**WEEK – 03**

3. Spring Core and Maven

**Superset ID: 6262264**

**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.

**Solution :**

***MainApp.class***

*package com.library;*

*import com.library.service.BookService;*

*import org.springframework.context.ApplicationContext;*

*import org.springframework.context.support.ClassPathXmlApplicationContext;*

*public class MainApp {*

*public MainApp() {*

*}*

*public static void main(String[] args) {*

*ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");*

*BookService bookService = (BookService)context.getBean("bookService");*

*bookService.addBook("The Alchemist");*

*}*

*}*

***BookRepository.class***

*package com.library.repository;*

*public class BookRepository {*

*public BookRepository() {*

*}*

*public void saveBook(String title) {*

*System.out.println("Book saved: " + title);*

*}*

*}*

***BookService.class***

*package com.library.service;*

*import com.library.repository.BookRepository;*

*public class BookService {*

*private BookRepository bookRepository;*

*public BookService() {*

*}*

*public void setBookRepository(BookRepository bookRepository) {*

*this.bookRepository = bookRepository;*

*}*

*public void addBook(String title) {*

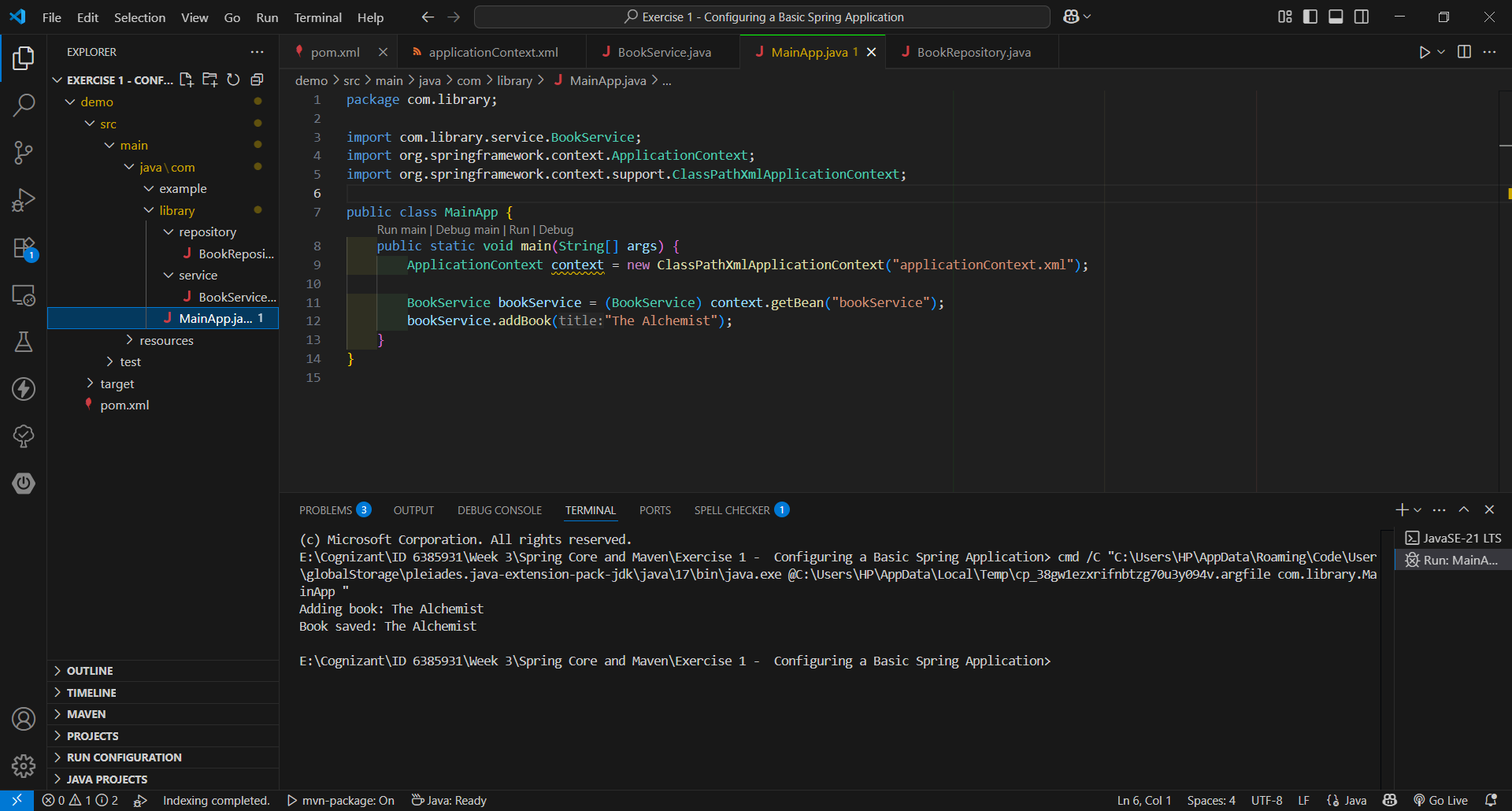
*System.out.println("Adding book: " + title);*

*this.bookRepository.saveBook(title);*

*}*

*}*

***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.

**SOLUTION:**

***BookRepository.class:***

*package com.library.repository;*

*public class BookRepository {*

*public void fetchBook() {*

*System.out.println("Fetching book from repository...");*

*}*

*}*

***BookService.class:***

*package com.library.service;*

*import com.library.repository.BookRepository;*

*public class BookService {*

*private BookRepository bookRepository;*

*// Setter method for Dependency Injection*

*public void setBookRepository(BookRepository bookRepository) {*

*this.bookRepository = bookRepository;*

*}*

*public void displayBook() {*

*System.out.println("Inside BookService...");*

*bookRepository.fetchBook();*

*}*

*}*

***LibraryManagementApplication.class:***

*package com.library;*

*import com.library.service.BookService;*

*import org.springframework.context.ApplicationContext;*

*import org.springframework.context.support.ClassPathXmlApplicationContext;*

*public class LibraryManagementApplication {*

*public static void main(String[] args) {*

*ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");*

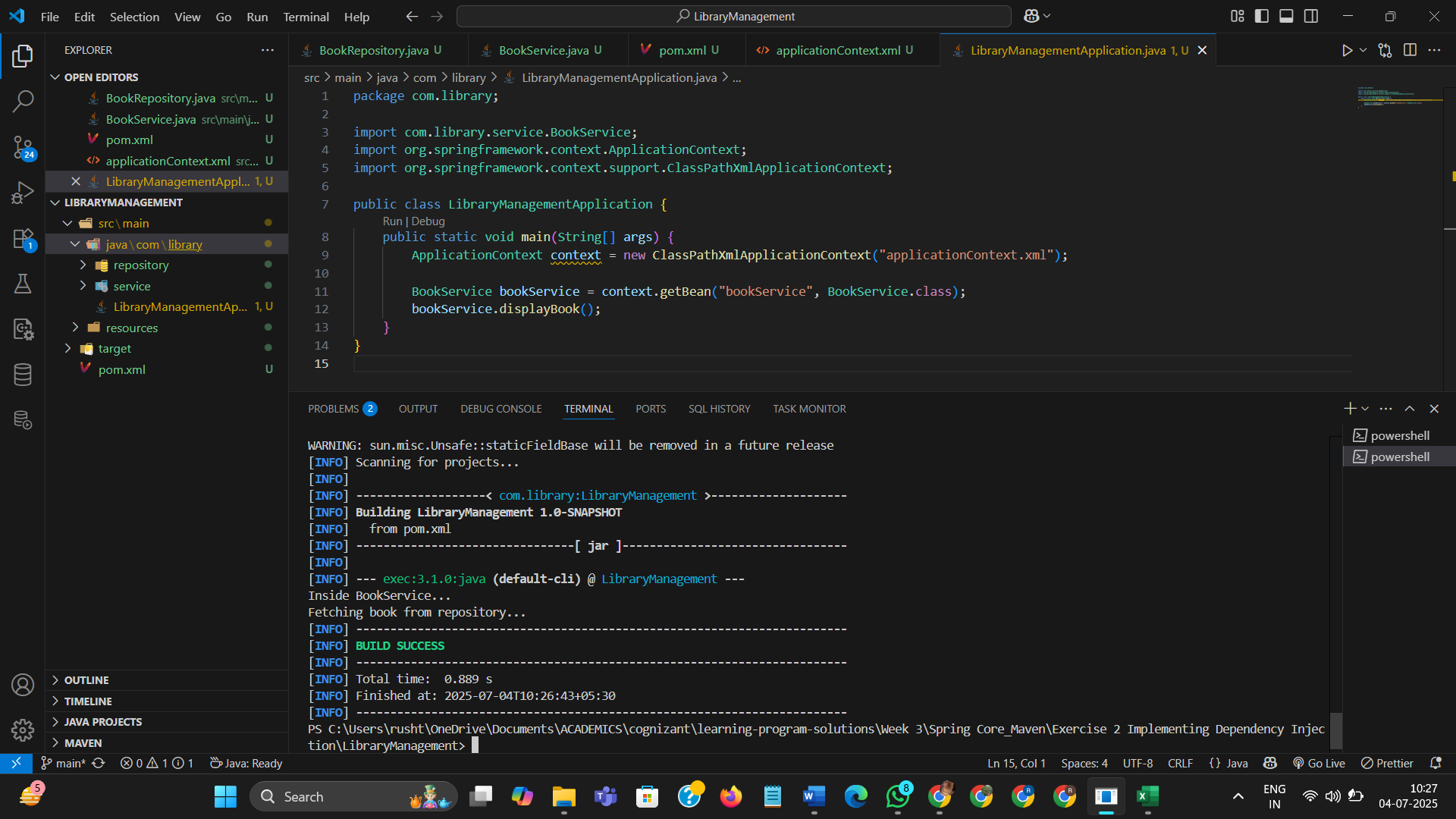
*BookService bookService = context.getBean("bookService", BookService.class);*

*bookService.displayBook();*

*}*

*}*

***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.

**SOLUTION:**

***BookService.class***

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

   private BookRepository bookRepository;

   public BookService() {

   }

   public void setBookRepository(BookRepository bookRepository) {

      this.bookRepository = bookRepository;

   }

   public void displayBook() {

      System.out.println("Inside BookService...");

      this.bookRepository.fetchBook();

   }

}

***BookRepository.class***

*package com.library.repository;*

*public class BookRepository {*

*public BookRepository() {*

*}*

*public void fetchBook() {*

*System.out.println("Fetching book from repository...");*

*}*

*}*

***LibraryManagementApplication.class:***

*package com.library;*

*import com.library.service.BookService;*

*import org.springframework.context.ApplicationContext;*

*import org.springframework.context.support.ClassPathXmlApplicationContext;*

*public class LibraryManagementApplication {*

*public LibraryManagementApplication() {*

*}*

*public static void main(String[] args) {*

*ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");*

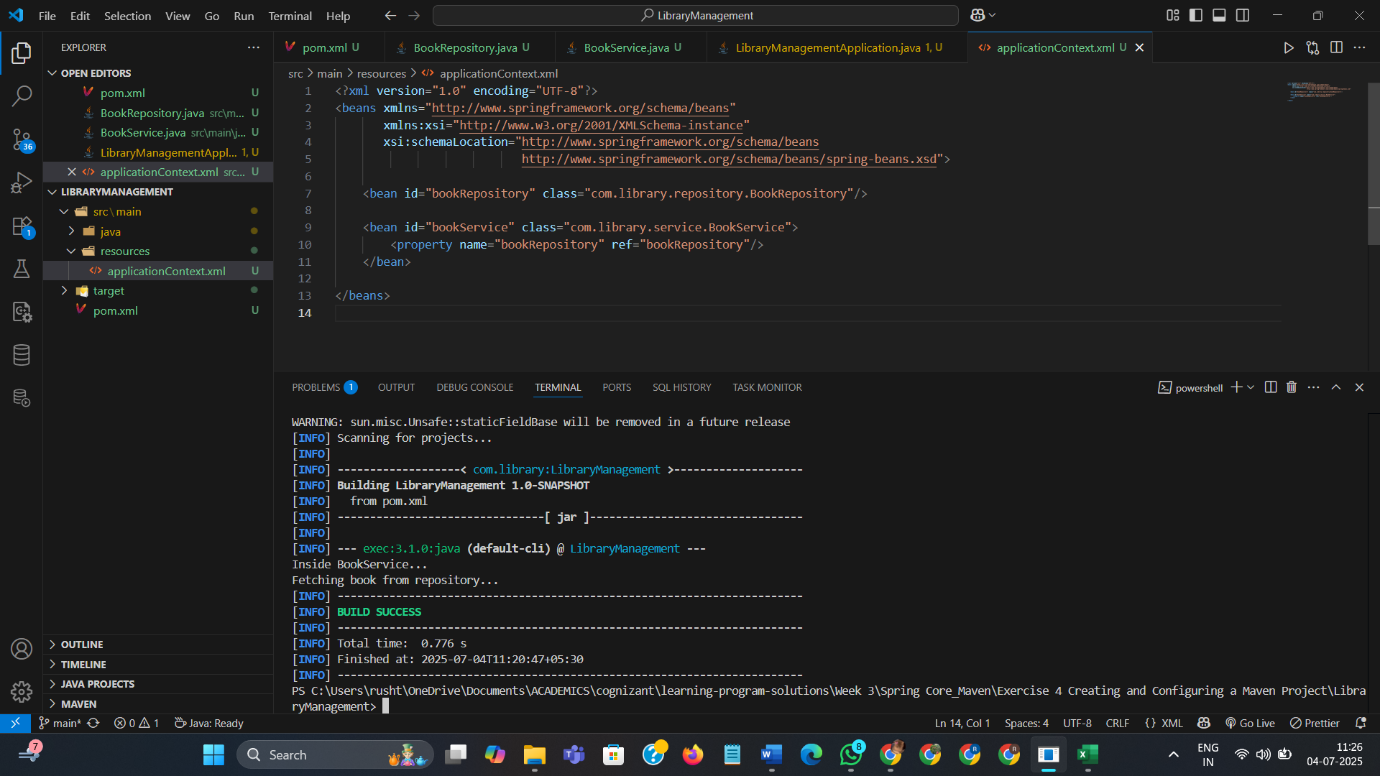
*BookService bookService = (BookService)context.getBean("bookService", BookService.class);*

*bookService.displayBook();*

*}*

*}*

***OUTPUT:***

**