**Exercise 1: Configuring a Basic Spring Application & Exercise 4: Creating and Configuring a Maven Project :**

**Project Details:**

* Project Name: LibraryManagement
* Tool Used: Eclipse IDE
* Build System: Maven
* Java Version: JDK 1.8
* Spring Version: 5.3.3

## Source Code:

### **BookRepository.java :**

**package** com.library.repository;

**public** **class** BookRepository {

**public** **void** saveBook() {

System.***out***.println("Book saved to database.");

}

}

### **BookService.java :**

**package** com.library.service;

**public** **class** BookService {

**public** **void** addBook() {

System.***out***.println("Book added through service.");

}

}

### **LibraryManagementApplication.java :**

**package** com.library.LibraryManagement;

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

service.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

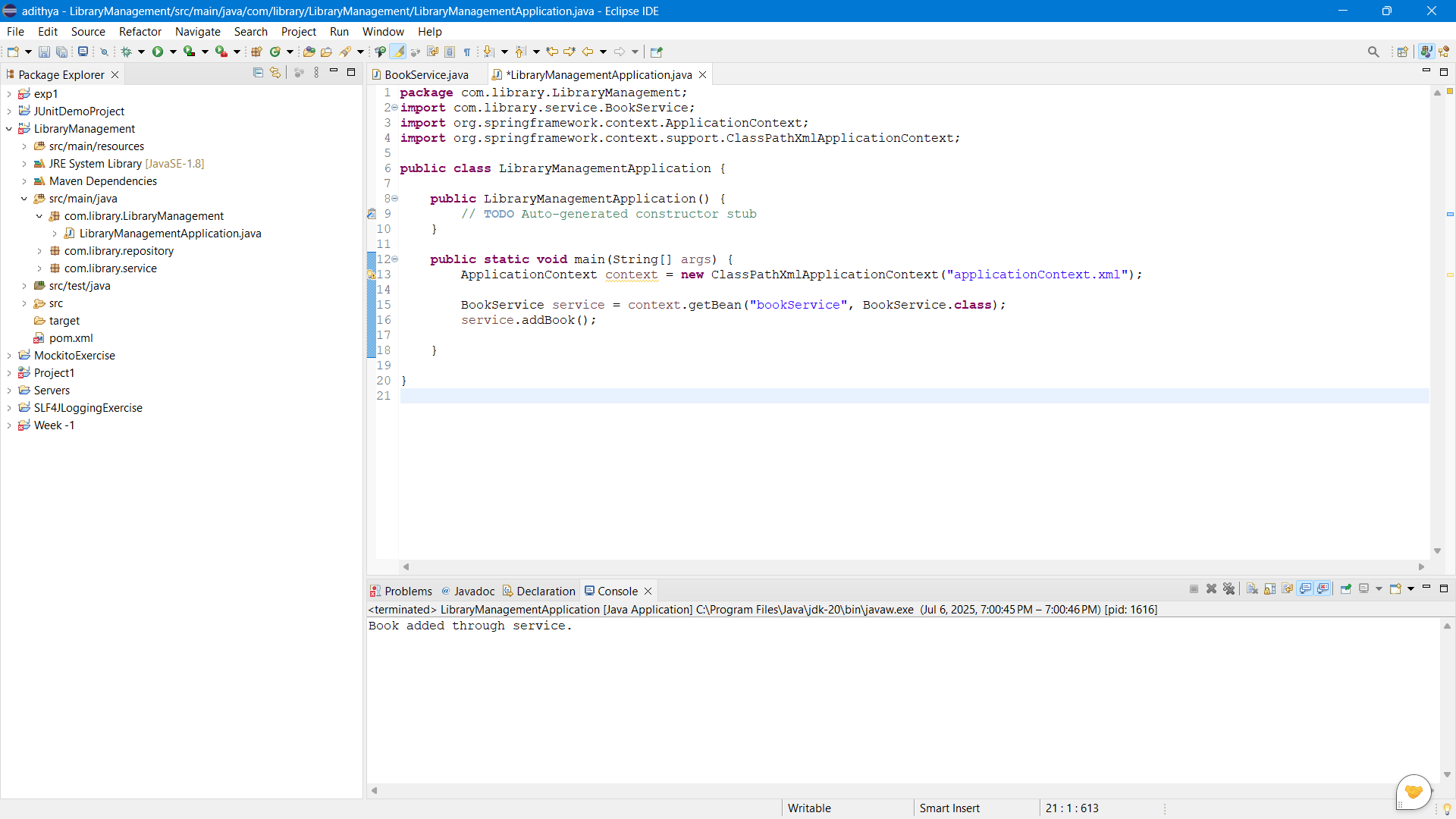
https://www.springframework.org/schema/beans/spring-beans.xsd">

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

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

</beans>

**Output :**



**Exercise 2: Implementing Dependency Injection & Exercise 5: Configuring the Spring IoC Container :**

## Source Code:

### **BookRepository.java :**

**package** com.library.repository;

**public** **class** BookRepository {

**public** **void** saveBook() {

System.***out***.println("Book saved to database.");

}

}

### **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("Book added through service.");

bookRepository.saveBook(); // Call repository method to confirm DI

}

}

## 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

https://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 7: Implementing Constructor and Setter Injection :**

## Source Code:

### **BookRepository.java :**

**package** com.library.repository;

**public** **class** BookRepository {

**public** **void** saveBook() {

System.***out***.println("Book saved to database.");

}

}

### **BookService.java :**

**package** com.library.service;

**import** com.library.repository.BookRepository;

**public** **class** BookService {

**private** BookRepository bookRepository; // Constructor Injection

**private** String libraryName; // Setter Injection

// Constructor for BookRepository injection

**public** BookService(BookRepository bookRepository) {

**this**.bookRepository = bookRepository;

}

// Setter for libraryName (demonstrate setter injection)

**public** **void** setLibraryName(String libraryName) {

**this**.libraryName = libraryName;

}

**public** **void** addBook() {

System.***out***.println("[" + libraryName + "] Book added through service.");

bookRepository.saveBook();

}

}

### **LibraryManagementApplication.java :**

**package** com.library.LibraryManagement;

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

service.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

https://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 injection -->

<constructor-arg ref="bookRepository"/>

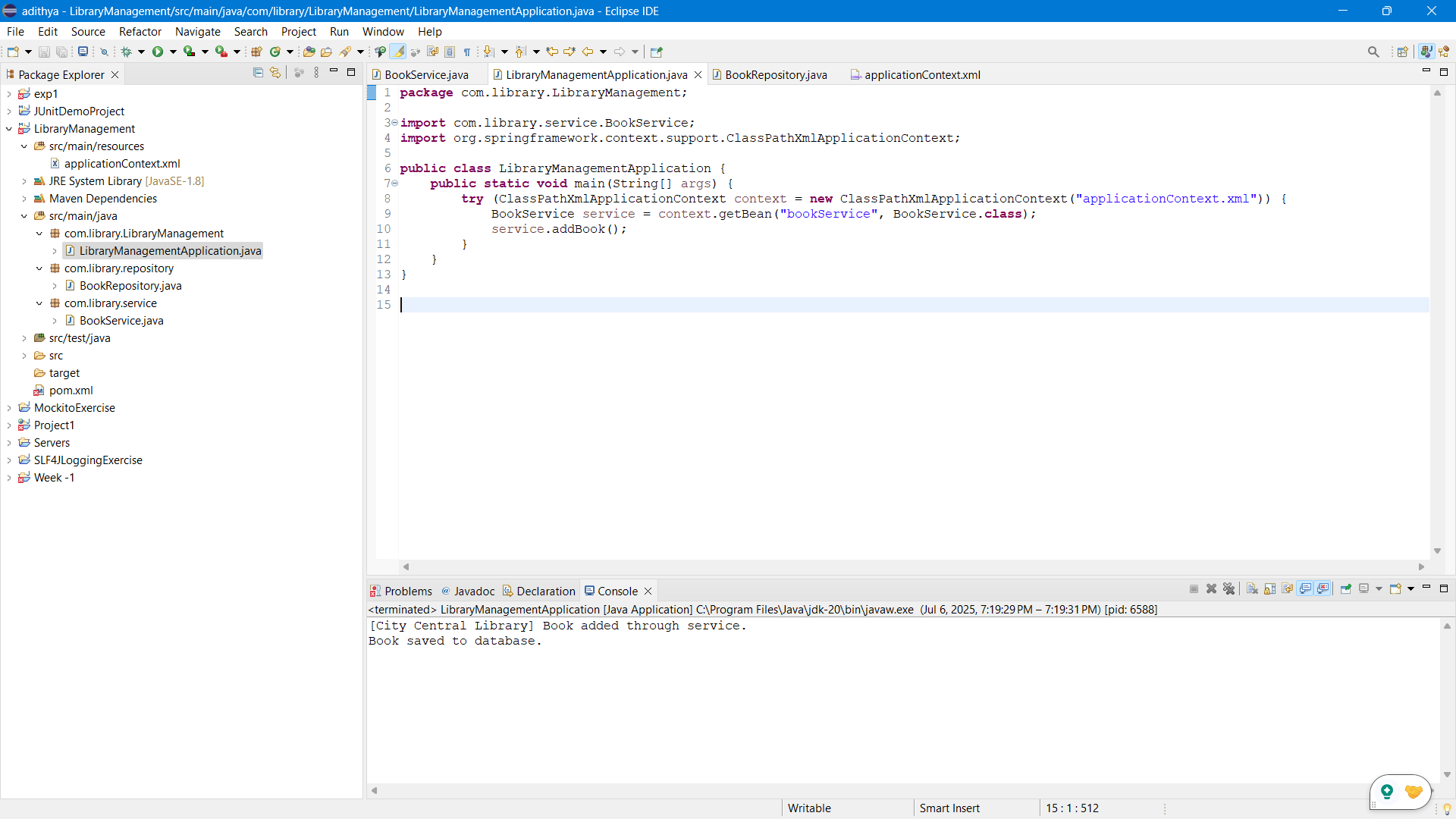
<!-- Setter injection -->

<property name="libraryName" value="City Central Library"/>

</bean>

</beans>

**Output :**



**Exercise 9: Creating a Spring Boot Application**

## Project Details:

* **Project Name**: LibraryManagement
* **Tool Used**: Eclipse IDE
* **Build System**: Maven
* **Java Version**: JDK 20
* **Framework**: Spring Boot 3.2.x
* **Database**: H2 In-Memory Database
* **Dependencies Used**:
  + Spring Web
  + Spring Data JPA
  + H2 Database

Folder Structure:

LibraryManagement/

├── src/

│ ├── main/

│ │ ├── java/com/library/LibraryManagement/

│ │ │ ├── LibraryManagementApplication.java

│ │ │ ├── controller/BookController.java

│ │ │ ├── entity/Book.java

│ │ │ └── repository/BookRepository.java

│ │ └── resources/

│ │ ├── application.properties

├── pom.xml

**application.properties:**

spring.datasource.url=jdbc:h2:mem:librarydb

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

server.port=9191

**SQL Queries to Add Book Records:**

INSERT INTO BOOK (title, author) VALUES ('Atomic Habits', 'James Clear');

INSERT INTO BOOK (title, author) VALUES ('Clean Code', 'Robert C. Martin');

INSERT INTO BOOK (title, author) VALUES ('The Pragmatic Programmer', 'Andrew Hunt');

## Source Code:

**Book.java :**

**package** com.library.LibraryManagement.entity;

**import** jakarta.persistence.\*;

@Entity

**public** **class** Book {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

**private** Long id;

**private** String title;

**private** String author;

**public** Book() {}

**public** Book(String title, String author) {

**this**.title = title;

**this**.author = author;

}

**public** Long getId() { **return** 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; }

}

**BookRepositary.java :**

**package** com.library.LibraryManagement.repository;

**import** com.library.LibraryManagement.entity.Book;

**import** org.springframework.data.jpa.repository.JpaRepository;

**public** **interface** BookRepository **extends** JpaRepository<Book, Long> {

}

**BookController.java :**

**package** com.library.LibraryManagement.controller;

**import** com.library.LibraryManagement.entity.Book;

**import** com.library.LibraryManagement.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;

@PostMapping

**public** Book addBook(@RequestBody Book book) {

**return** bookRepository.save(book);

}

@GetMapping

**public** List<Book> getAllBooks() {

**return** bookRepository.findAll();

}

@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);

}

}

**LibrarayManagementApplication.java:**

package com.library.LibraryManagement

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class LibraryManagementApplication {

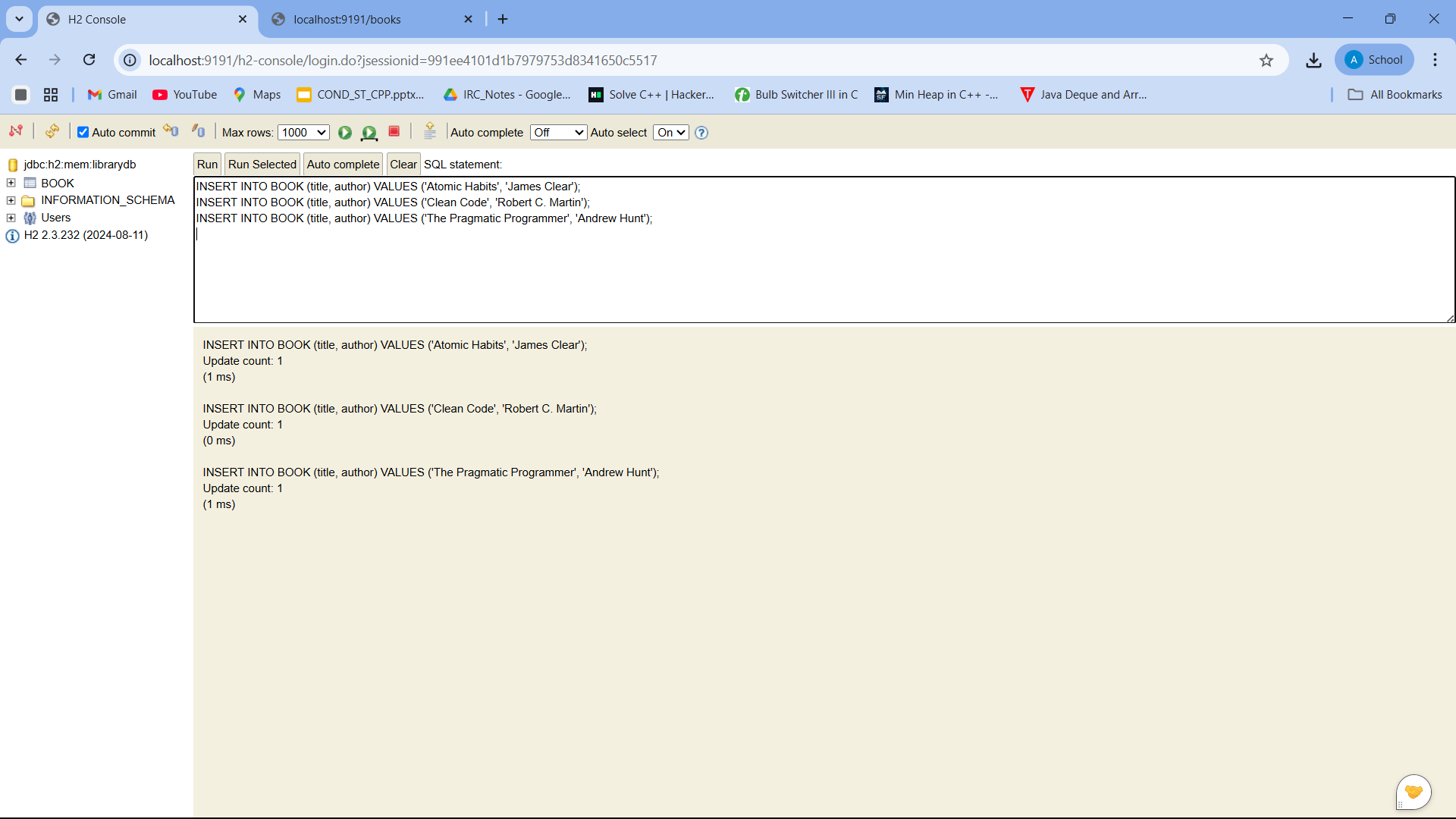
public static void main(String[] args) {

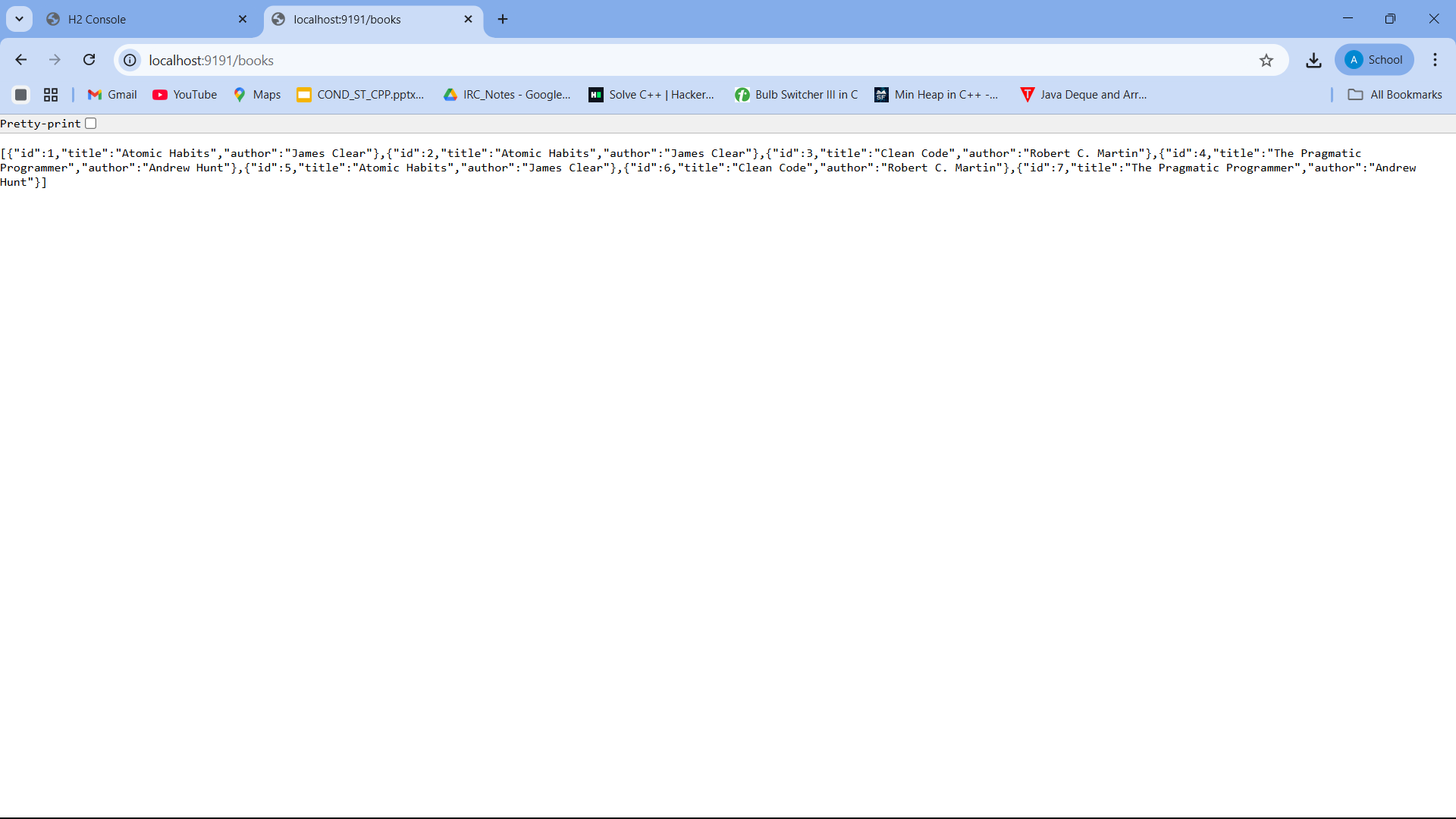
SpringApplication.run(LibraryManagementApplication.class, args);

}

}

**Output :**





**Implement services for managing Country** :

**Output :**

