Book Management System

Submitted by: Yash Bharda

1. Introduction

The Book Management System is a RESTful application developed using Java and Spring Boot. It allows users to manage a collection of books with functionalities like creating, reading, updating, and deleting book records. This backend system can be used for digital libraries, book inventory systems, or educational resource platforms.

2. Objective

The main objective of the Book Management System is to provide a structured and secure way to handle book information. The system allows users to manage book entries through simple API calls and view the API documentation using Swagger UI.

3. Technology Stack

- Java 21

- Spring Boot 3.2.5

- Spring Data JPA

- MySQL

- Maven

- Lombok

- Swagger (Springdoc OpenAPI 2.3.0)

- Postman (for API Testing)

4. System Architecture

The system follows a layered architecture:

• Controller Layer – Handles HTTP requests.

• Service Layer – Contains business logic.
• Repository Layer – Handles database interactions using JPA.
• Entity – Represents the Book table.

Swagger is integrated for API documentation.

5. Module Description

The system includes the following book-related functionalities:

- Add a new book (POST /books)

- View all books (GET /books)

- View a book by ID (GET /books/{id})

- Update a book by ID (PUT /books/{id})

- Delete a book by ID (DELETE /books/{id})

6. Database Design

The Book table contains the following fields:

- id (Primary Key, Long)

- title (String)

- author (String)

- price (Double)

- description (String)

- category (String)

The author table contains the following fields:

- id (Primary Key, Long)

- name (String)

7. Swagger Integration & Testing

Swagger UI is integrated using Springdoc OpenAPI. It allows developers to explore, test, and document APIs in a web interface.

URL: http://localhost:8080/swagger-ui/index.html

Postman is also used to manually test all CRUD endpoints.

8. Conclusion

The Book Management System provides a solid foundation for managing books in a structured and reliable way. With a RESTful architecture and Swagger UI integration, it is developer-friendly and easily extendable for larger applications.