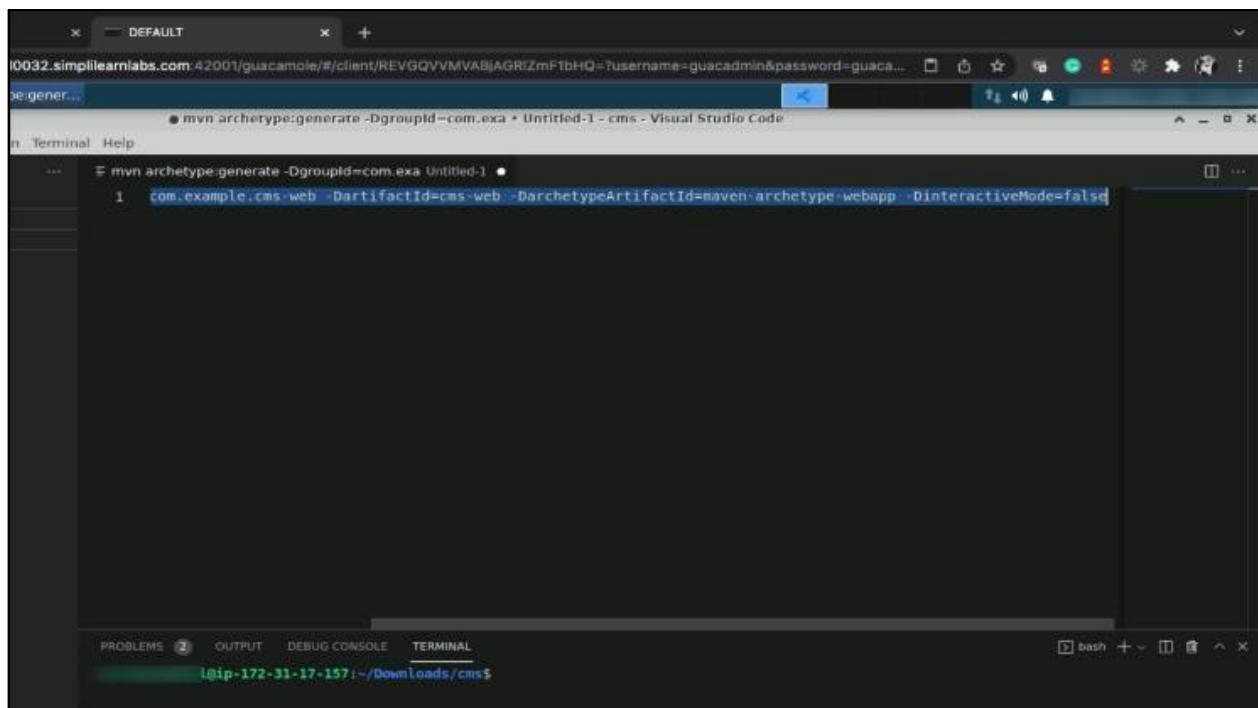


```
mvn archetype:generate -DgroupId=com.exa Untitled-1 -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false
```

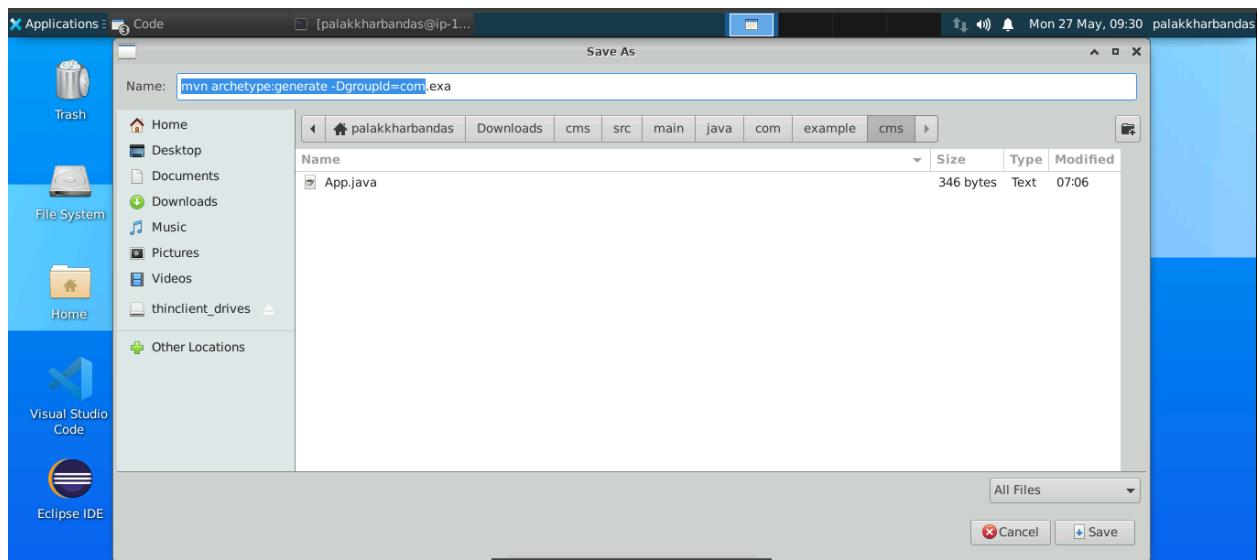
The **-DarchetypeArtifactId=maven-archetype-webapp** flag specifies the Maven archetype or project template for creating a new web application project. The **-DinteractiveMode=false** flag tells Maven to use non-interactive mode, which means that it won't prompt the user for input during the project creation process.

The complete command will look like: **mvn archetype:generate -DgroupId=com.example.cms-web -DartifactId=cms-web -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 -DinteractiveMode=false**

#### 1.8 Copy the following command:

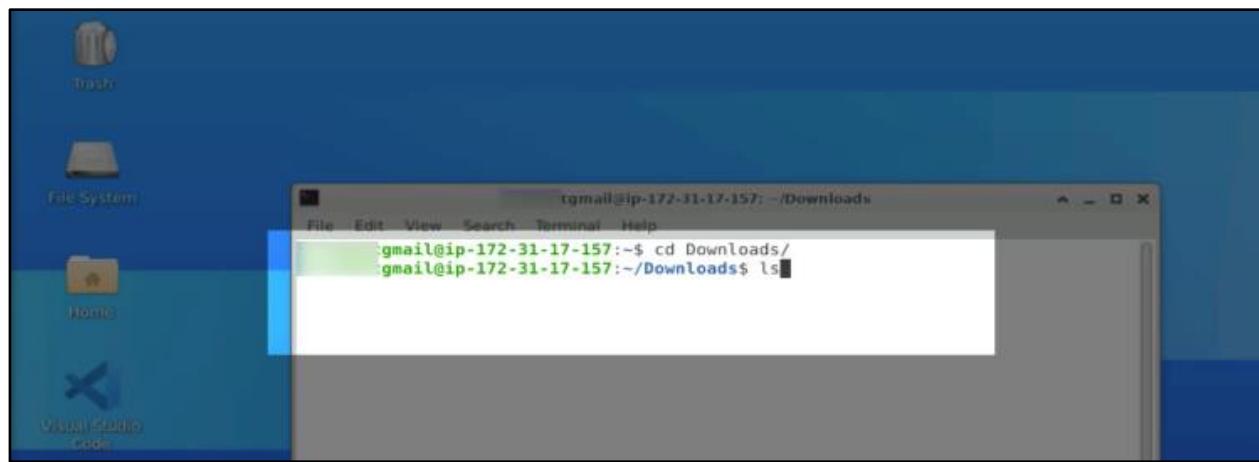


A screenshot of the Visual Studio Code interface, specifically focusing on the Terminal tab. The terminal window shows the command `mvn archetype:generate -DgroupId=com.example.cms-web -DartifactId=cms-web -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 -DinteractiveMode=false` copied into it. The terminal tab is highlighted at the bottom of the screen.



1.9 Enter the **Downloads** directory in the terminal window:

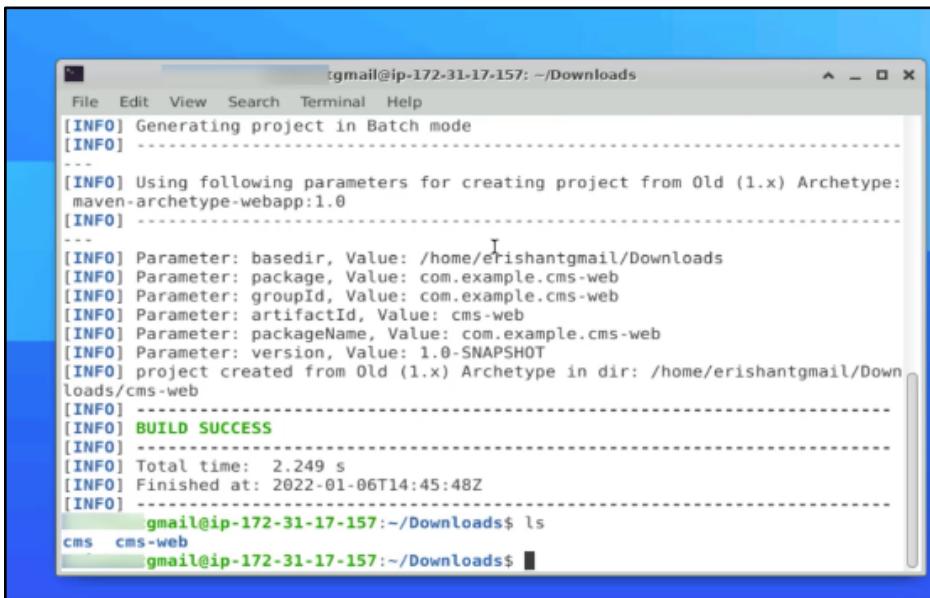
```
cd Downloads/
```



### 1.10 Paste the above command and press Enter:

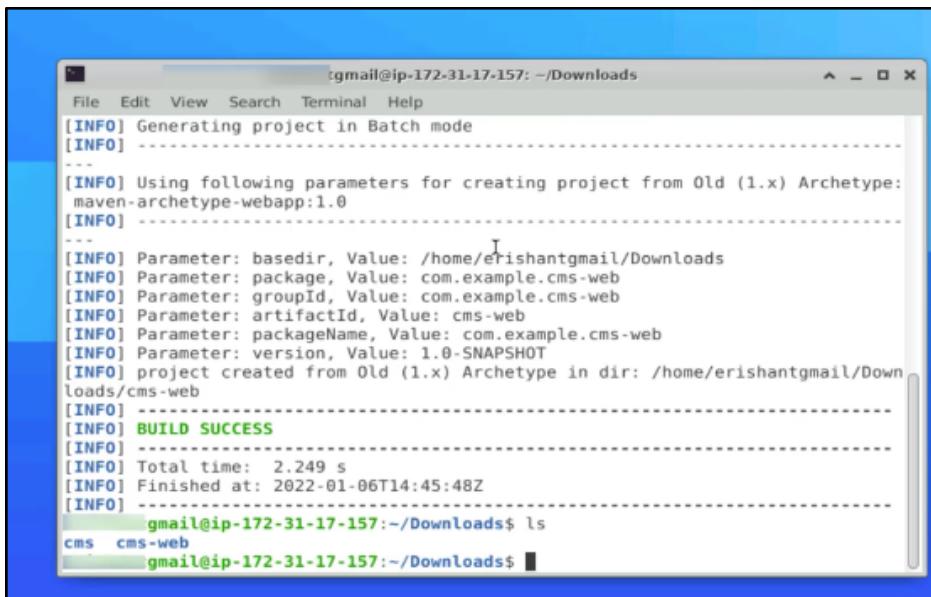
```
mvn archetype:generate -DgroupId=com.example.cms-web -DartifactId= cms-web - 
DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 - 
DinteractiveMode=false
```

```
palakkharbandas@ip-172-31-17-240:~/Downloads$ mvn archetype:generate -DgroupId=com.example.cms-web -DartifactId= cms-web -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 -DinteractiveMode=false
[INFO] Scanning for projects...
[INFO]
[INFO] -----< org.apache.maven:standalone-pom >-----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----[ pom ]-----
[INFO]
[INFO] >>> maven-archetype-plugin:3.2.1:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
[INFO] <<< maven-archetype-plugin:3.2.1:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO]
[INFO] --- maven-archetype-plugin:3.2.1:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Batch mode
[INFO]
```



The screenshot shows a terminal window titled "tgmail@ip-172-31-17-157: ~/Downloads". The window displays the output of a Maven command. The output includes information about the archetype being used (maven-archetype-webapp:1.0), parameters set (basedir, package, groupId, artifactId, packageName, version), and the successful creation of a project named "cms-web" in the specified directory. The terminal also shows the "BUILD SUCCESS" message and the total build time.

```
tgmail@ip-172-31-17-157: ~/Downloads
File Edit View Search Terminal Help
[INFO] Generating project in Batch mode
[INFO]
[INFO] -----
[INFO] Using following parameters for creating project from Old (1.x) Archetype:
[INFO] maven-archetype-webapp:1.0
[INFO] -----
[INFO] Parameter: basedir, Value: /home/erishantgmail/Downloads
[INFO] Parameter: package, Value: com.example.cms-web
[INFO] Parameter: groupId, Value: com.example.cms-web
[INFO] Parameter: artifactId, Value: cms-web
[INFO] Parameter: packageName, Value: com.example.cms-web
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: /home/erishantgmail/Downloads/cms-web
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 2.249 s
[INFO] Finished at: 2022-01-06T14:45:48Z
[INFO]
[INFO] -----
gmail@ip-172-31-17-157:~/Downloads$ ls
cms cms-web
gmail@ip-172-31-17-157:~/Downloads$
```



```
tgmail@ip-172-31-17-157: ~/Downloads
File Edit View Search Terminal Help
[INFO] Generating project in Batch mode
[INFO] -----
[INFO] Using following parameters for creating project from Old (1.x) Archetype:
maven-archetype-webapp:1.0
[INFO] -----
[INFO] Parameter: basedir, Value: /home/erishantgmail/Downloads
[INFO] Parameter: package, Value: com.example.cms-web
[INFO] Parameter: groupId, Value: com.example.cms-web
[INFO] Parameter: artifactId, Value: cms-web
[INFO] Parameter: packageName, Value: com.example.cms-web
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: /home/erishantgmail/Downloads/cms-web
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.249 s
[INFO] Finished at: 2022-01-06T14:45:48Z
[INFO] -----
gmail@ip-172-31-17-157:~/Downloads$ ls
cms cms-web
gmail@ip-172-31-17-157:~/Downloads$
```

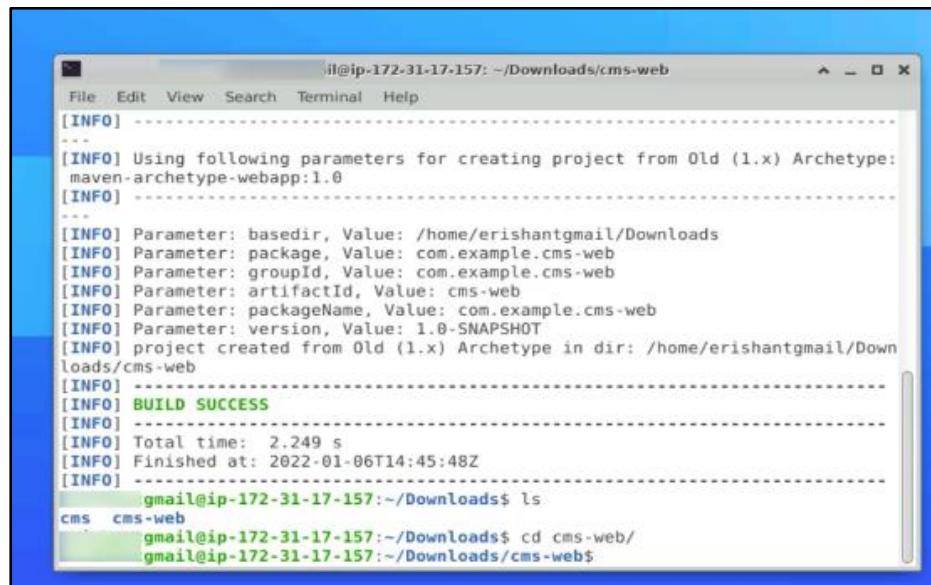
1.11 Use the **ls** command to list the directories, and the **cms** folder will be present



```
[INFO] -----
gmail@ip-172-31-17-157:~/Downloads$ ls
cms cms-web
gmail@ip-172-31-17-157:~/Downloads$
```

1.11 Use the **cd** command to navigate to the **cms-web** folder:

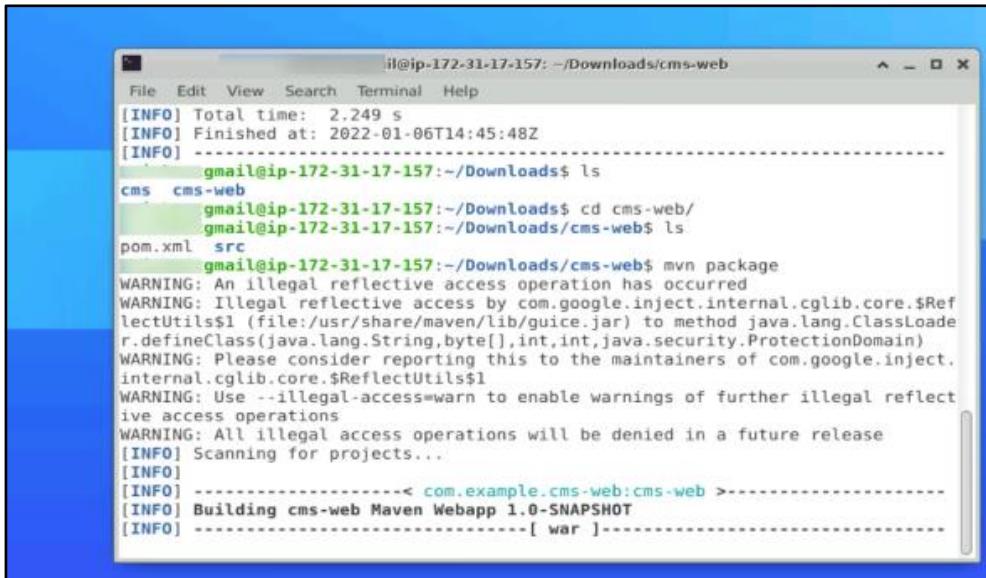
```
cd cms-web/
```



```
il@ip-172-31-17-157: ~/Downloads/cms-web
File Edit View Search Terminal Help
[INFO] -----
[INFO] Using following parameters for creating project from Old (1.x) Archetype:
maven-archetype-webapp:1.0
[INFO] -----
[INFO] Parameter: basedir, Value: /home/erishantgmail/Downloads
[INFO] Parameter: package, Value: com.example.cms-web
[INFO] Parameter: groupId, Value: com.example.cms-web
[INFO] Parameter: artifactId, Value: cms-web
[INFO] Parameter: packageName, Value: com.example.cms-web
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: /home/erishantgmail/Downloads/cms-web
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.249 s
[INFO] Finished at: 2022-01-06T14:45:48Z
[INFO] -----
gmail@ip-172-31-17-157:~/Downloads$ ls
cms cms-web
gmail@ip-172-31-17-157:~/Downloads$ cd cms-web/
gmail@ip-172-31-17-157:~/Downloads/cms-web$
```

### 1.12 Use the **mvn package** command to build the project:

**mvn package**



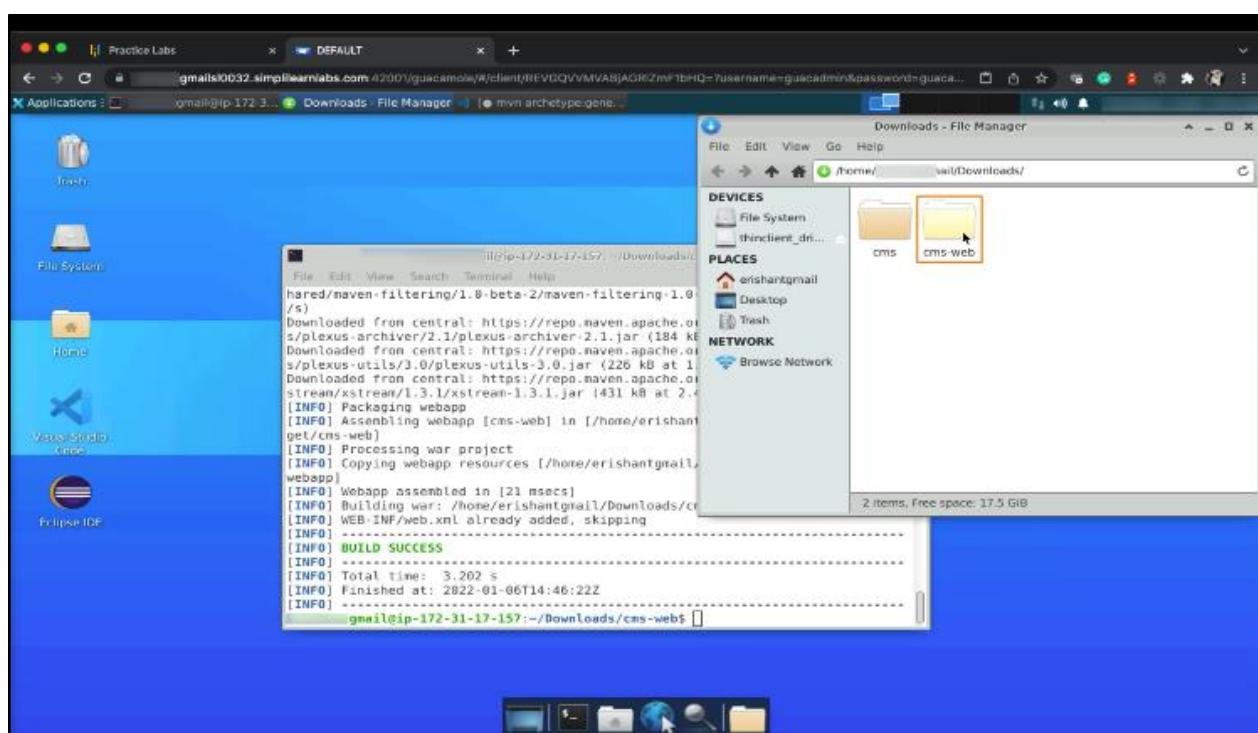
The screenshot shows a terminal window with a blue header bar. The terminal title is "jil@ip-172-31-17-157: ~/Downloads/cms-web". The window contains the following text:

```
jil@ip-172-31-17-157: ~/Downloads/cms-web
[INFO] Total time: 2.249 s
[INFO] Finished at: 2022-01-06T14:45:48Z
[INFO]
[INFO] -----> com.example.cms-web:cms-web <-----
[INFO] Building cms-web Maven Webapp 1.0-SNAPSHOT
[INFO] -----[ war ]-----
```

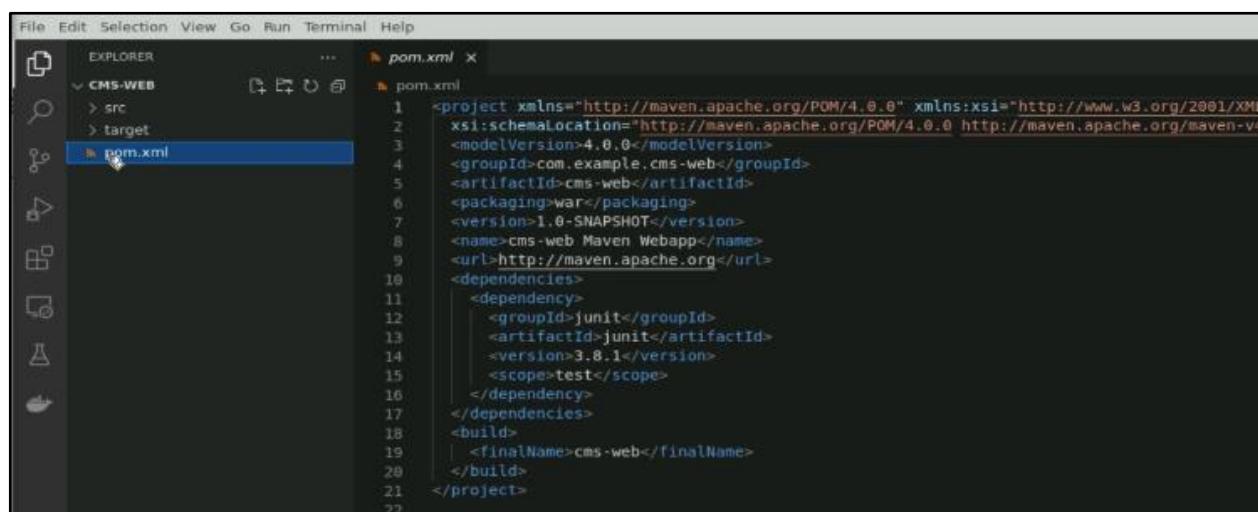
The terminal shows the user navigating to the project directory, listing files, and then executing the `mvn package` command. Maven outputs several warning messages about illegal reflective access before finally starting the build process.

## Step 2: Open the CMS project

### 2.1 Navigate to the **Downloads** directory and open the **cms-web** folder in the Visual Studio Code

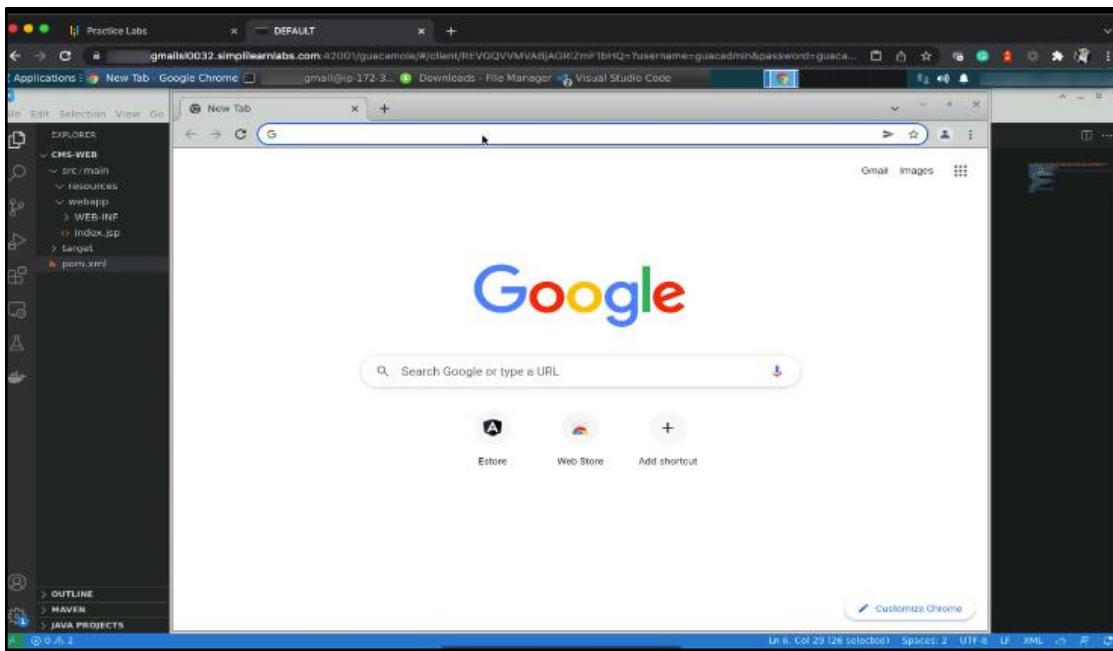


### 2.2 Open the **pom.xml** file. Check and ensure that the packaging is **war** and not **jar**.

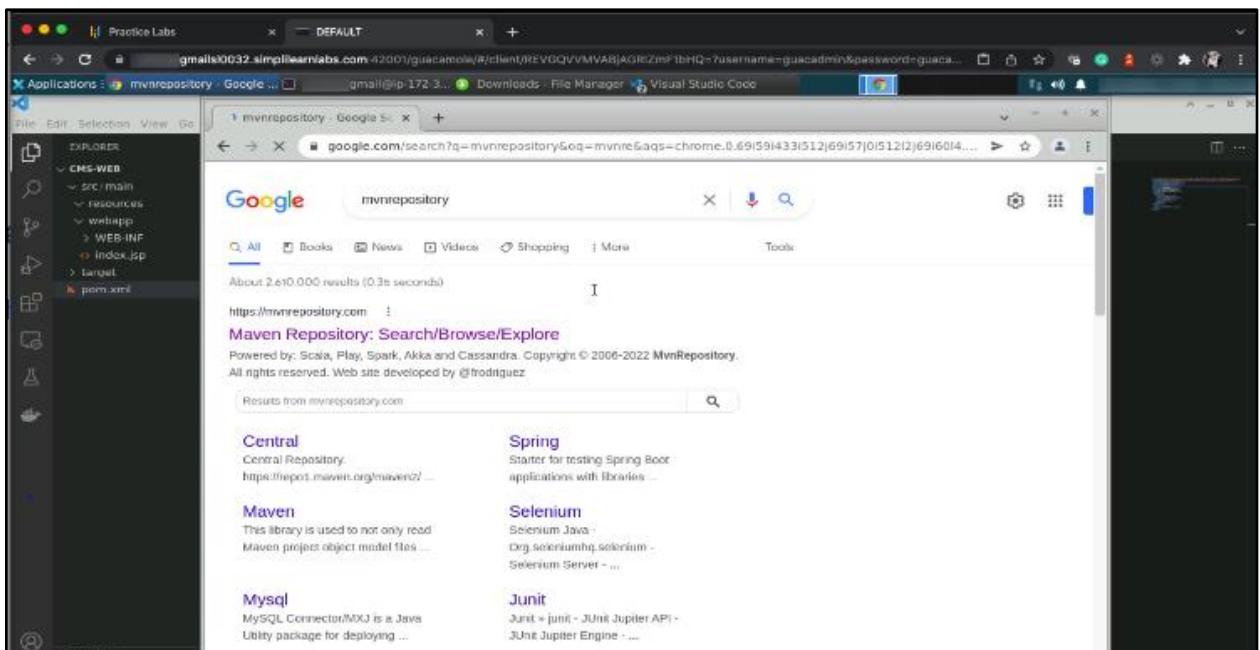


## Step 3: Add the dependency

### 3.1 Open the Google browser



### 3.2 Search for mvn repository



The screenshot shows the Maven Repository homepage. On the left, there's a sidebar with 'Indexed Artifacts (35.8M)' and a line graph showing the growth of projects from 2006 to 2018. Below that is a 'Popular Categories' section with links to Testing Frameworks & Tools, Android Packages, Logging Frameworks, Java Specifications, JSON Libraries, and JVM Languages. The main content area features a 'What's New in Maven' section with three items:

- ClickHouse JDBC Driver** (com.clickhouse » clickhouse-jdbc » 0.6.0-patch5) - 57 usages, Apache. Last Release on May 26, 2024.
- GradlePlugin** (ru.astrainteractive.gradleplugin » convention » 1.1.2) - 10 usages, Apache. Last Release on May 26, 2024.
- ClickHouse Java Client** (com.clickhouse » clickhouse-client » 0.6.0-patch5) - 10 usages, Apache. Last Release on May 26, 2024.

On the right, there's a sidebar for 'Indexed Repositories (1997)' with links to Central and Atlassian. There's also a CarMax advertisement.

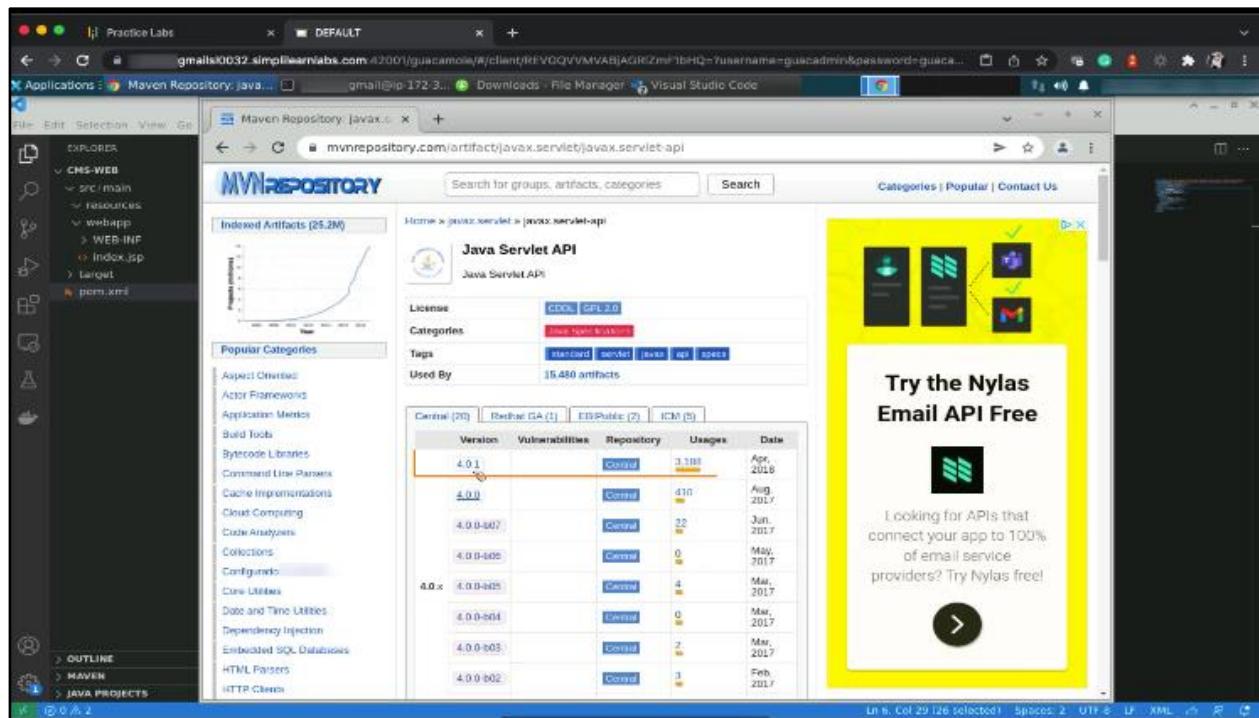
### 3.3 Search for **servlet** in the search box and select Java Servlet API

The screenshot shows a browser window with two tabs: 'Practice Labs' and 'Maven Repository: servlet'. The 'Maven Repository: servlet' tab is active, showing search results for 'servlet'. The search bar contains 'servlet'. The results are sorted by relevance and show three items:

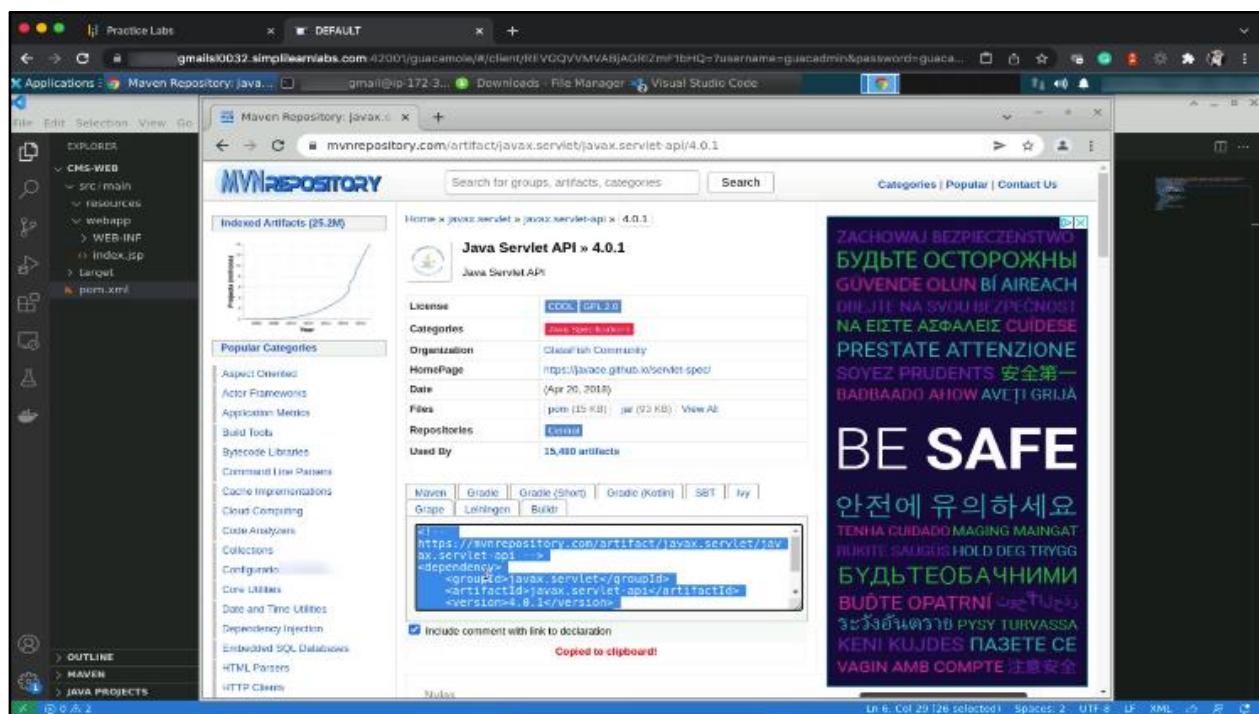
- Java Servlet API** (java.servlet » java.servlet-api) - 15,480 usages, Apache. Last Release on Apr 20, 2018.
- JavaServer(TM) Specification** (java.servlet » javax-servlet-api) - 11,422 usages, Apache. Last Release on Apr 17, 2018.
- Jetty :: Servlet Handling** (org.eclipse.jetty » jetty-servlet) - 2,190 usages, Apache. Last Release on Oct 12, 2025.

On the left, there's a file explorer showing a project structure with files like pom.xml, index.jsp, and web.xml. The Maven Repository sidebar on the right lists various repositories like Central, Sonatype, Atlassian, and more.

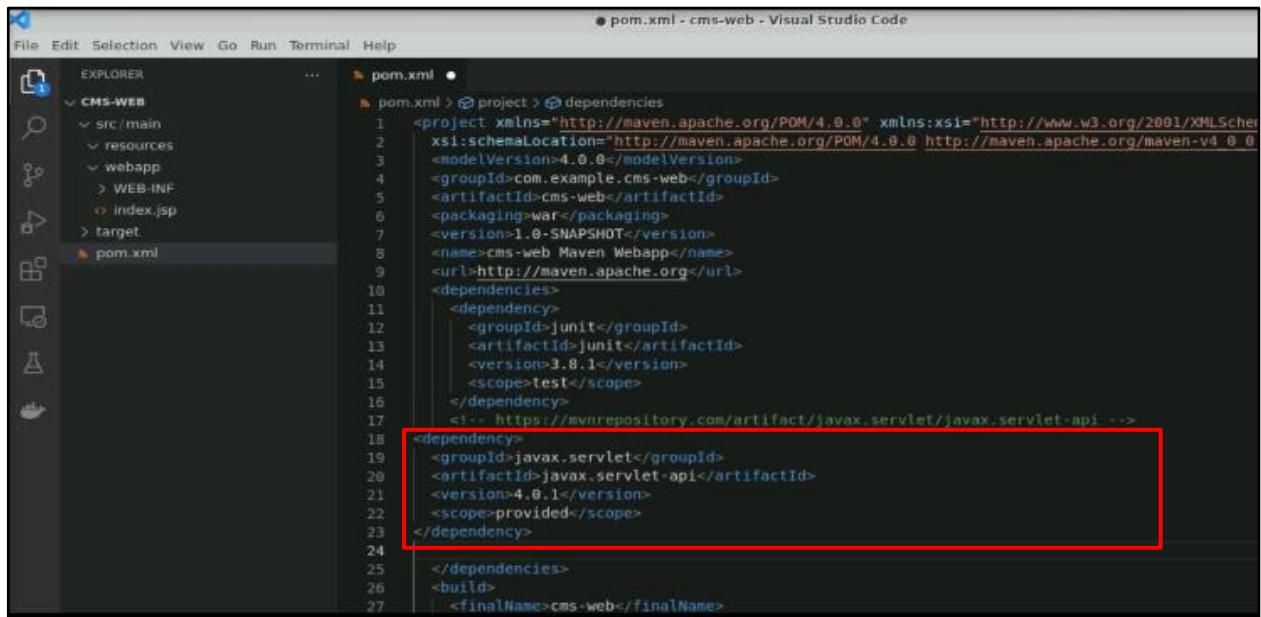
### 3.4 Select the latest version of Servlet, which is 4.0.1



### 3.5 Copy the dependency from the results shown



### 3.6 Paste it under the dependency in the pom.xml file



```
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>4.0.1</version>
    <scope>provided</scope>
</dependency>
```

This is how you can easily add dependencies for the Servlet API. The same process can be used for other dependencies, such as Spring.

By following these steps, you have successfully created a Java Web Project using Maven CLI, built the project, opened it in Visual Studio Code, and added the necessary dependencies.