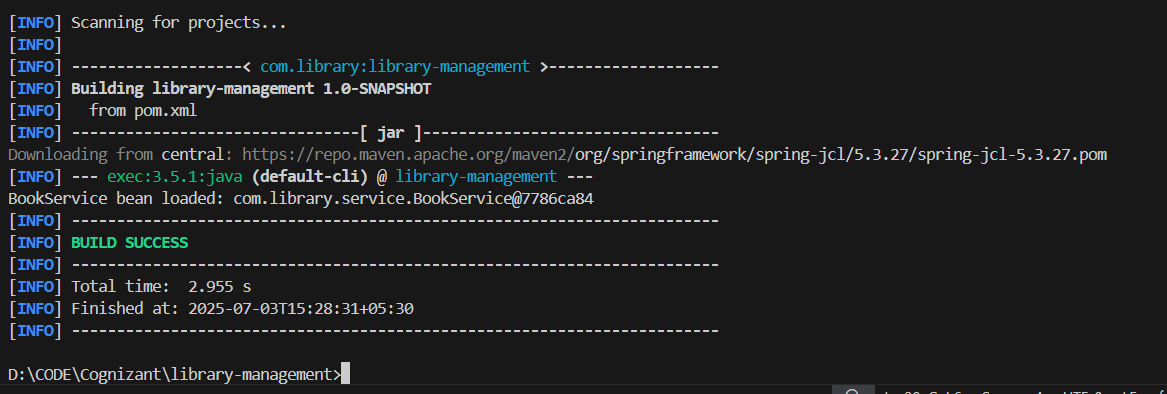
**Spring Core and Maven**

**Exercise 1:**

**Code for this exercise attached separately**

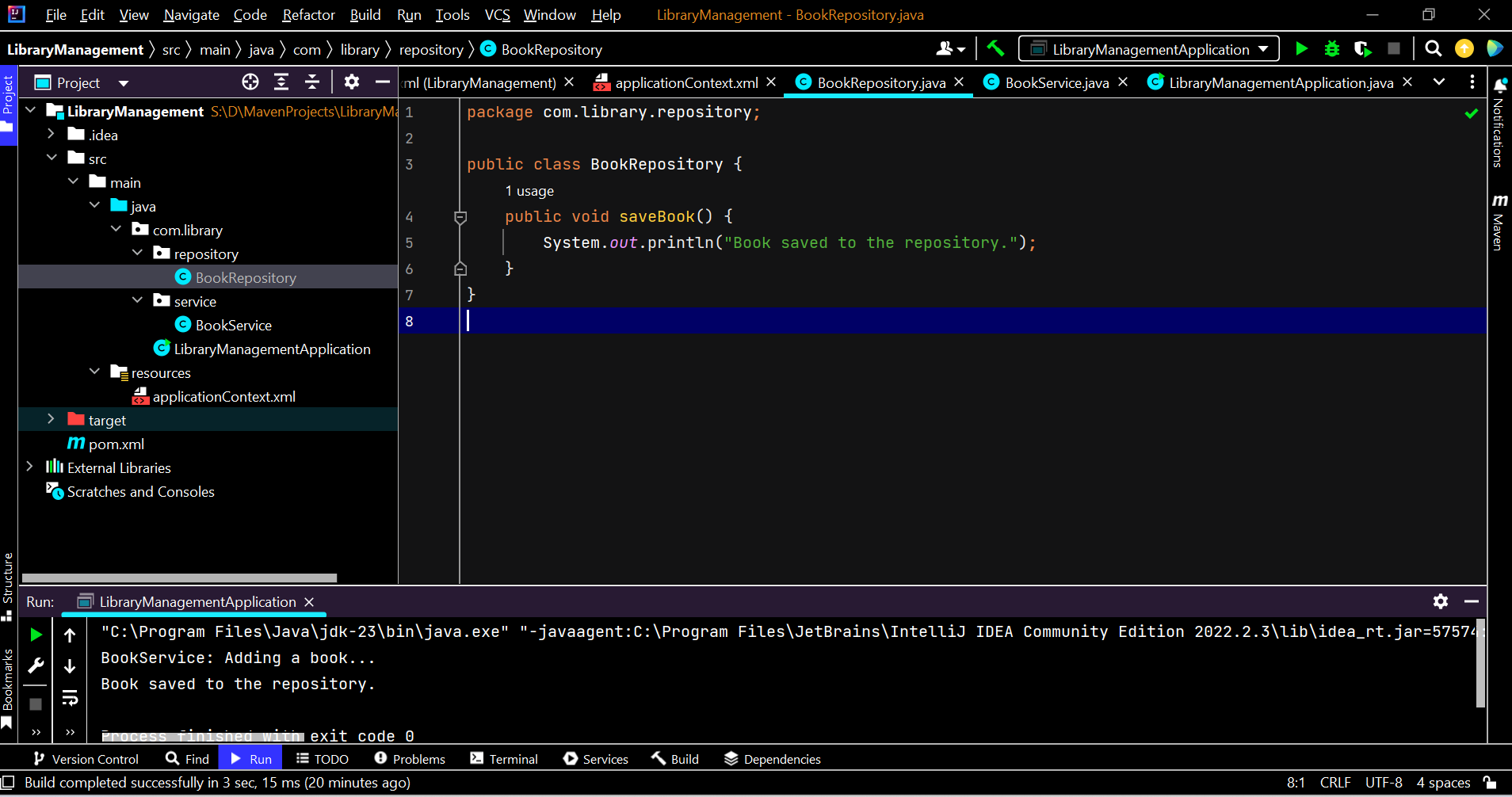
**Output**



**Exercise 2:**

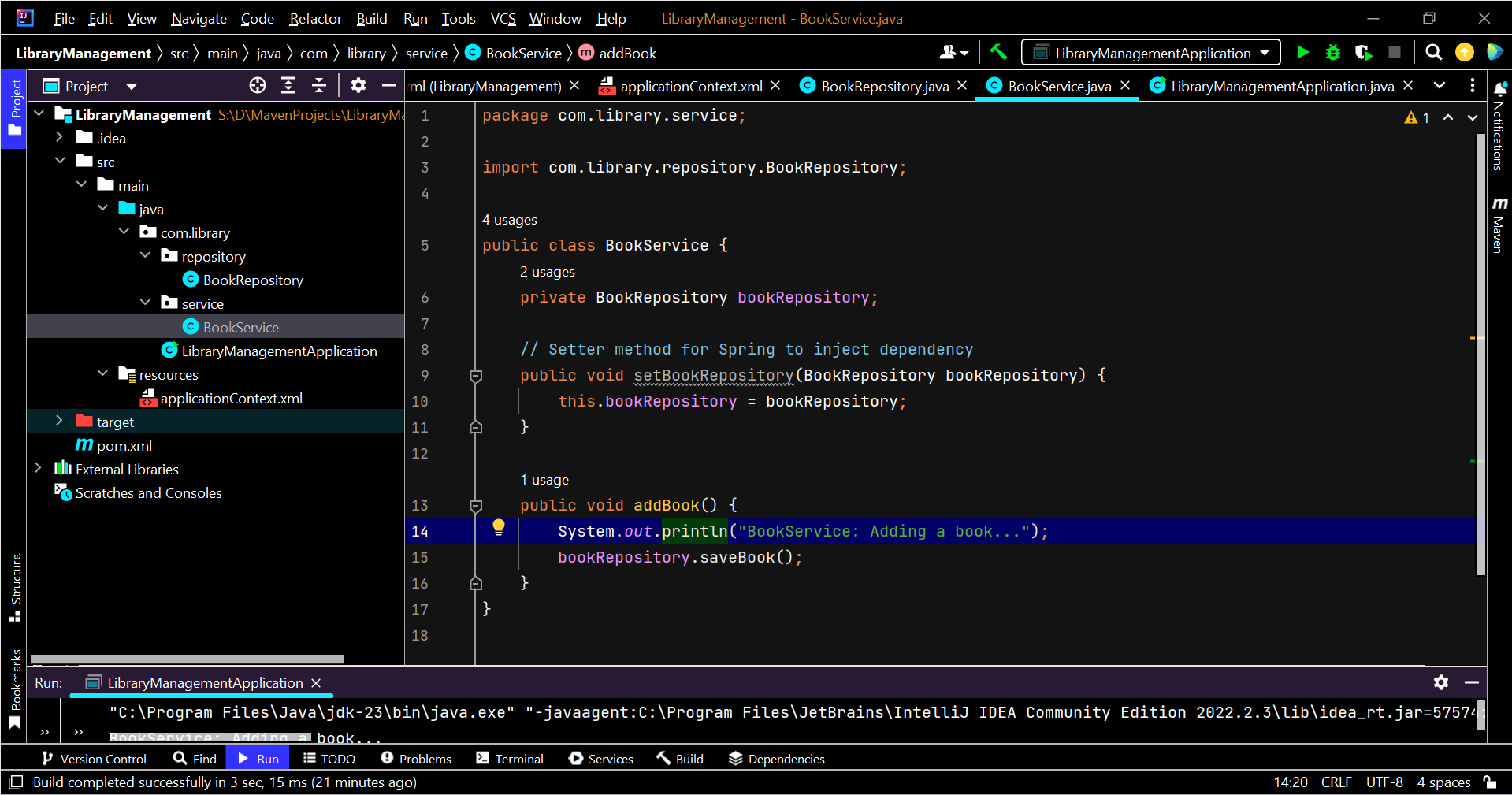
**Codes**

**BookRepository.java**

****

package com.library.repository;  
  
public class BookRepository {  
 public void saveBook() {  
 System.*out*.println("Book saved to the repository.");  
 }  
}

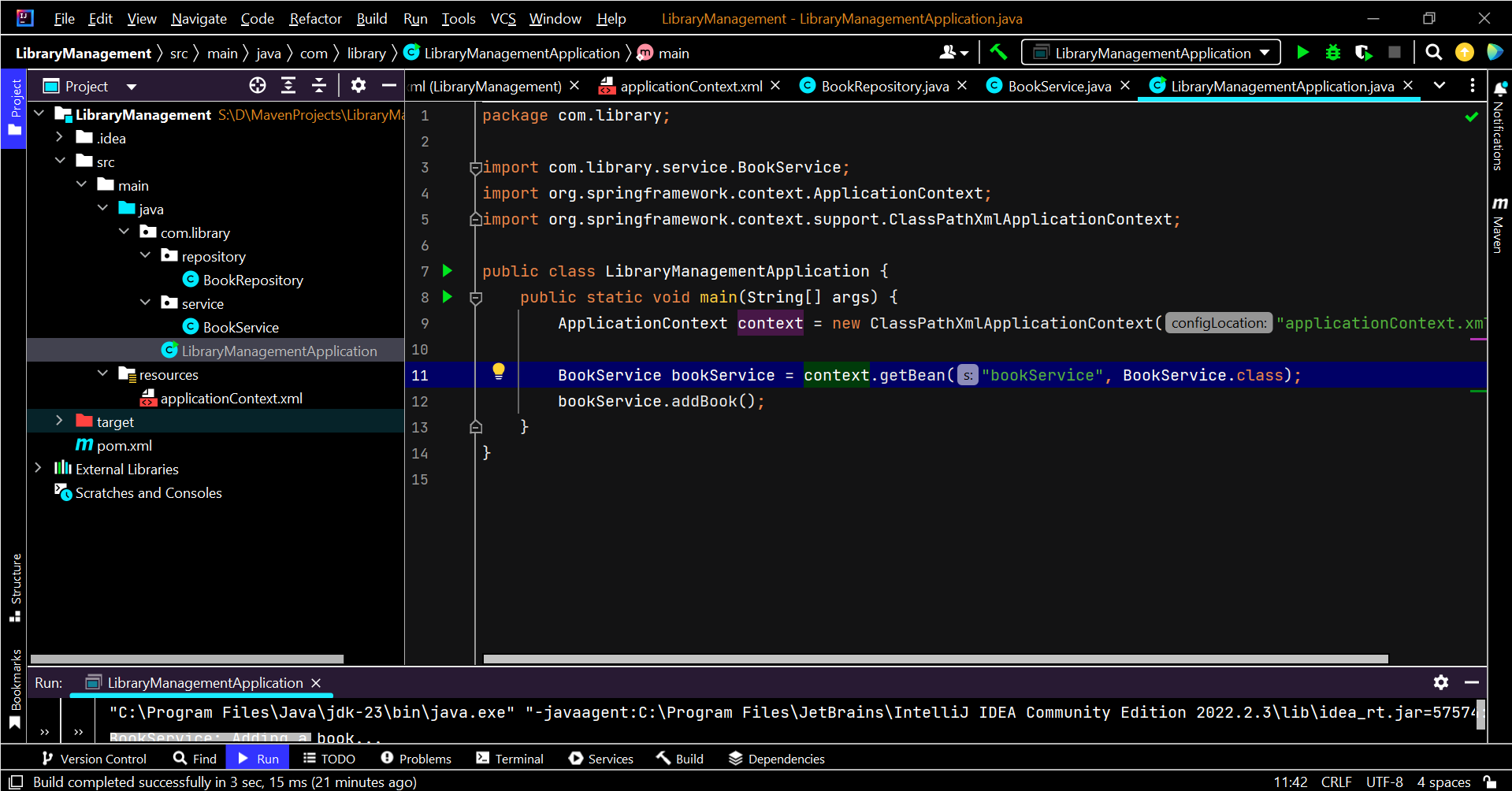
**BookService.java**

****

package com.library.service;  
  
import com.library.repository.BookRepository;  
  
public class BookService {  
 private BookRepository bookRepository;  
  
 // Setter method for Spring to inject dependency  
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 }  
  
 public void addBook() {  
 System.*out*.println("BookService: Adding a book...");  
 bookRepository.saveBook();  
 }  
}

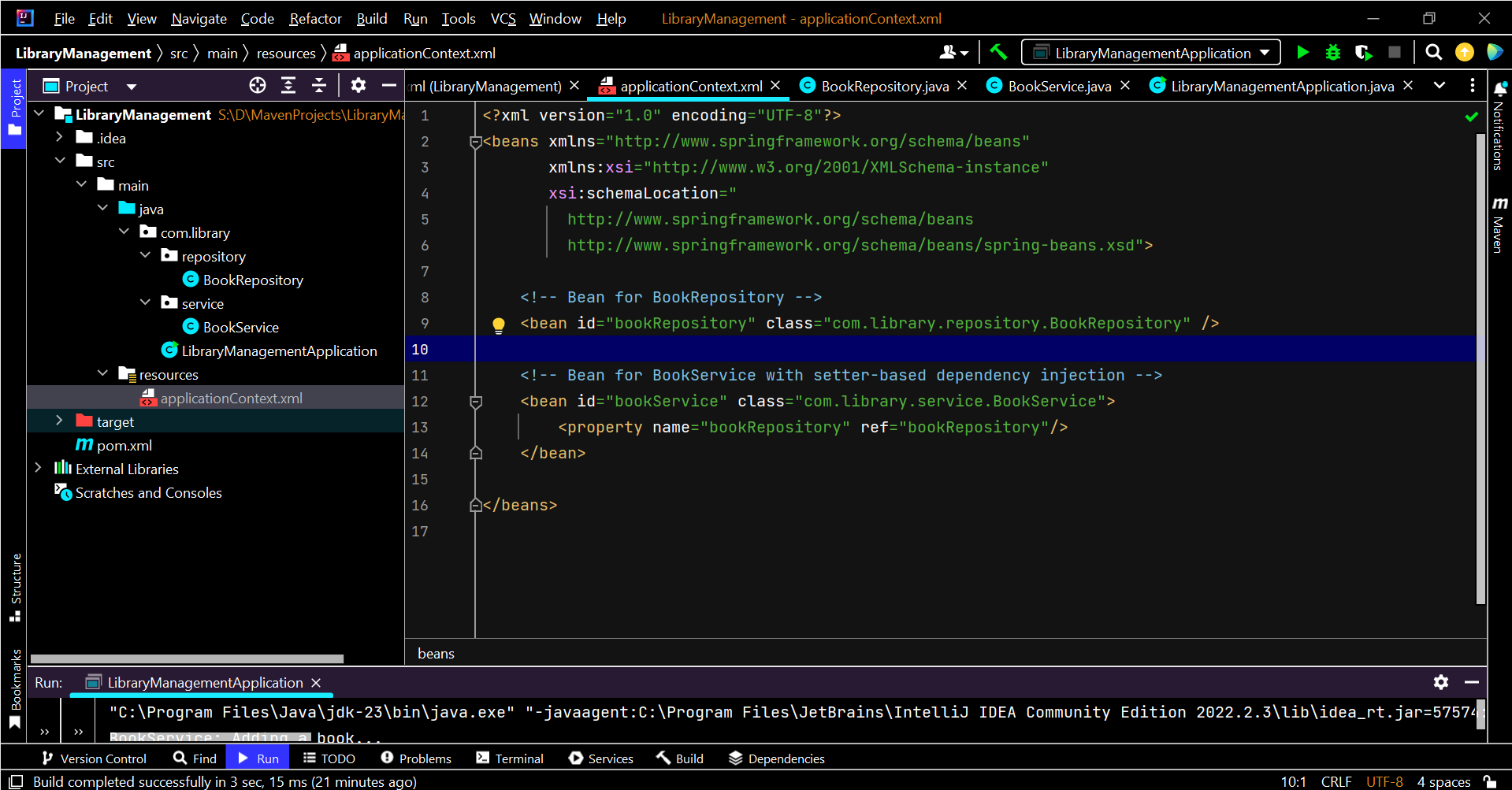
**LibraryManagementApplication.java**

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.addBook()**;** }  
}

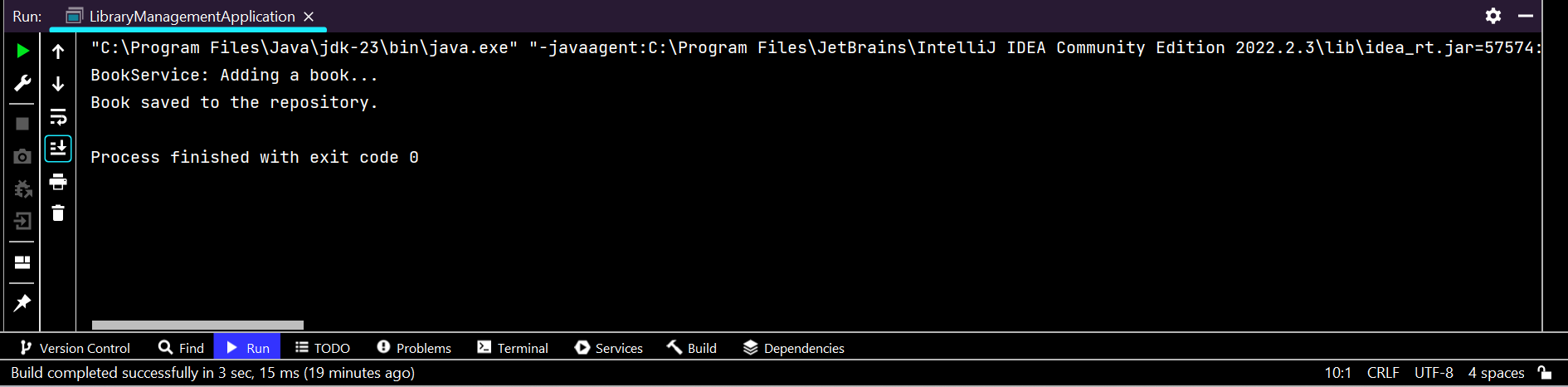
****

**ApplicationContext.xml**

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.addBook()**;** }  
}

****

**Output**

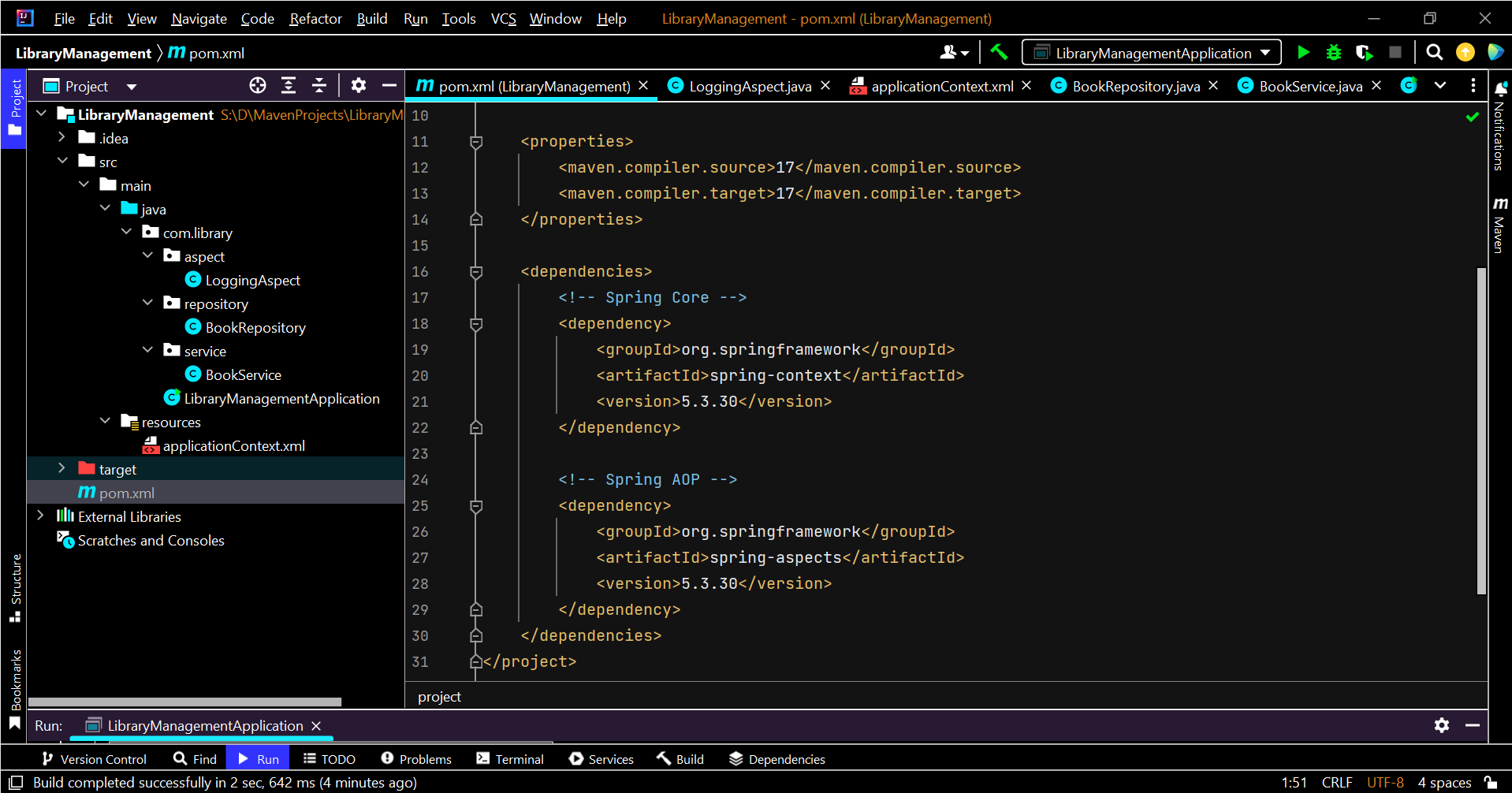
****

**Exercise 3:**

**Codes**

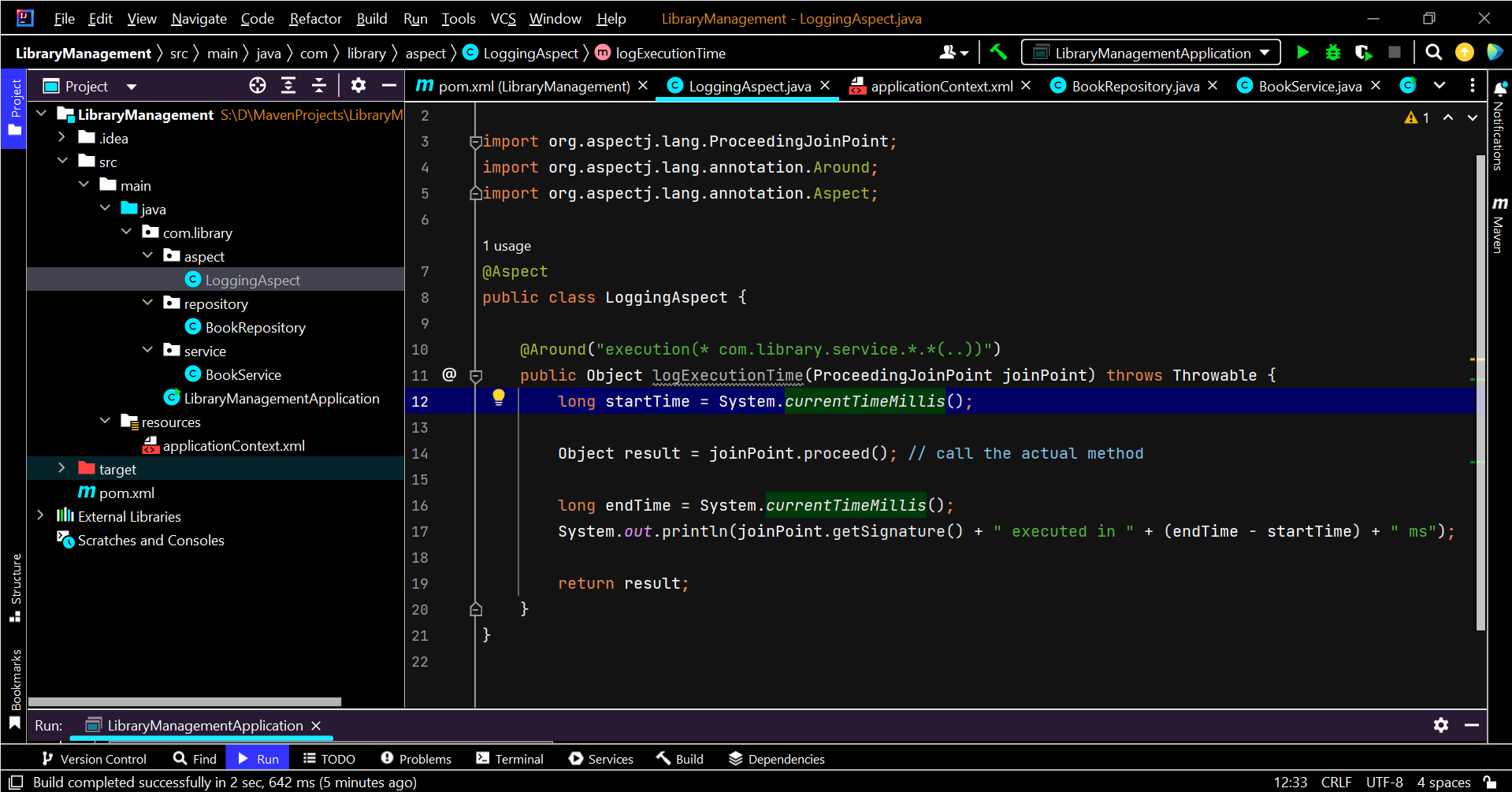
**Added section in pom.xml**

<dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aspects</artifactId>  
 <version>5.3.30</version>  
</dependency>



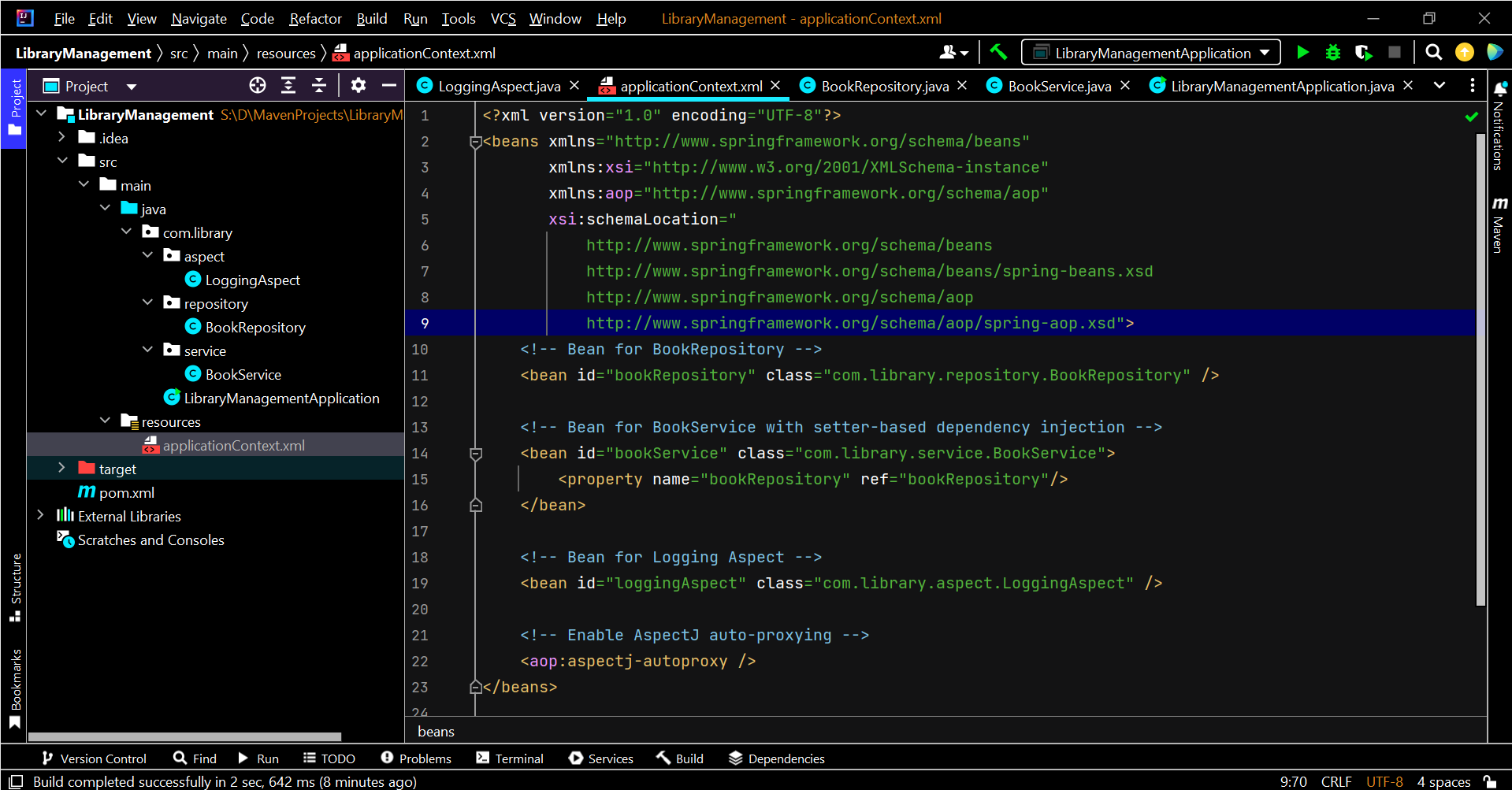
**LoggingAspect.java**

package com.library.aspect**;**import org.aspectj.lang.ProceedingJoinPoint**;**import org.aspectj.lang.annotation.Around**;**import org.aspectj.lang.annotation.Aspect**;**@Aspect  
public class LoggingAspect {  
  
 @Around("execution(\* com.library.service.\*.\*(..))")  
 public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {  
 long startTime = System.*currentTimeMillis*()**;** Object result = joinPoint.proceed()**;** // call the actual method  
  
 long endTime = System.*currentTimeMillis*()**;** System.*out*.println(joinPoint.getSignature() + " executed in " + (endTime - startTime) + " ms")**;** return result**;** }  
}



**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"  
 xmlns:aop="http://www.springframework.org/schema/aop"  
 xsi:schemaLocation="  
 http://www.springframework.org/schema/beans  
 http://www.springframework.org/schema/beans/spring-beans.xsd  
 http://www.springframework.org/schema/aop  
 http://www.springframework.org/schema/aop/spring-aop.xsd">  
  
 <!-- Bean for BookRepository -->  
 <bean id="bookRepository" class="com.library.repository.BookRepository" />  
  
 <!-- Bean for BookService with setter-based dependency injection -->  
 <bean id="bookService" class="com.library.service.BookService">  
 <property name="bookRepository" ref="bookRepository"/>  
 </bean>  
  
 <!-- Bean for Logging Aspect -->  
 <bean id="loggingAspect" class="com.library.aspect.LoggingAspect" />  
  
 <!-- Enable AspectJ auto-proxying -->  
 <aop:aspectj-autoproxy />  
  
</beans>

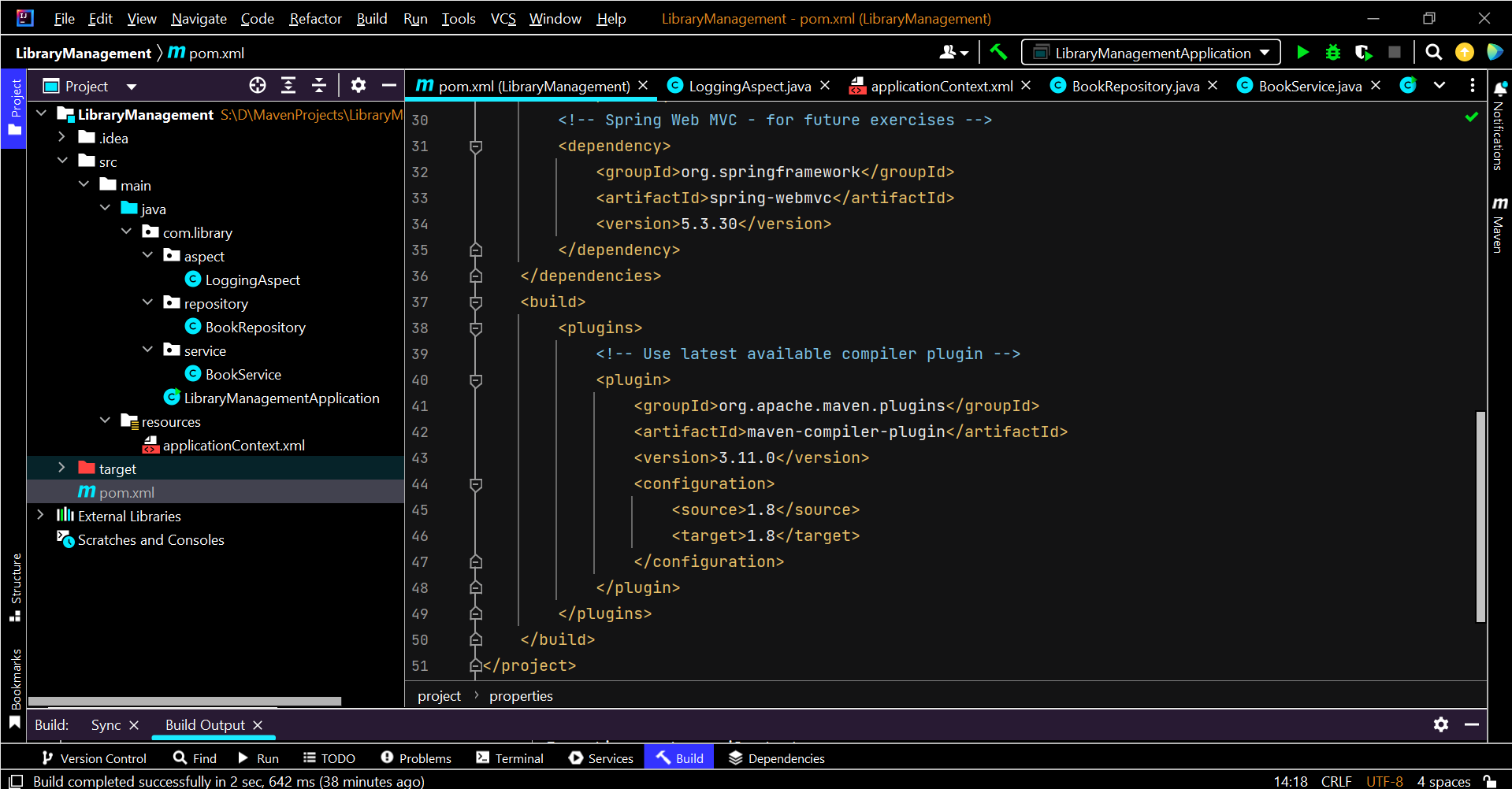


**Exercise 4:**

**Codes**

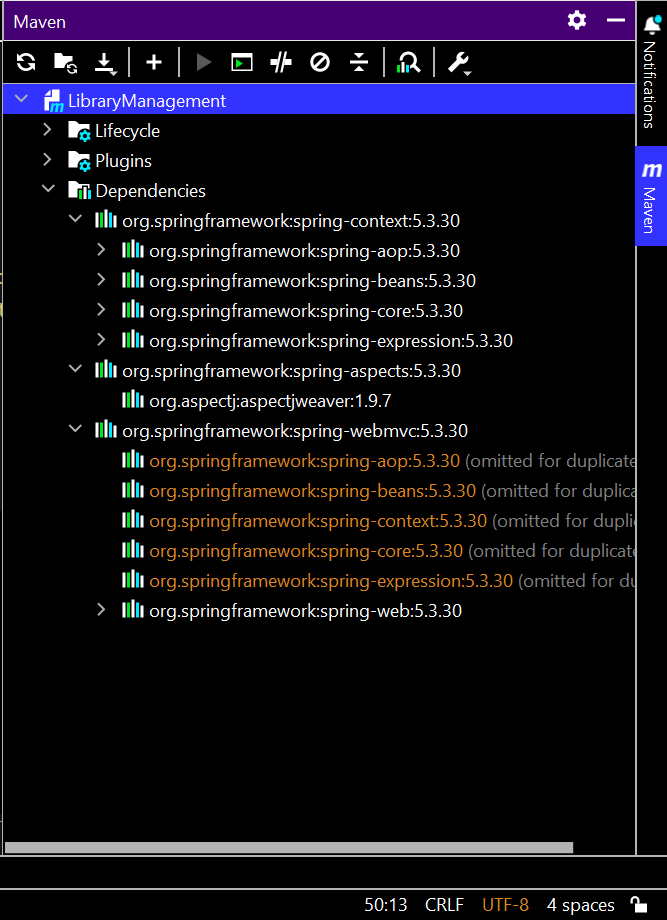
**Changes in pom.xml**

<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>  
  
 <properties>  
 <maven.compiler.source>17</maven.compiler.source>  
 <maven.compiler.target>17</maven.compiler.target>  
 </properties>  
  
 <dependencies>  
 <!-- Spring Core -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.30</version>  
 </dependency>  
  
 <!-- Spring AOP -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aspects</artifactId>  
 <version>5.3.30</version>  
 </dependency>  
 <!-- Spring Web MVC - for future exercises -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>5.3.30</version>  
 </dependency>  
 </dependencies>  
 <build>  
 <plugins>  
 <!-- Use latest available compiler plugin -->  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.11.0</version>  
 <configuration>  
 <source>1.8</source>  
 <target>1.8</target>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
</project>

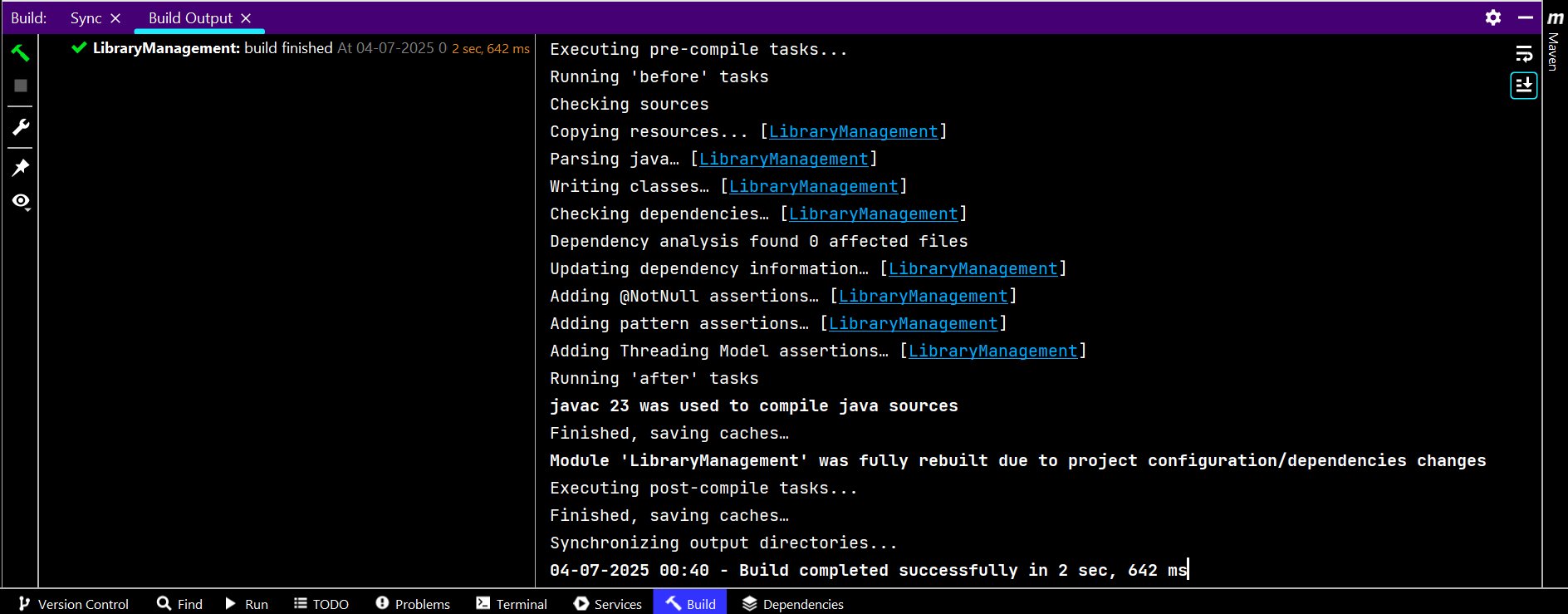


**Output**

**Dependencies included:**

****

**Build Successful**

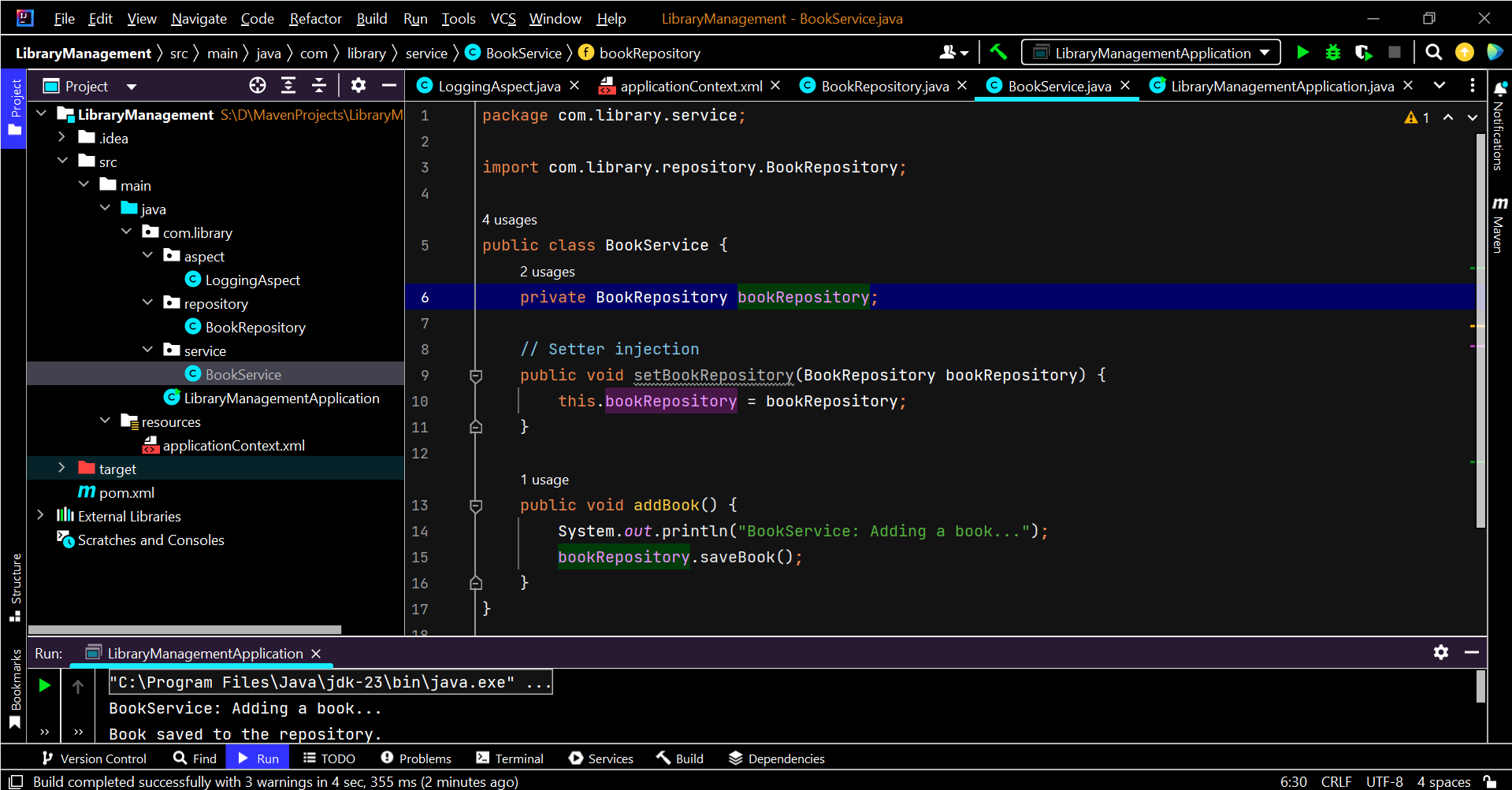
****

**Exercise 5:**

**Codes**

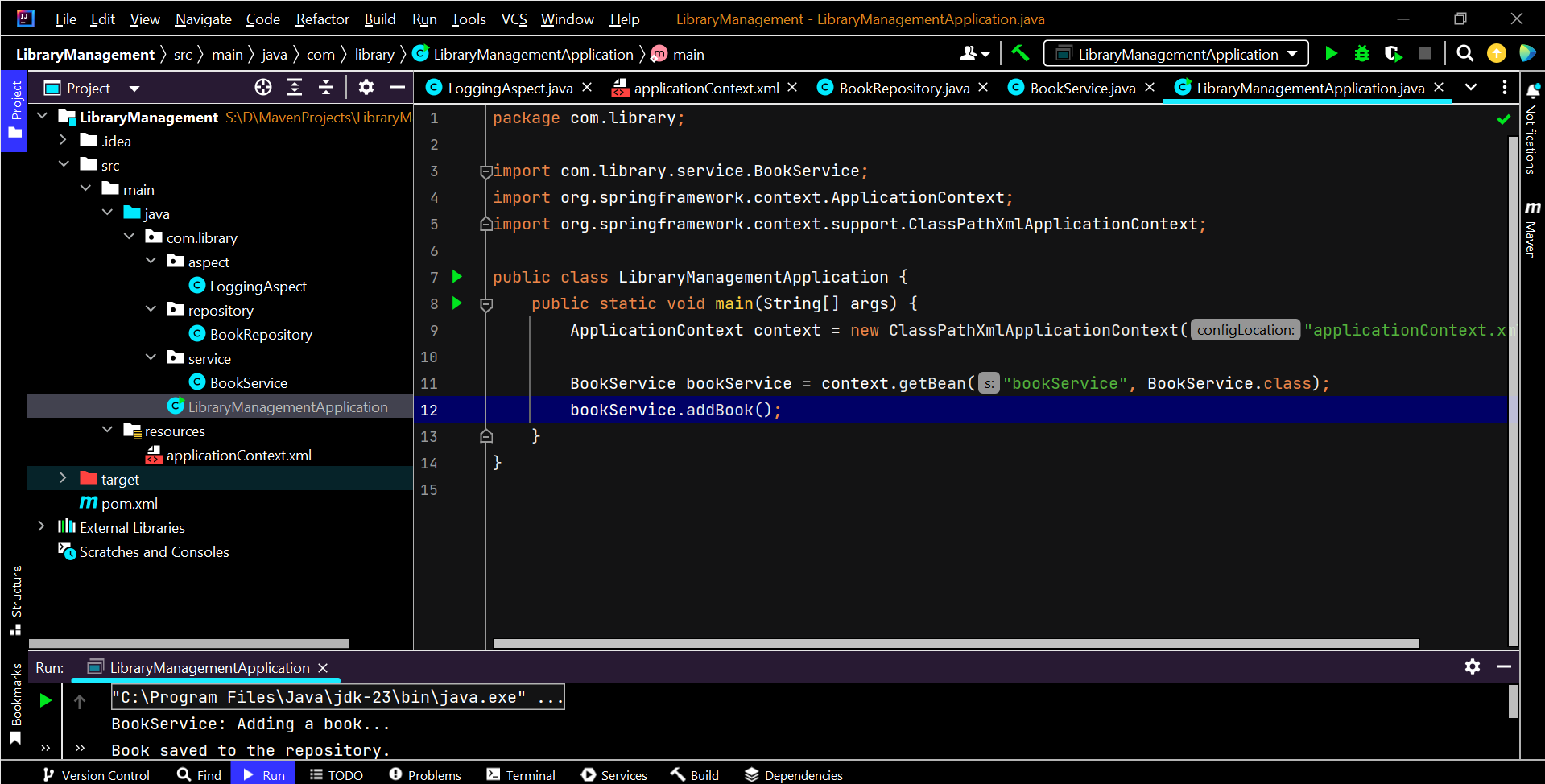
**BookService.java**

package com.library.service**;**import com.library.repository.BookRepository**;**public class BookService {  
 private BookRepository bookRepository**;** // Setter injection  
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository**;** }  
  
 public void addBook() {  
 System.*out*.println("BookService: Adding a book...")**;** bookRepository.saveBook()**;** }  
}



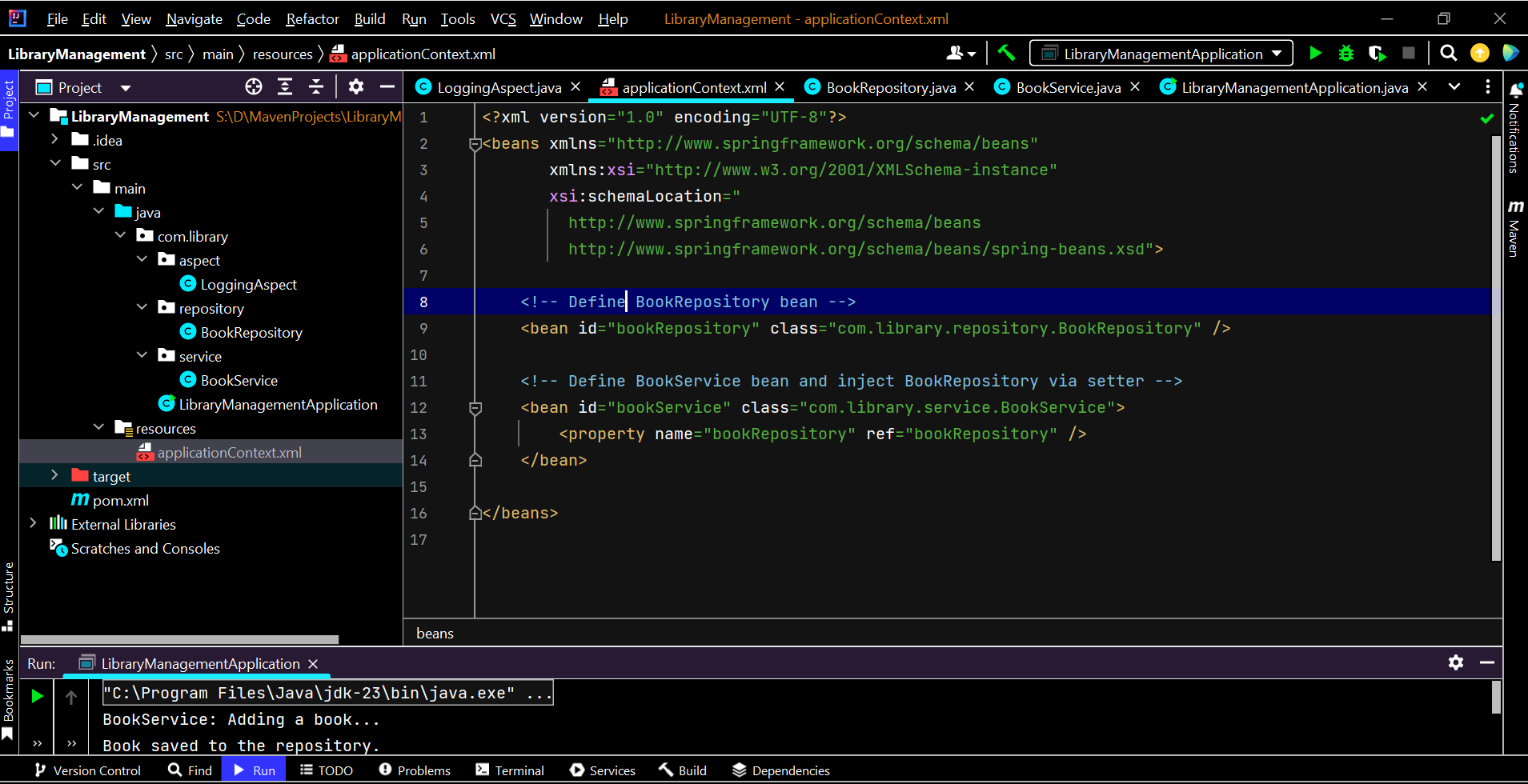
**LibraryManagementApplication.java**

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.addBook()**;** }  
}



**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">  
  
 <!-- Define BookRepository bean -->  
 <bean id="bookRepository" class="com.library.repository.BookRepository" />  
  
 <!-- Define BookService bean and inject BookRepository via setter -->  
 <bean id="bookService" class="com.library.service.BookService">  
 <property name="bookRepository" ref="bookRepository" />  
 </bean>  
  
</beans>



**Output**

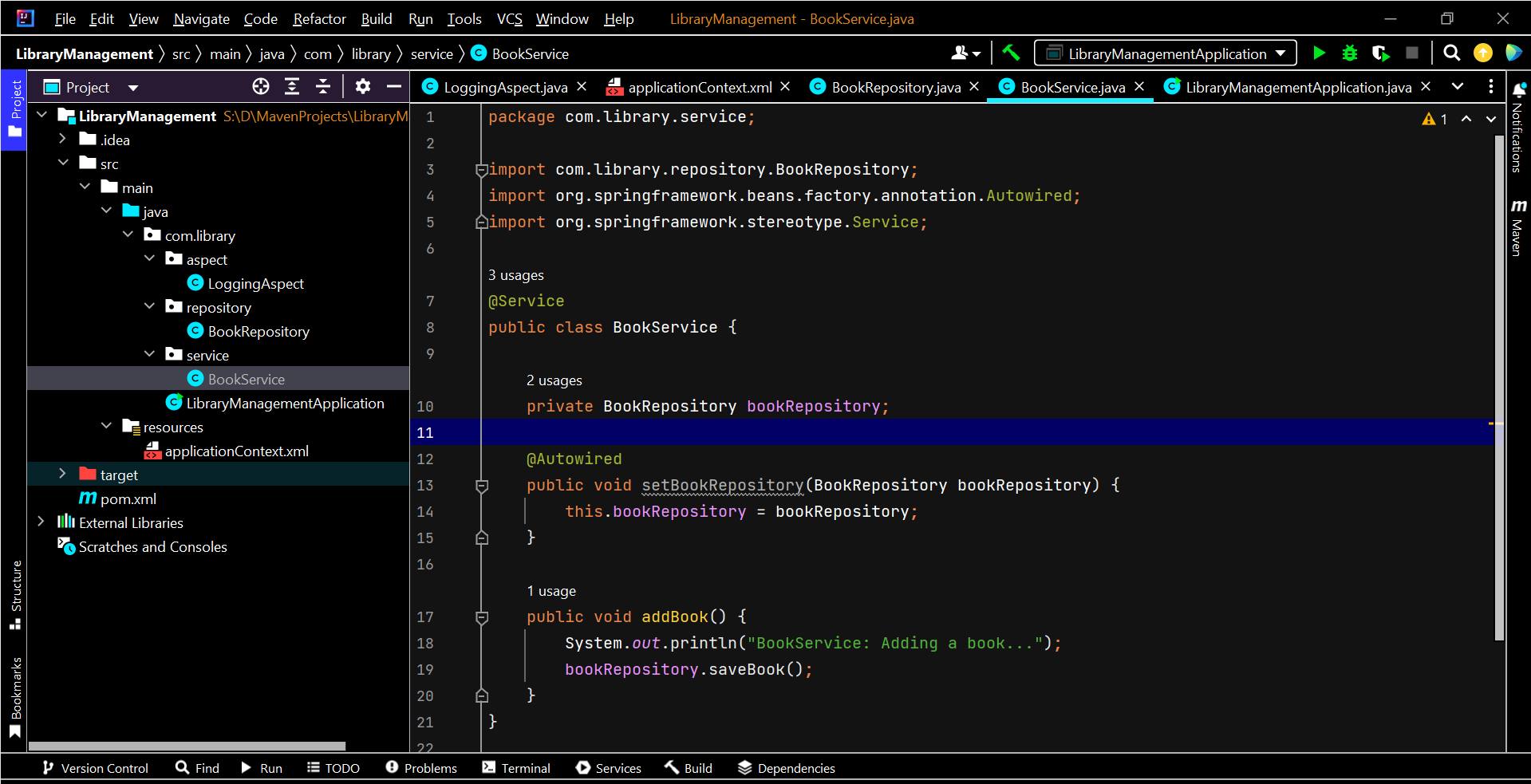
****

**Exercise 6:**

**Codes**

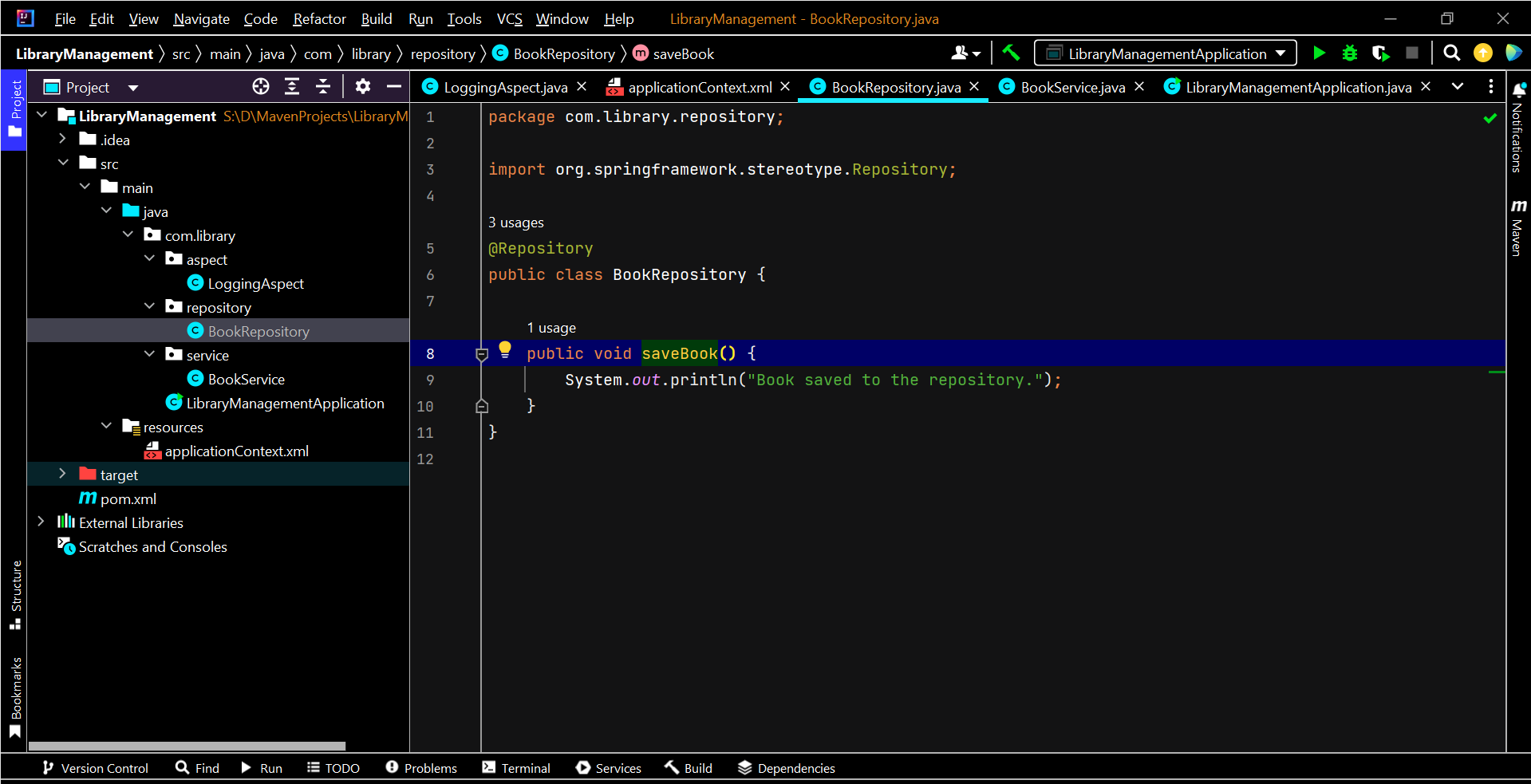
**BookService.java**

package com.library.service**;**import com.library.repository.BookRepository**;**import org.springframework.beans.factory.annotation.Autowired**;**import org.springframework.stereotype.Service**;**@Service  
public class BookService {  
  
 private BookRepository bookRepository**;** @Autowired  
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository**;** }  
  
 public void addBook() {  
 System.*out*.println("BookService: Adding a book...")**;** bookRepository.saveBook()**;** }  
}



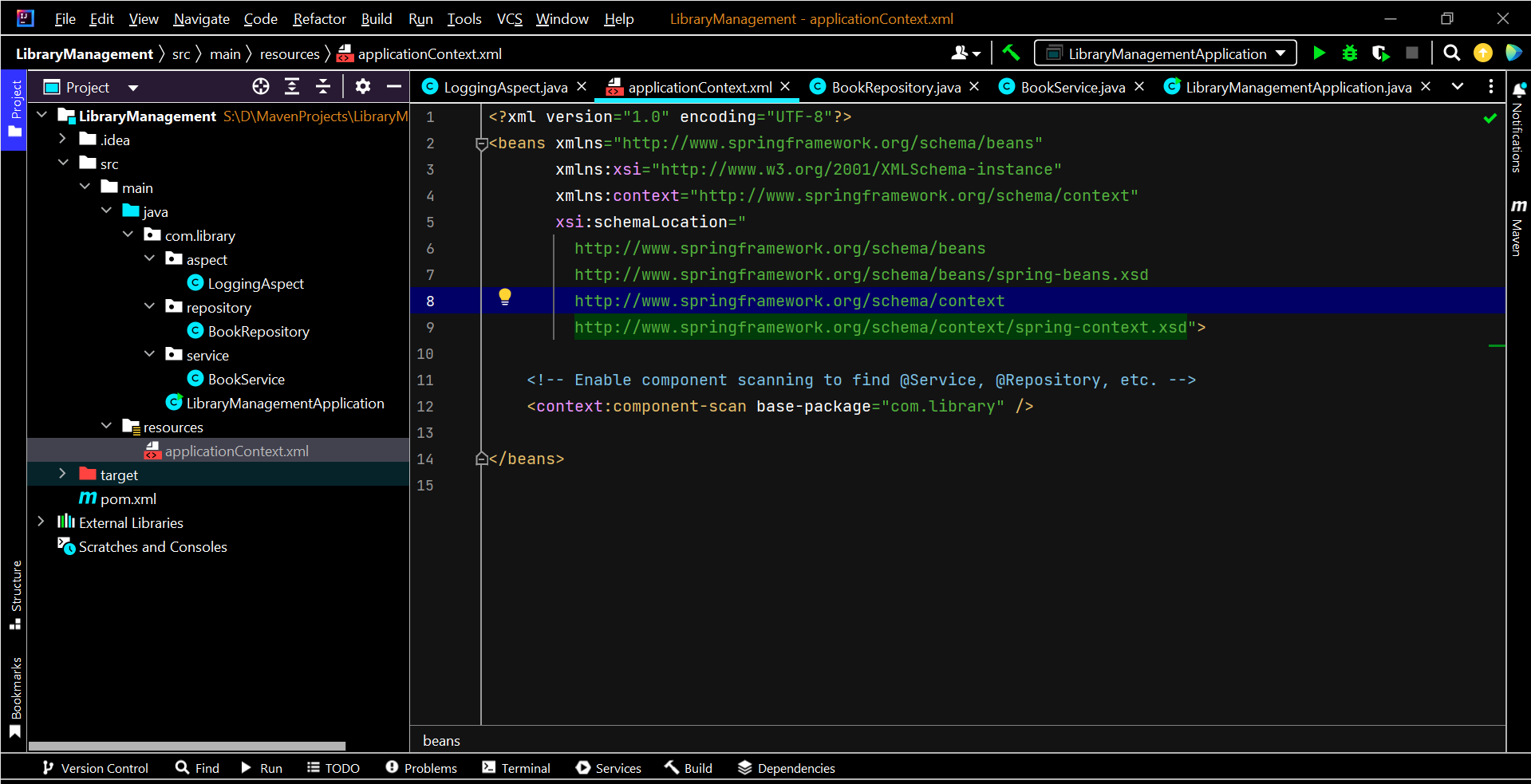
**BookRepository.java**

package com.library.repository**;**import org.springframework.stereotype.Repository**;**@Repository  
public class BookRepository {  
  
 public void saveBook() {  
 System.*out*.println("Book saved to the repository.")**;** }  
}



**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"  
 xmlns:context="http://www.springframework.org/schema/context"  
 xsi:schemaLocation="  
 http://www.springframework.org/schema/beans  
 http://www.springframework.org/schema/beans/spring-beans.xsd  
 http://www.springframework.org/schema/context  
 http://www.springframework.org/schema/context/spring-context.xsd">  
  
 <!-- Enable component scanning to find @Service, @Repository, etc. -->  
 <context:component-scan base-package="com.library" />  
  
</beans>



**Output**

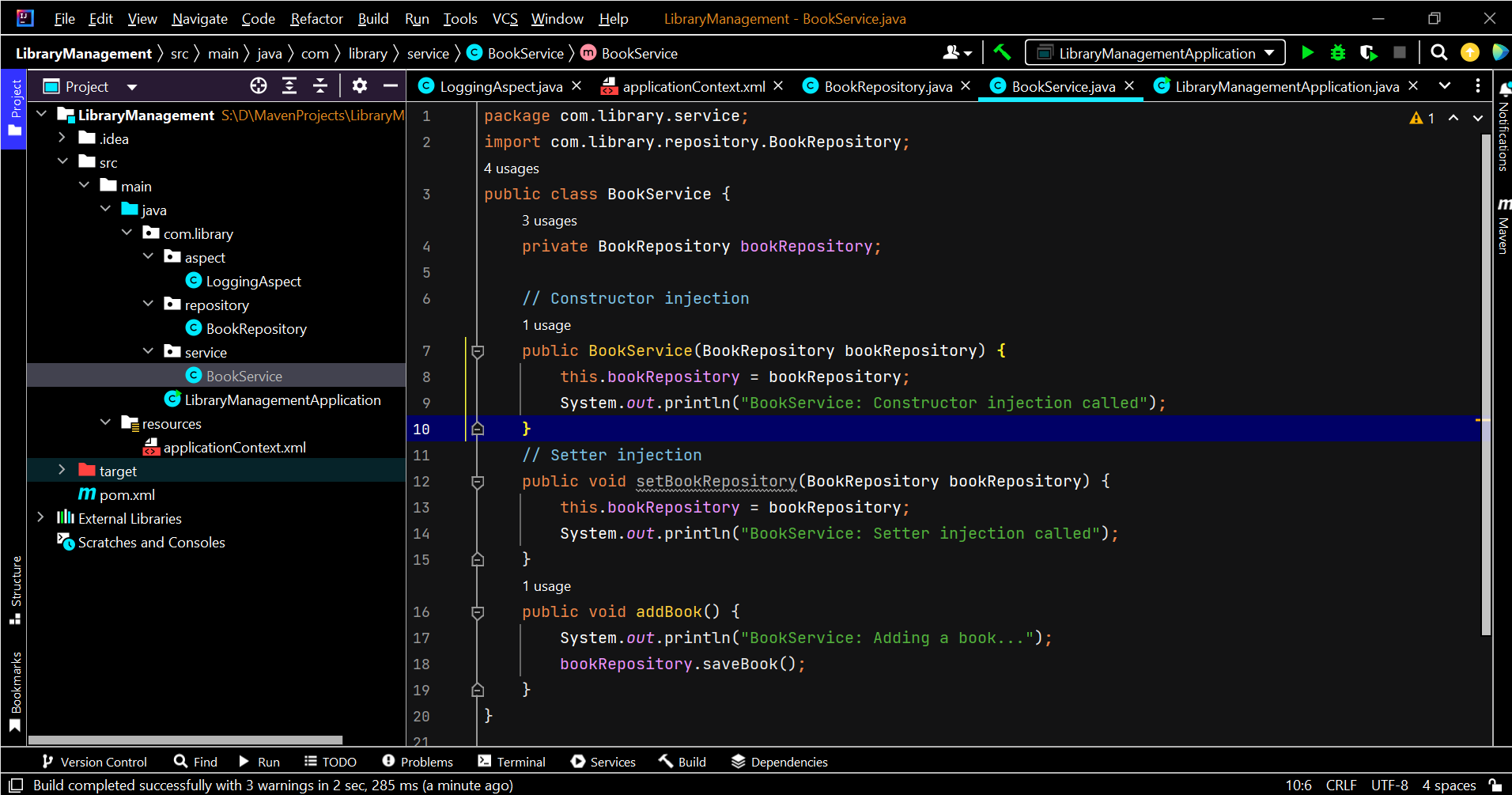
****

**Exercise 7:**

**Codes**

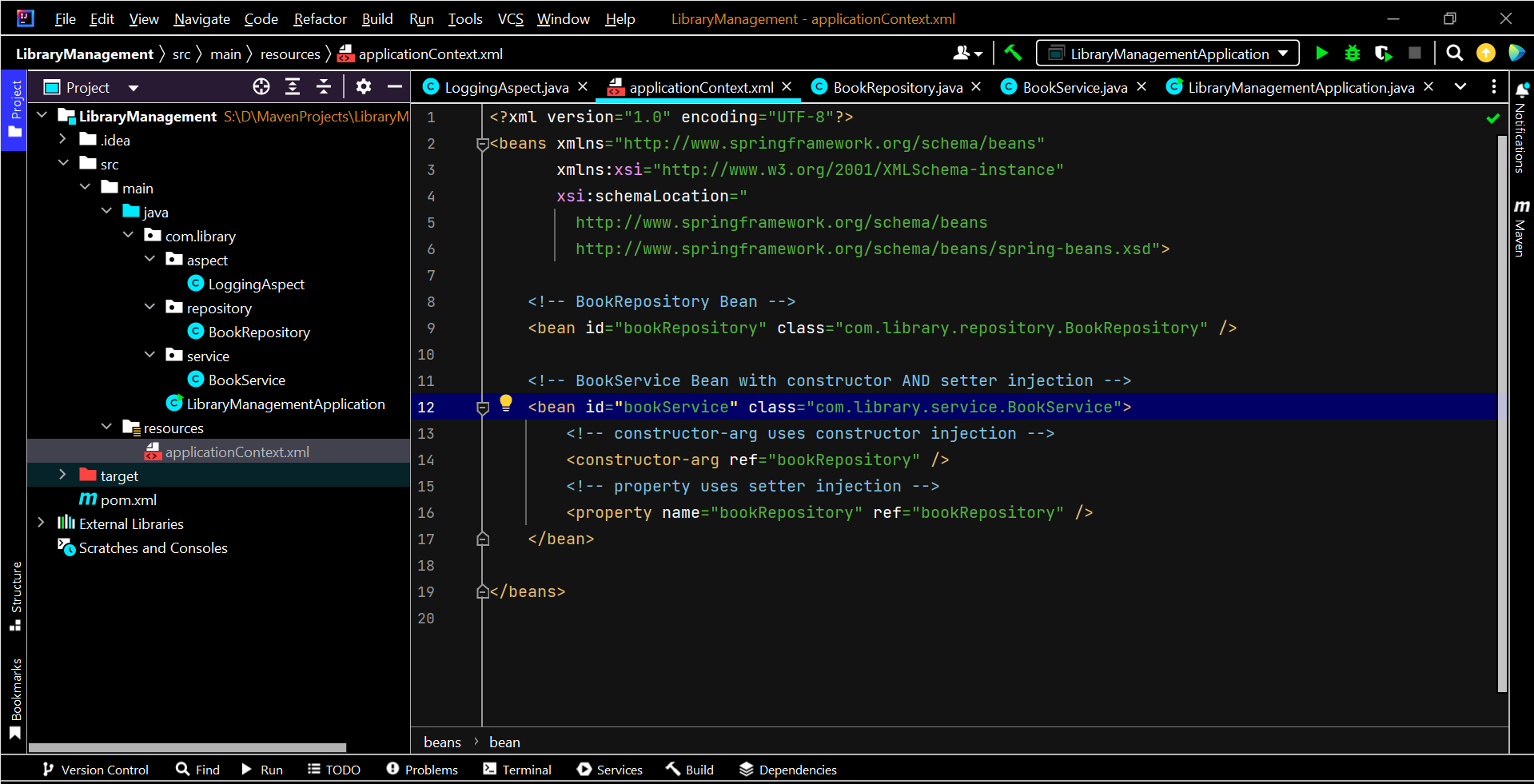
**BookService.java**

package com.library.service**;**import com.library.repository.BookRepository**;**public class BookService {  
  
 private BookRepository bookRepository**;** // Constructor injection  
 public BookService(BookRepository bookRepository) {  
 this.bookRepository = bookRepository**;** System.*out*.println("BookService: Constructor injection called")**;** }  
  
 // Setter injection  
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository**;** System.*out*.println("BookService: Setter injection called")**;** }  
  
 public void addBook() {  
 System.*out*.println("BookService: Adding a book...")**;** bookRepository.saveBook()**;** }  
}

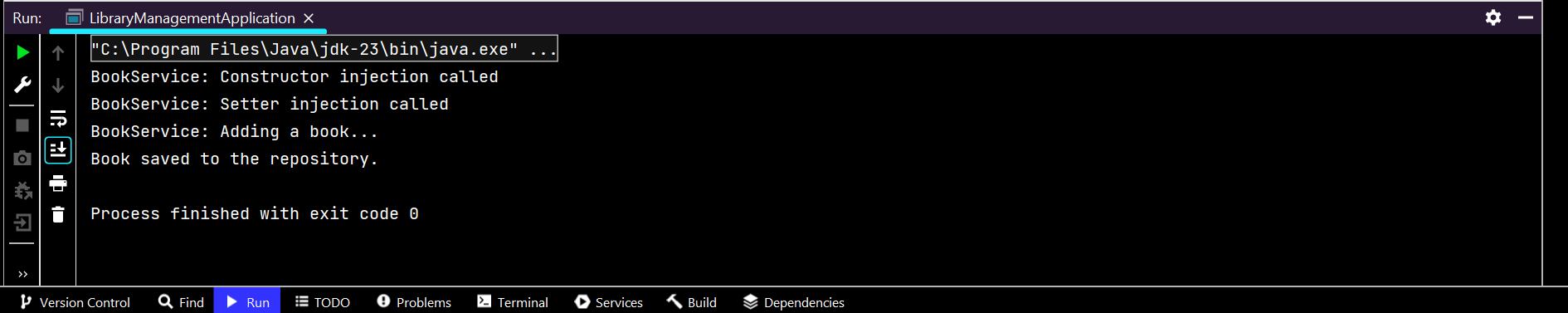


**applicationContext.xml**

<?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">  
  
 <!-- BookRepository Bean -->  
 <bean id="bookRepository" class="com.library.repository.BookRepository" />  
  
 <!-- BookService Bean with constructor AND setter injection -->  
 <bean id="bookService" class="com.library.service.BookService">  
 <!-- constructor-arg uses constructor injection -->  
 <constructor-arg ref="bookRepository" />  
 <!-- property uses setter injection -->  
 <property name="bookRepository" ref="bookRepository" />  
 </bean>  
  
</beans>



**Output**

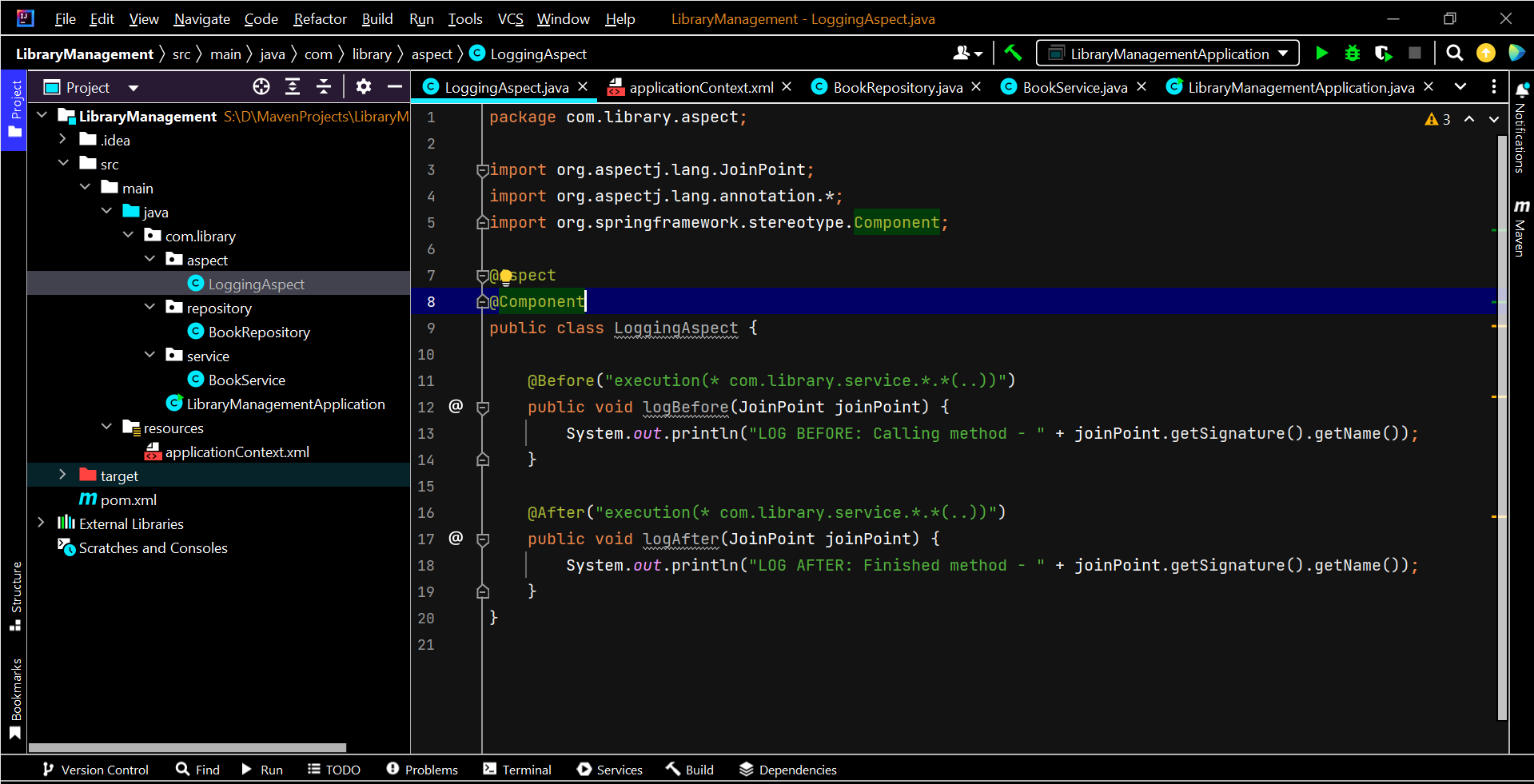
****

**Exercise -9**

**Codes**

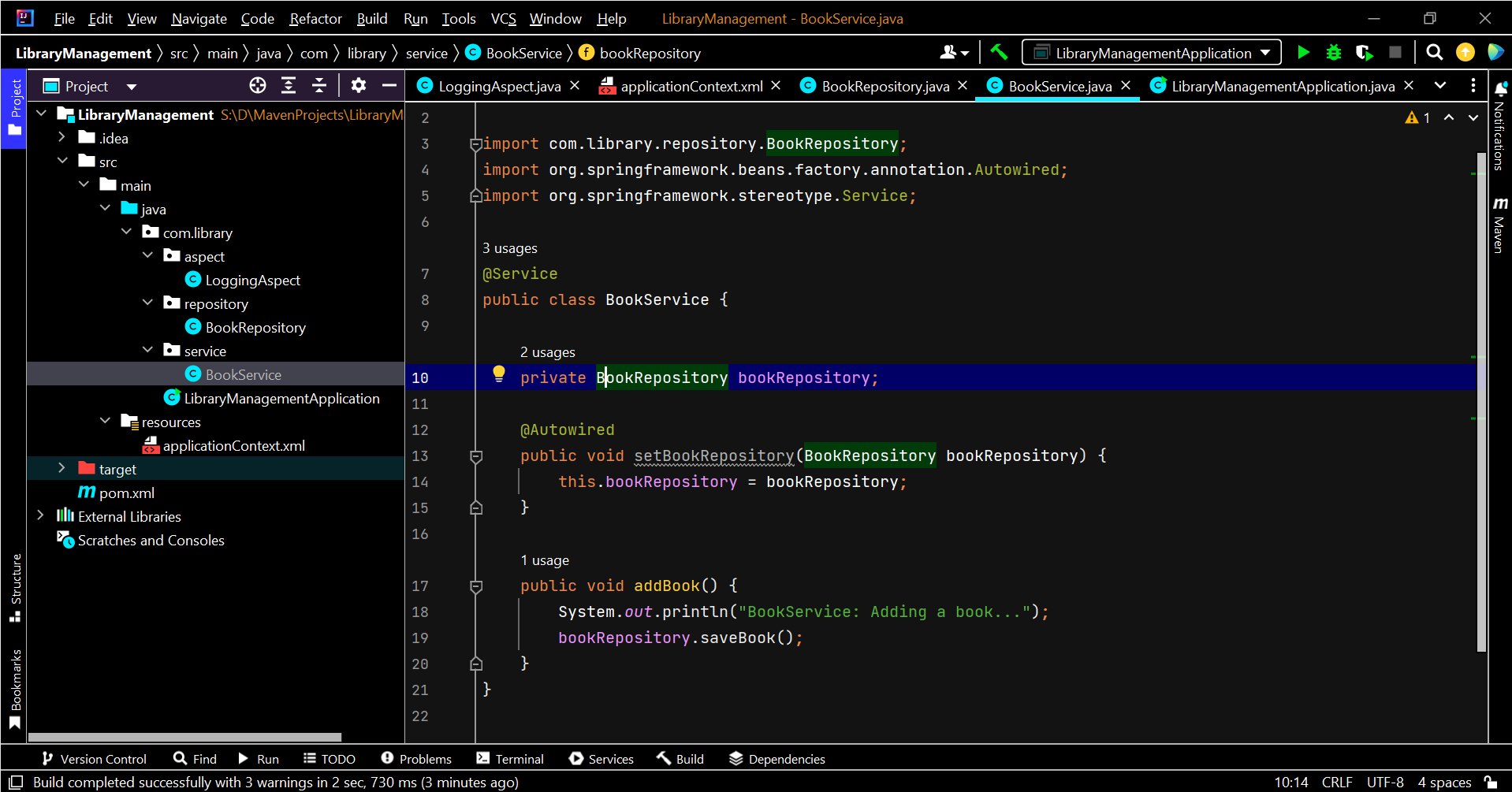
**LoggingAspect.java**

package com.library.aspect**;**import org.aspectj.lang.JoinPoint**;**import org.aspectj.lang.annotation.\***;**import org.springframework.stereotype.Component**;**@Aspect  
@Component  
public class LoggingAspect {  
  
 @Before("execution(\* com.library.service.\*.\*(..))")  
 public void logBefore(JoinPoint joinPoint) {  
 System.*out*.println("LOG BEFORE: Calling method - " + joinPoint.getSignature().getName())**;** }  
  
 @After("execution(\* com.library.service.\*.\*(..))")  
 public void logAfter(JoinPoint joinPoint) {  
 System.*out*.println("LOG AFTER: Finished method - " + joinPoint.getSignature().getName())**;** }  
}



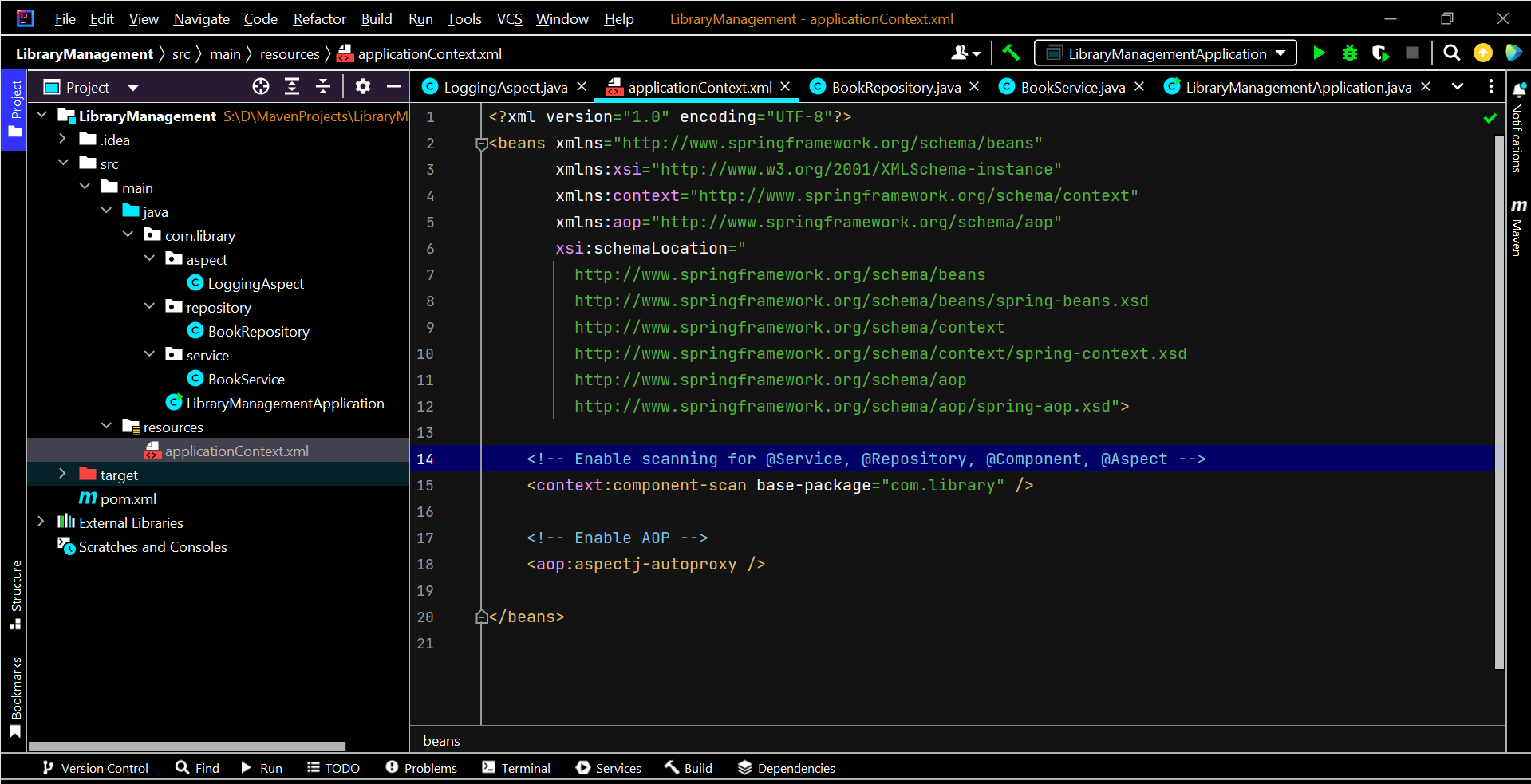
**BookService.java**

package com.library.service**;**import com.library.repository.BookRepository**;**import org.springframework.beans.factory.annotation.Autowired**;**import org.springframework.stereotype.Service**;**@Service  
public class BookService {  
  
 private BookRepository bookRepository**;** @Autowired  
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository**;** }  
  
 public void addBook() {  
 System.*out*.println("BookService: Adding a book...")**;** bookRepository.saveBook()**;** }  
}



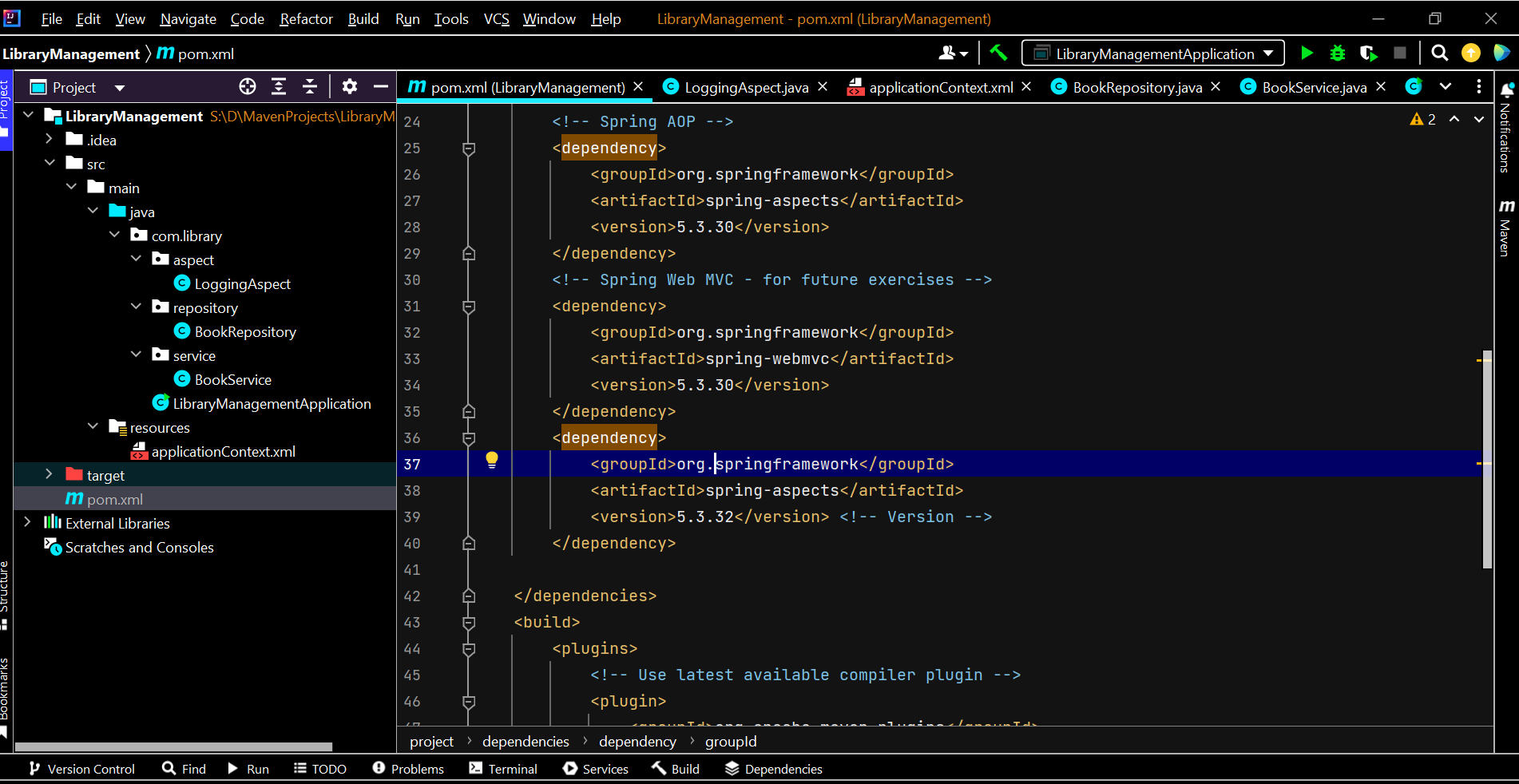
**applicationContext.java**

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xmlns:context="http://www.springframework.org/schema/context"  
 xmlns:aop="http://www.springframework.org/schema/aop"  
 xsi:schemaLocation="  
 http://www.springframework.org/schema/beans  
 http://www.springframework.org/schema/beans/spring-beans.xsd  
 http://www.springframework.org/schema/context  
 http://www.springframework.org/schema/context/spring-context.xsd  
 http://www.springframework.org/schema/aop  
 http://www.springframework.org/schema/aop/spring-aop.xsd">  
  
 <!-- Enable scanning for @Service, @Repository, @Component, @Aspect -->  
 <context:component-scan base-package="com.library" />  
  
 <!-- Enable AOP -->  
 <aop:aspectj-autoproxy />  
  
</beans>



**pom.xml(Section added)**

<dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aspects</artifactId>  
 <version>5.3.32</version> <!-- Version -->  
</dependency>



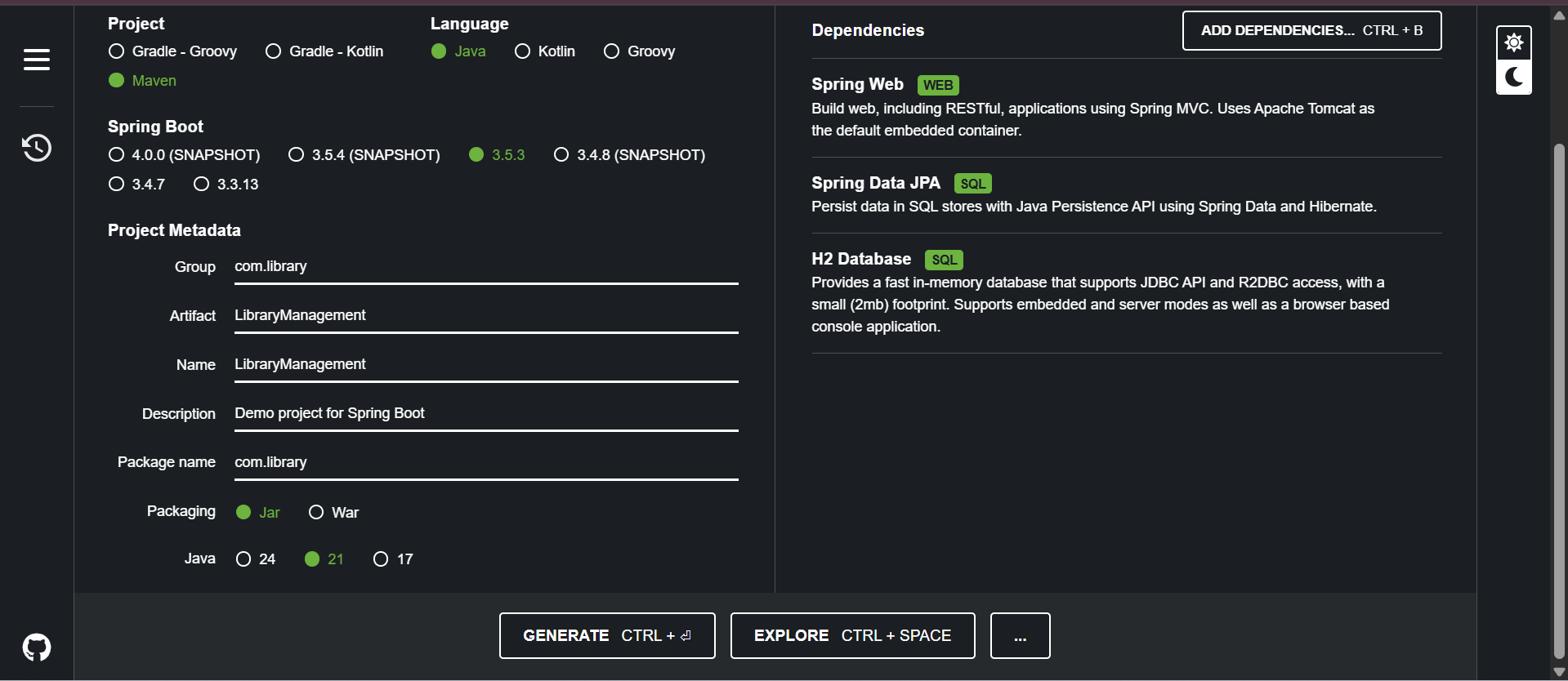
**Output**

****

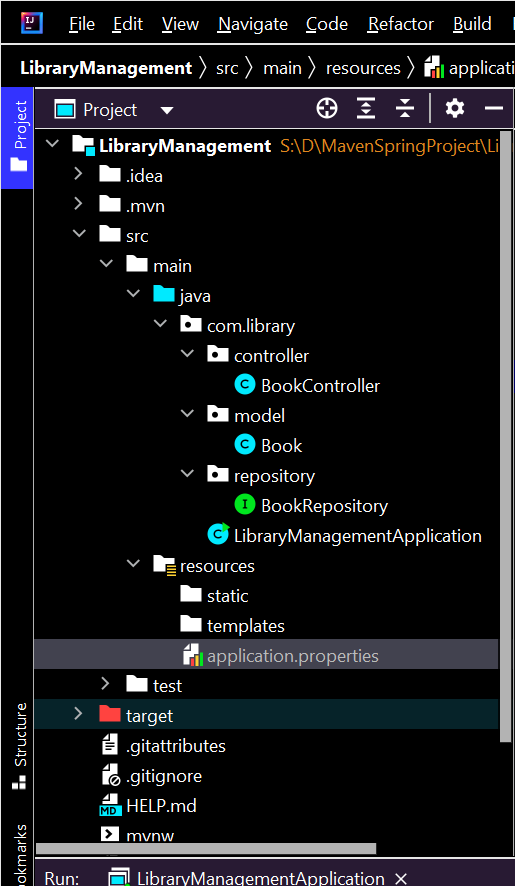
**Exercise 9:**

**Codes**

**Dependencies Added :**

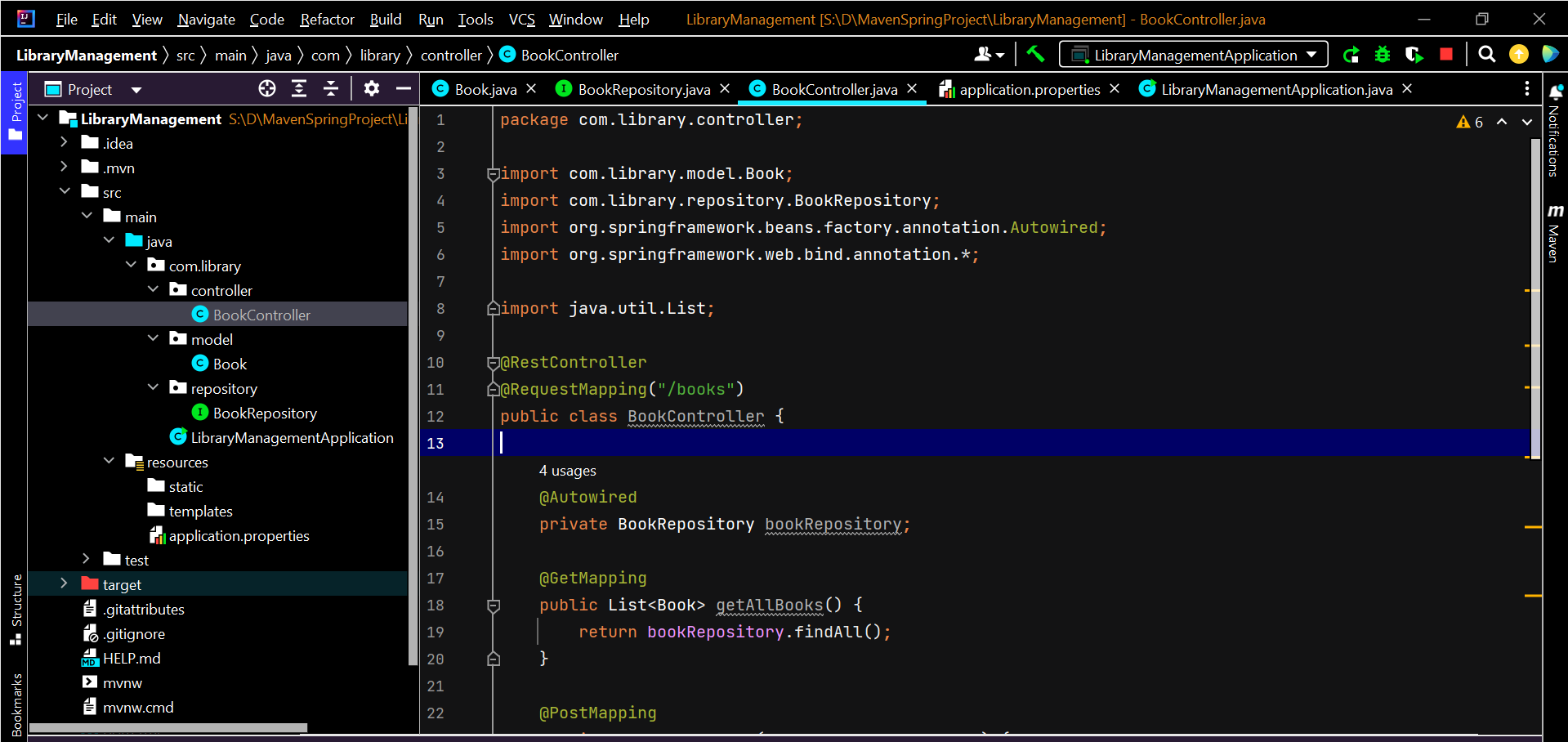
****

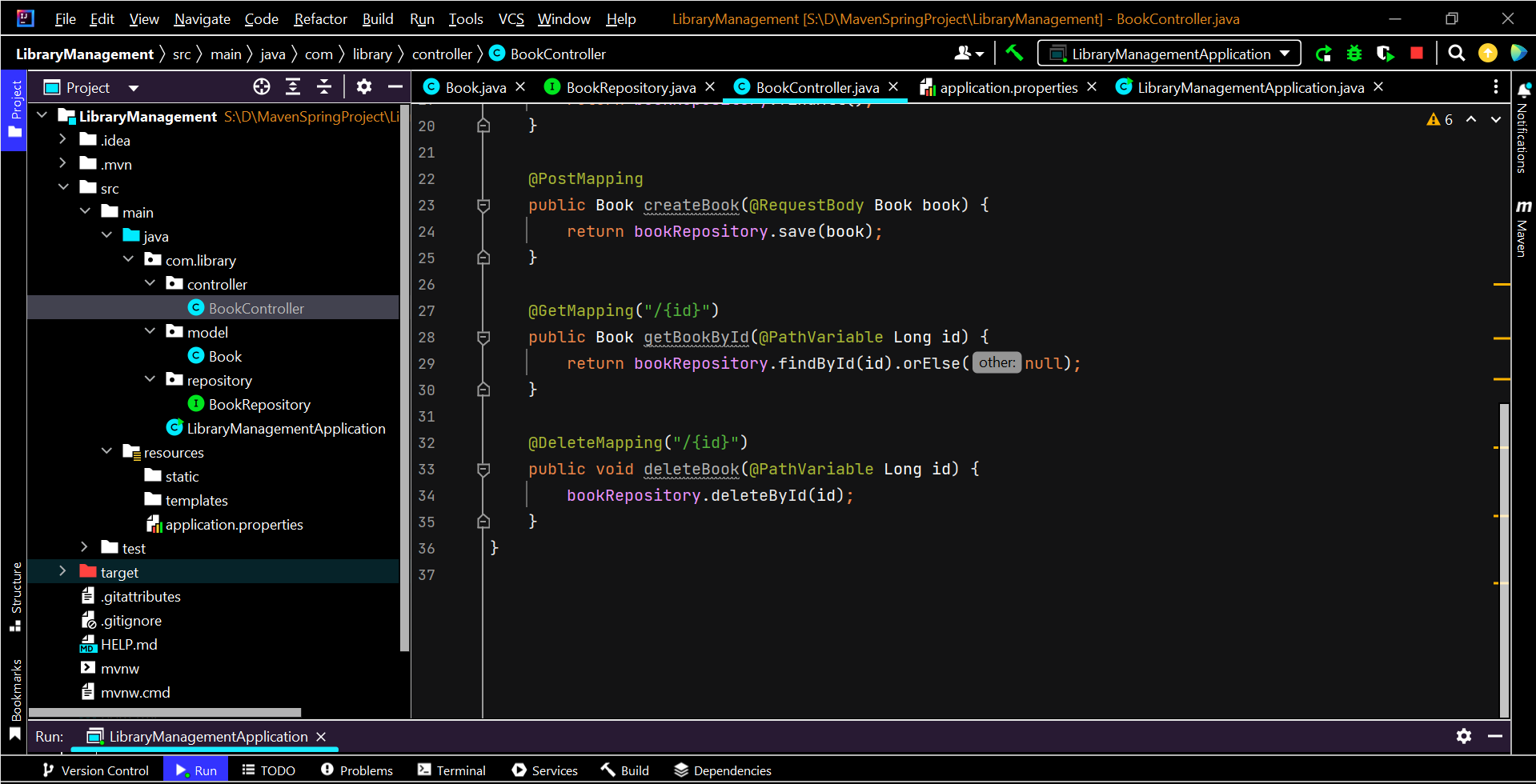
**Files Added:**

****

**BookController.java**

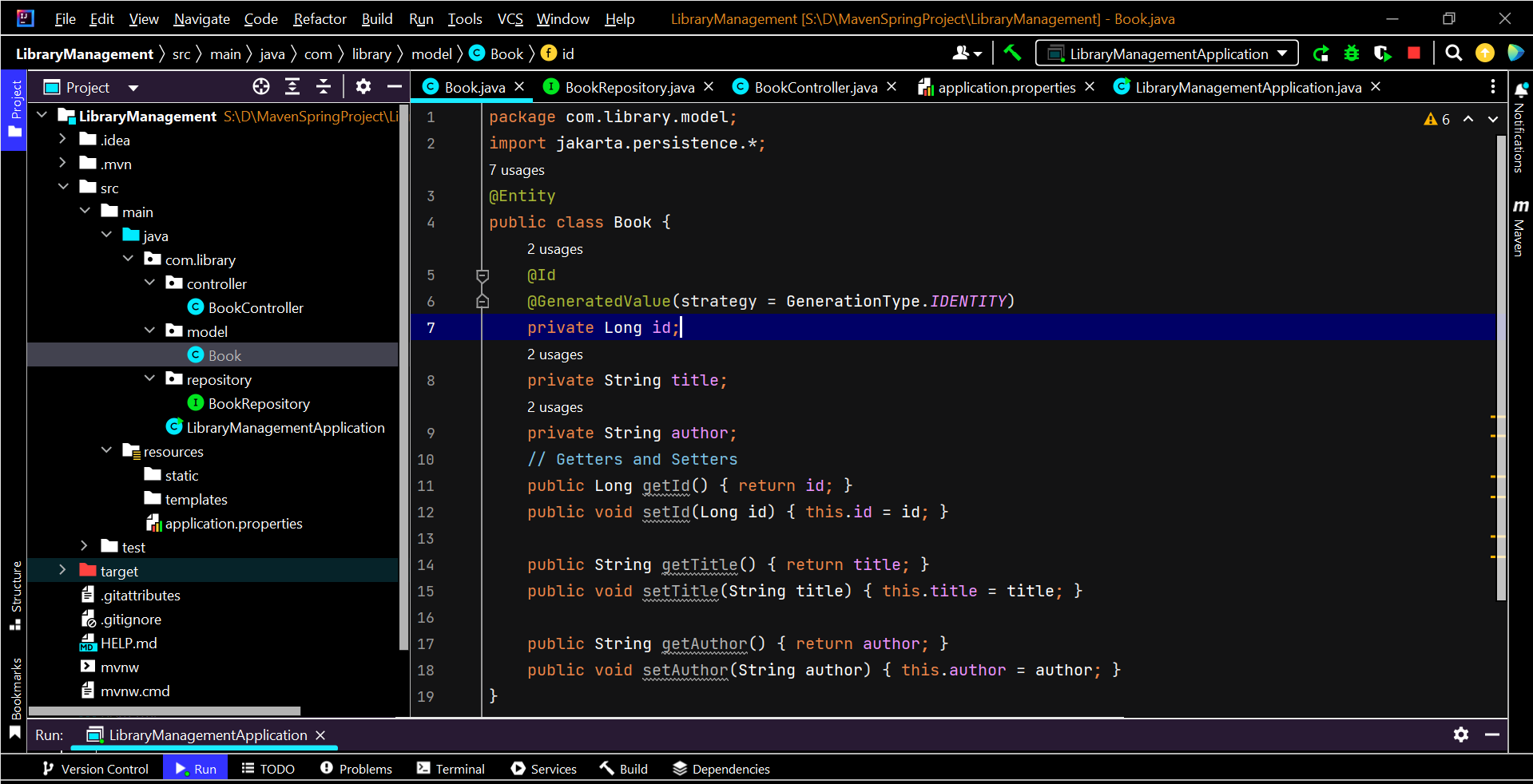
package com.library.controller**;**import com.library.model.Book**;**import com.library.repository.BookRepository**;**import org.springframework.beans.factory.annotation.Autowired**;**import org.springframework.web.bind.annotation.\***;**import java.util.List**;**@RestController  
@RequestMapping("/books")  
public class BookController {  
  
 @Autowired  
 private BookRepository bookRepository**;** @GetMapping  
 public List<Book> getAllBooks() {  
 return bookRepository.findAll()**;** }  
  
 @PostMapping  
 public Book createBook(@RequestBody Book book) {  
 return bookRepository.save(book)**;** }  
  
 @GetMapping("/{id}")  
 public Book getBookById(@PathVariable Long id) {  
 return bookRepository.findById(id).orElse(null)**;** }  
  
 @DeleteMapping("/{id}")  
 public void deleteBook(@PathVariable Long id) {  
 bookRepository.deleteById(id)**;** }  
}





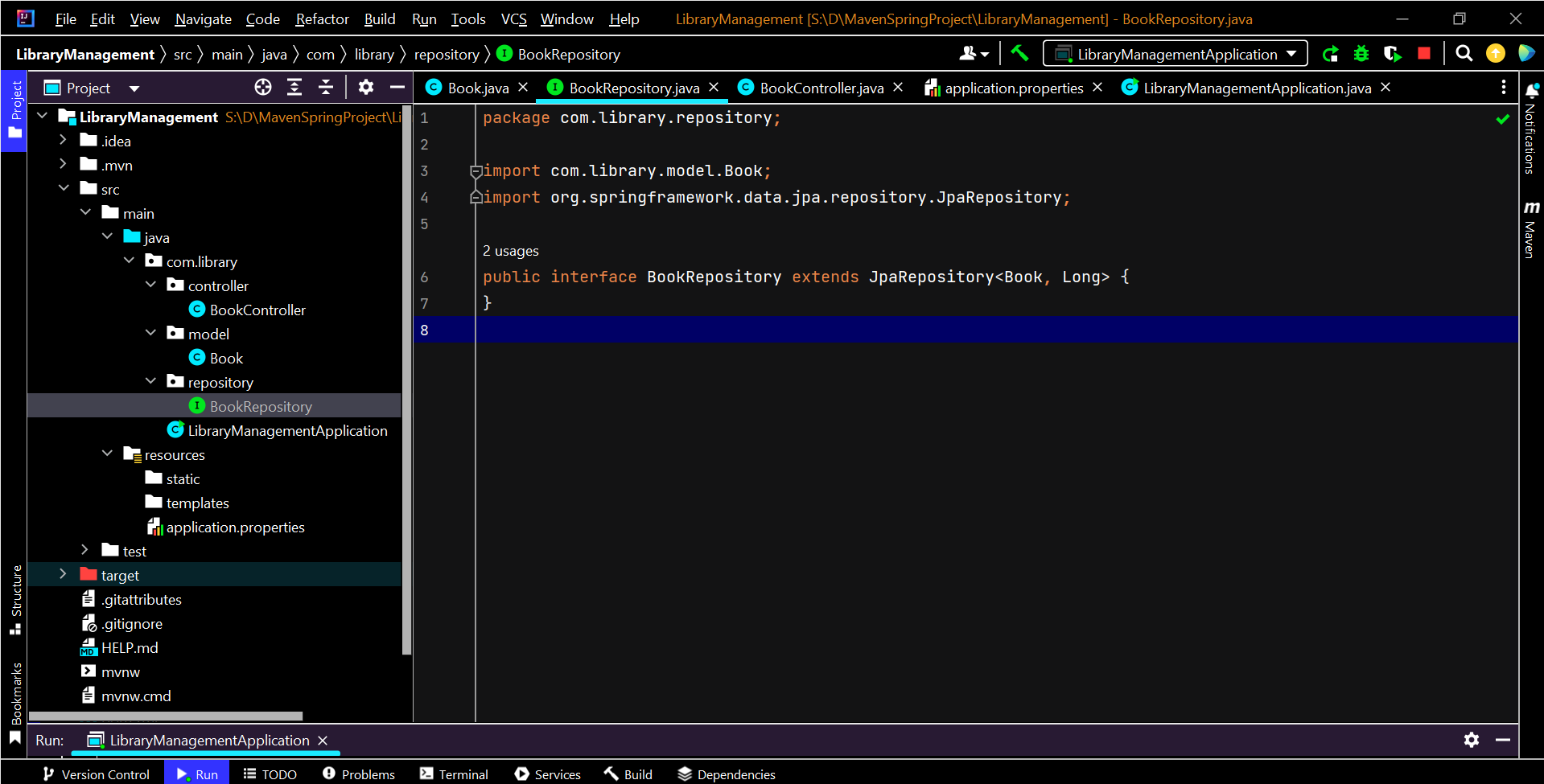
**Book.java**

package com.library.model**;**import jakarta.persistence.\***;**@Entity  
public class Book {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Long id**;** private String title**;** private String author**;** // Getters and Setters  
 public Long getId() { return id**;** }  
 public void setId(Long id) { this.id = id**;** }  
  
 public String getTitle() { return title**;** }  
 public void setTitle(String title) { this.title = title**;** }  
  
 public String getAuthor() { return author**;** }  
 public void setAuthor(String author) { this.author = author**;** }  
}



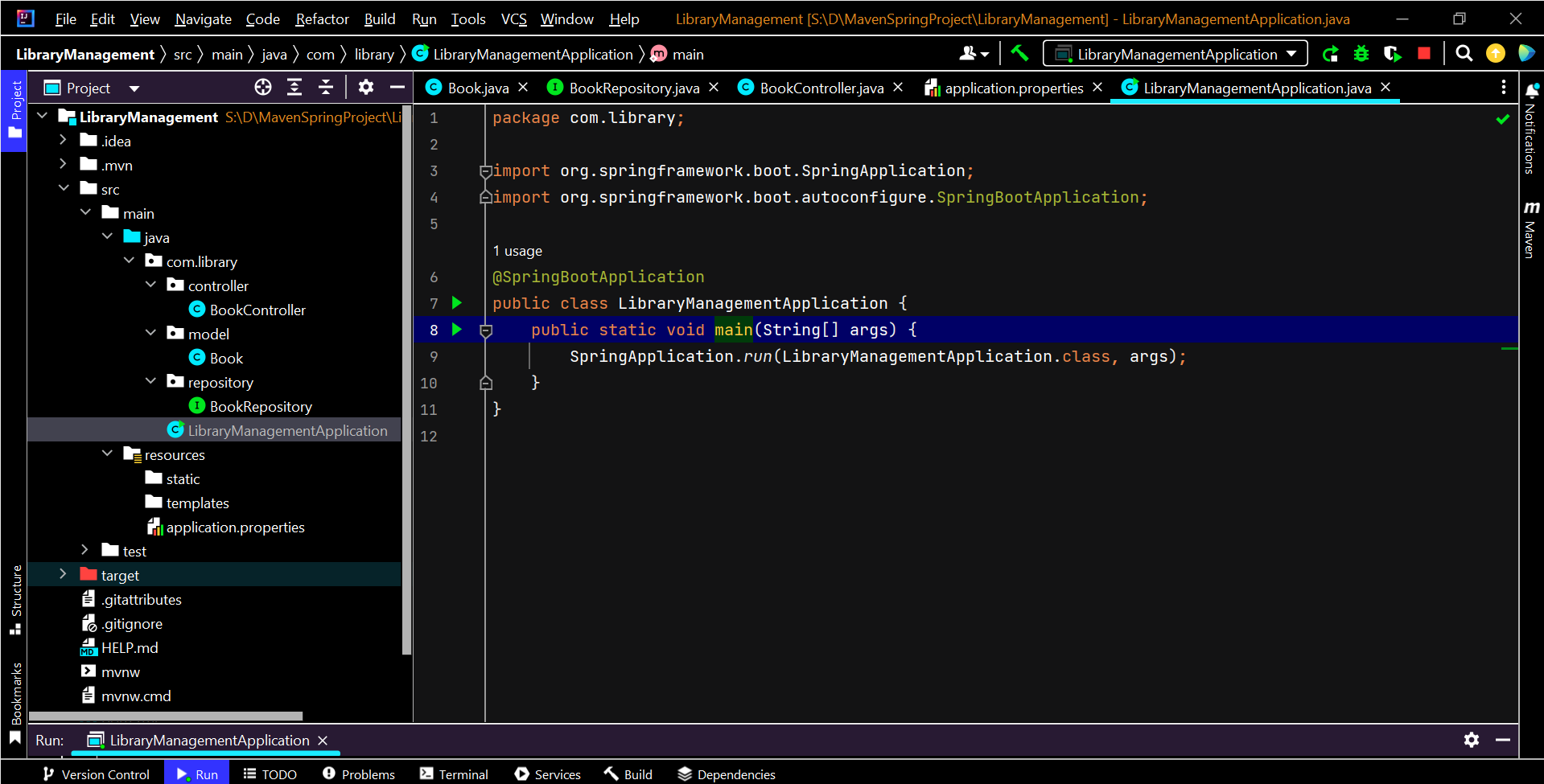
**BookRepository.java**

package com.library.repository;  
  
import com.library.model.Book;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface BookRepository extends JpaRepository<Book, Long> {  
}



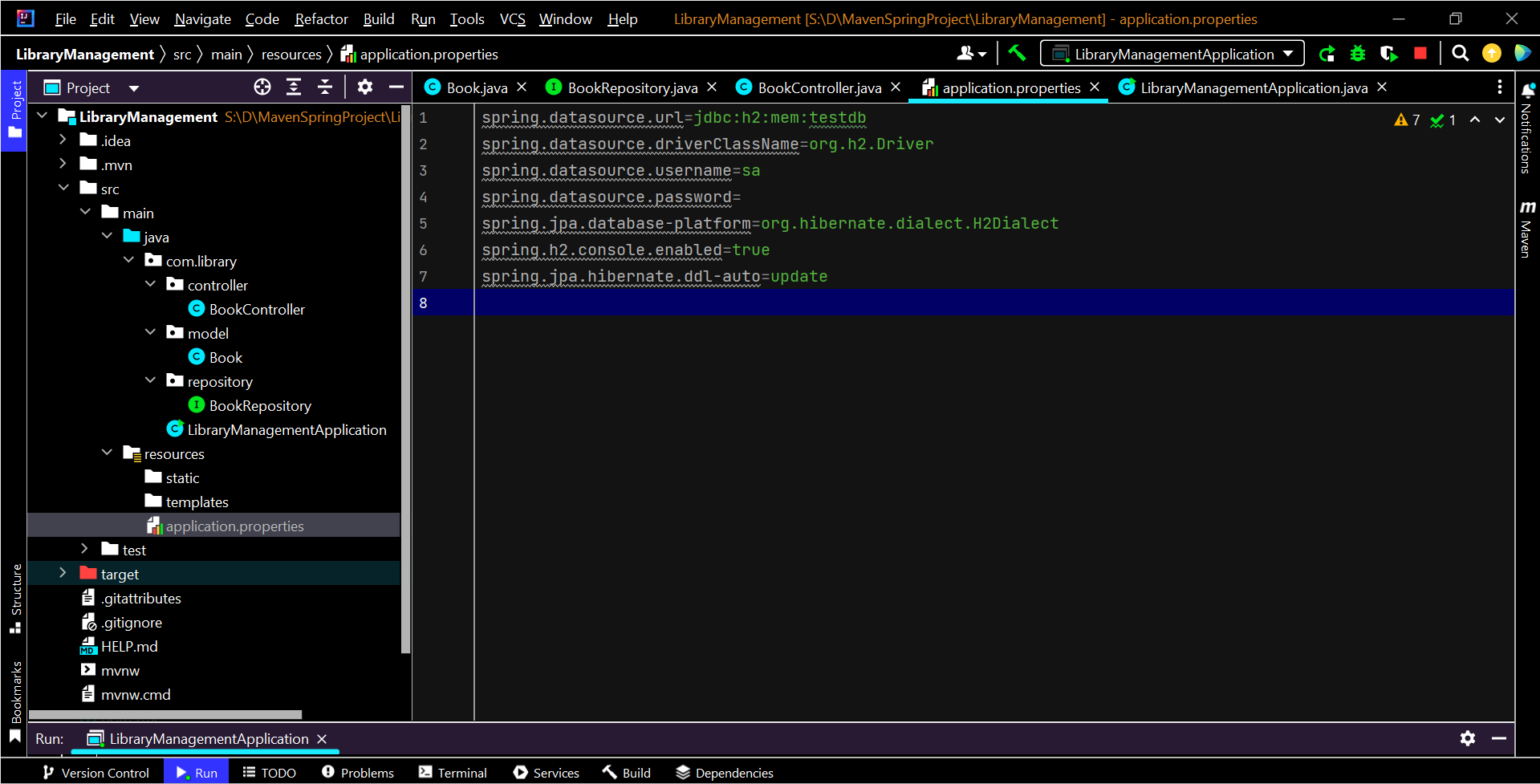
**LibraryManagementApplication.java**

package com.library**;**import org.springframework.boot.SpringApplication**;**import org.springframework.boot.autoconfigure.SpringBootApplication**;**@SpringBootApplication  
public class LibraryManagementApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(LibraryManagementApplication.class**,** args)**;** }  
}

****

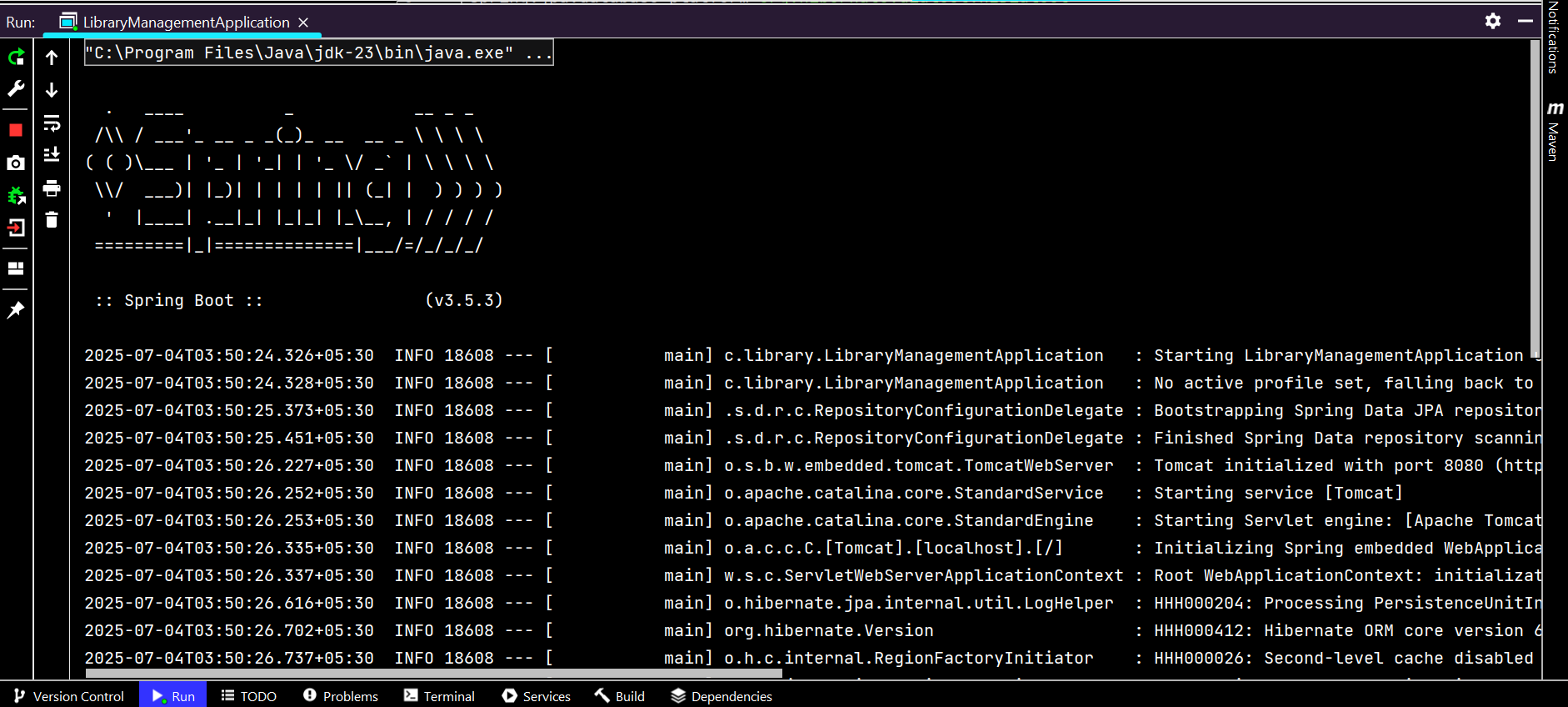
**application.properties**

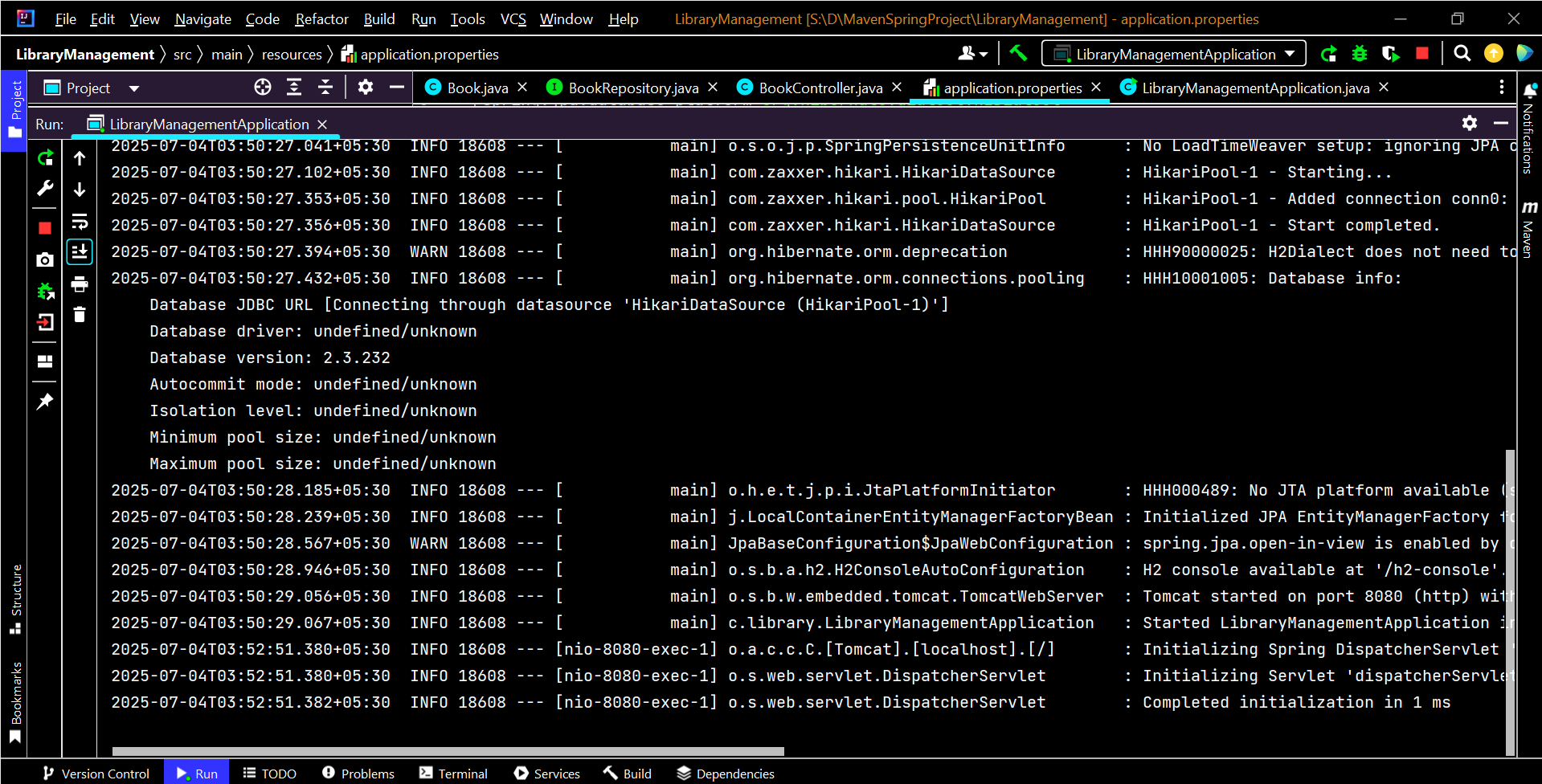
spring.datasource.url=jdbc:h2:mem:testdb  
spring.datasource.driverClassName=org.h2.Driver  
spring.datasource.username=sa  
spring.datasource.password=  
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect  
spring.h2.console.enabled=true  
spring.jpa.hibernate.ddl-auto=update



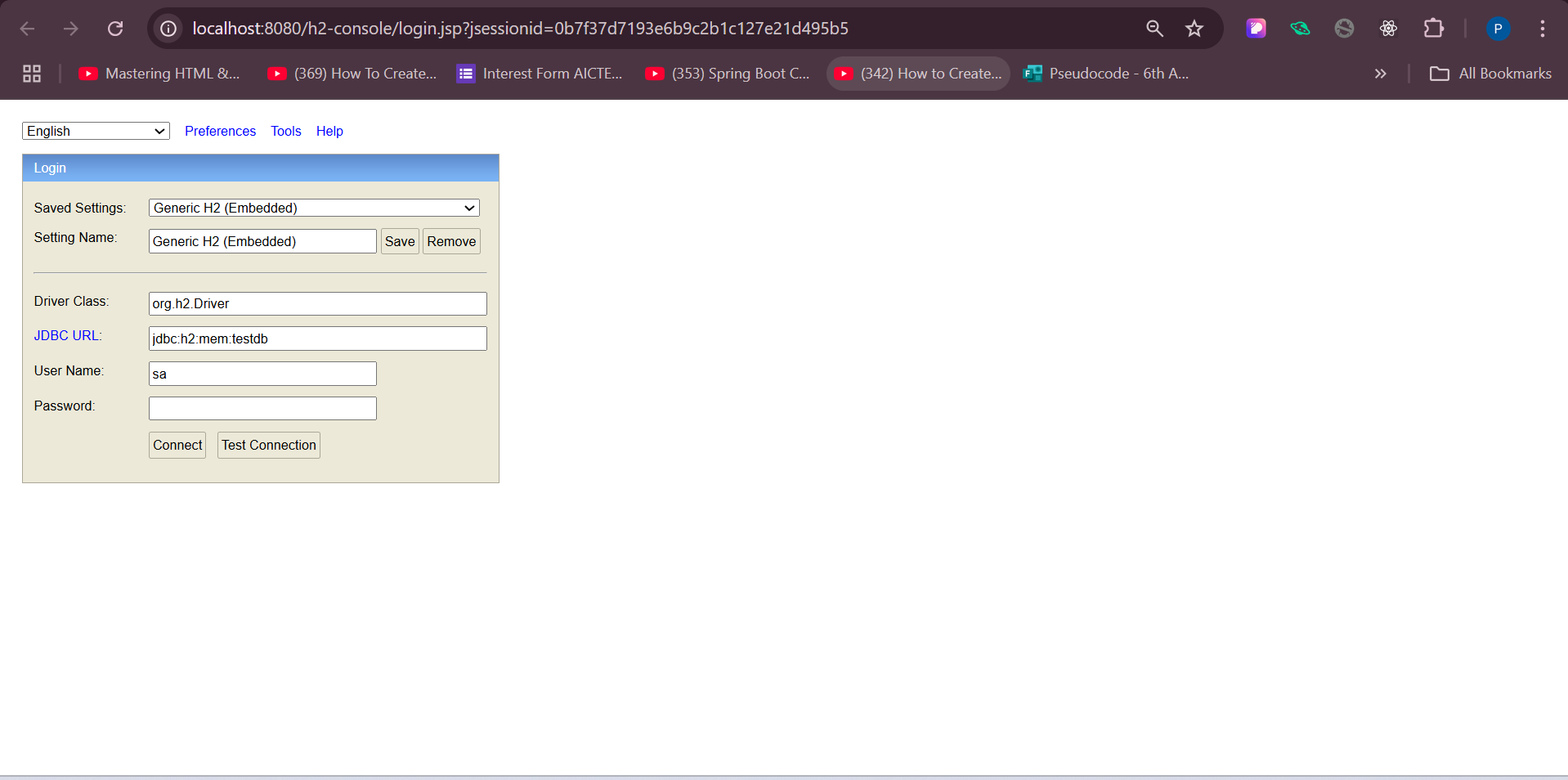
**Output**

**Application when run**

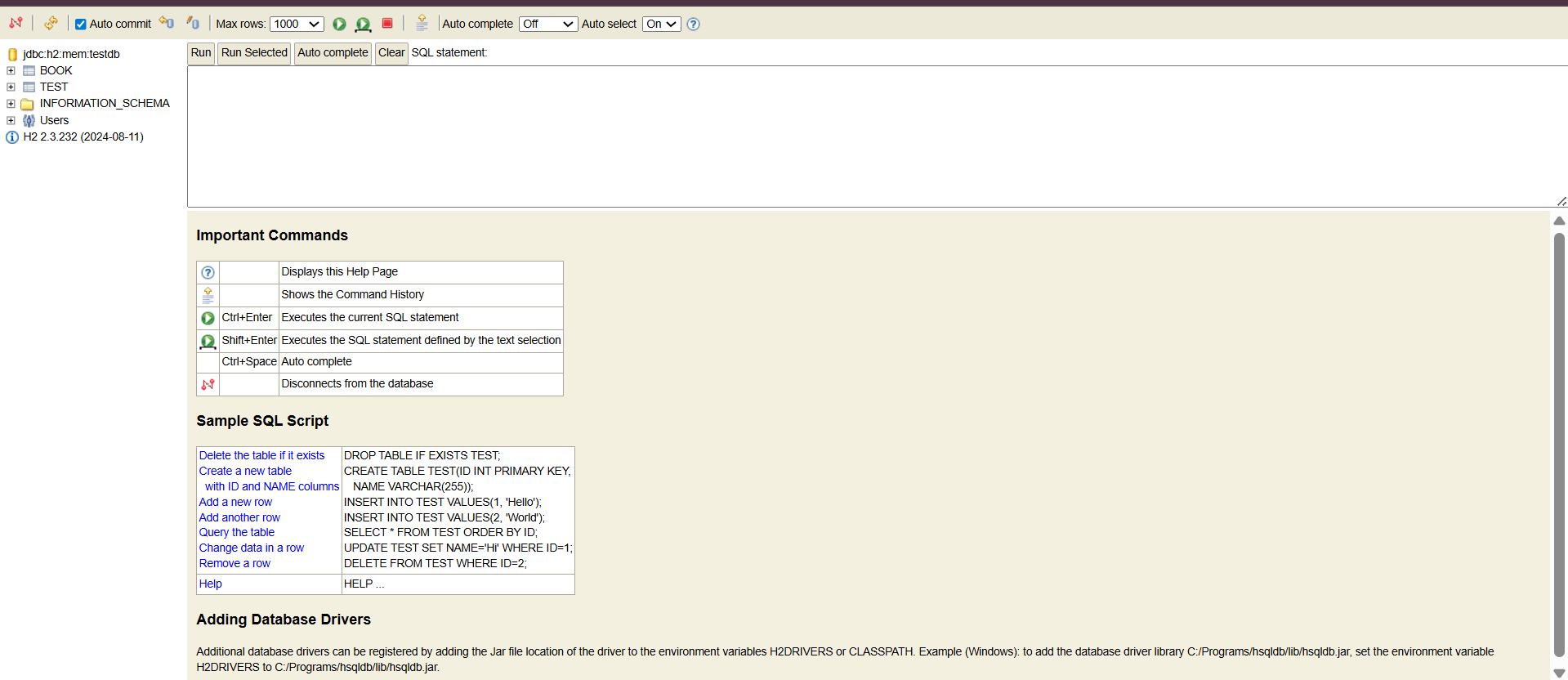
****

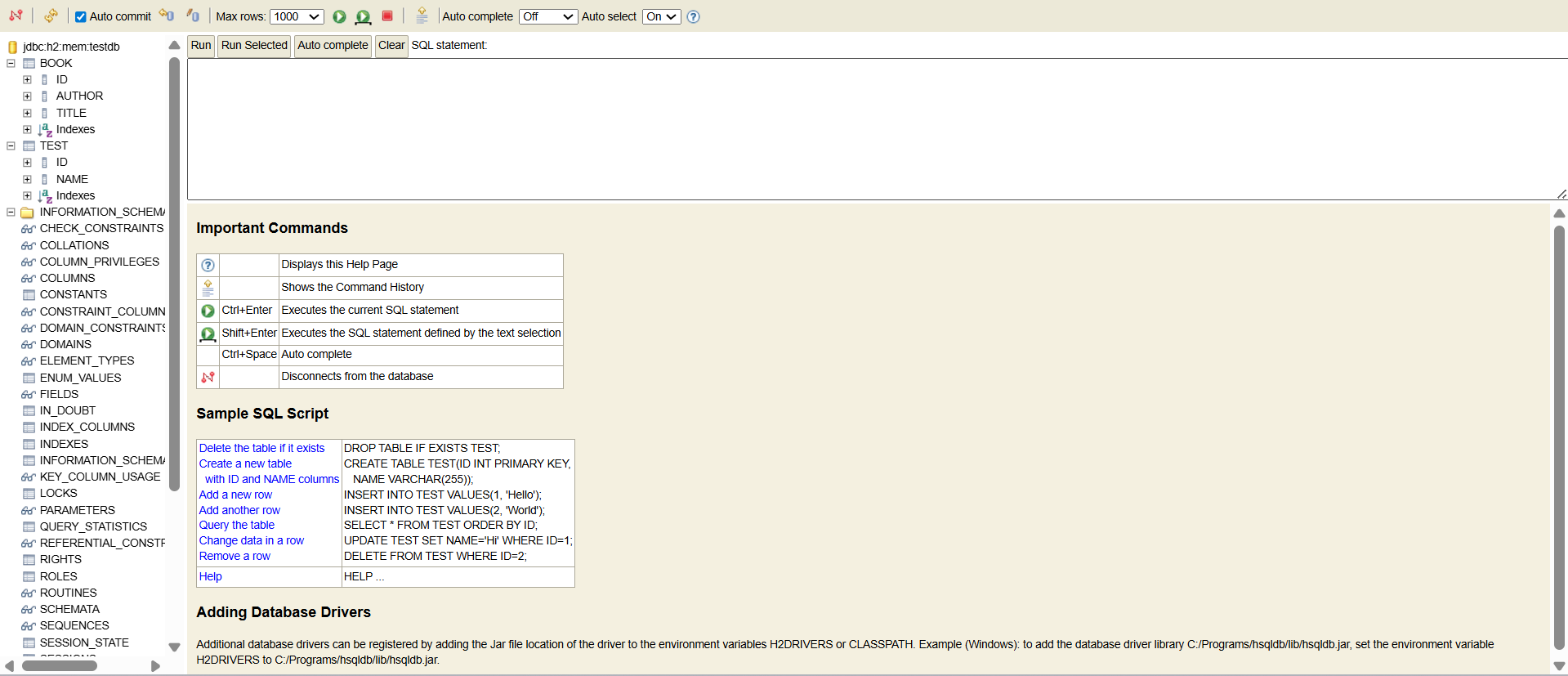
****

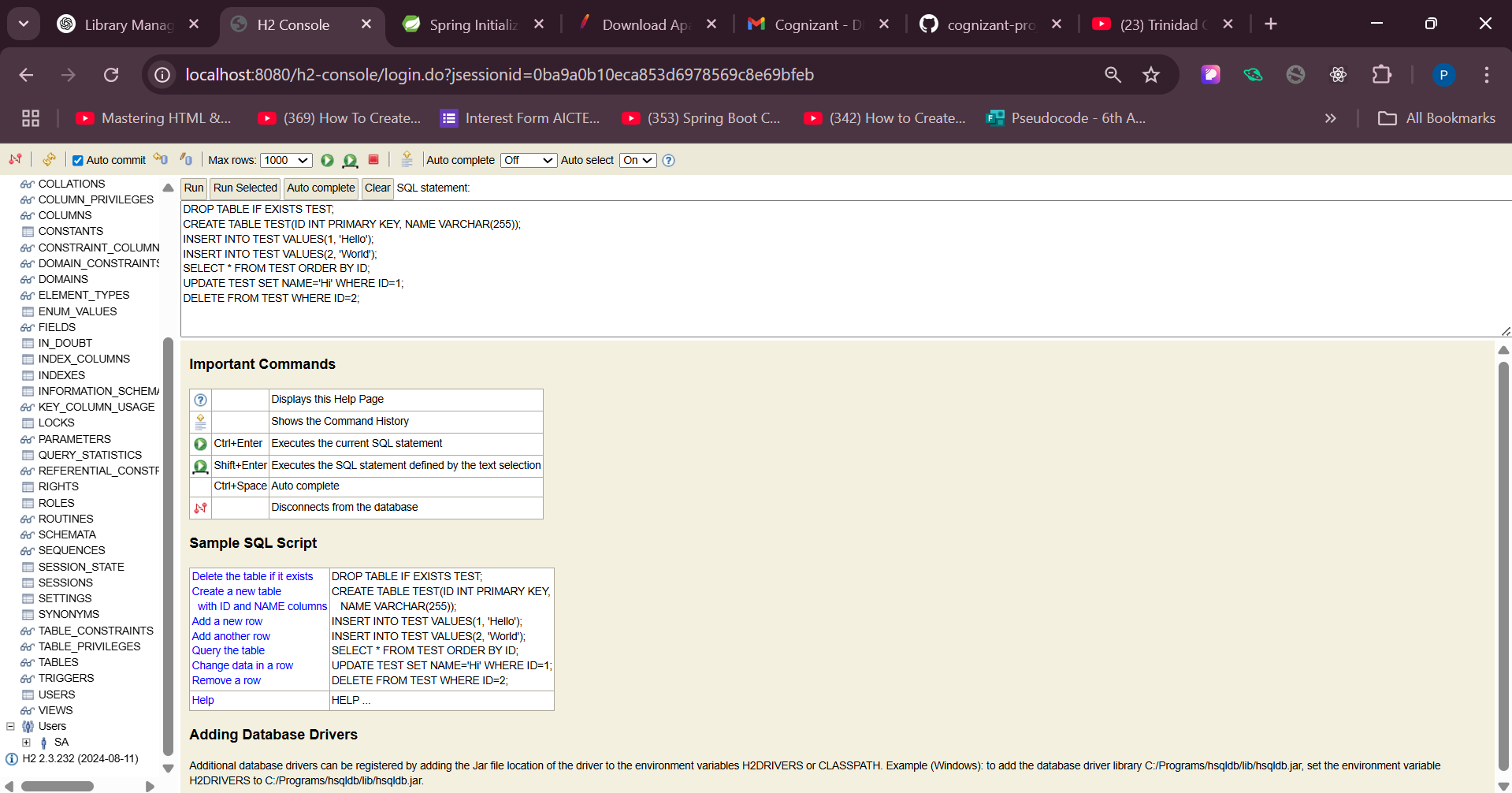
**SpringBoot Application**

****

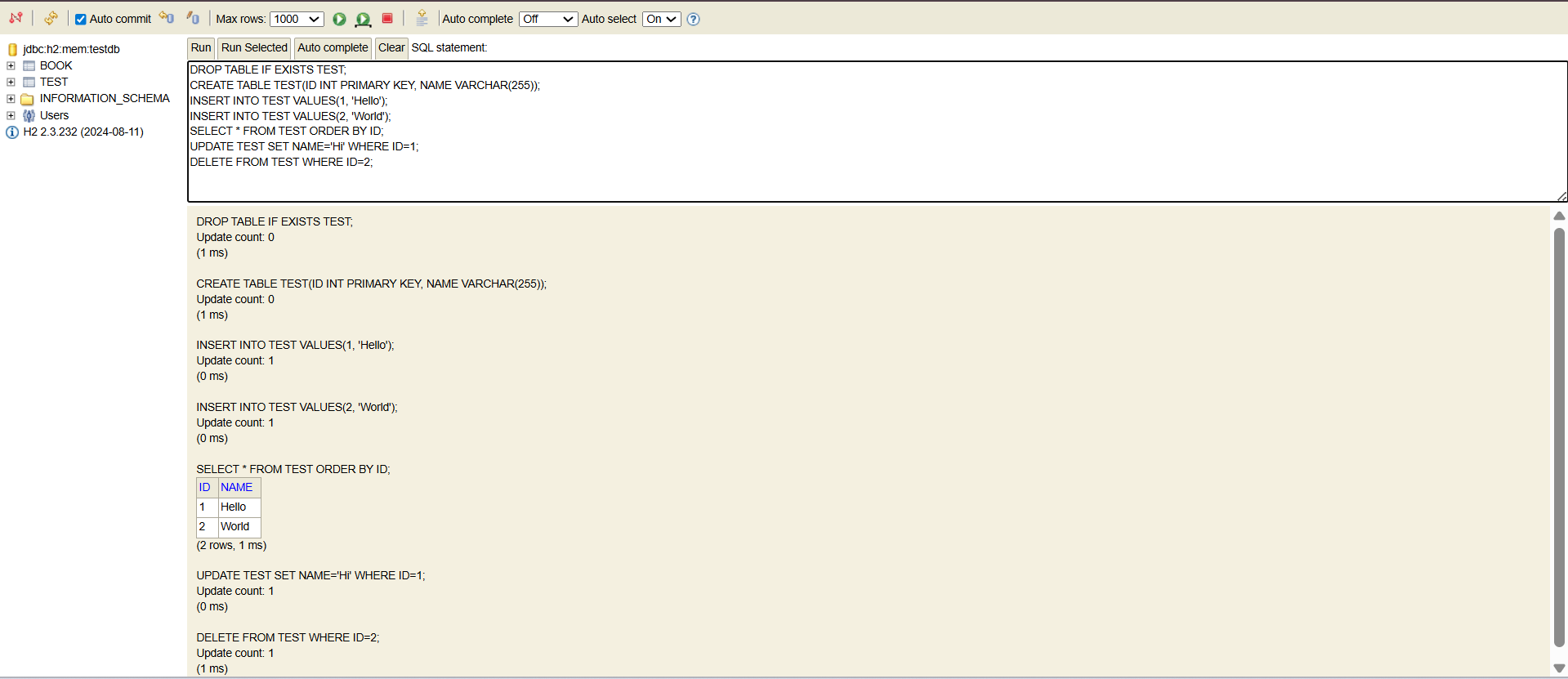
**On connect**

****

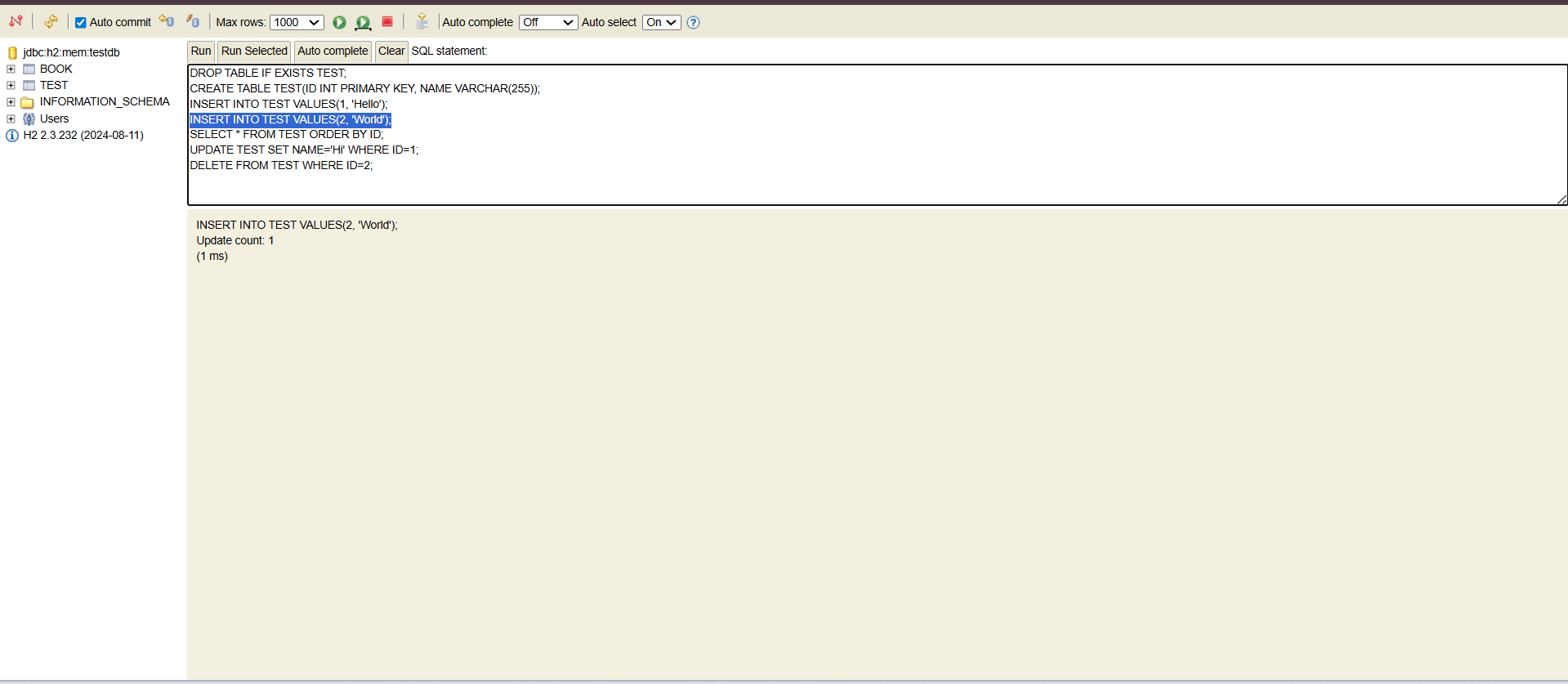
****

****

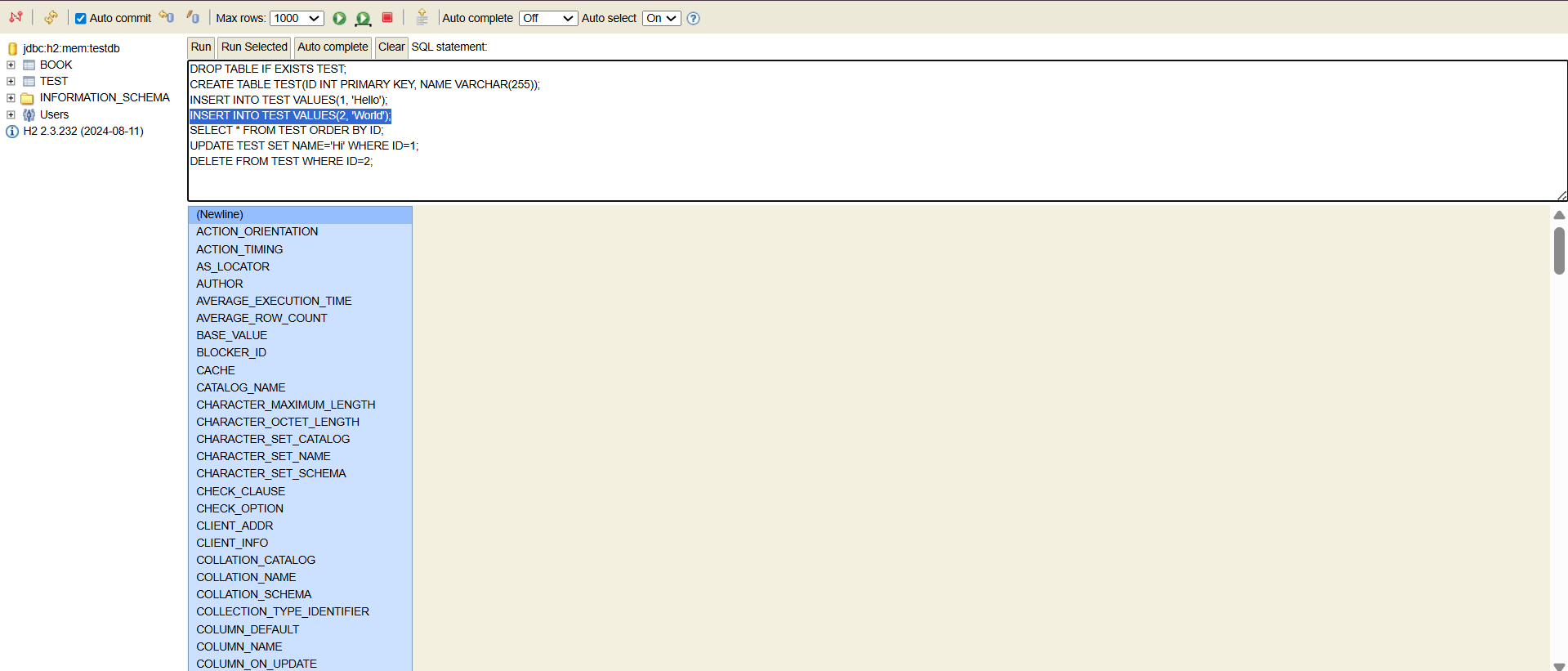
**On Run**

****

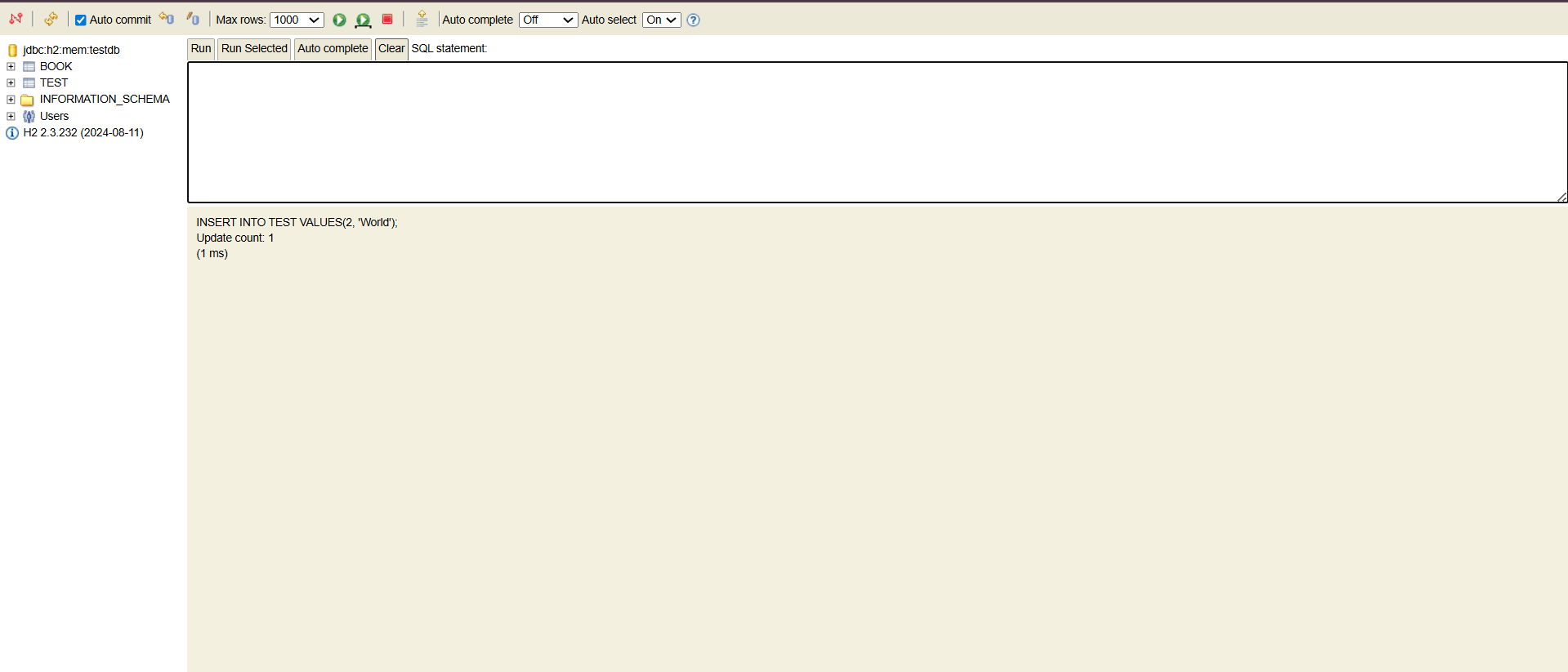
**On Run Selected**

****

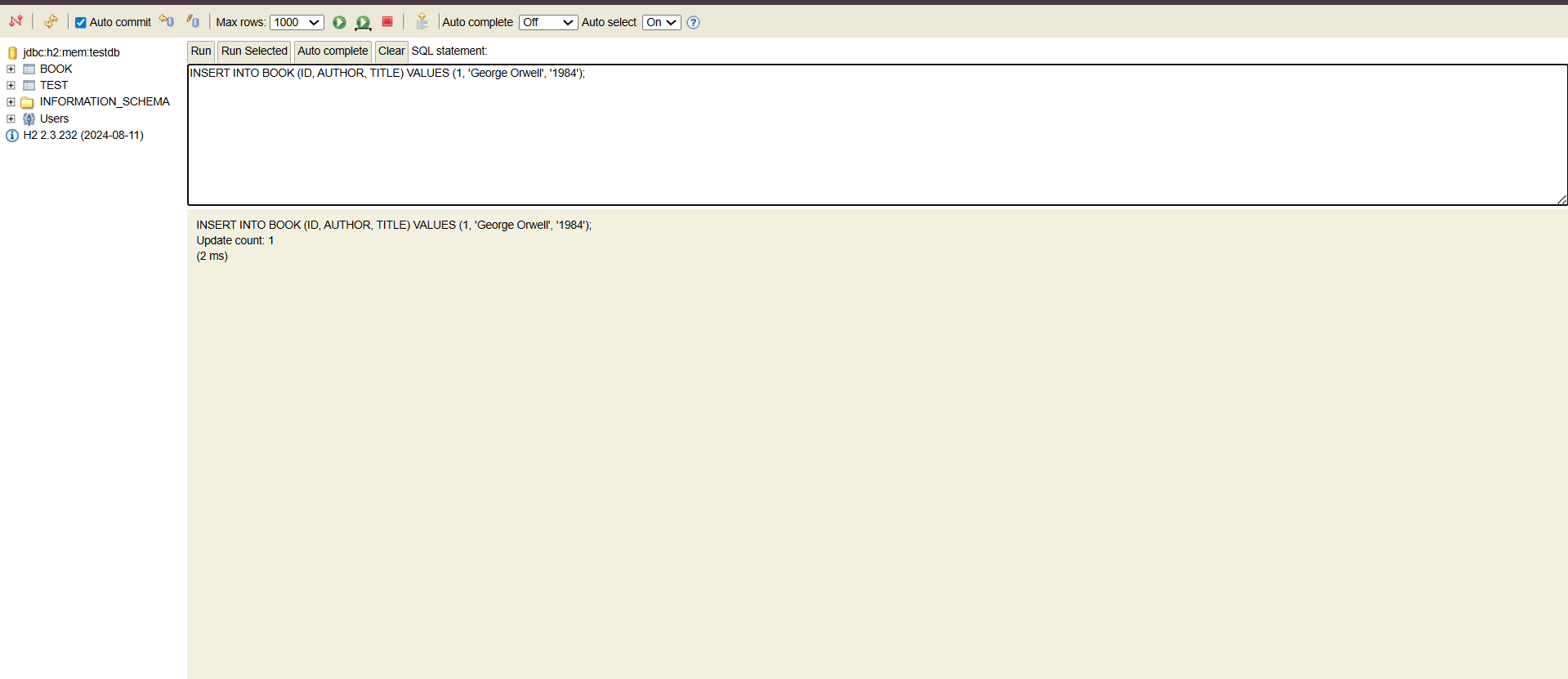
**On Auto Complete**

****

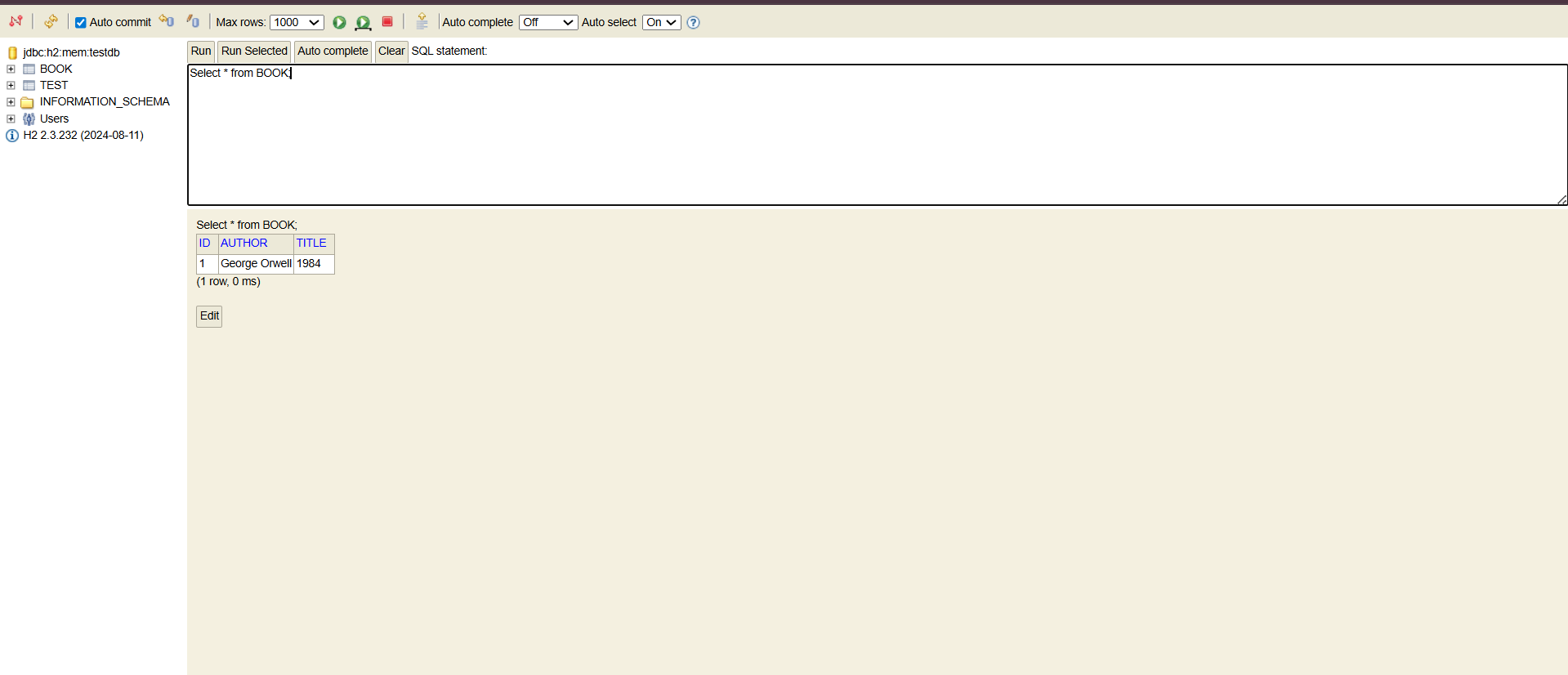
**On Clear**

****

**Inserting a Book details**

****

**Book Details stored in table**

****