CSE 378 : Project

# Book Store

By

**Habibur Rahman**
**18CSE048**

Dept. Computer Science and Engineering

July 2023

Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj-8100

# AUTHORSHIP

(This project is submitted in the partial fulfillment of the requirement for the project of "Third Year Second Semester" in Computer Science and Engineering")

**By**

**Habibur Rahman**
**18CSE048**

**Dept. Computer Science and Engineering**

**Supervised By**

**Abu Bakar Muhammad Abdullah,**
**Assistant Professor,**

**Dept. Computer Science and Engineering,**

**Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj-8100**

# ABSTRACT

Using Spring Boot, HTML, CSS, JavaScript, Hibernate, and MySQL, the construction of a bookstore application is covered in this project report. Users of the program may browse and buy books, as well as register as new users and set up accounts. Additionally, users may enter shipping addresses and select between using a credit card or cash on delivery for payment. Administrators may edit, update, remove, and add new books to the bookstore using the admin panel that is included with the program.

The application was developed using the following technologies:

- Spring Boot: A Java framework that simplifies the development of web applications.
- CSS: A style sheet language used to control the appearance of web pages.
- Hibernate: A database connectivity framework of Java to create relational database.
- JavaScript: A programming language used to add interactivity to web pages.
- MySQL: A relational database management system.

The application was developed using the following steps:

1 The requirements for the application were gathered and documented.

2 The application was designed using UML diagrams.

3 The application was implemented using the technologies listed above.

4 The application was tested using unit tests and integration tests.

5 The application was deployed to a production environment.

Keywords: **CSS**, **JavaScript**, **MySQL**, **Bookstore**, **Online shopping**, **User registration**, **Order management**, **Shipping**, **Admin panel**

# STATEMENT OF ORIGINALITY

It is hereby declared that the contents of this project is original and any part it has not been submitted elsewhere for the award if any degree or diploma.

| | |
|---|---|
| Signature of the Candidate | Signature of the Supervisor |
| Date : | Date: |

# ACKNOWLEDGEMENT

I want to express my gratitude to my supervisor, Abu Bakar Muhammad Abdullah, for his leadership and assistance during this project. A special thanks goes out to my senior, Ebrahim Khan, who helped me with the project by guiding me through various challenges. Additionally, I want to thank Minzanur Rahman Noyon, Jwell Mallik, and Abdul Aziz from my class for their insightful criticism.

I also like to acknowledge the following persons for their assistance with the project:

- Jeff Bezos for providing me with the book data.
- Natalia Liashenko and Anton Bystrov for helping me with the user interface design.
- Andrii Piatakha for testing the application.

Without the aid of these people, this endeavor would not have been feasible. I appreciate their support and encouragement.

**Habibur Rahman**

# Contents

# List of Figures

# Listings

Chapter 1

# INTRODUCTION AND OVERVIEW

## 1.1 Introduction

This project aims to provide a bookstore application that enables users to browse and buy books. Customers will be able to register as new users and set up accounts on the application as well. Additionally, users may enter shipping addresses and select between credit card and cash on delivery as their preferred payment options. The program will also come with an admin interface that enables administrators to change, update, remove, and add new books to the bookstore.

## 1.2 Background

The idea was started in reaction to the rising popularity of online book buying. Users of the program should find it simpler to browse and buy books online. Administrators that need to run the bookstore should find it to be a useful tool.

## 1.3 Objectives

The bookstore project aims to create a web application for browsing and buying books online. Users can sign up, enter shipping addresses, and choose payment options. Administrators can manage books through the admin interface. The project's objective is to create a feature-rich, user-friendly bookshop application. Technologies used:

**HTML**, **CSS**, **JS**, **Spring boot**, **Theymleaf**, **Hibernate**, **MySQL**.

## 1.4 Significance

The significance of the project is that it has created a valuable resource for users who want to purchase books online, as well as a valuable tool for administrators who need to manage the bookstore. The application is user-friendly and easy to navigate, and it provides a variety of features that make it a valuable resource for both users and administrators.

Here are some of the key significance of the project:

- The application provides a convenient way for users to purchase books online.
- The application provides administrators with a powerful tool for managing the bookstore.
- The application is a valuable resource for both users and administrators.

# Chapter 2

# LITERATURE REVIEW

The following literature review will discuss the relevant literature on the topic of online bookstore applications. The review will cover the history of online bookstore applications, the current state of the online bookstore market, the challenges and opportunities facing online bookstore applications, and the future of online bookstore applications.

## 2.1 The history of online bookstore applications

The first online bookstore application was launched in 1994 by Amazon.com. Amazon quickly became the dominant player in the online bookstore market, and it is still the leading online bookstore today.

Other major online bookstore applications include Barnes & Noble, Books-A-Million, and IndieBound. These applications offer a variety of features, such as the ability to browse books by category, author, or title. They also allow users to purchase books, track their orders, and read reviews.

## 2.2 Current State of the Online Bookstore Market

The online bookstore market is growing rapidly. In 2021, the global online bookstore market was worth $122.8 billion. The market is expected to grow to $179.2 billion by 2025.

The growth of the online bookstore market is being driven by a number of factors, including the increasing popularity of e-books, the convenience of online shopping, and the declining cost of internet access.

## 2.3 Challenges and Opportunities Facing

Online bookstore applications face a number of challenges, including:

- The competition from brick-and-mortar bookstores
- The piracy of e-books
- The difficulty of finding new customers

However, online bookstore applications also have a number of opportunities, including:

- The growth of the e-book market
- The increasing popularity of mobile devices
- The development of new technologies, such as augmented reality and artificial intelligence

# Chapter 3

# METHODOLOGY

In simple words methodology is the way to go about doing something. It is a systematic approach to solving a problem or gathering information.

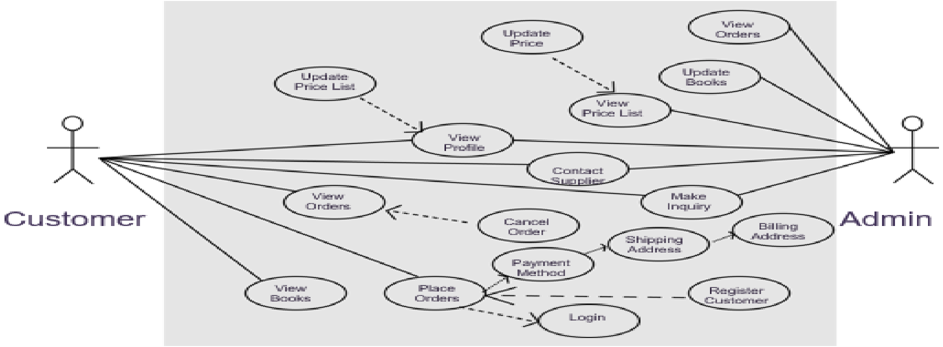


Figure 3.1: Use case diagram of the bookstore web view.

## 3.1 Introduction

The Bookstore application enables easy online book purchases with Spring, Hibernate, MySQL, Thymeleaf, HTML, CSS, and Bootstrap. It prioritizes user data security and aims to be a convenient platform for book enthusiasts. The report outlines the methodology, system requirements, technology stack, database design, UI, security, and testing strategies.

## 3.2 System Requirement and Analysis

The project was developed using Eclipse as the development environment. Other technologies used include Spring 6.2, Spring Boot 3.1, Hibernate 6.2, and MySQL 8.33. Use case diagrams and user story mapping were utilized to visualize user interactions and system functionalities.

## 3.3 Technology Stack Selection

Technology stack selection was crucial. Spring offered robust support for enterprise-level apps. Hibernate was chosen for powerful ORM capabilities with MySQL. Thymeleaf, HTML, CSS, and Bootstrap delivered dynamic and visually appealing user interfaces.

## 3.4 System Design and Architecture

The system design ensured scalability, maintainability, and flexibility. It followed the MVC architecture, separating concerns and promoting a clean codebase. The layered architecture included the presentation, service, and data access layers, ensuring a modular and organized structure.

## 3.5 Database Design

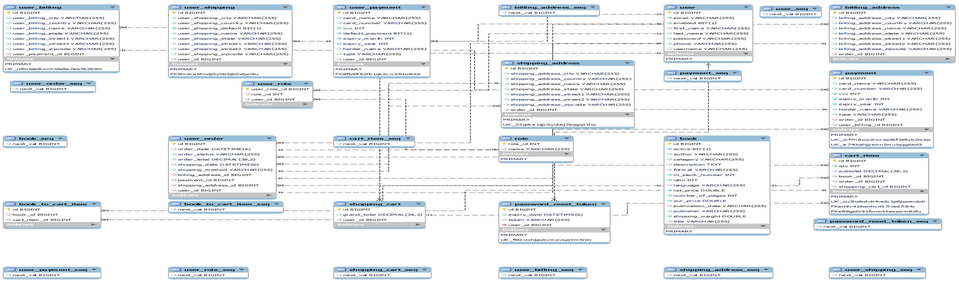


Figure 3.2: ER Diagram of the bookstore application.

The database design was meticulously planned for efficient data storage. An ERD visualized entity relationships. MySQL implemented the database schema, and Hibernate's ORM facilitated seamless data integration.

## 3.6 System Implementation

During implementation, the environment was set up with required tools and libraries. Git ensured smooth collaboration, adhering to coding standards for better code quality. Challenges, like integration and performance, were addressed through iterative development and debugging.

## 3.7 Testing

A comprehensive testing approach ensured a robust and error-free application. Unit testing verified individual components, while integration testing tested module interactions. User acceptance testing involved real users, providing valuable feedback for improvements.

## 3.8 User Interface and Experience (UI//UX) Design

The user interface (UI) design prioritized an intuitive and visually appealing experience. Skillful use of HTML, CSS, and Bootstrap resulted in a responsive and user-friendly interface. Thymeleaf templates enabled dynamic content generation and seamless integration with back-end data.

## 3.9 Project Management

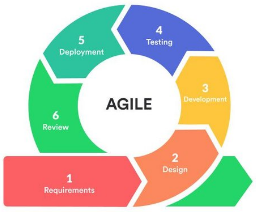


Figure 3.3: Agile Methodology

Effective project management practices were employed throughout the project's lifecycle. Agile methodology was followed, with regular sprints and reviews to track progress and adapt to changing requirements.

# Chapter 4

# IMPLEMENTATION

This section details the comprehensive development of the Bookstore application, integrating technologies like Spring Boot, Hibernate, MySQL, Thymeleaf, HTML, CSS, and Bootstrap.

Sub-sections explain backend and frontend development, highlighting the logic and design for functionality and user experience.

## 4.1 Development Environment Setup

In the implementation of the Bookstore application, we begin by detailing the setup of the development environment, which played a crucial role in facilitating a seamless development process. The following tools and technologies were utilized for the development environment:

- **Eclipse IDE**: Eclipse, a widely-used Integrated Development Environment (IDE), served as the primary development tool for the project.
- **Git and GitHub**: To enable version control and collaborative development.
- **Spring Framework 6**: The Spring framework, a powerful and widely adopted Java framework, formed the backbone of the application.
- **Spring Boot 3**: Spring Boot, an extension of the Spring framework, was leveraged to accelerate application development.
- **Spring Tool Suite 4**: Spring Tool Suite (STS), an Eclipse-based IDE tailored for Spring development, was used to enhance productivity.

- **Apache Maven**: Apache Maven, a powerful build automation tool, was employed to manage project dependencies, build processes, and generate executable artifacts.
- **MySQL**: The MySQL database management system was chosen for its efficiency, reliability, and widespread adoption.
- **Hibernate**: Hibernate, an Object-Relational Mapping (ORM) framework, was integrated to facilitate seamless interaction between the application and the MySQL database.
- **JUnit for Unit Testing** : JUnit, a powerful testing framework, was used to perform unit testing on the individual components and methods of the application.

## 4.2 Technical Integration

Technical integration is crucial for the Bookstore application's implementation. It combines various technologies and tools to create a cohesive system. Key technical integration during development include:

- **Spring Framework Integration** : Spring played a central role in the application's architecture. Integrating components achieved loose coupling, simplifying management.
- **Hibernate Integration** : Integrating Hibernate as the Object-Relational Mapping (ORM) framework allowed the application to interact with the MySQL database.
- **Frontend and Thymeleaf Integration** : Thymeleaf, as the templating engine, was seamlessly integrated with the frontend, consisting of HTML, CSS, and Bootstrap.
- **MySQL Database Integration :** MySQL served as the relational database management system for the Bookstore application.
- **JUnit Testing Integration :** JUnit, the testing framework, was integrated into the development workflow to ensure the quality and reliability of the application's code.
- **Git and GitHub Collaboration :** Git was integrated into the development process, enabling version control. GitHub served as the remote repository, facilitating code sharing and effective management of project changes.
- **Maven Build Automation :** Apache Maven streamlined the project's build process by managing dependencies and generating executable artifacts.

- **Spring Boot for Rapid Development :** Spring Boot 3 was utilized to accelerate application development by providing convention-over-configuration principles.

## 4.3 Backend Development

Backend development forms the foundation of the Bookstore application, handling data processing, business logic, and communication with the database. Key aspects of frontend development include:

- **Controllers and Request Handling :** Spring's controller classes were employed to handle incoming HTTP requests from the frontend.

Listing 4.1: Controller class to map http requests

```
@Controller
public class HomeController {

  @RequestMapping("/")
  public String index(){
    return "redirect:/home";
  }

  @RequestMapping("/home")
  public String home(){
    return "home";
  }

  @RequestMapping("/login")
  public String login(){
    return "login";
  }
}
```

- **Services and Business Logic :** Business logic was encapsulated in service classes, promoting modularity and separation of concerns.
- **Repositories and Database Interactions :** Hibernate's JPA (Java Persistence API) annotations were used in repository classes to define the database interactions.
- **RESTful API Design :** RESTful principles were followed to design the backend API, ensuring a consistent and intuitive interface for frontend interactions.

## 4.4 Frontend Development

Frontend development prioritized an intuitive and visually appealing user interface for enhanced experience. Thymeleaf, HTML, CSS, and Bootstrap facilitated dynamic content and responsive design. Key aspects of frontend development include:

- **Thymeleaf Templating :** Thymeleaf, as the templating engine, played a pivotal role in rendering dynamic data from the backend to the frontend.

Listing 4.2: Example of thymeleaf templating

```
<div class="card">
    <div class="card-body">
        <h5 class="card-title" th:text="${book.title}">Book Title</h5>
        <p class="card-text" th:text="${book.author}">Author Name</p>
        <p class="card-text" th:text="${book.price}">Price</p>
    </div>
</div>
```

- **HTML, CSS, and Bootstrap :** HTML provided the structural foundation of the user interface, while CSS and Bootstrap was utilized for styling and layout customization.

Listing 4.3: Example of BootStrap

```
    <div class="container">
        <div class="row">
            <div class="col-lg-4 col-md-6 col-sm-12">
                <div class="card">
                    <div class="card-body">
                        <h5 class="card-title">Book Title</h5>
                        <p class="card-text">Author Name</p>
                        <p class="card-text">Price</p>
                    </div>
                </div>
            </div>
            <!-- Additional cards for other books -->
        </div>
    </div>
```

- **Responsive Design** The application was designed to be responsive, ensuring that the user interface adapts to various screen sizes and devices.
- **User Interaction and Event Handling :** JavaScript and jQuery were incorporated to add interactive elements and event handling to the frontend with features such as form validation, dynamic content updates, and handling user interactions like button clicks.

Listing 4.4: Example of using JQuery and JS

```
function checkBillingAddress() {
  if($("#theSameAsShippingAddress").is(":checked")) {
    $(".billingAddress").prop("disabled", true);
  } else {
    $(".billingAddress").prop("disabled", false);
  }
}
```

- **Integration with Backend Endpoints :** Frontend components were integrated with backend API endpoints to communicate with the server and fetch dynamic data.

## 4.5 Database Integration

Data integration in the Bookstore application involves the seamless communication between the frontend and backend components, as well as the interaction with the MySQL database using Hibernate. This section outlines the key aspects of data integration:

Listing 4.5: Database Configuration

```
spring.thymeleaf.cache=false
#set here configurations for the database connection
spring.datasource.url=jdbc:mysql://localhost:3306/bookstoredatabase
# Username and secret
spring.datasource.username=root
spring.datasource.password=root
# Keep the connection alive if idle for a long time (needed in production
spring.datasource.dbcp2.test-while-idle = true
spring.datasource.dbcp2.validation-query = SELECT 1
# Use spring.jpa.properties.* for Hibernate native properties (the prefix
is striped before adding them to the entity manager).
# Show or not log for each sql query
spring.jpa.show-sql = true
# Hibernate ddl auto (create, create-drop, update) : with "update" the
database schema
# will be automatically updated accordingly to java entities
spring.jpa.hibernate.ddl-auto=update
# Allows Hibernate to generate SQL optimized for a particular DBMS
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
}
```

- **Communication between Frontend and Backend :** Data exchange between the frontend and backend was achieved through RESTful API endpoints.

- **Data Binding with Thymeleaf :** Thymeleaf facilitated data binding between the frontend and backend by seamlessly integrating dynamic data into HTML templates.

- **Database Interaction using Hibernate :** Hibernate acted as the intermediary layer between the backend application and the MySQL database.

## 4.6 User Authentication and Security

Ensuring user authentication and data security are paramount considerations in the development of the Bookstore application. This section outlines the implementation of user authentication mechanisms and security measures to safeguard user data:

- **User Registration and Login** : User authentication was achieved through a secure registration and login system.
- **Password Hashing** : To protect user passwords from unauthorized access, a strong cryptographic hashing algorithm (e.g., bcrypt) was utilized to convert plain text passwords into irreversible hash values.
- **Session Management** : Session management was employed to maintain user sessions after successful login.
- **Authorization and Role-Based Access** : Role-based access control (RBAC) managed user permissions.
- **Secure Communication (HTTPS)** : To protect data transmitted between the application and the server, the use of HTTPS (HTTP Secure) was enforced.

## 4.7 Performance Optimization

Performance optimization is crucial for the Bookstore application to deliver a seamless user experience and handle potential traffic spikes efficiently. This section outlines the key performance optimization techniques implemented:

- **Database Indexing** : To enhance database query performance, appropriate indexes were created on columns commonly used in search and filtering operations.
- **Optimized Database Queries** : Careful consideration was given to database queries to ensure they were optimized for performance.
- **Asynchronous Processing** : Asynchronous processing was used for time-consuming

tasks like email notifications and report generation.

- **Connection Pooling** : Connection pooling was implemented to reuse database connections and avoid the overhead of creating new connections for each user request.
- **Image and Resource Optimization** : Images and static resources were optimized to reduce their size and improve page load times.

## 4.8 Version Control and Collaboration

Version control and collaboration were vital for the Bookstore application's development. They enabled efficient teamwork, code management, and seamless integration of contributions. Here's an overview of the practices employed:

- **Git Version Control** : Git, a distributed version control system, managed the project's source code.
- **Branching Strategy** : A branching strategy organized development efforts.
- **Pull Requests and Code Reviews** : Pull requests were used for merging feature branches into the main branch, ensuring code quality and adherence to standards.
- **Issue Tracking** : GitHub's issue tracking system was utilized to manage project tasks and track bug fixes and feature requests.
- **Continuous Integration (CI)** : A CI pipeline was set up for automated building and testing whenever changes were pushed to the repository.

# Chapter 5

# FEATURES AND DISCUSSION

The Bookstore application offers a range of user-centric features that enhance the book browsing, selection, and purchasing experience. Key features include:

## 5.1 ADMIN

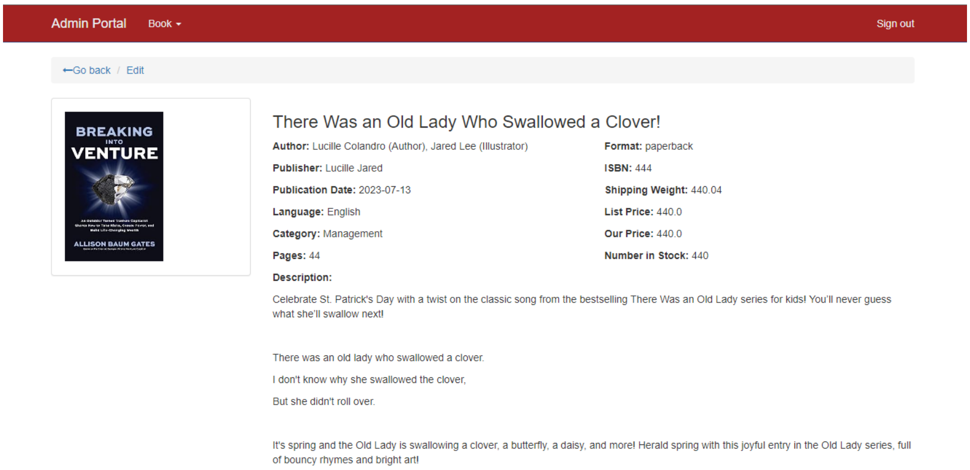
Admin panel is used for adding and updating the books information

### Admin Login

Before adding and updating information admin will login the application first with role as admin

### Finding and Updating Books Information

Admin can update and find books informations.



There Was an Old Lady Who Swallowed a Clover!

**Author:** Lucille Colandro (Author), Jared Lee (Illustrator)

**Format:** paperback

**Publisher:** Lucille Jared

**ISBN:** 444

**Publication Date:** 2023-07-13

**Shipping Weight:** 440.04

**Language:** English

**List Price:** 440.0

**Category:** Management

**Our Price:** 440.0

**Pages:** 44

**Number in Stock:** 440

**Description:**

Celebrate St. Patrick's Day with a twist on the classic song from the bestselling There Was an Old Lady series for kids! You'll never guess what she'll swallow next!

There was an old lady who swallowed a clover.
I don't know why she swallowed the clover,
But she didn't roll over.

It's spring and the Old Lady is swallowing a clover, a butterfly, a daisy, and more! Herald spring with this joyful entry in the Old Lady series, full of bouncy rhymes and bright art!

Figure 5.1: Viewing Book and its Details

## 5.2 User Interface(BookStore)

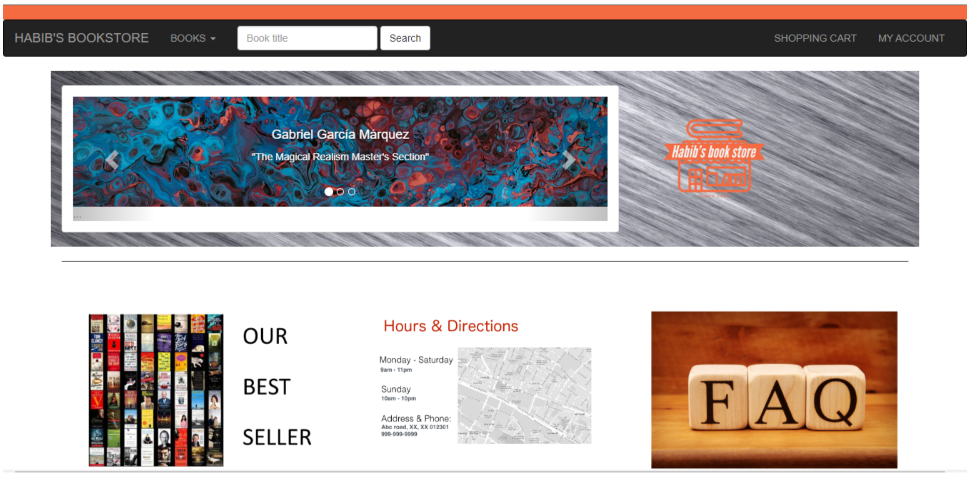


Figure 5.2: User interface homepage

BookStore interface is flexible to use and user friendly, which is easy to use it has some main features. Such as-

- **User Registration and Login :** Users can create new accounts or log in to access personalized features and track their orders.
- **Book Catalog :** A comprehensive catalog of books is available, allowing users to browse and search for books by title, author, genre, or keywords.

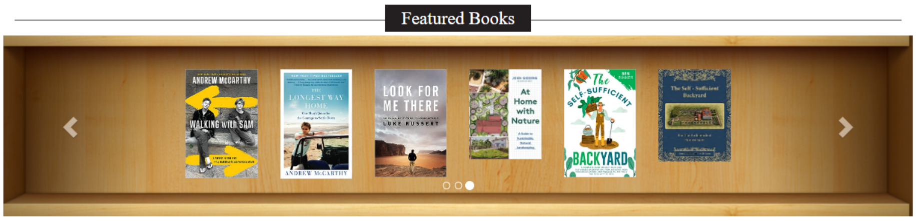
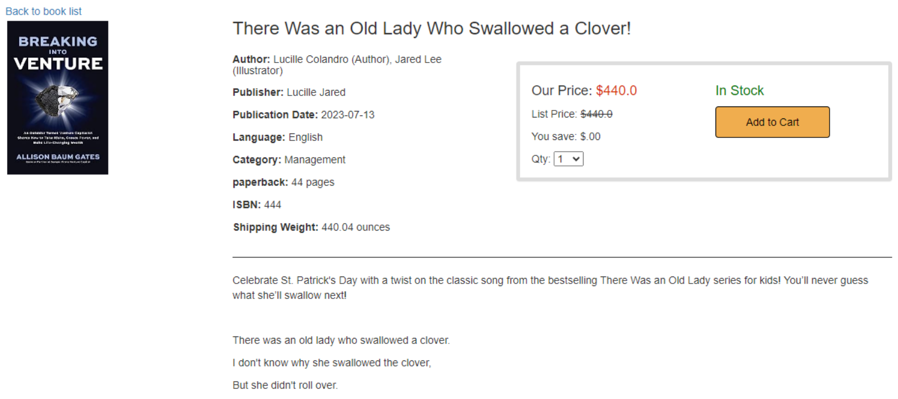


Figure 5.3: Featured Books on Book Shelf

- **Book Details :** Each book listing provides detailed information, including the book's cover image, title, author, description, and price.



There Was an Old Lady Who Swallowed a Clover!

**Author:** Lucille Colandro (Author), Jared Lee (Illustrator)

**Publisher:** Lucille Jared

**Publication Date:** 2023-07-13

**Language:** English

**Category:** Management

**paperback:** 44 pages

**ISBN:** 444

**Shipping Weight:** 440.04 ounces

Our Price: $440.0

List Price: ~~$440.0~~

You save: $.00

Qty: 1

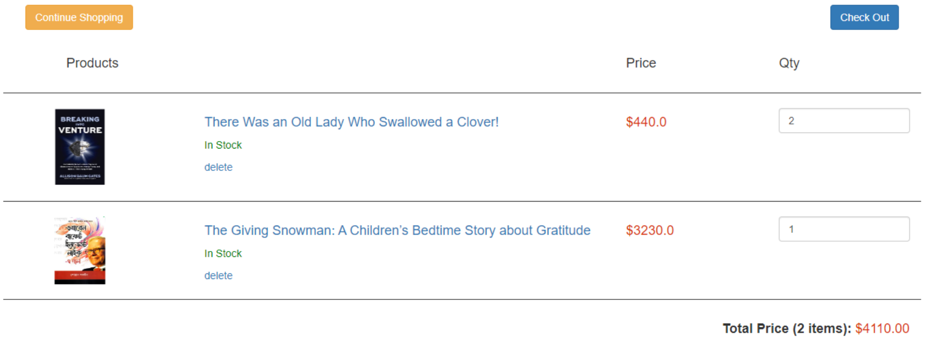
In Stock

Celebrate St. Patrick's Day with a twist on the classic song from the bestselling There Was an Old Lady series for kids! You'll never guess what she'll swallow next!

There was an old lady who swallowed a clover.
I don't know why she swallowed the clover,
But she didn't roll over.

Figure 5.4: Book Details Page

- **Shopping Cart :** Users can add books to their shopping cart for future purchases. They can view and manage the items in their cart before proceeding to checkout.

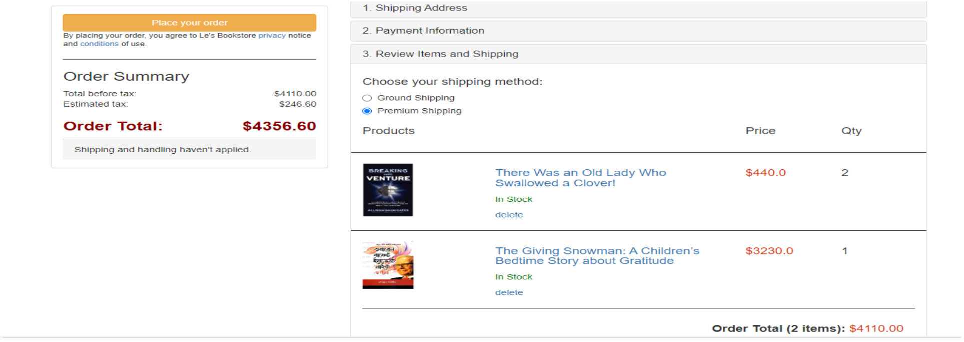


| Products | | Price | Qty |
|---|---|---|---|
| | There Was an Old Lady Who Swallowed a Clover!<br>In Stock<br>delete | $440.0 | 2 |
| | The Giving Snowman: A Children's Bedtime Story about Gratitude<br>In Stock<br>delete | $3230.0 | 1 |

**Total Price (2 items):** $4110.00

Figure 5.5: Showing Shopping Cart

- **Secure Checkout :** A secure and user-friendly checkout process allows users to review their order, select payment options, and enter shipping information.



By placing your order, you agree to Le's Bookstore privacy notice and conditions of use.

Order Summary

| | |
|---|---|
| Total before tax: | $4110.00 |
| Estimated tax: | $246.60 |
| **Order Total:** | **$4356.60** |

Shipping and handling haven't applied.

1. Shipping Address
2. Payment Information
3. Review Items and Shipping

Choose your shipping method:
○ Ground Shipping
● Premium Shipping

| Products | | Price | Qty |
|---|---|---|---|
| | There Was an Old Lady Who Swallowed a Clover!<br>In Stock<br>delete | $440.0 | 2 |
| | The Giving Snowman: A Children's Bedtime Story about Gratitude<br>In Stock<br>delete | $3230.0 | 1 |

**Order Total (2 items):** $4110.00

Figure 5.6: Placing Orders from the Cart

- **Order History :** Users have access to their order history, allowing them to track past purchases and view order details.

- **Responsive Design** The application is designed to be responsive, adapting to different screen sizes and devices, providing an optimal user experience across desktops, tablets, and smartphones.

## 5.3 Discussion

The Bookstore application successfully integrates Spring, Hibernate, Thymeleaf, and Bootstrap to provide a user-friendly platform for browsing and purchasing books. The backend utilizes controllers, services, and repositories, while the frontend incorporates Thymeleaf templates, HTML, CSS, and Bootstrap for a responsive design.

Security measures, such as user authentication, password hashing, and role-based access, ensure data protection. Performance optimization techniques, including caching and database indexing, enhance the application's responsiveness. Collaboration via Git and GitHub enables efficient code management and continuous integration practices ensure code reliability.

# Chapter 6

# CONCLUSION AND FUTURE SCOPE

The application maybe more acceptable to the user. So in future work the limitations should be removed and application become more user friendly.

## 6.1 Conclusion

The Bookstore app, using Spring, Hibernate, Thymeleaf, and Bootstrap, provides a secure and user-friendly platform for online book browsing and purchasing. Robust security, performance optimization, and collaboration through Git and GitHub enhance the application's quality.

## 6.2 Future Scope

While the Bookstore application is already feature-rich and fully functional, there are several avenues for future enhancement and expansion:

- Advanced Search Functionality
- User Profiles and Wish-list
- Recommendation Engine
- Multiple Payment Options
- Social Media Integration
- Mobile Application

By continuously refining and expanding the Bookstore application, it can remain a competitive and sought-after platform for book lovers, catering to their evolving needs and preferences.

# References

- Fincer java book store - The project uses a number of Spring Boot features, including JPA, Thymeleaf, and Spring Security.
- Nisib Book Store - The project uses a number of different technologies, including MySQL, Lombok, and Swagger.
- Step-by-Step Guide: Developing a Bookstore API with Spring Boot and MySQL for CRUD Operations - This article provides a step-by-step guide on how to develop a book store API using Spring Boot and MySQL.
- Spring boot simple project - Book Store Management — MySql — Thymeleaf —JPA
- java eee - API Documentation and Tag Reference
- Spring 6.0.11
- A simple e-commerce project A course on udemay

# Appendix A

# Appendix Chapter

**Project title : Book Store**

Appendix with supplementary details for Online Bookstore App using Spring framework. The following items offer additional insights and details supporting the main project report:

1. Introduction and Overview: This section provides a complete overview of the Book Store Application, detailing its development process and the technologies utilized.
2. Literature Review : This section provides a short literature review about online bookstore.
3. Methodology : Methodology section describes in which way the project is implemented.
4. Implementation: This section offers a brief overview of both backend and frontend implementations of the application. It includes insights into the relationship among entities and dynamic implementations of the admin and user panels.
5. Features and discussion: Here the visualizing implementation is shown.
6. References: A list of external references, or articles used for guidance and inspiration during the development process is included in this section.

The appendix contains essential materials, such as the database schema, code snippets, performance metrics, and screenshots, offering deeper insights into the technical aspects of the project. Readers can refer to the appendix for a comprehensive understanding of the application's implementation and user interface. We believe that the included materials enhance the overall project report.