Frontend Coding Challenge Documentation

Name: Kashish K. Jadhav (FSD Java + Angular Training Batch 2)

Book Management System — Project Documentation

1. Project Overview

The **Book Management System** is a full-stack web application designed to manage book records effectively. It supports user registration, login using JWT authentication, and CRUD operations on books. The app is intuitive, responsive, and secure, ideal for libraries or individual use.

2. Objectives

- Provide a simple interface for managing book data
- Ensure secure access using JWT-based authentication
- Implement a responsive UI for ease of use
- Follow scalable architecture using Angular + Spring Boot

3. Technology Stack

Layer	Technology Used
Frontend	Angular 17, TypeScript, HTML, CSS
Backend	Spring Boot (v3+), Java 17, REST API
Database	MySQL
Security	Spring Security + JWT Authentication
Build Tool	Maven
IDEs	VS Code (Frontend)
Version Control	Git + GitHub

4. Backend Structure (Spring Boot)

controller

Handles REST API endpoints like /api/auth/**, /api/books/**

model

Contains data entities like Book, User, and Role

repository

Interfaces extending JpaRepository for DB operations

service

Contains business logic (e.g., BookService, UserService)

security

JWT authentication logic (JwtFilter, JwtUtil, SecurityConfig)

config

CORS configuration (CorsConfig.java)

5. Frontend Structure (Angular)

auth/

Components for login and registration

books/

Components for displaying, adding, editing, deleting books

shared/

Reusable components like Navbar, Footer

core/

Services and interceptors (e.g., AuthService, TokenInterceptor)

pipes/

Custom pipe for filtering books (e.g., search by title, author, etc.)

guards/

Route guards (optional for authenticated route protection)

6. Database

RDBMS: MySQL

Script File: BookManagement_Script.sql

Tables:

- users stores user credentials
- books stores all book records

You can find the DB script in the database-script/ folder in the GitHub repo.

7. Security (JWT)

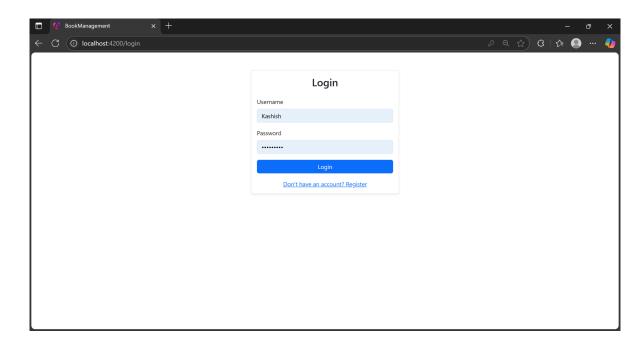
- On login, a JWT token is generated and returned to the frontend.
- The token is stored in localStorage and sent via an Authorization Header for protected routes.
- The backend validates token expiry and roles before processing requests.

8. UI Screenshots

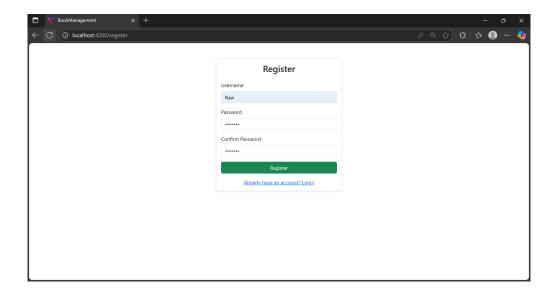
Backend Project Structure:



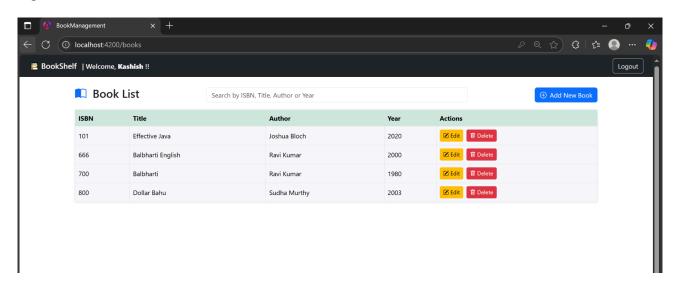
W Login Page



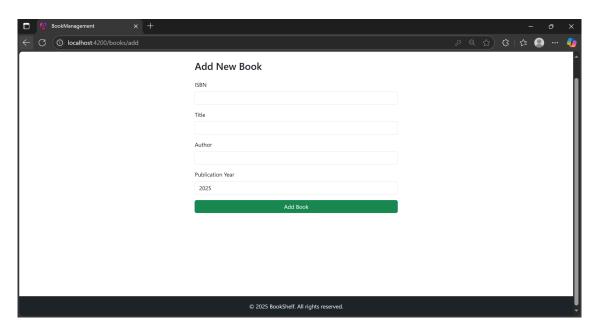
Register Page

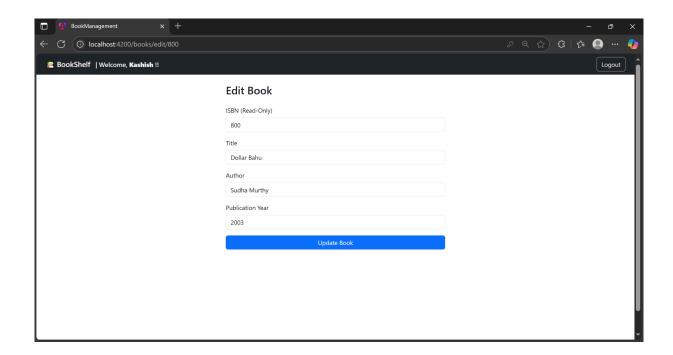


See Book List

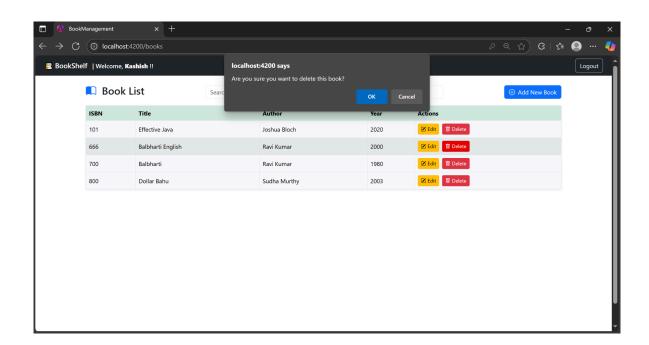


+ Add Book

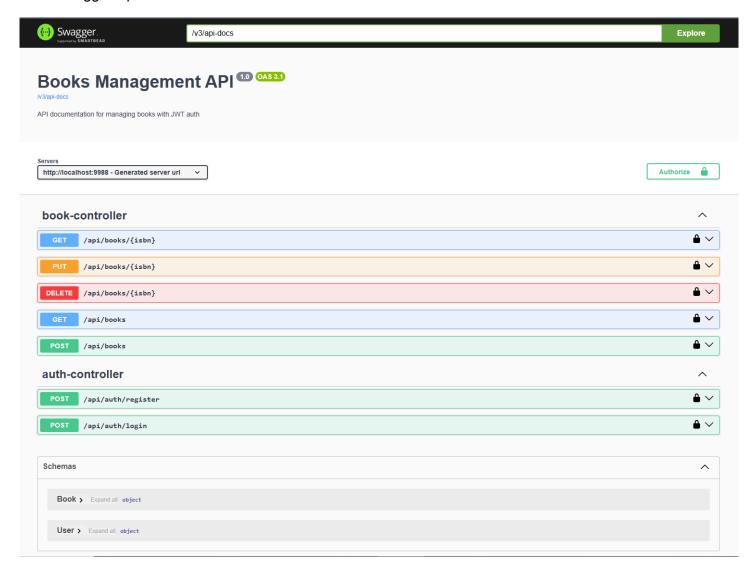




Delete Confirmation



Swagger Api controller interface



9. Notable Features

- Responsive design using Bootstrap 5
- Custom pipe for live search filtering
- Proper error handling and validations
- Navigation guard via token presence
- Token interceptor auto-attaches JWT

10. Conclusion

The Book Management System demonstrates the practical use of modern web technologies to build a full-stack secure and user-friendly application.



- GitHub Backend Repo: <u>BooksManagementSystem</u>
- GitHub Frontend Repo: <u>BookManagementFrontend</u>