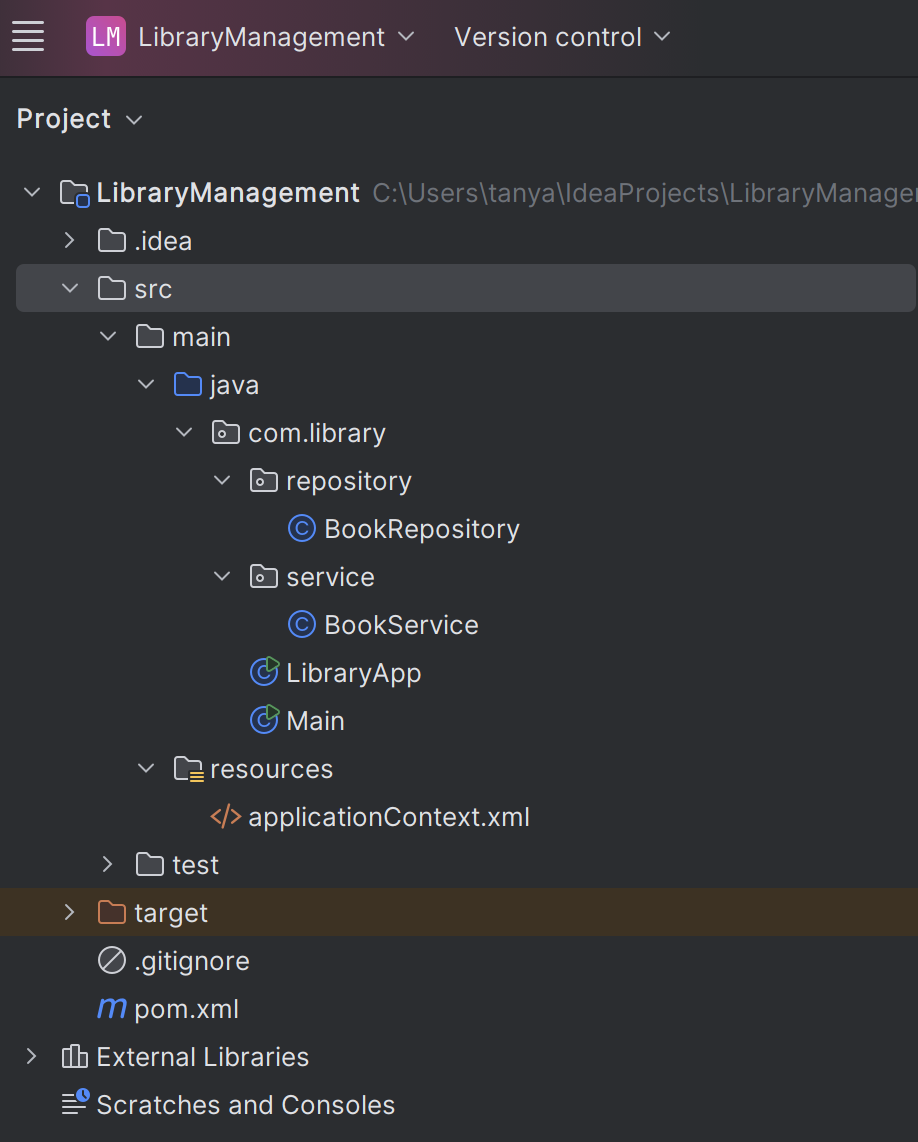
# Spring Core Maven

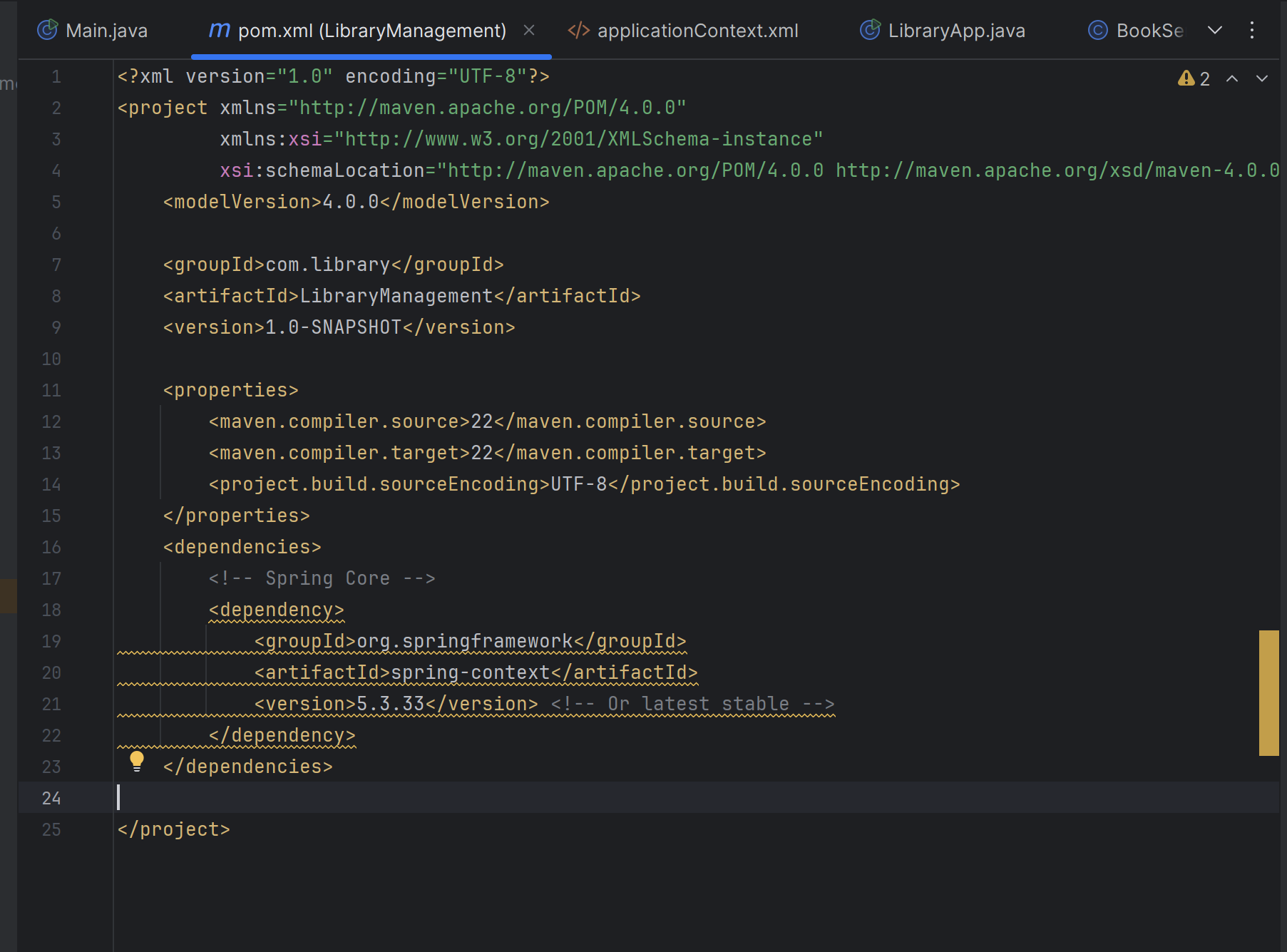
**Exercise 1: Configuring a Basic Spring Application**

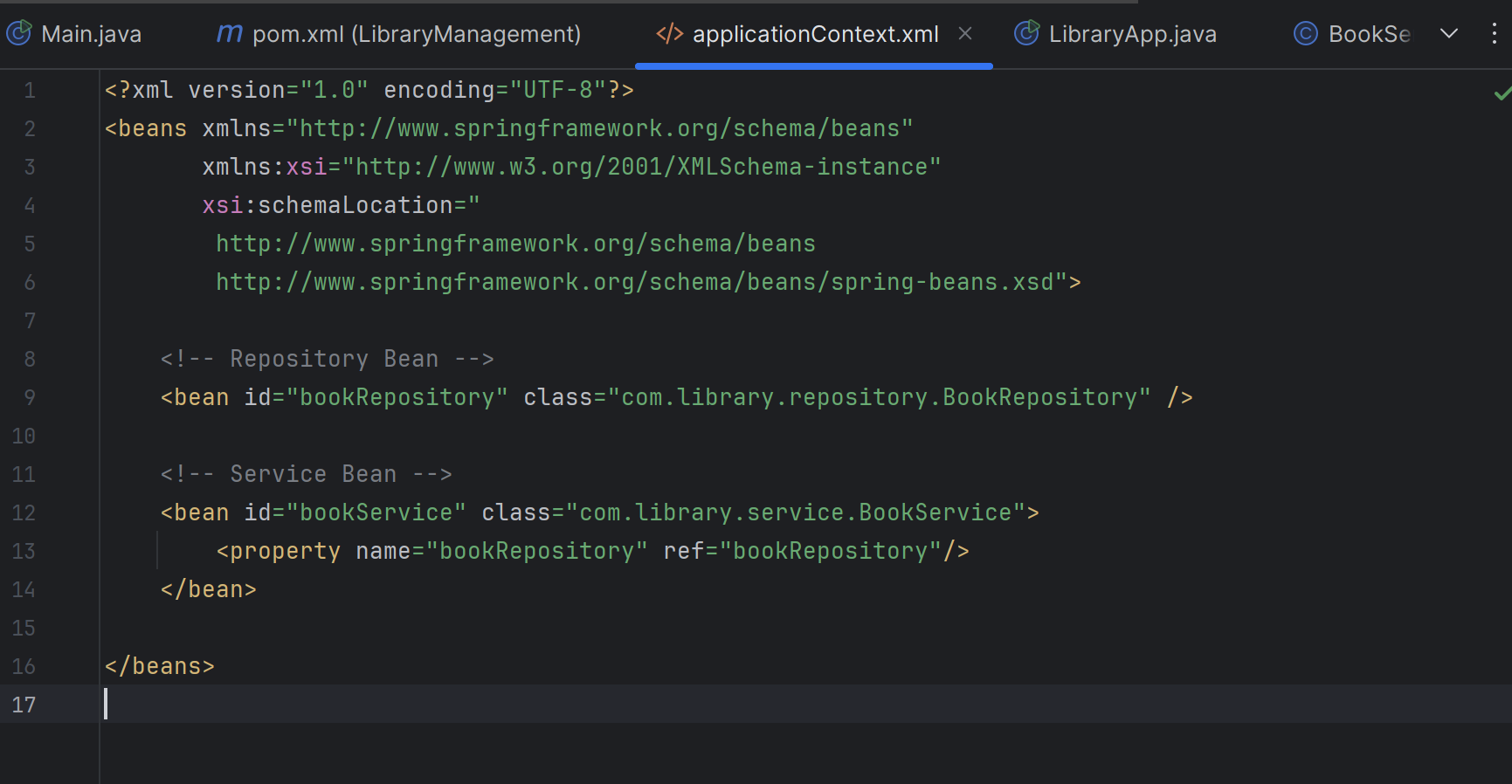
**Scenario:**

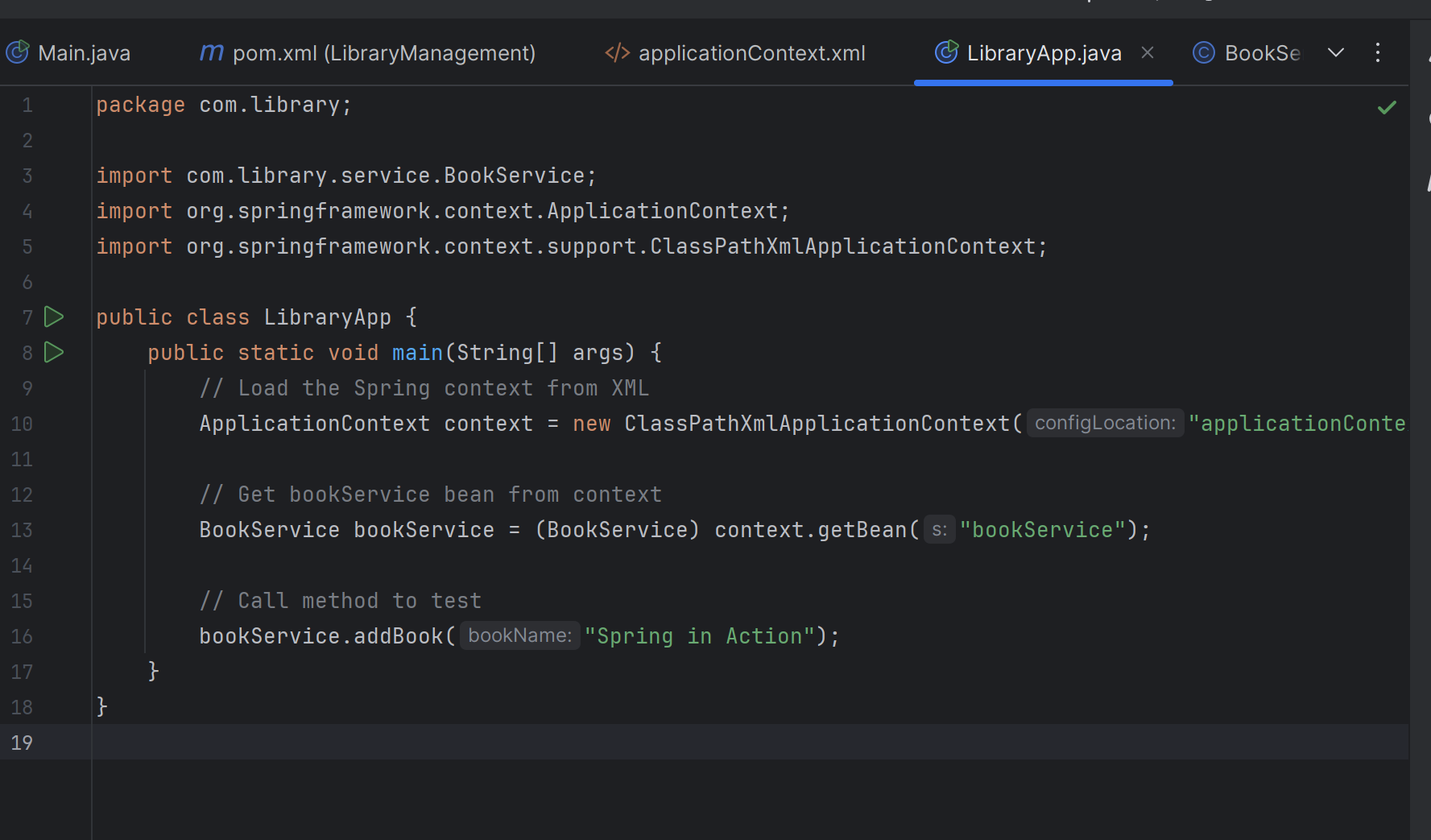
Your company is developing a web application for managing a library. You need to use the Spring Framework to handle the backend operations.

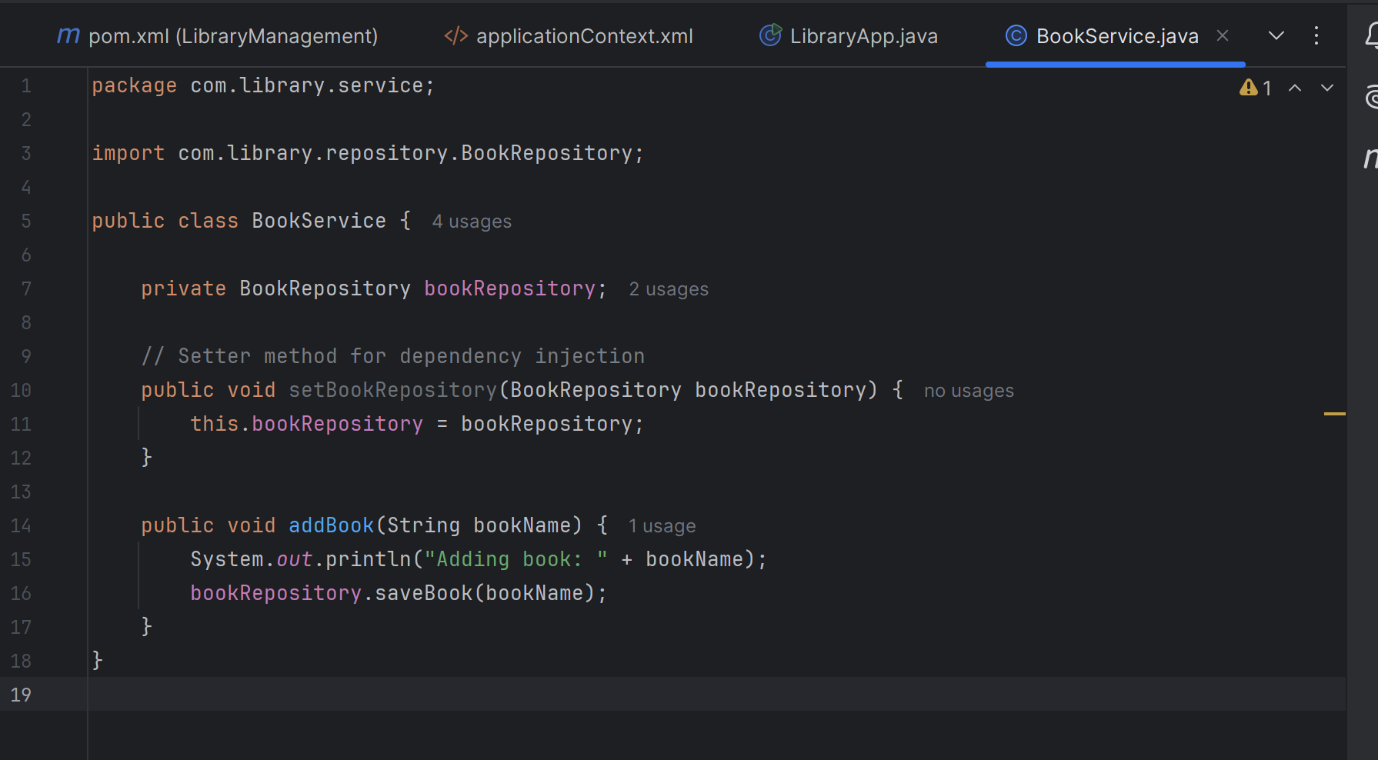
1. **Set Up a Spring Project:**
   * Create a Maven project named LibraryManagement.
   * Add Spring Core dependencies in the pom.xml file.
2. **Configure the Application Context:**
   * Create an XML configuration file named **applicationContext.xml** in the **src/main/resources** .
   * Define beans for **BookService** and **BookRepository** in the XML file.
3. **Define Service and Repository Classes:**
   * Create a package **com.library.service** and add a class **BookService**.
   * Create a package **com.library.repository** and add a class **BookRepository**.
4. **Run the Application:**Create a main class to load the Spring context and test the configuration.

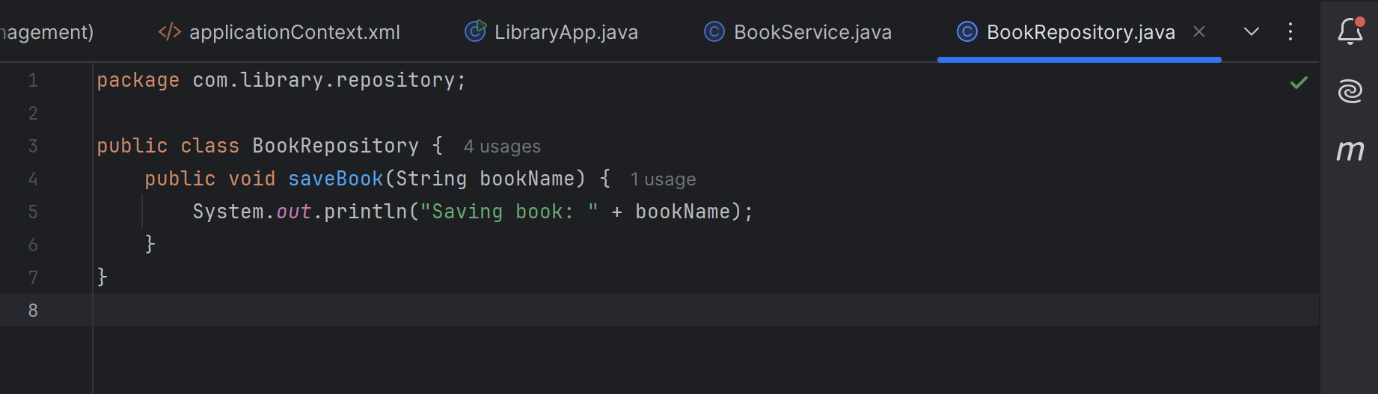
****

****

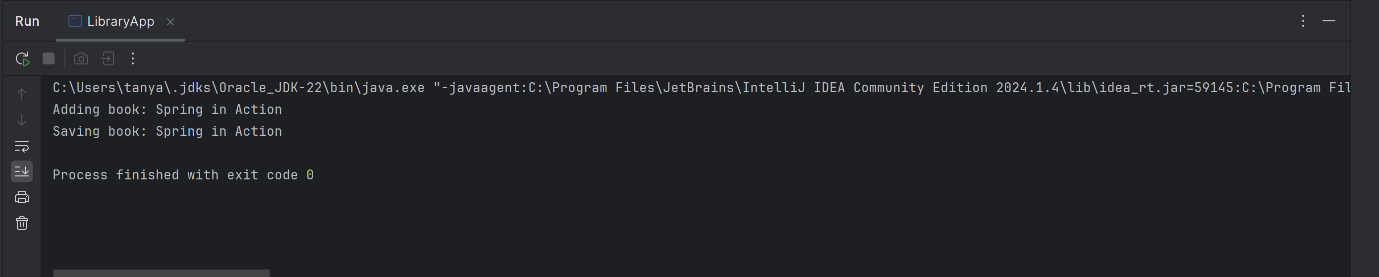
****

****

****

****

**OUTPUT**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Exercise 2: Implementing Dependency Injection**

**Scenario:**

In the library management application, you need to manage the dependencies between the BookService and BookRepository classes using Spring's IoC and DI.

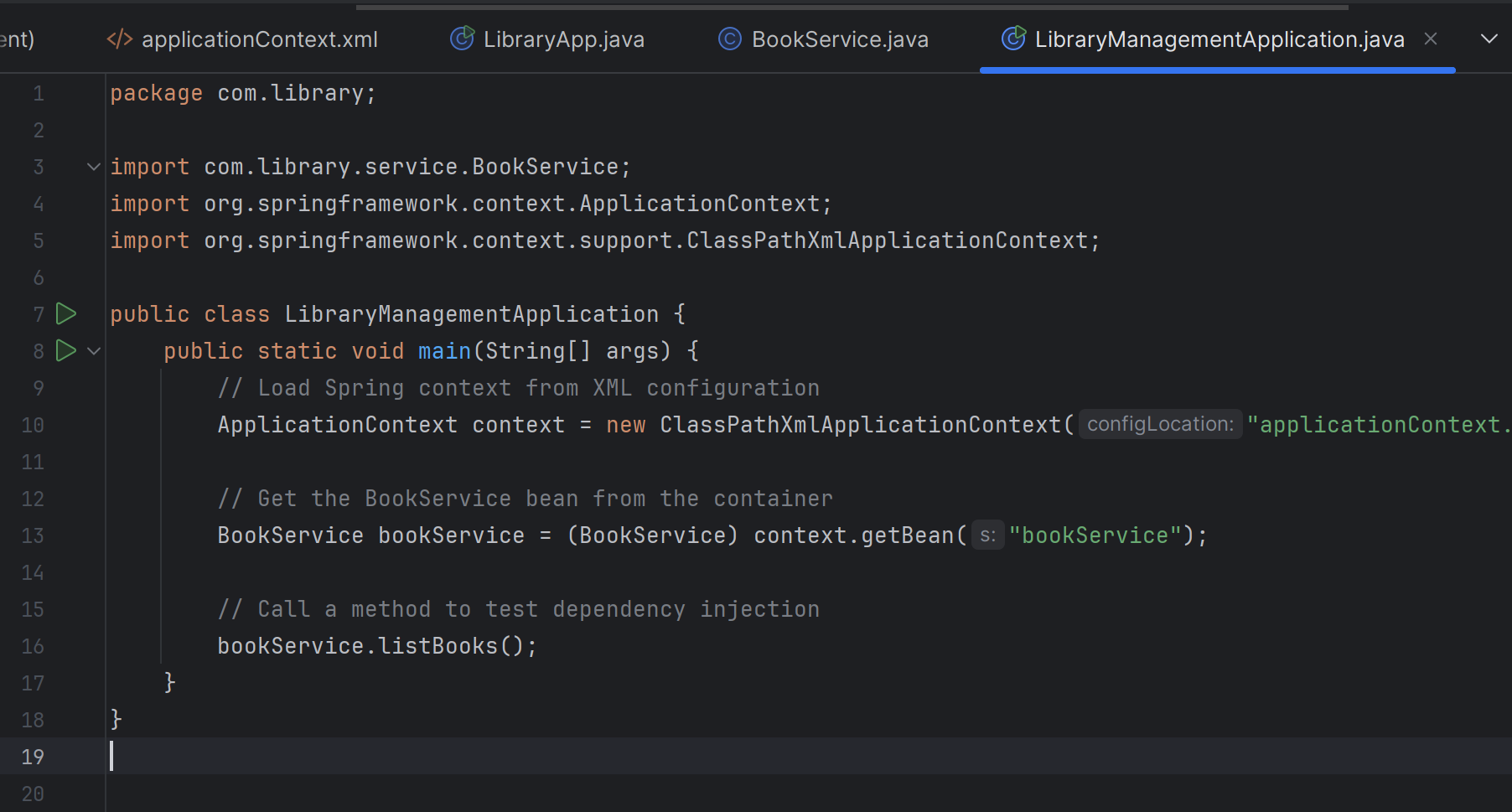
**Steps:**

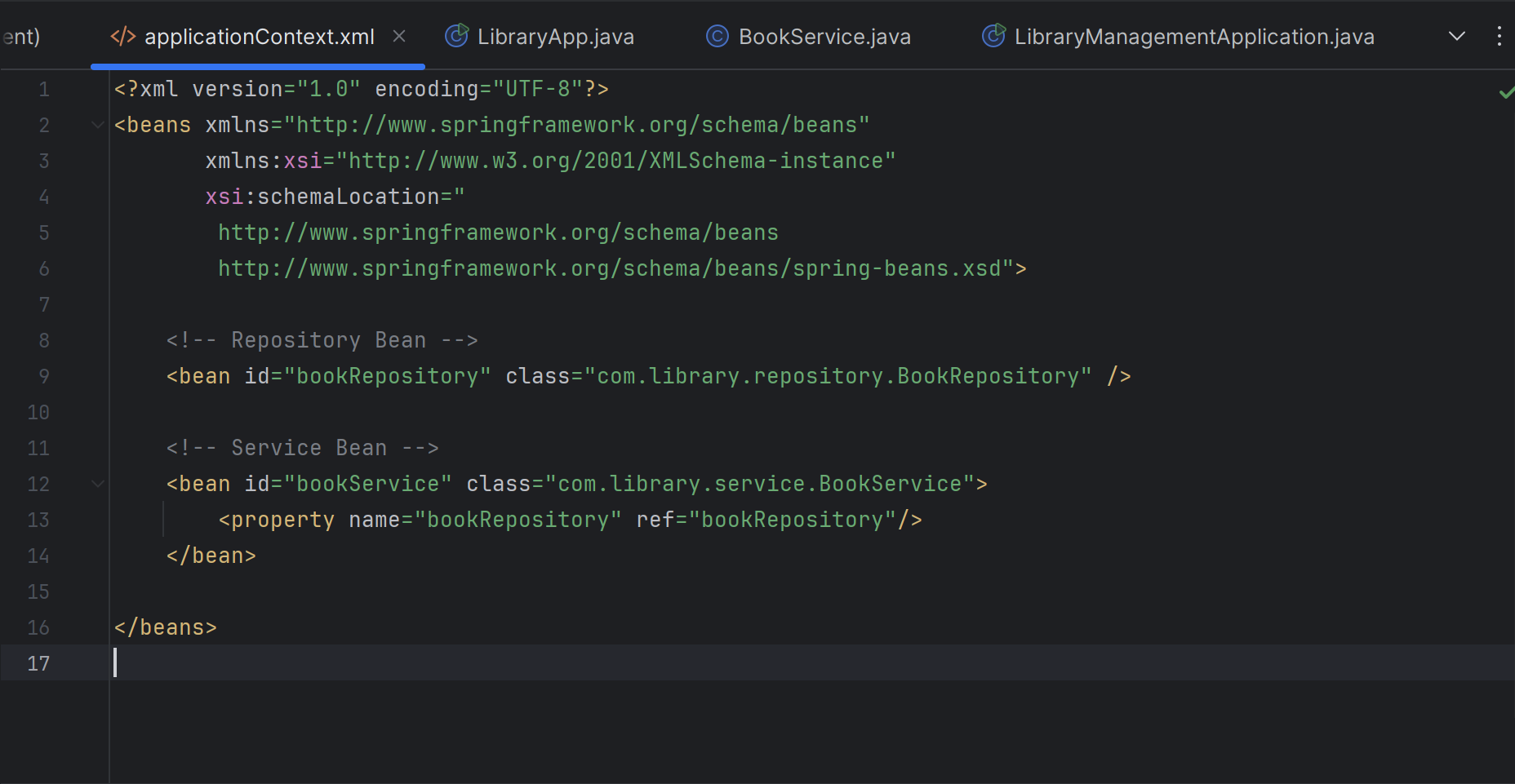
1. Modify the XML Configuration:
   * Update applicationContext.xml to wire BookRepository into BookService.
2. Update the BookService Class:
   * Ensure that BookService class has a setter method for BookRepository.
3. Test the Configuration:
   * Run the LibraryManagementApplication main class to verify the dependency injection.

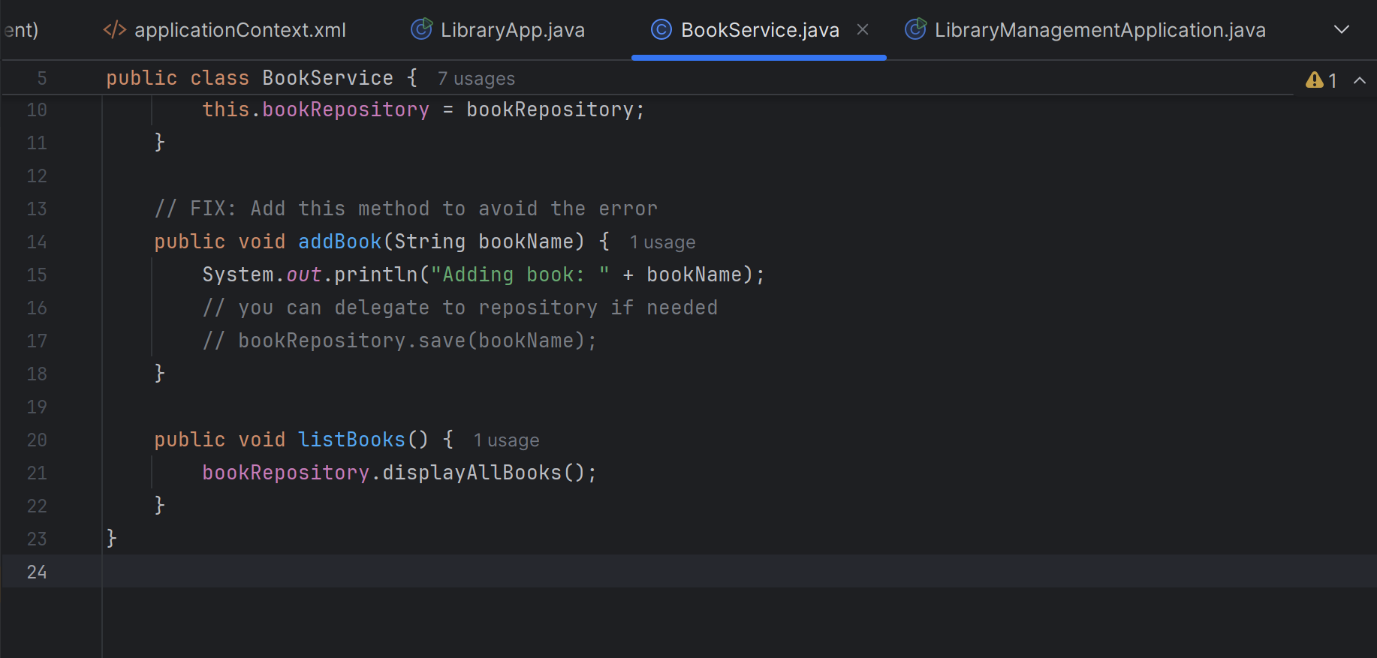
**SOLUTION**

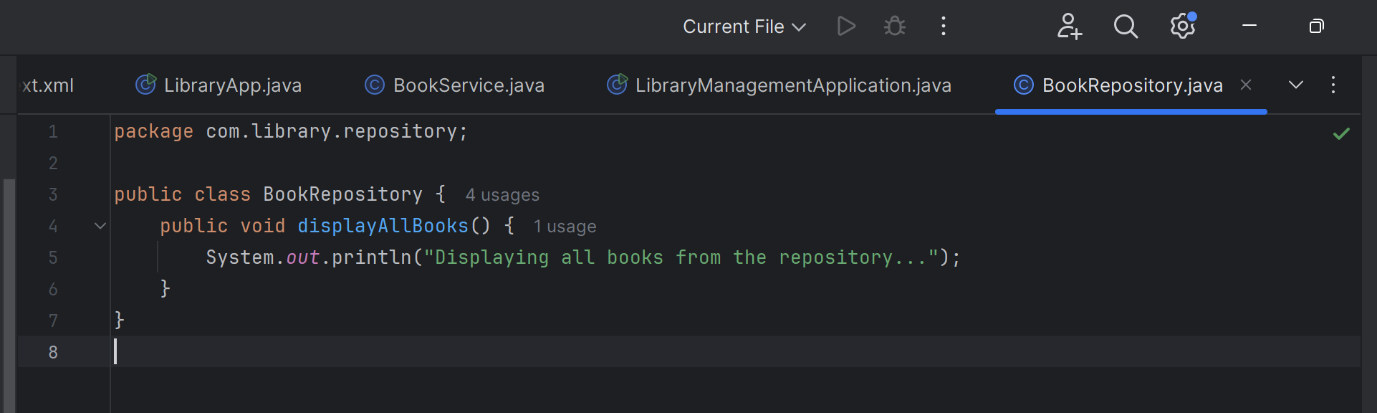
We need to make some changes in the previous example to implement the dependency injection.

Create a main class named LibraryManagementApplication.java to load the Spring context and test the DI.

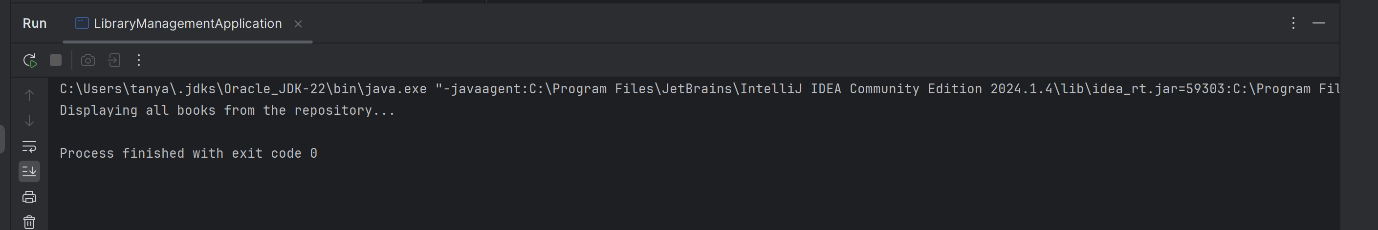






****

**OUTPUT:**

****

**--------------------------**

**Exercise 4: Creating and Configuring a Maven Project**

**Scenario:**

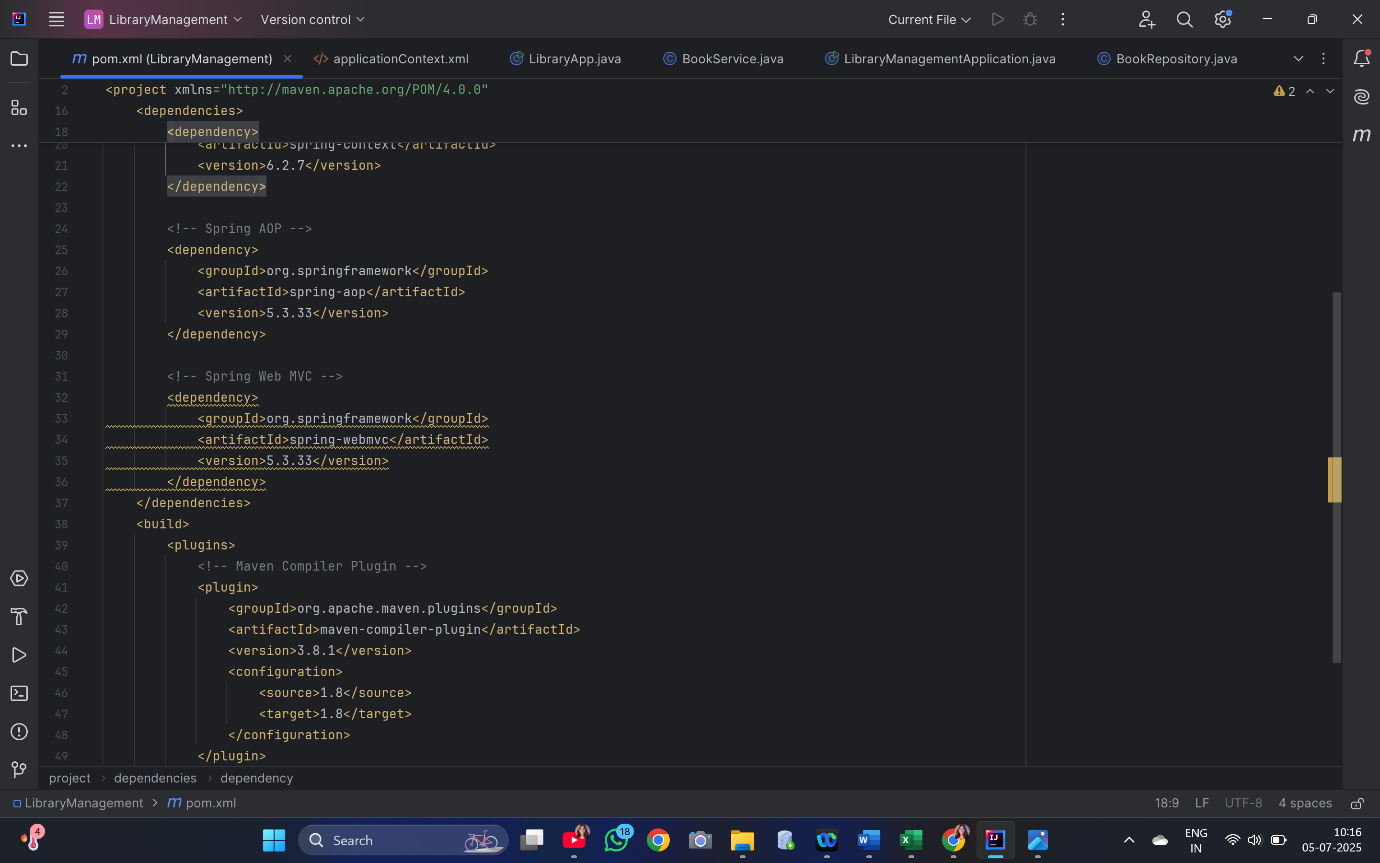
You need to set up a new Maven project for the library management application and add Spring dependencies.

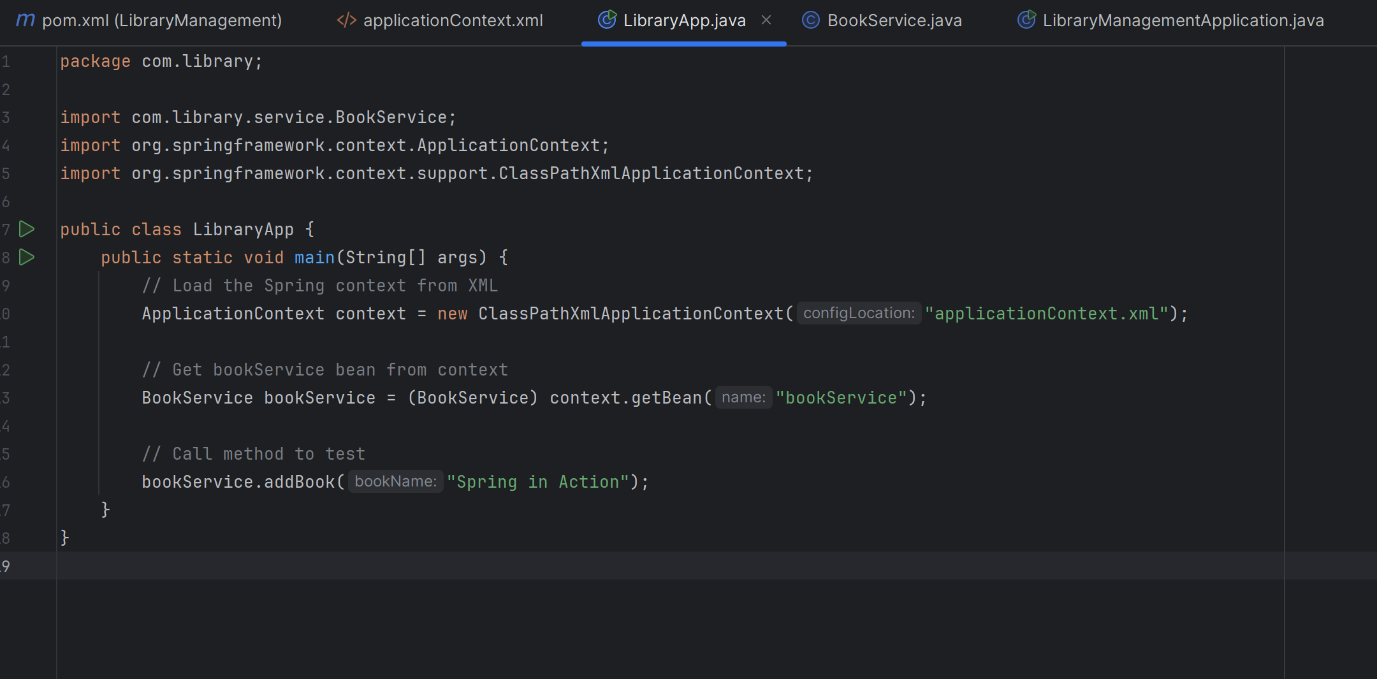
**Steps:**

1. **Create a New Maven Project:**
   * Create a new Maven project named **LibraryManagement**.
2. **Add Spring Dependencies in pom.xml:**
   * Include dependencies for Spring Context, Spring AOP, and Spring WebMVC.
3. **Configure Maven Plugins:**
   * Configure the Maven Compiler Plugin for Java version 1.8 in the pom.xml file.

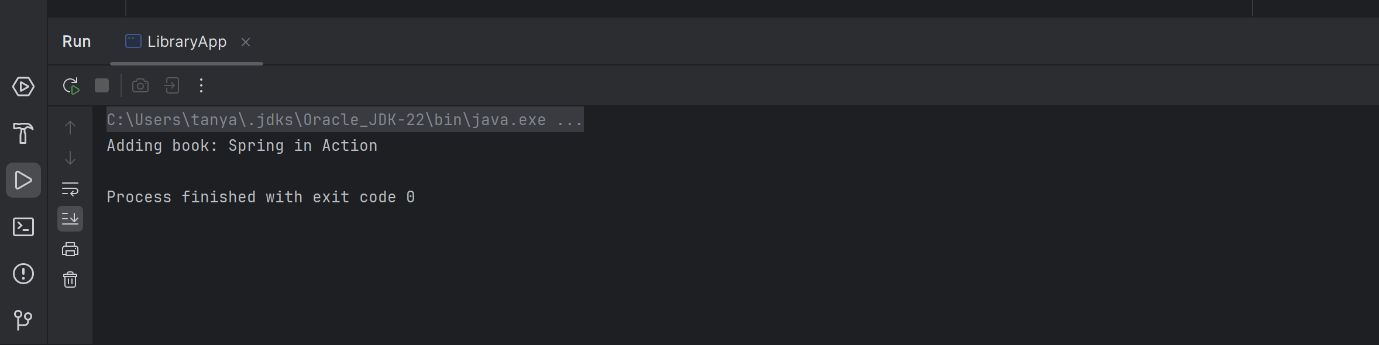
**SOLUTION:**

Add spring dependencies in pom.xml , also Add Maven Compiler Plugin.

****

****

**OUTPUT:**

****

**--------------------**