Project Structure Explanation

1. src/main/java

- Contains the main application code**.**
- In this project:

  com.cognizant.spring_learn.SpringLearnApplication.java

  - This is the entry point of the Spring Boot application.
  - Contains the main() method which bootstraps the Spring context and starts the embedded Tomcat server.
  - Includes System.out.println statements to confirm application start.

2. src/main/resources

- Contains configuration files and static resources.
- In this project:

  application.properties

  - Configures application-specific properties:
  - spring.application.name=spring-learn
  - server.port=8083

3. src/test/java

- Contains unit and integration tests**.**
- In this project:

  com.cognizant.spring_learn.SpringLearnApplicationTests.java

  - Tests if the Spring context loads without errors.
  - Uses @SpringBootTest for integration testing.

Main Class: SpringLearnApplication.java

```
@SpringBootApplication
public class SpringLearnApplication {
   public static void main(String[] args) {
      System.out.println("SpringLearnApplication is starting.");
      SpringApplication.run(SpringLearnApplication.class, args);
      System.out.println("SpringLearnApplication started successfully.");
   }
}
```

- Prints startup and success messages.
- Calls SpringApplication.run() to launch the application.
- The @SpringBootApplication annotation enables:
    - @Configuration — Marks this as a configuration class.
    - @EnableAutoConfiguration — Automatically configures Spring components based on dependencies.
    - @ComponentScan — Scans the package and its sub-packages for Spring components.

pom.xml Explanation

The Maven POM defines the project dependencies and build configuration.

Key sections:

- **Parent:**

  ```
  <parent>
     <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.5.3</version>
  </parent>
  ```

    - Inherits default Spring Boot settings.
- **Dependencies:**
    - spring-boot-starter-web
        - Provides Spring MVC, embedded Tomcat, JSON serialization, etc.
    - spring-boot-devtools
        - Enables live reload and other development-time tools.
    - spring-boot-starter-test

        - Includes JUnit and testing utilities.
- **Build plugin:**
    - spring-boot-maven-plugin
        - Allows building and running Spring Boot applications with Maven.

Dependency Hierarchy

- In Eclipse → pom.xml → *Dependency Hierarchy* tab:
    - Shows the full tree of dependencies.
    - Example:

      spring-boot-starter-web
      ├── spring-webmvc
      ├── spring-web
      ├── jackson-databind
      └── tomcat-embed-core

    - Useful to see transitive dependencies and version conflicts.