**WEEK – 3**

**SPRING CORE AND MAVEN**

**HANDS - ON**

**CODE:-**

1. **applicationContext.xml:-**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<!-- Repository Bean -->

<bean id="bookRepository" class="com.library.repository.BookRepository"/>

<!-- Service Bean with DI using setter method -->

<bean id="bookService" class="com.library.service.BookService">

<property name="bookRepository" ref="bookRepository"/>

</bean>

</beans>

1. **BookRepository.java:-**

package com.library.repository;

public class BookRepository {

public void saveBook() {

System.out.println("Book saved to the repository.");

}

}

1. **BookService.java:-**

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook() {

System.out.println("Adding book using BookService...");

bookRepository.saveBook();

}

}

1. **MainApp.java:-**

package com.library;

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

public static void main(String[] args) {

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

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

bookService.addBook();

}

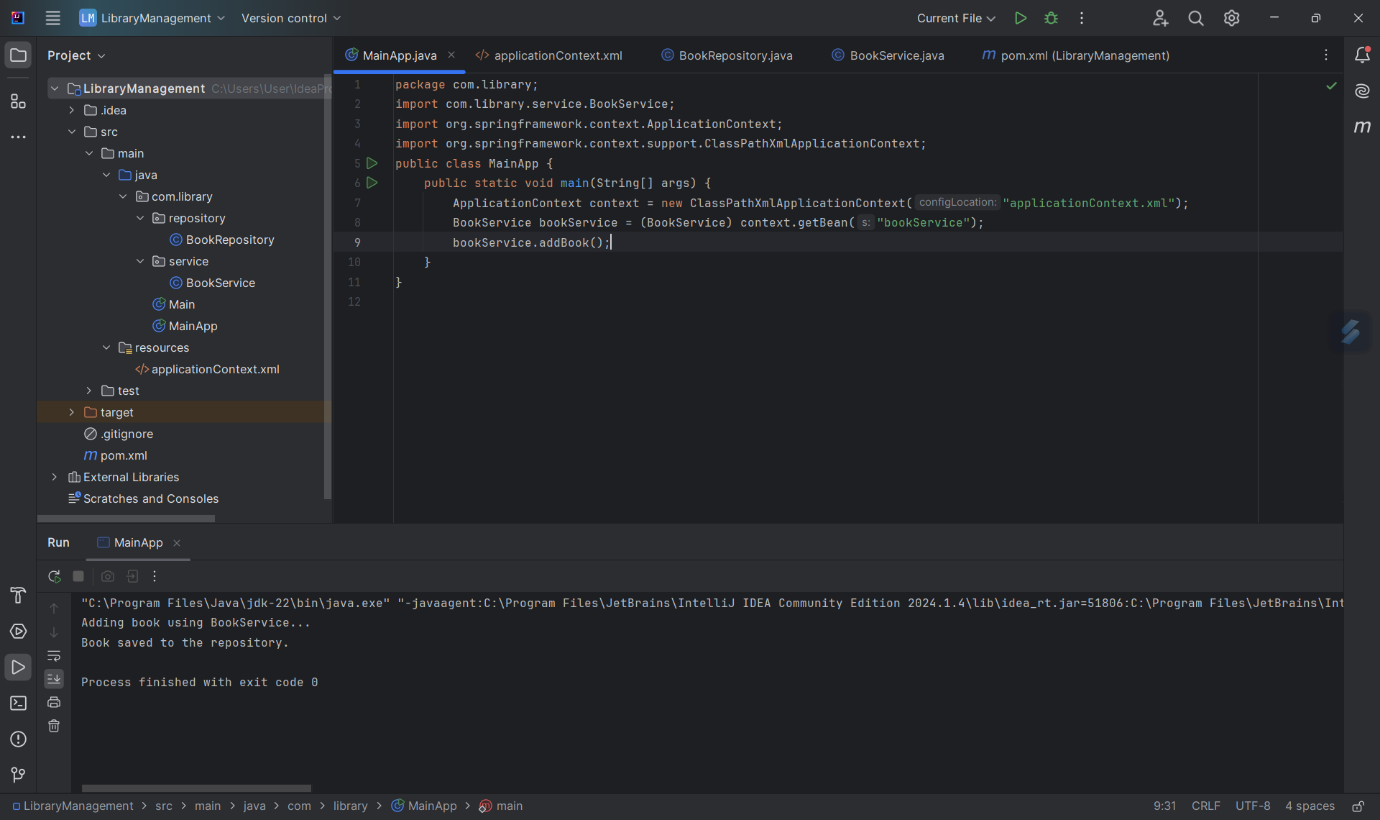
}

**Q. How is this differ from exercise 1:-**

🡪 **Exercise 1**: Here we learn how to build a Spring project and define components (beans).

**Exercise 2**: Now after we have the components, here we see how Spring injects one into another automatically.

**OUTPUT**

****