**HandsOn 1 : Create a Spring Web Project using Maven**

**Steps :-**

1. **Project Initialization Using Spring Initializr -**

I created a new Spring Boot project using <https://start.spring.io>, configuring it with the following details:

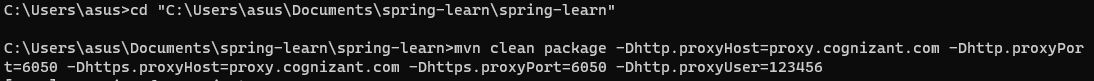
* **Group**: com.cognizant
* **Artifact**: spring-learn
* **Dependencies**: Spring Web, Spring Boot DevTools

1. **Project Setup and Build using Maven –**

After downloading the .zip file, I extracted it into my local workspace. The extracted folder contained the Maven project structure, including the **pom.xml** and **src** directories.

Then, I opened the terminal and navigated into the project directory.

To build the project, I ran the following Maven command:

****

A screen shot of a computer

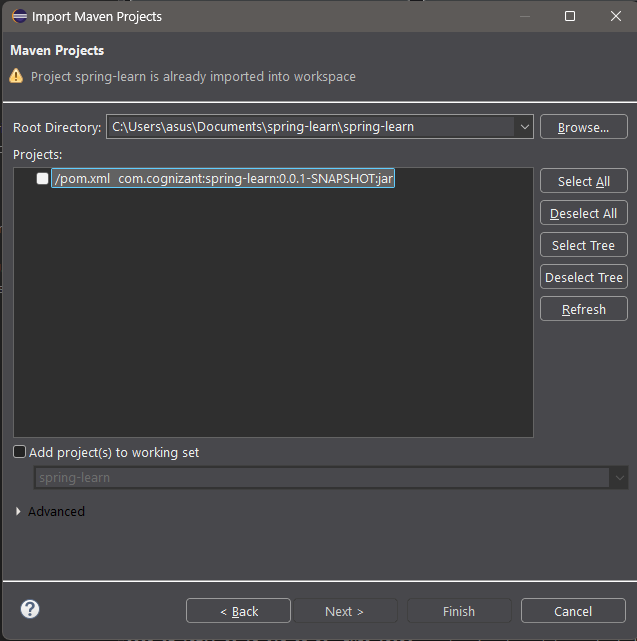
AI-generated content may be incorrect.

1. **Importing Project into Eclipse -**

I imported the project into Eclipse via:

File → Import → Maven → Existing Maven Projects

This recognized the pom.xml and auto-configured the Maven dependencies.



1. **Exploring the Project Structure –**

The standard Maven directory structure includes:

* **src/main/java** – Source code
* **src/main/resources** – Configuration files
* **src/test/java** – Unit test classes
* **pom.xml** – Maven configuration and dependencies

A screenshot of a computer program

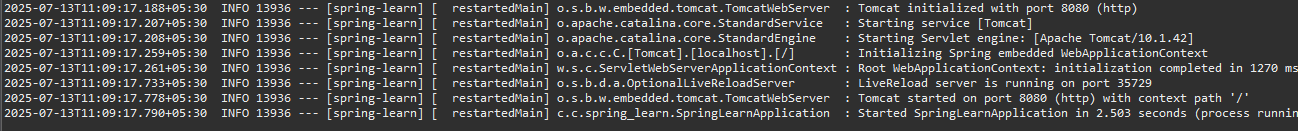
AI-generated content may be incorrect.

1. **Running the Spring Boot Application –**

I executed the SpringLearnApplication.java class using:

Right-click → Run As → Java Application

The console confirmed that the Spring Boot app started successfully with embedded Tomcat running on port 8080.

****

1. **Understanding SpringLearnApplication.java and Annotations –**

The main class contains:

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

*@SpringBootApplication*

public class SpringLearnApplication {

public static void main(String[] args) {

SpringApplication.*run*(SpringLearnApplication.class, args);

}

}

* **@SpringBootApplication:** Combines @Configuration, @EnableAutoConfiguration, and @ComponentScan
* **SpringApplication.run(...):** Starts the embedded server and initializes the Spring context

1. **Project Dependencies (pom.xml) –**

In Eclipse, I explored the **Dependency Hierarchy** tab to view how Spring Boot manages nested and transitive dependencies.

