**SME Walkthrough: Spring Web Project (IntelliJ)**

**1. src/main/java**

This folder contains the main source code of the application. It includes the primary application class (SpringLearnApplication.java) and will also include future packages such as controller, service, model, etc.

**2. src/main/resources**

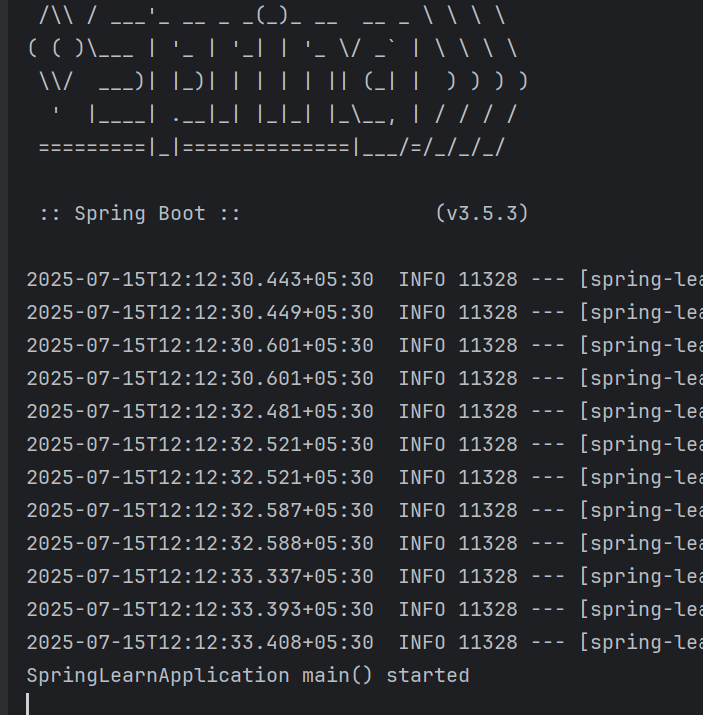
This folder is used to store configuration files and static resources. It includes files like application.properties or application.yml for configuration, as well as templates and static folders for frontend resources.

**3. src/test/java**

This directory is designated for writing unit and integration test cases. By default, it contains a test class for SpringLearnApplication which verifies if the Spring context loads properly.

**4. SpringLearnApplication.java**

This is the main entry point of the Spring Boot application. It contains the main() method where SpringApplication.run() is invoked to launch the application. A print/log statement can be added to verify execution.



**5. Purpose of @SpringBootApplication annotation**

@SpringBootApplication is a meta-annotation that combines:  
- @Configuration: Declares the class as a source of Spring bean definitions.  
- @EnableAutoConfiguration: Enables Spring Boot’s auto-configuration based on dependencies.  
- @ComponentScan: Automatically scans the package and registers annotated components as beans.

**6. pom.xml**

This is the Maven Project Object Model (POM) file used to manage project dependencies, plugins, and metadata.  
  
**1. Walkthrough all the configuration defined in XML file:**  
- The <dependencies> section lists all required libraries like spring-boot-starter-web, spring-boot-devtools, etc.  
- The <build> section may include plugins or packaging instructions.  
- The <properties> section sets values like Java version or encoding.

**2. Open 'Dependency Hierarchy' and show the dependency tree:** 