**WEEK - 3 : HANDS-ON EXERCISE**

**Spring Core and 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.

**Step 1: Project Setup**

* Create a new Maven project named 'LibraryManagement' in Eclipse.
* Update the 'pom.xml' file to include Spring Core dependency:

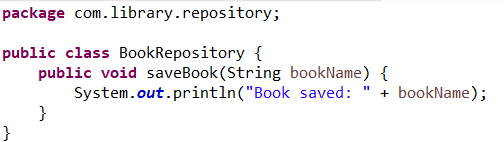


**Step 2: Create Packages and Java Classes**

* Create Packages and Classes
* com.library.repository
* com.library.service
* com.library
* Create the following Java classes:
* BookRepository.java (in repository package)
* BookService.java (in service package)
* LibraryApp.java (in com.library package)

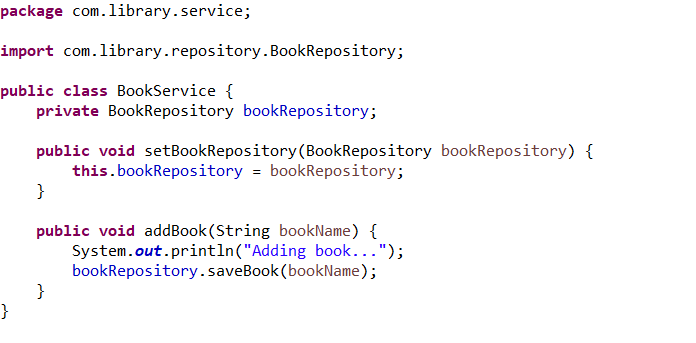
**BookRepository.java**

**CODE:**



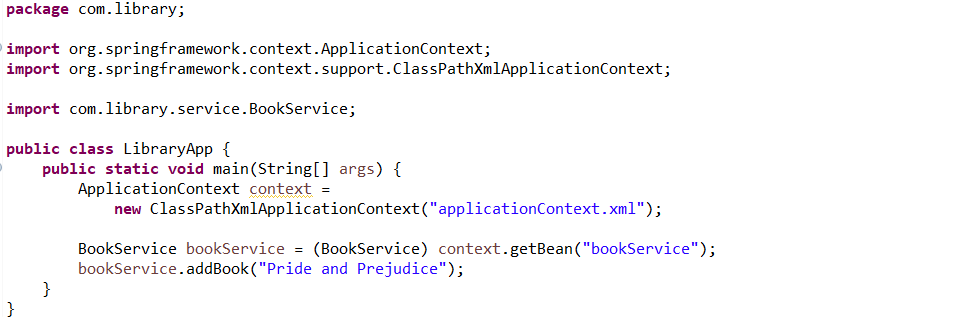
**BookService.java**

**CODE:**



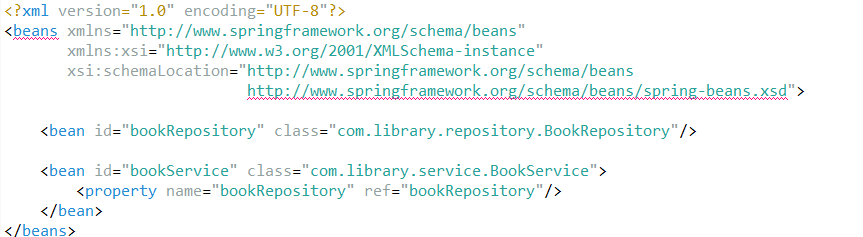
**LibraryApp.java**

**CODE:**

****

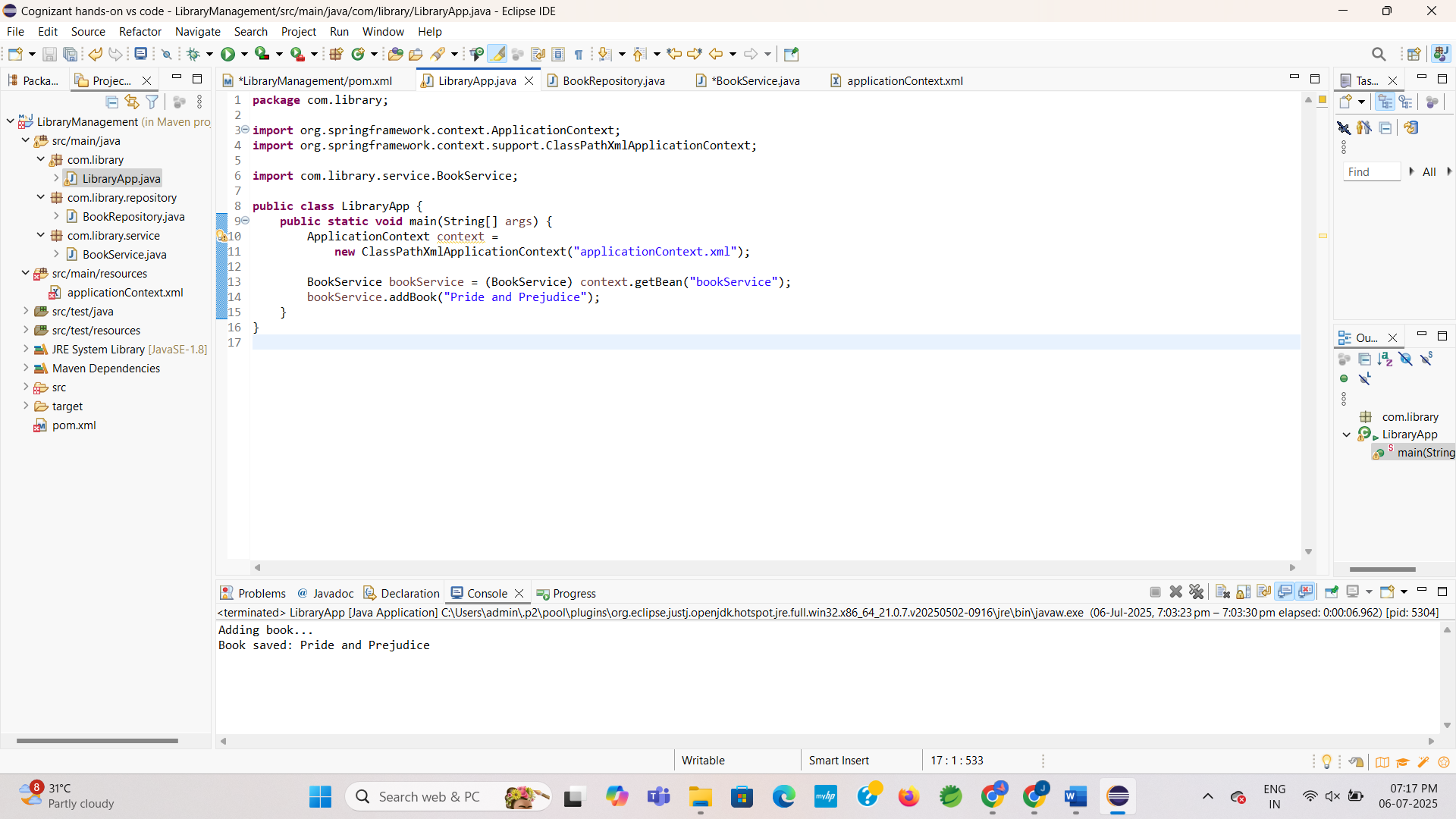
**Step 3: applicationContext.xml**

* Create 'applicationContext.xml' file in 'src/main/resources' directory and add the following:



**Step 4: Running the Application**

**OUTPUT:**



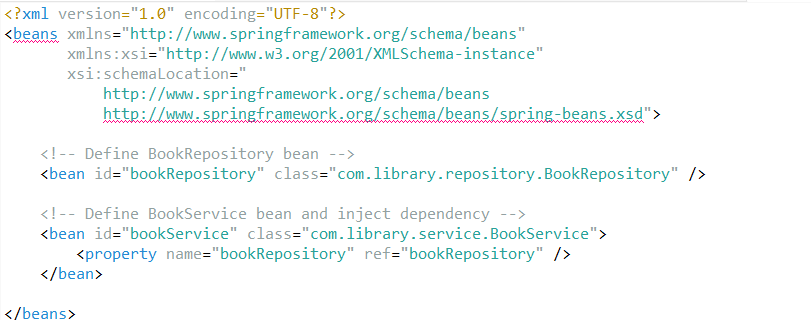
**Exercise 2: Implementing Dependency Injection**

**Scenario:**

In the same library management application, you now need to manually wire the dependency between BookService and BookRepository using Spring's XML-based Dependency Injection.

**Step 1: Modify applicationContext.xml**

* Instead of using @Autowired and @Service, you'll configure the beans manually.
* Update your applicationContext.xml like this:

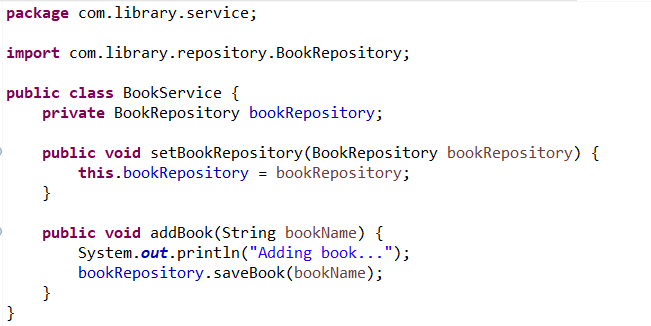


**Step 2: Update BookService.java**

* Remove annotations like @Autowired and @Service, and include a **setter method** for bookRepository.

**BookService.java**

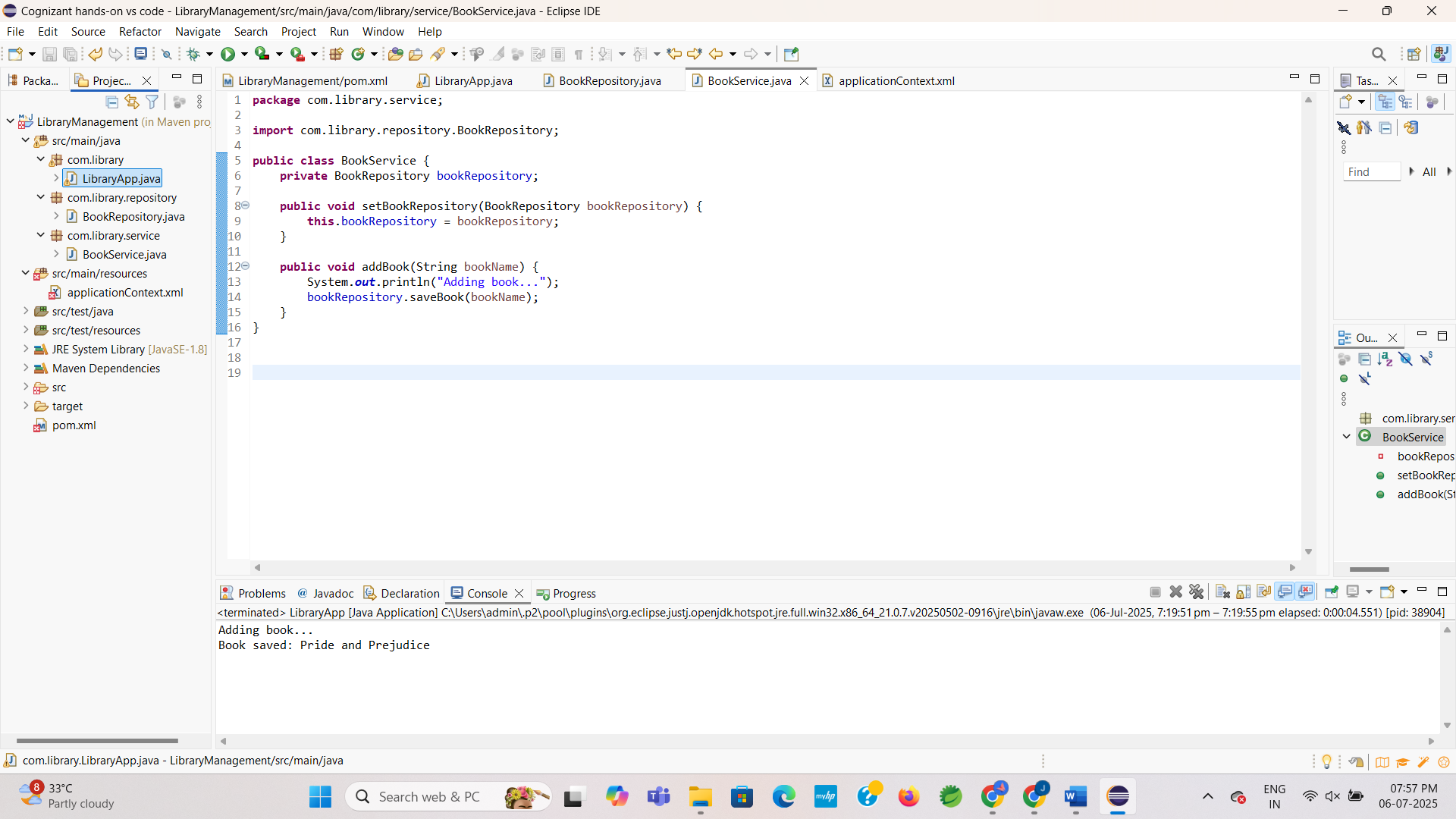
**CODE:**



**Step 3: Run the Application**

* Use the same MainApp.java to test the DI setup

**OUTPUT:**



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

**Scenario:**

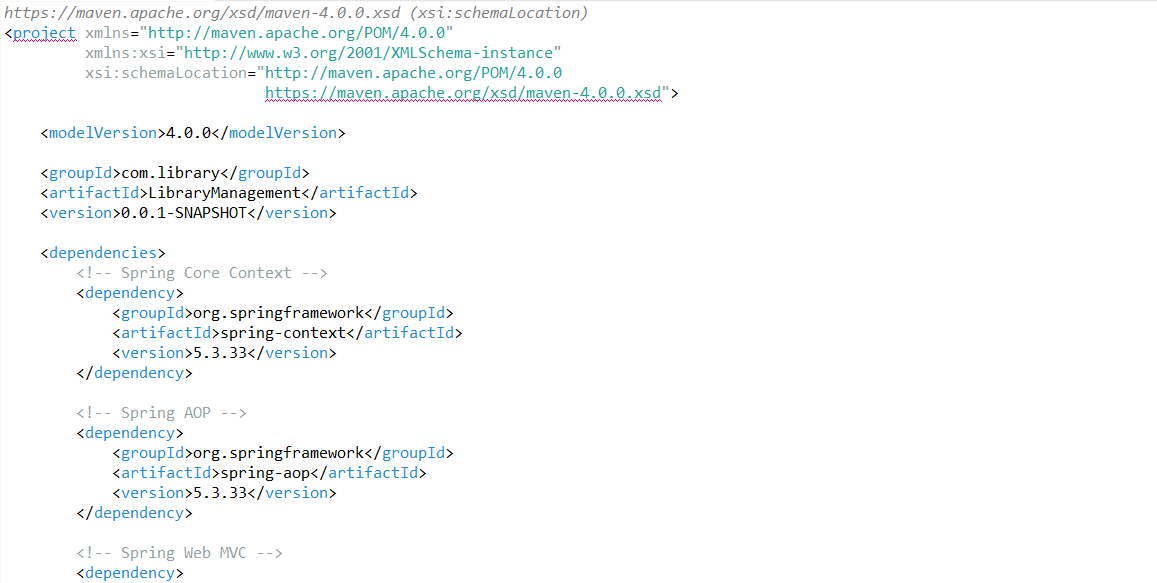
You need to set up a new Maven project for the library management application and add Spring dependencies.

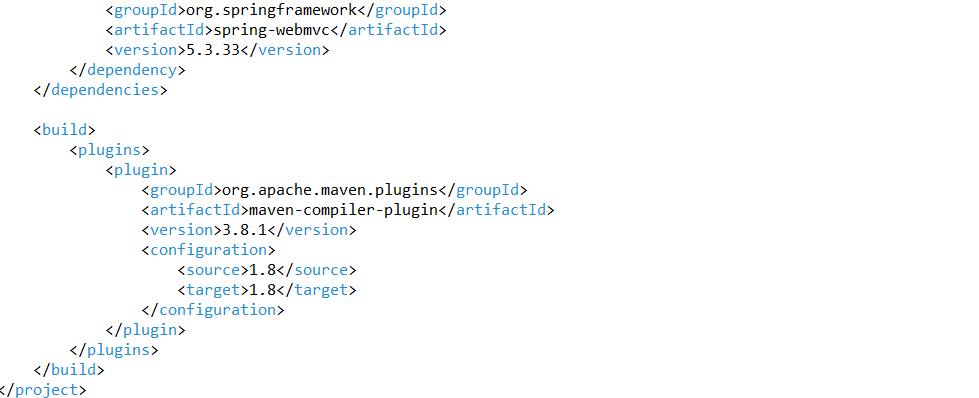
**Step 1: Create a New Maven Project**

* This step was already done in Exercise 1. We are continuing in the same project: LibraryManagement.

**Step 2: Add Spring Dependencies**

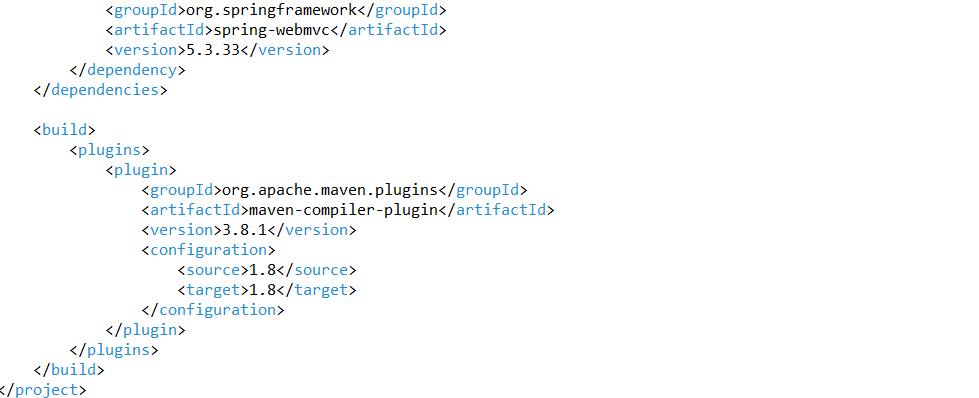
* You need to **add two more Spring dependencies** to your existing <dependencies> block:





**Step 3: Add Maven Compiler Plugin**

* Scroll **below the <dependencies> section** and add the <build> block like this:



**Step 4: Update Maven Project**

* After saving the pom.xml, update your project in Eclipse:

1. Right-click the project → Maven → Update Project
2. Click OK