Exercise 4: Creating and Configuring a Maven Project

1. Create a New Maven Project

I made a new folder named `LibraryManagement` and created Maven project inside it. The folder contains `src/main/java`, `src/main/resources` and `src/test/java`.

2. Add Spring Dependencies in `pom.xml`

I added the Spring dependencies for **Context**, **AOP**, and **WebMVC** in the `pom.xml` file.

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
         http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>
    <groupId>com.library</groupId>
    <artifactId>LibraryManagement</artifactId>
    <version>1.0-SNAPSHOT</version>

    <dependencies>
        <!-- Spring Core Context -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>5.3.9</version>
        </dependency>

        <!-- Spring AOP -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-aop</artifactId>
            <version>5.3.9</version>
        </dependency>

        <!-- Spring WebMVC -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-webmvc</artifactId>
            <version>5.3.9</version>
        </dependency>
    </dependencies>
```

---

3. Configure Maven Compiler Plugin

I added the maven compiler plugin at the end of the `pom.xml` file to make sure it uses java 1.8.

```
   <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.8.1</version>
                <configuration>
                    <source>1.8</source>
                    <target>1.8</target>
                </configuration>
            </plugin>
        </plugins>
    </build>

</project>
```