**Exercise 1: Configuring a Basic Spring Application**

**BookRespitory.java:**

package com.library.repository;

import java.util.ArrayList;

import java.util.List;

public class BookRepository {

private List<String> books = new ArrayList<>();

public void saveBook(String bookName) {

books.add(bookName);

System.out.println("Book '" + bookName + "' saved to the database.");

}

public String findBook(String bookName) {

if (books.contains(bookName)) {

return "Book found: " + bookName;

} else {

return "Book not found: " + bookName;

}

}

public List<String> getAllBooks() {

return books;

}

}

**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(String bookName) {

bookRepository.saveBook(bookName);

}

public void getBook(String bookName) {

String result = bookRepository.findBook(bookName);

System.out.println(result);

}

public void showAllBooks() {

System.out.println("All Books:");

for (String book : bookRepository.getAllBooks()) {

System.out.println("- " + book);

}

}

}

**MainApp.java:**

package com.library;

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

Import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.util.Scanner;

public class MainApp {

public static void main(String[] args) {

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

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

Scanner scanner = new Scanner(System.in);

int choice;

do {

System.out.println("\n=== Library Menu ===");

System.out.println("1. Add Book");

System.out.println("2. Search Book");

System.out.println("3. Show All Books");

System.out.println("4. Exit");

System.out.print("Enter your choice: ");

choice = scanner.nextInt();

scanner.nextLine(); // consume newline

switch (choice) {

case 1:

System.out.print("Enter book name to add: ");

String bookNameToAdd = scanner.nextLine();

bookService.addBook(bookNameToAdd);

break;

case 2:

System.out.print("Enter book name to search: ");

String bookNameToSearch = scanner.nextLine();

bookService.getBook(bookNameToSearch);

break;

case 3:

bookService.showAllBooks();

break;

case 4:

System.out.println("Exiting Library App.");

break;

default:

System.out.println("Invalid choice. Try again.");

}

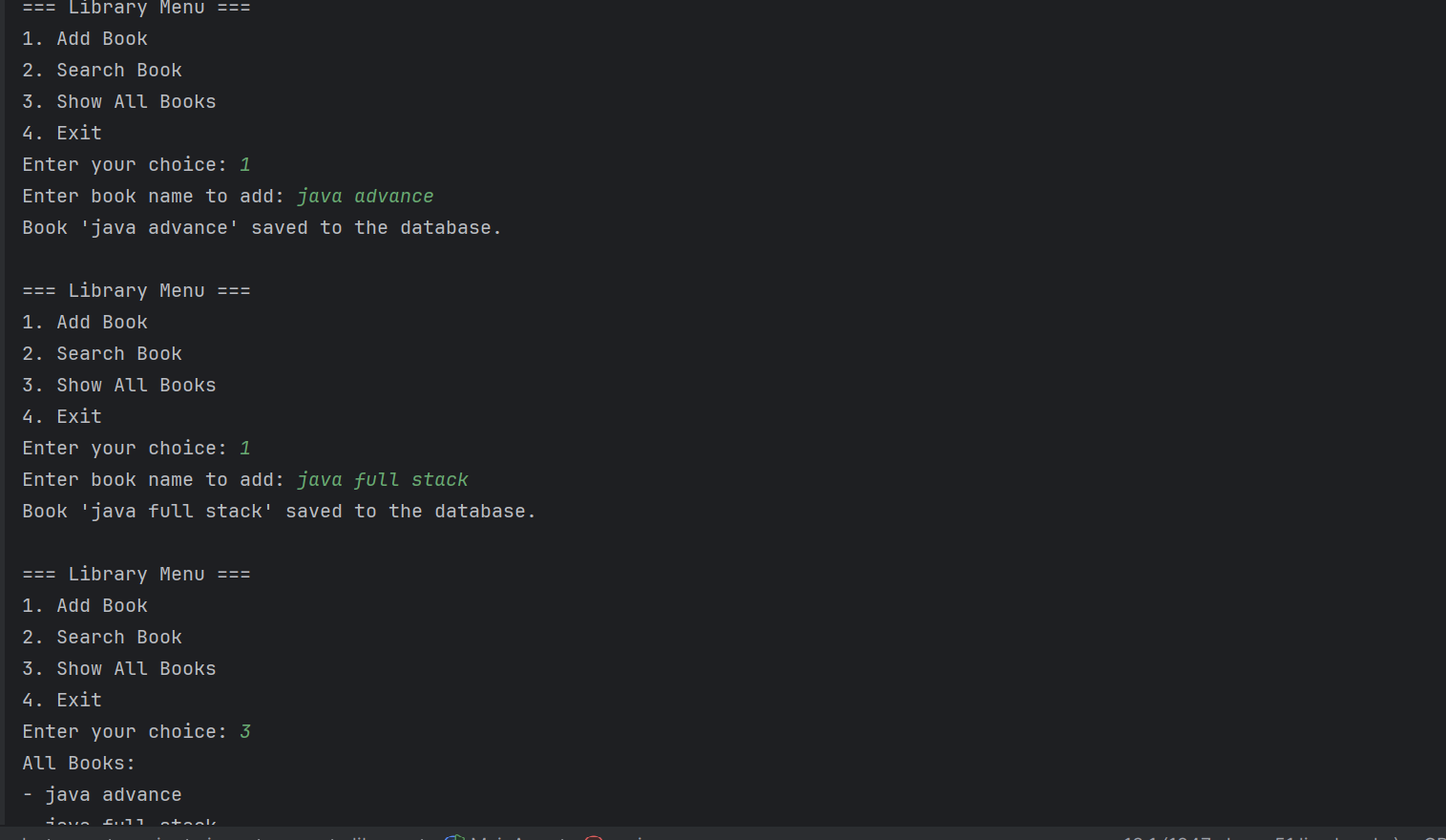
} while (choice != 4);

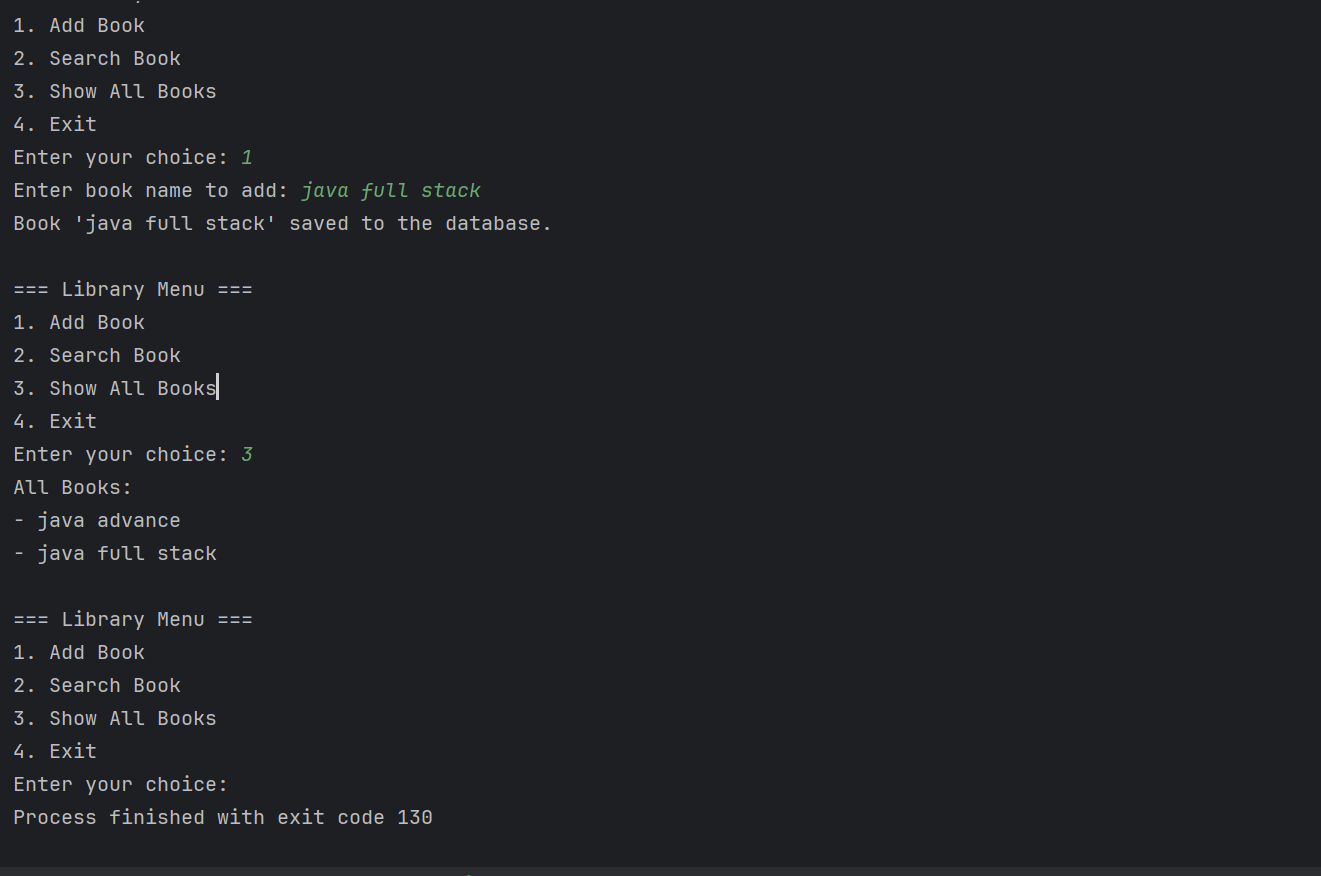
scanner.close();

}

}

**Output:**





**Exercise 2: Implementing Dependency Injection**

**BookRepository:**

package com.library.repository;

import java.util.ArrayList;

import java.util.List;

public class BookRepository {

private List<String> books = new ArrayList<>();

public void saveBook(String bookName) {

books.add(bookName);

System.out.println("Book '" + bookName + "' saved to the database.");

}

public String findBook(String bookName) {

if (books.contains(bookName)) {

return "Book found: " + bookName;

} else {

return "Book not found: " + bookName;

}

}

public List<String> getAllBooks() {

return books;

}

}

**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(String bookName) {

bookRepository.saveBook(bookName);

}

public void getBook(String bookName) {

String result = bookRepository.findBook(bookName);

System.out.println(result);

}

public void showAllBooks() {

System.out.println("All Books:");

for (String book : bookRepository.getAllBooks()) {

System.out.println("- " + book);

}

}

}

**MainApp.java:**

package com.library;

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.util.Scanner;

public class MainApp {

public static void main(String[] args) {

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

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

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a book to add: ");

String book = scanner.nextLine();

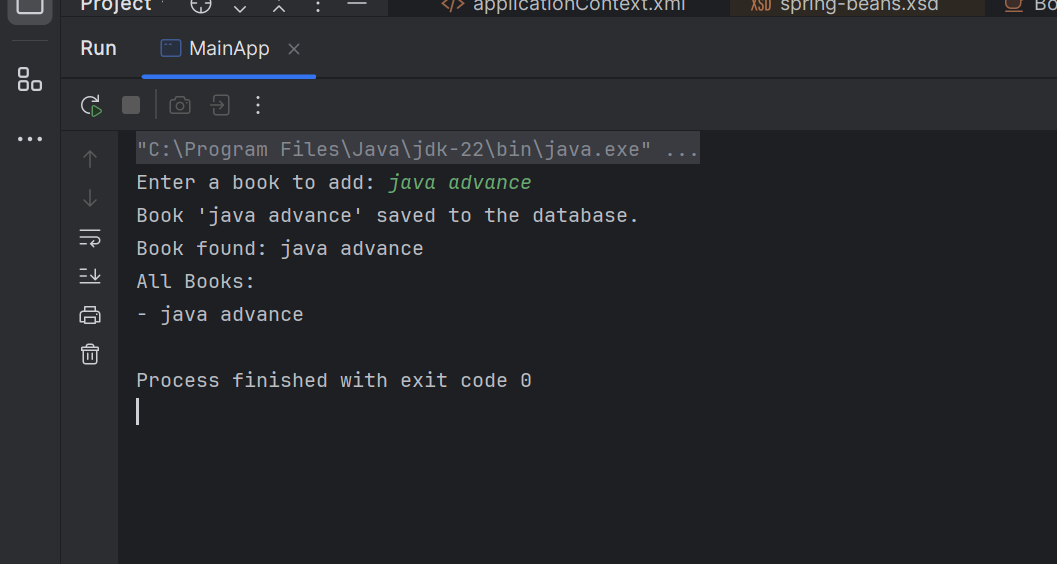
bookService.addBook(book);

bookService.getBook(book);

bookService.showAllBooks();

scanner.close();}}

**Output:**



**Spring Data JPA - Quick Example:**

**Sql:**

CREATE SCHEMA ormlearn;

USE ormlearn;

CREATE TABLE country (

co\_code VARCHAR(2) PRIMARY KEY,

co\_name VARCHAR(50)

);

INSERT INTO country VALUES ('IN', 'India');

INSERT INTO country VALUES ('US', 'United States of America');

**application.properties:**

# Logging

logging.level.org.springframework=info

logging.level.com.cognizant=debug

logging.level.org.hibernate.SQL=trace

logging.level.org.hibernate.type.descriptor.sql=trace

logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %-20.20thread %5p %-25.25logger{25} %25M %4L %m%n

# Database

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn

spring.datasource.username=root

spring.datasource.password=root

# Hibernate

spring.jpa.hibernate.ddl-auto=validate

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

**com.cognizant.ormlearn.model:**

package com.cognizant.ormlearn.model;

import javax.persistence.\*;

@Entity

@Table(name = "country")

public class Country {

@Id

@Column(name = "co\_code")

private String code;

@Column(name = "co\_name")

private String name;

// Getters and Setters

public String getCode() { return code; }

public void setCode(String code) { this.code = code; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

@Override

public String toString() {

return "Country [code=" + code + ", name=" + name + "]";

}

}

**com.cognizant.ormlearn.repository:**

package com.cognizant.ormlearn.repository;

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

import org.springframework.stereotype.Repository;

import com.cognizant.ormlearn.model.Country;

@Repository

public interface CountryRepository extends JpaRepository<Country, String> {

}

**com.cognizant.ormlearn.service**

package com.cognizant.ormlearn.service;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.repository.CountryRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import javax.transaction.Transactional;

import java.util.List;

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

@Transactional

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

}

**OrmLearnApplication.java:**

package com.cognizant.ormlearn;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.service.CountryService;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

import java.util.List;

@SpringBootApplication

public class OrmLearnApplication {

private static final Logger LOGGER = LoggerFactory.getLogger(OrmLearnApplication.class);

private static CountryService countryService;

public static void main(String[] args) {

ApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);

countryService = context.getBean(CountryService.class);

LOGGER.info("Inside main");

testGetAllCountries();

}

private static void testGetAllCountries() {

LOGGER.info("Start");

List<Country> countries = countryService.getAllCountries();

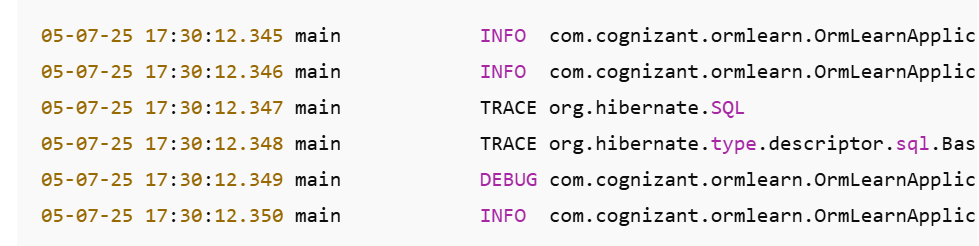
LOGGER.debug("countries={}", countries);

LOGGER.info("End");

}

}

**OUTPUT:**



**Difference between JPA, Hibernate and Spring Data JPA** 

public Integer addEmployee(Employee employee){

Session session = factory.openSession();

Transaction tx = null;

Integer employeeID = null;

try {

tx = session.beginTransaction();

employeeID = (Integer) session.save(employee);

tx.commit();

} catch (HibernateException e) {

if (tx != null) tx.rollback();

e.printStackTrace();

} finally {

session.close();

}

return employeeID;

}

**EmployeeService.java**

java

CopyEdit

@Servicepublic class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public Employee addEmployee(Employee employee) {

return employeeRepository.save(employee);

}

}

**OUTPUT:**

