**Exercise – 9**

**Main.java**

package com.lib.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);

    }

}

**bookServe.java**

package com.lib.LibraryManagement.service;

import com.lib.LibraryManagement.model.Book;

import com.lib.LibraryManagement.repository.BookRepository;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class bookServe {

    private bookRepo bookRepository;

    public bookServe (bookRepo bookRepository) {

        this.bookRepository = bookRepository;

    }

    public List<Book> getAllBooks(){

        return bookRepository.findAll();

    }

    public Book getBookById(int id){

        return bookRepository.findById(id).orElse(null);

    }

    public void addBook(Book book){

        bookRepository.save(book);

    }

    public void updateBook(Book book){

        bookRepository.save(book);

    }

    public void deleteBookById(int id){

        bookRepository.deleteById(id);

    }

}

**bookRepo Interface**

package com.lib.LibraryManagement.repository;

import com.lib.LibraryManagement.model.Book;

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

public interface bookRepo extends JpaRepository<Book,Integer> {

}

**Book.java**

package com.lib.LibraryManagement.model;

import jakarta.persistence.\*;

@Entity

public class Book {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private int bookId; //primary key

    private String bookName;

    private String author;

    public Book() {

    }

    public Book(int bookId, String bookName, String author) {

        this.bookId = bookId;

        this.bookName = bookName;

        this.author = author;

    }

    public int getBookId() {

        return bookId;

    }

    public void setBookId(int bookId) {

        this.bookId = bookId;

    }

    public String getBookName() {

        return bookName;

    }

    public void setBookName(String bookName) {

        this.bookName = bookName;

    }

    public String getAuthor() {

        return author;

    }

    public void setAuthor(String author) {

        this.author = author;

    }

}

**BookController.java**

package com.lib.LibraryManagement.controller;

import com.lib.LibraryManagement.model.Book;

import com.lib.LibraryManagement.service.BookService;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

@RequestMapping("/api")

public class BookController {

    private BookService service;

    public BookController(BookService service) {

        this.service = service;

    }

    @GetMapping("/books")

    public List<Book> getAllBooks() {

        return service.getAllBooks();

    }

    @GetMapping("/books/{bookid}")

    public Book getBookById(@PathVariable int bookid) {

        return service.getBookById(bookid);

    }

    @PostMapping("/books")

    public void addBook(@RequestBody Book book) {

        service.addBook(book);

    }

    @PutMapping("/books")

    public void updateBook(@RequestBody Book book) {

        service.updateBook(book);

    }

    @DeleteMapping("/books/{bookid}")

    public void deleteProduct(@PathVariable int bookid) {

        service.deleteBookById(bookid);

    }

}

**AppTest.java**

package com.lib;

import static org.junit.Assert.assertTrue;

import org.junit.Test;

/\*\*

 \* Unit test for simple App.

 \*/

public class AppTest

{

    /\*\*

     \* Rigorous Test :-)

     \*/

    @Test

    public void shouldAnswerWithTrue()

    {

        assertTrue( true );

    }

}

**application.properties**

spring.application.name=LibraryManagement

spring.datasource.url = jdbc:h2:krupa

spring.datasource.driverClassName = org.h2.Driver

spring.jpa.show-sql = true

spring.jpa.hibernate.ddl-auto = update

**pom.xml**

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

<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.3.2</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.library</groupId>

<artifactId>LibraryManagement</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>LibraryManagement</name>

<description>Library Management System</description>

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>