# E-Book Management System

**Java and Spring Framework Lab (CS3603-1)**

***Mini Project Report***

***submitted by***

**Bhushith Umesh Amin**

(NNM22CS042)

**Deepit S Nayak**

(NNM22CS058)

***Under the Guidance of***

**Mr. AKHILRAJ V. GADAGKAR**
Assistant Professor Gd.-II

*In partial fulfillment of the requirements for the award of*

*the Degree of*

**Bachelor of Technology in Computer Science and Engineering**

*from*

**NITTE (Deemed to be University), Mangalore**

Department of Computer Science and Engineering

NMAM Institute of Technology, Nitte - 574110


2024-2025

NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

# CERTIFICATE

*Certified that the Mini project work entitled*

***E-Book Management system***

*is a bonafide work carried out by*

Bhushith Umesh Amin (NNM22CS042)

Deepit S Nayak (NNM22CS058)

*in partial fulfillment of the requirements for the award of*

**Bachelor of Engineering Degree in Computer Science and Engineering**

**prescribed by NITTE (Deemed to be University)**

during the year **2024-2025**.

*The project report has been approved as it satisfies*

*the academic requirements in respect of the project*

*work prescribed for the Bachelor of Engineering*

*Degree.*

_____

**Signature of the Guide**

# ABSTRACT

The E-Book Management System is a comprehensive solution designed to simplify the management of book inventories for both administrators and users. The system leverages Spring Boot, Thymeleaf, and Bootstrap to provide a robust backend and a user-friendly frontend interface.

The platform offers CRUD (Create, Read, Update, Delete) operations for book records, enabling administrators to manage book data effectively. A secure admin login feature ensures restricted access to administrative functions. For users, the system provides intuitive search and filtering options, enhancing their ability to explore and find books based on various criteria like title, author, or category.

The application employs an H2 in-memory database for storing book information, ensuring seamless and efficient data handling. It also showcases responsive design principles, making the interface accessible across multiple devices.

This project highlights modern software development practices by integrating structured database management, scalable service architecture, and a dynamic user experience. Furthermore, it lays the groundwork for future improvements, such as integrating a persistent database, advanced role-based access control, and features like payment systems and personalized recommendations.

The system is a reliable tool for managing e-books, providing convenience to users and ensuring streamlined operations for administrators.

# TABLE OF CONTENTS

# INTRODUCTION

Managing books in the digital era has become a critical aspect of libraries, bookstores, and personal collections. With the proliferation of e-books and the need for effective cataloging, a streamlined solution is essential. The E-Book Management System is developed to address these challenges, offering a web-based platform that simplifies book management tasks.

This system enables administrators to perform CRUD (Create, Read, Update, Delete) operations on book records, allowing them to manage inventory efficiently. It also provides a user-friendly interface for customers to browse the available collection, search for books by title or category, and access detailed information about each book.

Built using Spring Boot and Thymeleaf, the application combines powerful backend functionality with dynamic and responsive frontend design. It uses an in-memory H2 database for lightweight and efficient data handling, ensuring smooth operation during development and testing phases.

This project not only caters to the need for streamlined book management but also provides a foundation for future enhancements like user authentication, advanced search filters, and real-time inventory tracking. It is a versatile and scalable solution for modern-day book management needs.

# PROBLEM STATEMENT

Traditional methods of managing books can be inefficient, especially when dealing with a large number of records. Manual methods are prone to errors and lack real-time search capabilities. The absence of a centralized system can lead to delays and mismanagement of book data. Therefore, there is a need for an automated system to manage books effectively, providing robust features for both administrators and end-users.

# OBJECTIVES

The primary objective of the E-Book Management System is to streamline the management of book inventories while providing an intuitive interface for both administrators and users. The system's goals include:

**1. Simplified Book Management**
Enable administrators to easily add, edit, view, and delete book records through a user-friendly interface, reducing the complexity of inventory management.

**2. Efficient Search and Retrieval**
Provide users with the ability to search for books using various parameters, such as title, author, or category, ensuring quick access to desired information.

**3. Organized Categorization**
Facilitate better organization of books by categorizing them based on their genre or subject, making it easier for users to navigate through the collection.

**4. Scalable and Reliable Infrastructure**
Leverage modern technologies such as Spring Boot and H2 Database to create a robust and scalable backend, ensuring seamless data handling and operational efficiency.

**5. Responsive and Dynamic User Interface**
Incorporate Thymeleaf templates for dynamic content rendering and CSS-based responsive designs to deliver a visually appealing and consistent user experience across devices.

**6. Support for CRUD Operations**
Implement core CRUD functionalities to enable administrators to manage book details, such as title, author, ISBN, price, quantity, and other attributes, efficiently.

**7. Future-Ready Design**
Lay the groundwork for future enhancements such as user authentication, book recommendations, purchase functionality, and integration with third-party e-commerce platforms.

**8. Centralized Data Management**
Provide a centralized system to store all book data securely and retrieve it effortlessly when needed, improving accuracy and data consistency.

**9.Admin and User Role Differentiation**

Incorporate distinct interfaces and features for administrators and end-users, ensuring that each role has access to tools tailored to their specific needs.

By achieving these objectives, the E-Book Management System not only optimizes the process of managing books but also ensures a smooth experience for end-users, fostering greater satisfaction and usability.

# METHODOLOGY

The **E-Book Management System** was developed using a structured and systematic approach to ensure efficiency and scalability. The following steps outline the methodology employed in the project:

### 1. Requirement Analysis

- Conducted a detailed analysis of functional requirements for both user and admin interfaces.
- Identified the need for CRUD operations, search functionality, and secure admin access.
- Evaluated user experience requirements to ensure ease of navigation and responsiveness.

### 2. Technology Selection

- **Backend**: Choose **Spring Boot** for its simplicity, flexibility, and integration capabilities.
- **Frontend**: Used **Thymeleaf** as the template engine for dynamic web content rendering.
- **Database**: Implemented **H2 Database**, an in-memory database, for its simplicity during development and testing.
- **Styling and Responsiveness**: Incorporated **CSS** and **Bootstrap** for a polished and adaptive user interface.

### 3. System Development

- **Modular Design**: Developed the application using a modular approach to simplify development and testing.
    - Created separate controllers for handling user and admin functionalities.
    - Implemented a service layer (BookService) to handle business logic and interact with the database via the repository layer (BookRepository).
- **User Interface Implementation**:
    - Designed pages for users to view and search books dynamically.
    - Developed admin-specific pages for book management operations like adding, editing, updating, and deleting.
- **Admin Login Setup**: Implemented a secure access mechanism for administrators to perform inventory management tasks.

### 4. Testing and Validation

- Conducted unit testing on individual modules such as book creation, search functionality, and admin operations.
- Performed integration testing to validate interactions between frontend and backend components.
- Verified data consistency and ensured the application met functional and non-functional requirements.

### 5. Deployment

- Configured the application to run on port 8081.
- Enabled H2 console for real-time monitoring and debugging during development.

### Additional Details

- **Dynamic Search Feature**: Designed to allow users to search books using keywords for titles, authors, and categories.
- **Book Categorization**: Enabled browsing books based on categories, simplifying the user experience.
- **Error Handling**: Implemented error-handling mechanisms to manage invalid operations, such as attempting to edit or delete non-existent books.
- **Logging and Debugging**: Configured logging levels (DEBUG and TRACE) for Hibernate queries to facilitate debugging and ensure efficient SQL execution.

This comprehensive methodology ensured the development of a robust and scalable e-book management system capable of meeting user and admin needs effectively.

# SYSTEM DESIGN

The **E-Book Management System** is designed to provide a seamless and user-friendly platform for managing books while incorporating a secure admin interface for inventory control. The system uses a layered architecture to ensure modularity, scalability, and maintainability. Below is an in-depth description of the system design:

## 1. Architecture

The system is built on a **three-layered architecture**, comprising:

#### a. Presentation Layer (Frontend)

- **Technologies Used**: HTML, CSS, Bootstrap, and Thymeleaf.
- **Functionality**:
    - Provides user-friendly interfaces for interacting with the system.
    - Includes dynamic web pages for:
        - Displaying books for users.
        - Search and filtering options.
        - Admin functionalities like adding, updating, and deleting books.
    - Ensures responsive design for accessibility across devices.

#### b. Business Logic Layer (Service Layer)

- **Technologies Used**: Spring Boot with Service Classes.
- **Functionality**:
    - Encapsulates all business logic related to book management.
    - Handles operations such as book creation, updating inventory, and executing search queries.
    - Acts as an intermediary between the presentation layer and the database layer.

#### c. Data Access Layer (Repository Layer)

- **Technologies Used**: Spring Data JPA with H2 Database.
- **Functionality**:
    - Manages database interactions through the `BookRepository` interface.
    - Handles CRUD operations and query execution using JPA methods.

## 2. Database Design

The system uses the **H2 in-memory database** for development and testing purposes.

#### Key Tables

- **Book Table**: Stores book-related data such as ID, title, author, ISBN, price, quantity, category, and publisher.
- **Admin Table**: Stores admin credentials for secure access (future expansion).

#### Relationships

The system is currently designed with a flat table structure, making it simple yet effective for managing book-related data. In future iterations, relationships can be extended to include user profiles and purchase history.

## 3. User Roles

#### User Interface

- Accessible to all visitors.
- Features:
    - Browse books.
    - Search books by title, author, or category.
    - View book details.

#### Admin Interface

- Restricted to administrators with secure login credentials.
- Features:
    - Add new books to the inventory.
    - Update book details (e.g., price, quantity, or description).
    - Delete books from the inventory.
    - Manage book categories for better organization.

## 4. Key System Features

#### a. Search and Filtering

- Users can search books using keywords (e.g., title, author).

- Books can be filtered by category or other attributes.

#### b. CRUD Operations

- **Create**: Admins can add new books with all relevant details (e.g., title, author, price).
- **Read**: Users can view the list of books, along with details like price and availability.
- **Update**: Admins can edit book details.
- **Delete**: Admins can remove books that are no longer available.

#### c. Secure Admin Access

- Admin login is implemented to restrict access to sensitive functionalities like inventory management.

#### d. Real-Time Database Interaction

- H2 Database provides an in-memory platform for efficient data handling during development.
- The Spring JPA configuration ensures smooth database queries and updates.

## 5. System Flow

- **User Actions**:
  - Users visit the homepage to browse or search books.
  - Upon searching, the system retrieves results from the database using optimized queries.
- **Admin Actions**:
  - Admins log in to the system to manage book inventory.
  - Actions such as adding, editing, or deleting books trigger updates to the database via service and repository layers.
- **Data Handling**:
  - Data is dynamically fetched and rendered on the user and admin interfaces using Thymeleaf templates.

This system design ensures a seamless experience for both users and administrators, with a clear separation of responsibilities between the different layers. It provides a solid foundation for future enhancements and scalability.

# IMPLEMENTATION DETAILS

The **E-Book Management System** is implemented using **Spring Boot** as the core framework, providing a robust backend structure, and **Thymeleaf** for rendering dynamic front-end templates. Below are the detailed implementation aspects of the system:

## 1.Core Features

- **CRUD Operations for Books**:
  - Managed by BookService and BookRepository.
  - Enables administrators to **Create**, **Read**, **Update**, and **Delete** book records.
  - Operations include adding new books, updating details (e.g., price, stock), and removing outdated entries.
  - Seamlessly integrates with the database using **Spring Data JPA** for efficient data handling.
- **Search Functionality**:
  - Implements findByTitleContainingIgnoreCase and similar repository methods to allow flexible searching.
  - Search by keyword (e.g., title, author, or category) for efficient book retrieval.
  - Results are dynamically displayed to users with pagination for better user experience.
- **Category-Based Filtering**:
  - Books are organized into categories, allowing users to browse by specific genres or themes.
  - Uses repository methods like findByCategory to retrieve relevant results.

## 2.Admin Login

- **Secured Access to Administrative Pages**:
  - Access to administrative features (like inventory management) is restricted to authenticated users.
  - Future iterations can implement Spring Security for role-based authentication and session management.
  - Ensures unauthorized users cannot alter or view sensitive inventory data.
- **Admin Dashboard**:

- Provides intuitive navigation for managing books.
- Admin can quickly add, update, or remove books using streamlined forms and action buttons.

### 3.Database Configuration

- **H2 In-Memory Database Setup**:
  - Used for development and testing to provide a lightweight and fast environment.
  - Configured with:
    - URL: jdbc:h2:mem:bookstoredb.
    - Driver: org.h2.Driver.
    - JPA Properties: Enables automatic schema generation and SQL visibility.
- **Entity Mapping**:
  - Book entities are mapped to the database schema using annotations like @Entity, @Id, and @GeneratedValue.
  - Supports fields like title, author, ISBN, price, quantity, category, and image URL.
- **Scalability Considerations**:
  - The database can easily transition to persistent databases like MySQL, PostgreSQL, or MongoDB for production use.

### 4.User Interface

- **User-Friendly Design**:
  - Designed using **HTML**, **CSS**, and **Bootstrap** for responsiveness and aesthetic appeal.
  - Ensures smooth navigation for both users and admins.
- **Dynamic Web Pages**:
  - Integrated with **Thymeleaf** to dynamically render data from the backend.
  - Supports features such as:
    - Displaying book details in a grid format.
    - Form-based input for adding and editing books.
- **Bootstrap Integration**:
  - Ensures responsive design across various devices (desktop, tablet, mobile).
  - Includes modern UI elements like cards, buttons, and forms for enhanced

usability.

- **Search and Filter**:
  - The search bar allows users to quickly locate books by title.
  - Filters enable browsing by author or category.

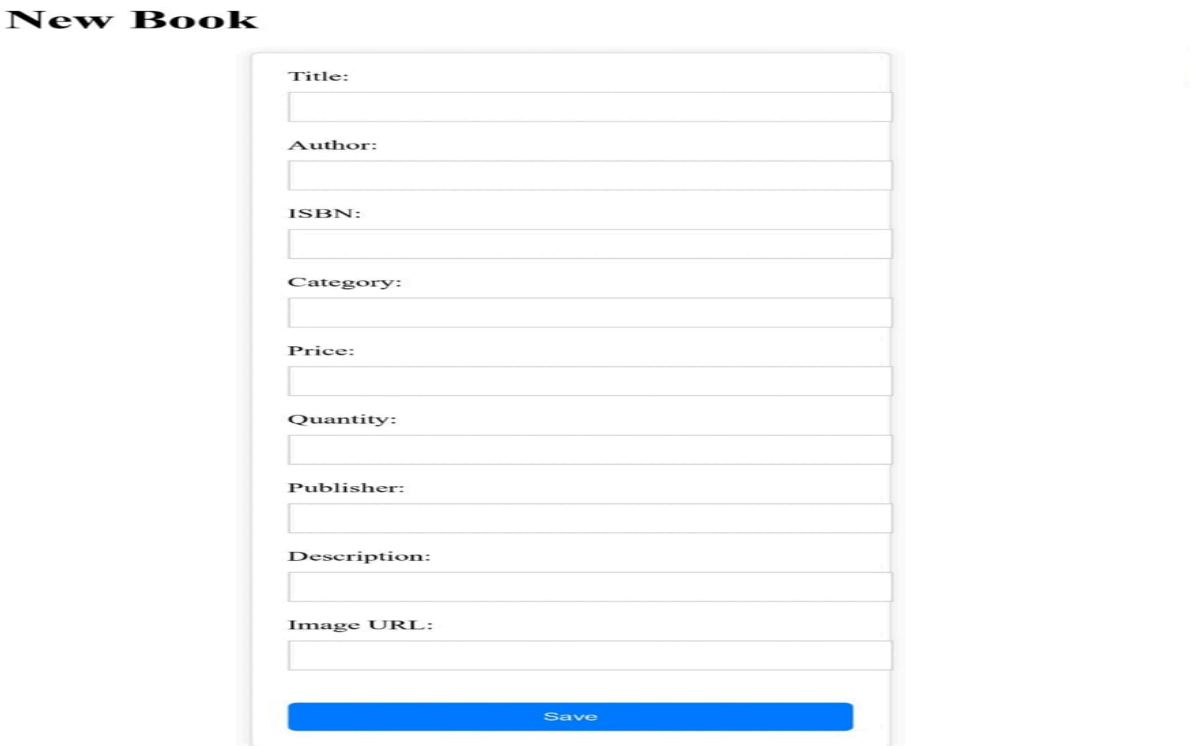
### 5.Additional Functionalities

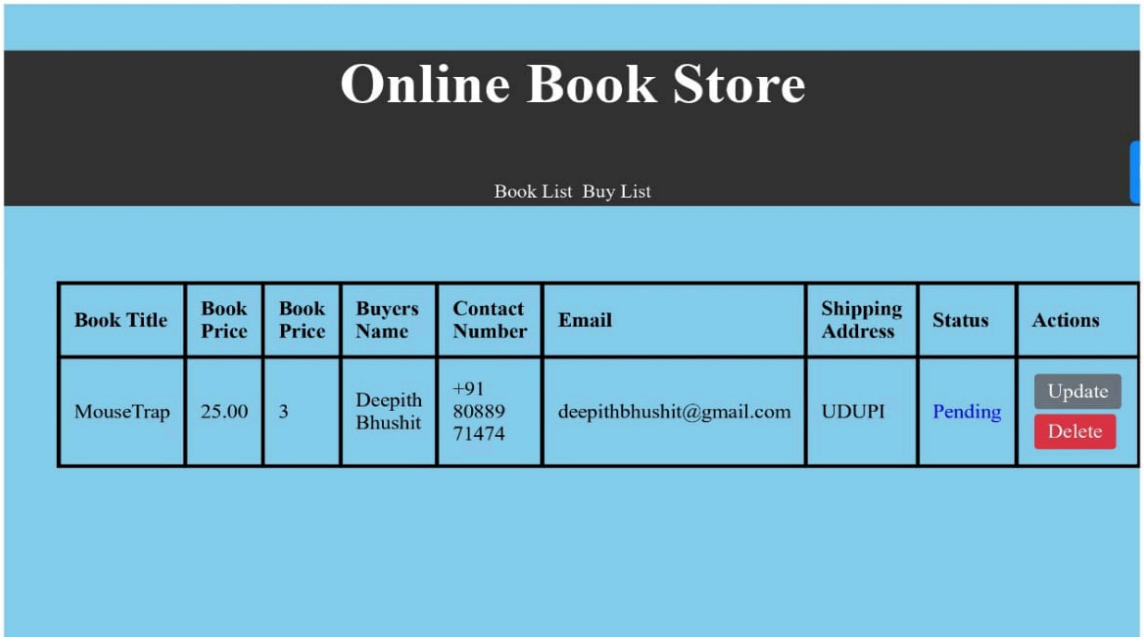
- **Real-Time Updates**: Admin changes (e.g., adding or editing books) are immediately reflected on the user-facing interface.
- **Error Handling**: Graceful handling of scenarios like missing books (e.g., invalid book ID) with user-friendly error messages.
- **Static Resource Management**: Uses static folder for storing CSS, JS, and image assets, ensuring fast loading of UI components.

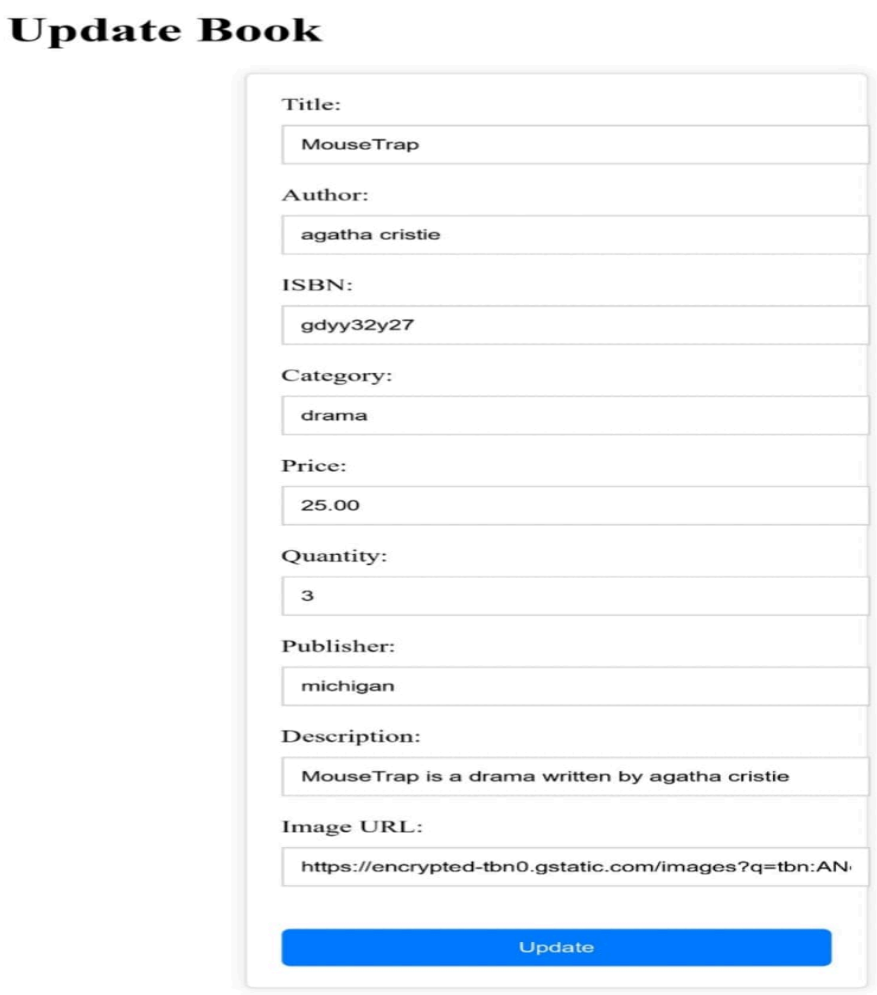
The implementation ensures a robust, modular, and user-friendly system that balances ease of use with efficient data handling. The inclusion of scalable technologies ensures the system is ready for future enhancements.

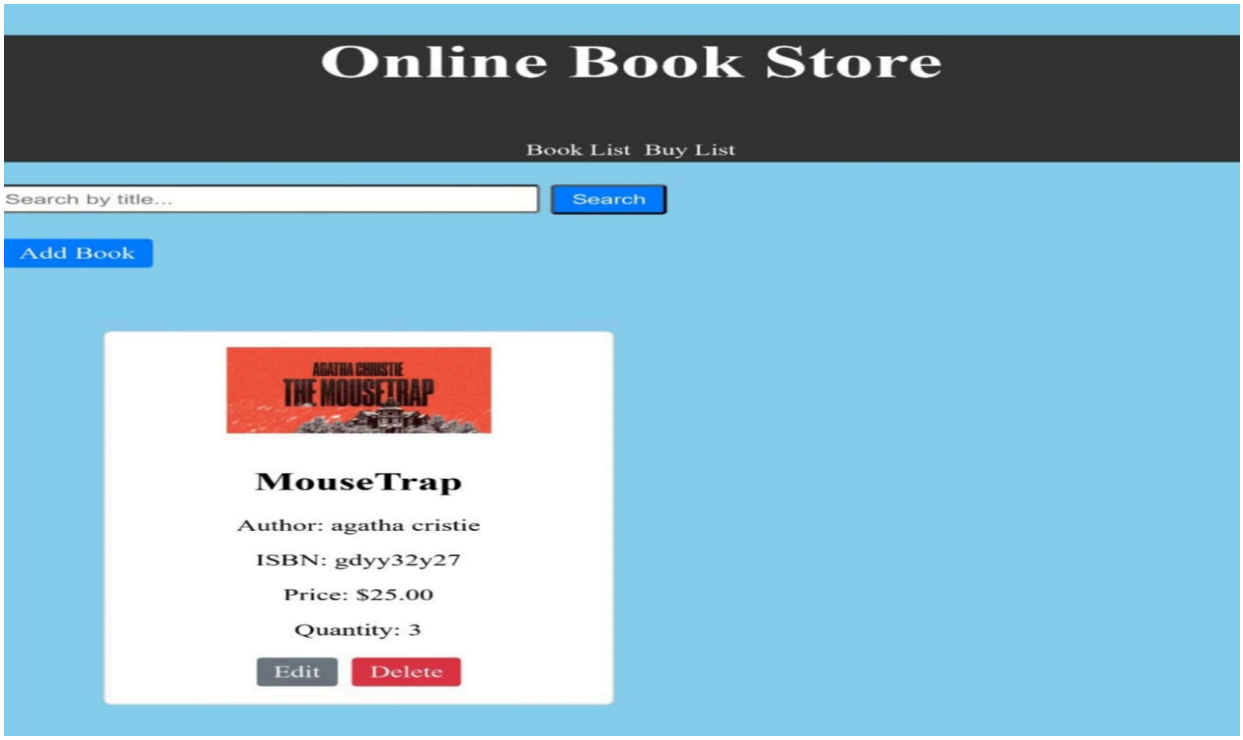
# RESULTS

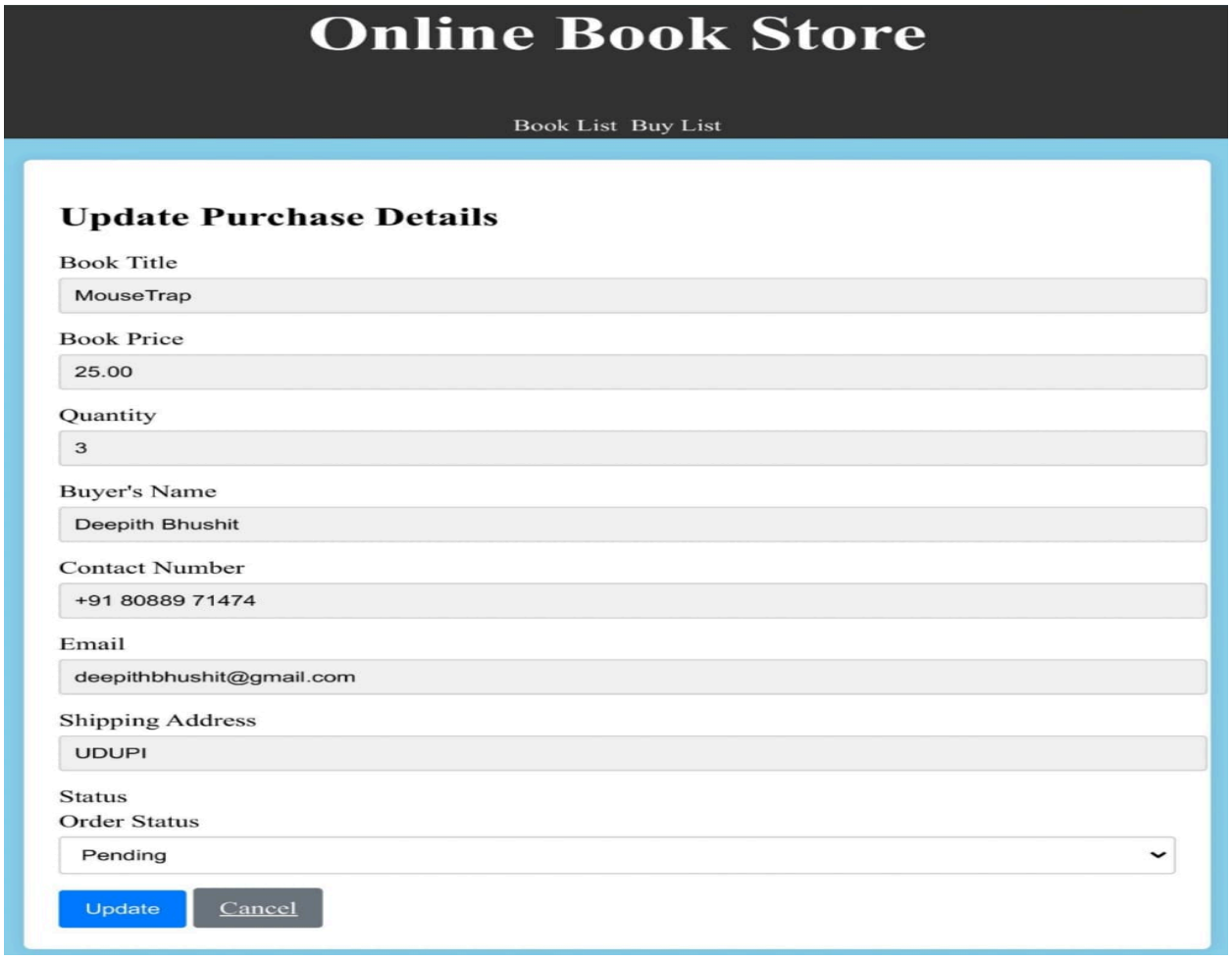

Online Book Store
Admin Login
Username:
Password:
Login
© 2024 Online Book Store. All rights reserved.


New Book
Title:
Author:
ISBN:
Category:
Price:
Quantity:
Publisher:
Description:
Image URL:
Save


Online Book Store
Book List Buy List
Book Title
Book Price
Book Price
Buyers Name
Contact Number
Email
Shipping Address
Status
Actions
MouseTrap
25.00
3
Deepith Bhushit
+91 80889 71474
deepithbhushit@gmail.com
UDUPI
Pending
Update
Delete


Update Book
Title:
MouseTrap
Author:
agatha cristie
ISBN:
gdyy32y27
Category:
drama
Price:
25.00
Quantity:
3
Publisher:
michigan
Description:
MouseTrap is a drama written by agatha cristie
Image URL:
https://encrypted-tbn0.gstatic.com/images?q=tbn:AN
Update


Online Book Store
Book List Buy List
Search by title...
Search
Add Book
AGATHA CHRISTIE
THE MOUSETRAP
MouseTrap
Author: agatha cristie
ISBN: gdyy32y27
Price: $25.00
Quantity: 3
Edit
Delete

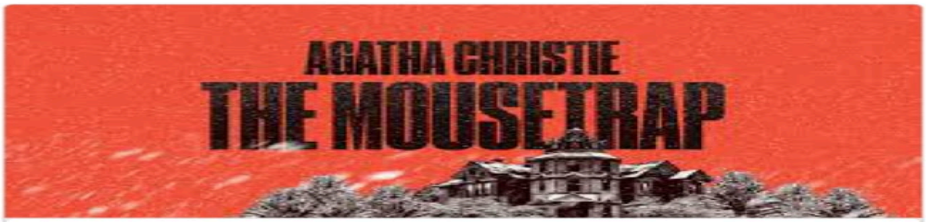

Online Book Store
Book List Buy List
Update Purchase Details
Book Title
MouseTrap
Book Price
25.00
Quantity
3
Buyer's Name
Deepith Bhushit
Contact Number
+91 80889 71474
Email
deepithbhushit@gmail.com
Shipping Address
UDUPI
Status
Order Status
Pending
Update
Cancel

BookStore

Welcome to the Online Book Store

Explore our vast collection of books and find your next great read!

Books Collection




MouseTrap

Author: agatha cristie

ISBN: gdyy32y27

Price: $25.00

Quantity: 3

Buy Now



BookStore

Complete Your Purchase

Book Title

MouseTrap

Book Price

25.00

Quantity

3

Your Name

Contact Number

Email Address

Shipping Address

Enter your address

Buy Now

# CONCLUSION AND FUTURE WORK

The **E-Book Management System** has successfully streamlined book inventory management, incorporating core functionalities such as CRUD operations, secure admin logins, and intuitive user interfaces. This ensures an efficient and scalable platform for handling e-book data.

The use of technologies like Spring Boot, Thymeleaf, and H2 Database has facilitated a robust and responsive design. These choices ensure ease of use for users and maintainability for developers.

The system effectively addresses the gap between traditional book management and digital automation, enhancing accessibility and usability for administrators and readers alike.

**Future Enhancements**:

- **Advanced User Management**: Introduce user roles, including customer profiles and personalized recommendations.
- **Payment Gateway Integration**: Implement secure online payment systems to allow users to purchase e-books directly.
- **Data Persistence**: Transition from an in-memory database to a persistent database for long-term storage.
- **Cloud Deployment**: Deploy the system to cloud platforms for better scalability and accessibility.
- **Mobile Application Development**: Extend functionality to mobile platforms, ensuring greater accessibility for users on the go.

The project lays a solid foundation for a fully functional e-library, showcasing its potential for further innovation and addressing the growing demand for digitized book management systems. With continuous updates and feature enhancements, the system can become an all-encompassing platform for e-book management in various domains.

# REFERENCES


- Spring Boot Documentation - https://spring.io/projects/spring-boot
- Thymeleaf Documentation - https://www.thymeleaf.org
- H2 Database Documentation - https://h2database.com
- Bootstrap CSS Framework - https://getbootstrap.com