VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY

INTERNATIONAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

WEB APPLICATION DEVELOPMENT

READORA - WHERE STORIES BEGIN

Course name: Web Application Development

Course ID: IT093IU

Theory class lecturer: Assoc. Prof. Nguyen Van Sinh

Laboratory class lecturer: MSc. Nguyen Trung Nghia

No.	Full name	Student ID	Role	Contribution
1	Hoàng Xuân Dũng	ITITWE22009	Back-end Developer	50%
2	Nguyễn Tiến Anh	ITITDK22128	Front-end Developer	50%

A report submitted to the School of Computer Science and Engineering in partial fulfillment
of the requirements for the Final Project
in Web Application Development course

Semester 1, 2025 – 2026

International University
School of Computer Science and Engineering

Table of Contents

List of Figures

List of Tables

CHAPTER 1: ACKNOWLEDGMENTS

First and foremost, we would like to thank our team members for this initiative. Their commitment, encouragement, and suggestions were crucial to our project's success. Their contributions have improved this report and made the process of finishing it more rewarding.

We also appreciate everyone else who contributed to this project and offered their assistance in seeing it through to completion. Their suggestions and insights were priceless and really aided us in completing the job. We appreciate their support and have recognized all of their donations, no matter how minor.

We would especially like to thank our mentors, Assoc. Prof. Nguyen Van Sinh and MSc. Nguyen Trung Nghia for their encouraging advice, helpful criticism, and kind supervision throughout the research. Their kindness and tolerance were much appreciated, and they made a lasting impression on us.

We realize that without their help, the project would not have been able to be completed. We sincerely appreciate each one of them. We are appreciative that we had the chance to work on this project, and we can't wait to use the lessons we've learned in the future. We're also looking forward to the opportunities that lie ahead.

Finally, we dedicate this study to everyone who believes in the power of collaboration and teamwork. This paper serves as an example of what may be achieved when several ideas are directed toward the same objective. It is our goal that by sharing our work, others may be inspired to explore and innovate on their own.

CHAPTER 2: EXECUTIVE SUMMARY

2.1. Overview

The decision to develop an online book store platform stems from several important factors aligned with current consumer behavior, digital transformation trends, and the growing demand for convenient access to reading materials. Books remain one of the most valued forms of knowledge and entertainment, and unlike many fast-consumed digital products, they maintain long-term relevance and cultural significance. This makes the book industry a stable and sustainable market for online business development.

First, the rising popularity of online shopping has significantly changed the way people discover and purchase books. Modern consumers increasingly prefer browsing, comparing, and buying items through digital platforms rather than visiting physical bookstores. This shift is driven by the convenience of home delivery, diverse product availability, and the ability to access detailed information before making a purchase. The online book store project leverages this trend by offering a wide selection of titles across multiple genres, ensuring that every reader can find a book that matches their interests and needs.

Second, the continuous expansion of educational activities, self-learning habits, and professional skill development has contributed to a growing demand for books. Readers today search for not only entertainment but also knowledge that can support their academic, career, and personal growth. By providing rich book descriptions, pricing details, and comprehensive metadata such as author information, publication year, category, and ratings, the system enhances the user experience and helps readers make informed decisions. This focus on user-centered design allows the platform to appeal to a broad range of customers—from casual readers to students, educators, and professionals.

Furthermore, remote learning and digital education have become more prominent in recent years, especially with the increase of online courses and virtual classrooms. This creates additional demand for academic textbooks, reference materials, and specialized publications. The online book store addresses this need by ensuring that administrators can quickly update the catalog, add new academic titles, and adjust inventory based on market demand. This flexibility helps maintain a fresh and relevant product lineup, which is essential for long-term competitiveness.

Another compelling factor is the relatively high value of specialized or premium books, such as imported editions, limited collections, and academic resources. Compared to low-cost, fast-moving consumer products, the book market offers more stable profit margins, particularly when customers purchase multiple items in a single transaction. This higher average order value strengthens the business model and provides a solid foundation for sustainable revenue growth.

Lastly, the movement toward digital literacy and the global emphasis on accessible knowledge contribute to an environment where online bookstores play an important social role. By making reading materials easily available to a wide audience, the platform supports lifelong learning and promotes a culture of reading. Additionally, the system's administrative features—such as adding new books, managing prices, and updating product information—enable the store to operate efficiently while maintaining transparency and consistency across all listings.

In summary, the strategic choice to develop an online book store is grounded in strong market demand, evolving consumer habits, and the long-term relevance of books as both educational tools and sources of personal enrichment. By meeting these needs through a dynamic, user-friendly platform, the project positions itself to grow sustainably and stand out in the increasingly competitive e-commerce landscape.

2.2. Objectives and Scopes

2.2.1 Objectives

Expanding the Customer Base: In the current digital business environment, expanding the customer base has become a critical objective for online book retailers. By developing a user-friendly and accessible e-commerce platform, the Book Store application aims to attract a wider range of customers beyond geographical limitations. Offering a diverse catalog of books, personalized browsing experiences, and convenient online purchasing helps increase visibility and encourages new users to engage with the platform, thereby driving growth in sales and overall profitability.

Improving Customer Service: Providing better customer service is one of the key objectives of this project. Since the online Book Store operates continuously, users are able to browse products, access detailed book information, and manage orders at any time. By presenting clear product descriptions, categories, and order tracking features, the system enables customers to make informed decisions easily. Improved accessibility and responsiveness contribute to a higher level of customer satisfaction and help the business remain competitive in the e-commerce market.

Strengthening Business–Customer Relationships: The Book Store application aims to enhance the relationship between the business and its customers through direct and efficient interaction. Features such as user accounts, order history, and secure authentication allow the system to better understand customer preferences and purchasing behavior. By maintaining consistent communication and reliable service, the platform helps build trust and long-term relationships, thereby expanding the company’s market reach.

Increasing Online Sales Performance: Another important objective of this project is to boost sales within the online Book Store. This is achieved by optimizing the shopping flow from product discovery to checkout, ensuring that the website is intuitive and easy to navigate. Improving conversion rates through effective search, filtering, and a streamlined checkout process encourages more users to complete purchases. Additionally, features such as promotions and repeat customer engagement increase the likelihood of higher order values and repeat transactions.

Enhancing Customer Satisfaction: Enhancing customer satisfaction is a fundamental objective of the Book Store system. This includes delivering a smooth shopping experience, addressing customer needs promptly, and minimizing potential issues during the purchasing process. Ensuring accurate book information, transparent pricing, and reliable order processing helps meet customer expectations. By responding effectively to user actions and reducing friction, the platform aims to create a positive and satisfying customer experience.

Reducing Shopping Cart Abandonment: Reducing shopping cart abandonment is a common objective in e-commerce systems and is also a focus of this project. By simplifying cart management, providing clear pricing details, and ensuring a secure and straightforward checkout process, the Book Store application minimizes reasons for users to leave before completing their purchase. These improvements help increase customer retention and overall sales efficiency.

Enhancing User Experience: The primary objective of the Book Store application is to deliver an excellent user experience across both desktop and mobile devices. This includes intuitive navigation, fast response times, and visually clear interfaces. Ensuring that all core features are easy to access and function correctly allows users to interact with the system effortlessly. By enhancing the overall user experience, the platform aims to increase conversion rates, encourage repeat usage, and strengthen customer loyalty.

2.2.2. Scopes

Electronic Data Exchange: The Book Store application supports electronic data exchange through a RESTful architecture, enabling digital communication between system components. The React-based frontend interacts with the Express and Node.js backend to retrieve product catalogs, manage user accounts, update shopping carts, and process orders. These data exchanges may also involve limited third-party services, such as image hosting, email notifications, and sandboxed payment services, allowing information to flow efficiently between different systems and stakeholders.

Technology-Facilitated Transactions: All business transactions within the system are fully technology-facilitated using the MERN stack. Users interact with the application through a web browser via a single-page application (SPA), while the backend services and MongoDB database manage business logic, data persistence, and automation. This approach replaces traditional face-to-face interactions and enables scalable, efficient, and continuous online buying and selling processes.

Transaction Enablement and Order Management: The system is designed to enable and support core e-commerce transactions, including adding, updating, and removing items from the shopping cart, placing orders, and tracking order status. In addition, administrative users are provided with tools to manage and process customer orders. These features aim to improve checkout efficiency, enhance transaction transparency, and support higher conversion rates within the online store.

Customer Experience: Providing a positive and reliable customer experience is a central focus of this project. The application emphasizes fast page load times, intuitive website navigation, and a clear product discovery flow to reduce user frustration and abandonment. High-quality product images, detailed descriptions, and customer reviews or ratings are included to build trust and assist users in making informed purchasing decisions. A streamlined checkout process further contributes to user satisfaction and repeat purchase behavior.

Content and Operations Management: The scope of the project includes administrative functionalities that support day-to-day content and operational management. Authorized administrators are able to perform create, read, update, and delete (CRUD) operations on books and categories, manage customer orders, and view basic store metrics. These capabilities help maintain catalog accuracy, ensure smooth operations, and support effective decision-making for the business.

Security and Access Control: The application incorporates essential security mechanisms to protect user data and sensitive operations. This includes password hashing, JSON Web Token (JWT)–based authentication, and role-based access control to differentiate between customer and administrator privileges. Input validation and standard protections against common web vulnerabilities are applied to enhance overall system reliability and data security.

Limited External Integrations: The system supports a limited set of external integrations that are essential to core functionality, such as image storage services, email delivery, and sandboxed payment processing. More complex integrations, including production-level payment gateways, PCI compliance, and real-time shipping or logistics services, are intentionally excluded from the main scope to maintain focus on core application objectives.

Testing and Quality Assurance: The project scope includes testing and quality assurance activities to ensure system stability and correctness. Unit and integration tests are implemented for critical backend endpoints and key frontend components. In addition, code linting and stable build configurations are applied to reduce errors and support reliable development and deployment processes.

Deployment and Environment Configuration: The project provides environment configuration files, run scripts, and documentation to support deployment in both development and production environments. Optional containerization using Docker may be included to standardize setup and improve portability across different systems.

Out of Scope: The following features are explicitly excluded from the scope of this project: live production payment gateways and PCI compliance, real-time shipping and fulfillment integrations, native mobile applications, multi-vendor marketplace functionality, and advanced enterprise-level inventory management or business intelligence systems. These exclusions help define a clear and achievable minimum viable product (MVP).

2.3. Key Achievements and Milestones

Enhanced Customer Shopping Experience: Customers are able to browse, search, and explore detailed book descriptions, including author information, genres, prices, and summaries, through a clean and user-friendly interface. This smooth browsing experience helps users make informed purchasing decisions, increases customer satisfaction, and encourages repeat purchases on the book website.

Flexible and Secure Payment System: The system successfully supports multiple payment methods, allowing customers to complete transactions conveniently and securely. The payment service ensures safe, reliable, and breach-free processing of transactions, helping to preserve financial integrity and build strong user trust in the platform.

Wishlist Creation: Before registering, I made use of the Wishlist tool to bookmark my favorite goods. simplified the procedure of registering as a user after being a visitor.

Book Review and Comment System: After purchasing books, customers can leave comments and reviews to share their opinions and reading experiences. These user-generated reviews enhance the credibility of books, assist new customers in choosing suitable titles, and provide valuable feedback for administrators to improve content quality and book selection.

Administrative Management and Content Expansion: Administrators effectively manage book listings, update book information, and oversee customer reviews to maintain platform quality. The successful introduction of new book titles and categories contributes to content expansion, increased user engagement, and potential revenue growth for the website.

2.4. Summary of Project Outcomes

The online book-selling website has successfully positioned itself as a convenient and reliable platform for readers seeking a one-stop destination to explore, purchase, and review books across multiple genres. By offering detailed book descriptions, author information, user reviews, and a clean, user-friendly interface, the platform delivers a satisfying shopping experience that supports informed decision-making. As a result, the website has attracted increasing user engagement and fostered a growing community of readers who actively interact through purchases and book reviews.

E-commerce in the context of online book retailing encompasses a wide range of activities, including digital product presentation, online ordering, electronic payments, and customer feedback management. The coordination of digital data flows—such as orders, payment confirmations, and review submissions—ensures smooth end-to-end transaction processing. Technology-enabled interactions, including secure online payment systems and integrated databases, play a critical role in supporting efficient operations, data consistency, and reliable financial management. These elements contribute directly to customer trust, operational efficiency, and long-term platform sustainability.

The primary objectives of the book e-commerce platform include expanding the customer base, improving customer support and engagement, and establishing long-term relationships with readers. By enhancing user experience through responsive design, multiple payment options, and an interactive review system, the platform aims to increase conversion rates, reduce cart abandonment, and encourage repeat purchases. Additionally, the project supports brand awareness growth and traffic generation while maintaining high levels of customer satisfaction. Collectively, these outcomes demonstrate the platform’s potential for sustainable development and its contribution to the evolving digital retail ecosystem.

CHAPTER 3: INTRODUCTION

3.1. Goal

3.1.1 Customer Service

Clear Contact Information: Ensure that the book-store website clearly displays customer service contact channels, including email addresses, support forms, and help-center links. Easy access to contact information allows customers to quickly reach support staff with questions related to books, orders, payments, or account issues.

Responsive Customer Support: Respond to customer inquiries promptly and professionally across multiple channels such as email, live chat, and social media. Timely resolution of issues related to orders, payments, or book availability helps build trust and enhances overall customer satisfaction.

Order Notifications and Updates: Provide customers with real-time electronic notifications regarding their order status. Inform users when an order is confirmed, processed, shipped, and delivered. Where applicable, include tracking numbers or links so customers can easily monitor their book deliveries.

Simple Returns and Refunds: Simplify the return and refund process to ensure a positive customer experience. Clearly communicate return policies for books, provide an easy-to-follow procedure, and process refunds efficiently. Allow customers to initiate return or refund requests directly through their user accounts or by contacting customer support.

Proactive Customer Feedback: Encourage customers to share feedback, ratings, and comments after purchasing or reading a book. Collecting user reviews helps improve book recommendations, enhances content quality, and supports informed purchasing decisions for future customers.

Continuous Service Improvement: Regularly assess customer service performance by monitoring response times, resolution rates, customer satisfaction levels, and feedback. Use this information to identify areas for improvement and continuously enhance service quality and operational efficiency.

User-Friendly Interface: Design and maintain an intuitive, responsive, and device-compatible interface that supports a smooth customer service experience. Ensure that users can easily navigate the website, access support features, and manage their accounts across different devices and operating systems.

Diverse Book Collection: Offer a wide variety of books across multiple genres, authors, price ranges, and reader preferences. A rich and well-organized catalog enhances customer satisfaction and supports diverse user needs, contributing to higher engagement and repeat purchases.

3.1.2 Platform Performance

Transaction and Visitor Optimization: Ensuring optimal platform performance is essential for delivering a smooth and efficient online book-shopping experience. Performance optimization supports higher search engine rankings, increased visitor traffic, and reliable transaction processing. The project aims to achieve page load times of under 3 seconds by optimizing images, scripts, and backend processes, thereby enhancing usability and encouraging user retention.

Efficient Transaction Processing: Improve transaction handling to guarantee secure, seamless, and error-free book purchases. Optimized payment workflows and database interactions help reduce transaction delays while simultaneously supporting increased website traffic and higher order volumes.

Stability and Scalability: Design and maintain the book-store platform to support growth in users, book listings, and transaction volumes. The system should remain stable during traffic spikes or peak shopping periods. Implement scalable infrastructure solutions such as load balancing, horizontal scaling, and a robust database architecture to ensure long-term performance and reliability.

Advanced Search and Filtering: Provide an efficient search and filtering system that allows users to quickly find books by title, author, genre, price range, or popularity. Optimized indexing and query handling enable accurate and fast search results, improving user experience and increasing conversion rates.

Checkout Process Simplification: Streamline the checkout process to minimize cart abandonment and improve conversion rates. Reduce the number of steps required to complete a purchase, optimize payment gateway integration, and enhance database response times. Ensure that the checkout process is both user-friendly and secure.

Data Protection and Security: Implement strong security measures to protect customer data, including personal information and payment details. Utilize encrypted communication channels, secure authentication mechanisms, regular security updates, and compliance with best practices to maintain data integrity and customer trust.

3.2. A Brief Overview of The Report’s Structure

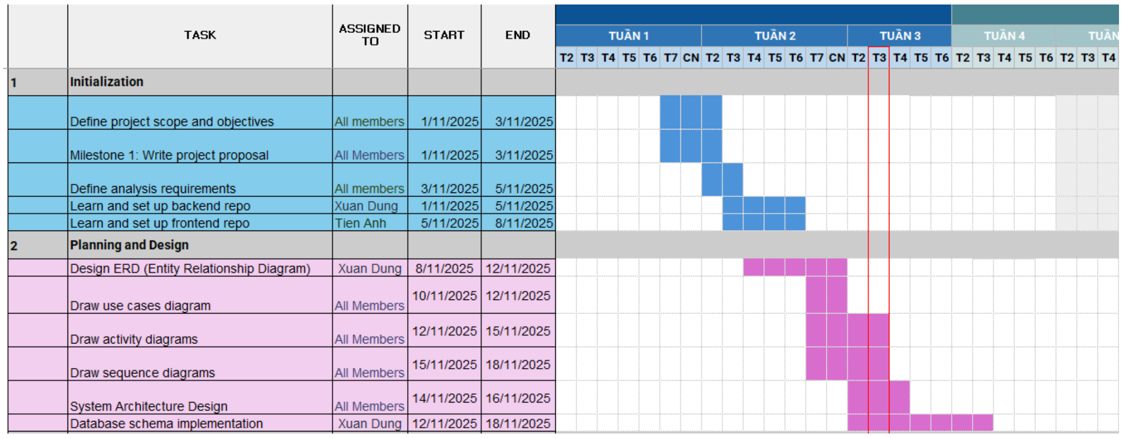
A summary, an introduction, project planning, requirements analysis, design and architecture, development, testing and quality assurance, deployment and implementation, user documentation, project evaluation, lessons learned, conclusion, appendices, references, and acknowledgments collectively form the fifteen chapters of this report. Each chapter presents an overview, objectives, progress updates, system architecture, team member contributions, customer requirements analysis, and outcomes related to a specific area of the online book-store e-commerce project.

The report systematically documents the planning, design, and implementation of a book-selling platform that enables users to browse detailed book information, complete secure online purchases, and share reviews through comments and ratings. An appendix section is included to provide additional references, technical resources, and supporting materials relevant to the project.

The final chapter summarizes the key findings and outcomes from all chapters, highlighting the overall achievements and challenges encountered throughout the project lifecycle. The references section acknowledges the data sources, technical documentation, and learning materials used in developing the report, ensuring transparency, credibility, and academic integrity. Finally, the acknowledgments section expresses appreciation to the individuals and groups whose guidance, collaboration, and support contributed significantly to the successful completion of the Book-Store project.

CHAPTER 4: PROJECT PLANNING

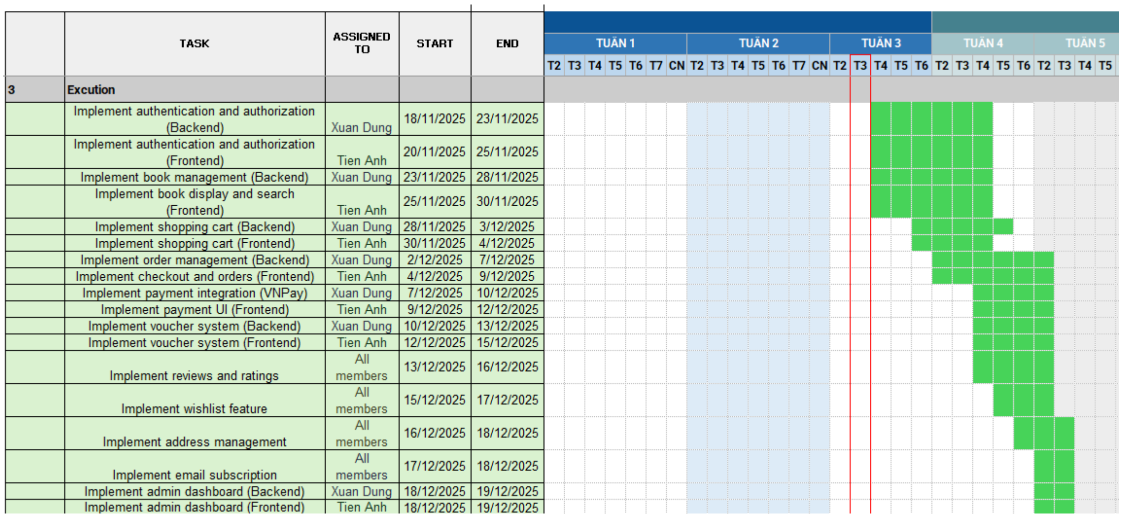
4.1. Project Timeline and Milestones



	TASK	ASSIGNED TO	START	END
1	**Initialization**			
	Define project scope and objectives	All members	1/11/2025	3/11/2025
	Milestone 1: Write project proposal	All Members	1/11/2025	3/11/2025
	Define analysis requirements	All members	3/11/2025	5/11/2025
	Learn and set up backend repo	Xuan Dung	1/11/2025	5/11/2025
	Learn and set up frontend repo	Tien Anh	5/11/2025	8/11/2025
2	**Planning and Design**			
	Design ERD (Entity Relationship Diagram)	Xuan Dung	8/11/2025	12/11/2025
	Draw use cases diagram	All Members	10/11/2025	12/11/2025
	Draw activity diagrams	All Members	12/11/2025	15/11/2025
	Draw sequence diagrams	All Members	15/11/2025	18/11/2025
	System Architecture Design	All Members	14/11/2025	16/11/2025
	Database schema implementation	Xuan Dung	12/11/2025	18/11/2025

TUẦN 1
TUẦN 2
TUẦN 3
TUẦN 4
TUẦN
T2 T3 T4 T5 T6 T7 CN T2 T3 T4 T5 T6 T7 CN T2 T3 T4 T5 T6 T2 T3 T4 T5 T6 T2 T3 T4

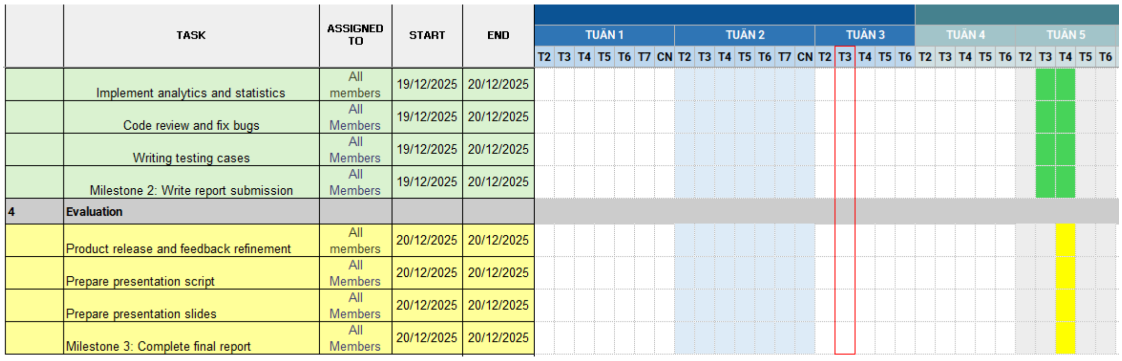
Figure 1: Gantt Chart part 1



	TASK	ASSIGNED TO	START	END
3	**Excution**			
	Implement authentication and authorization (Backend)	Xuan Dung	18/11/2025	23/11/2025
	Implement authentication and authorization (Frontend)	Tien Anh	20/11/2025	25/11/2025
	Implement book management (Backend)	Xuan Dung	23/11/2025	28/11/2025
	Implement book display and search (Frontend)	Tien Anh	25/11/2025	30/11/2025
	Implement shopping cart (Backend)	Xuan Dung	28/11/2025	3/12/2025
	Implement shopping cart (Frontend)	Tien Anh	30/11/2025	4/12/2025
	Implement order management (Backend)	Xuan Dung	2/12/2025	7/12/2025
	Implement checkout and orders (Frontend)	Tien Anh	4/12/2025	9/12/2025
	Implement payment integration (VNPay)	Xuan Dung	7/12/2025	10/12/2025
	Implement payment UI (Frontend)	Tien Anh	9/12/2025	12/12/2025
	Implement voucher system (Backend)	Xuan Dung	10/12/2025	13/12/2025
	Implement voucher system (Frontend)	Tien Anh	12/12/2025	15/12/2025
	Implement reviews and ratings	All members	13/12/2025	16/12/2025
	Implement wishlist feature	All members	15/12/2025	17/12/2025
	Implement address management	All members	16/12/2025	18/12/2025
	Implement email subscription	All members	17/12/2025	18/12/2025
	Implement admin dashboard (Backend)	Xuan Dung	18/12/2025	19/12/2025
	Implement admin dashboard (Frontend)	Tien Anh	18/12/2025	19/12/2025

TUẦN 1
TUẦN 2
TUẦN 3
TUẦN 4
TUẦN 5
T2 T3 T4 T5 T6 T7 CN T2 T3 T4 T5 T6 T7 CN T2 T3 T4 T5 T6 T2 T3 T4 T5 T6 T2 T3 T4 T5

Figure 2: Gantt Chart part 2



	TASK	ASSIGNED TO	START	END
	Implement analytics and statistics	All members	19/12/2025	20/12/2025
	Code review and fix bugs	All Members	19/12/2025	20/12/2025
	Writing testing cases	All Members	19/12/2025	20/12/2025
	Milestone 2: Write report submission	All Members	19/12/2025	20/12/2025
4	**Evaluation**			
	Product release and feedback refinement	All members	20/12/2025	20/12/2025
	Prepare presentation script	All Members	20/12/2025	20/12/2025
	Prepare presentation slides	All Members	20/12/2025	20/12/2025
	Milestone 3: Complete final report	All Members	20/12/2025	20/12/2025

TUẦN 1
TUẦN 2
TUẦN 3
TUẦN 4
TUẦN 5
T2 T3 T4 T5 T6 T7 CN T2 T3 T4 T5 T6 T7 CN T2 T3 T4 T5 T6 T2 T3 T4 T5 T6 T2 T3 T4 T5 T6

Figure 3: Gantt Chart part 3

The project officially starts on 01/11/2025 and will end at the beginning of December, estimated time: 18/12/2025. Over the course of 7 weeks (midterm weeks excluded), our project will undergo four primary phases: project initiation, project planning and design, project execution and project evaluation

- **Milestone 1**: Project proposal submission: by this milestone, the project plan with a detailed timeline, feasibility study and requirements gathering need to be finished. The design phase is ongoing.
- **Milestone 2**: Midterm report submission: by this milestone, the requirements analysis (Use case, class, Sequence diagrams) needs to be finished. The development of the product needs to be finished by 40% and documented in the midterm report.
- **Milestone 3**: Final report submission, the product needs to be ready for the soft launch, with functional testing and detailed technical documents.

CHAPTER 5: REQUIREMENTS ANALYSIS

5.1. Description of Project Requirements

5.1.1. Functional Requirements

Req.ID	Requirement Name	Detailed Description
01	Registration and Authentication of Users	The system shall allow users to securely create accounts, sign in, and manage their profile information. To enhance usability and accessibility, the platform supports features such as Google login integration and password recovery mechanisms.
02	Product Management	Administrators are responsible for managing the product catalog, including adding new books, updating existing entries, and removing outdated items. To ensure customers receive complete and accurate information, each product record is required to contain essential details such as the book title, description, price, discount information, category, and high-quality cover images.
03	Application of Advanced Search and Filtering	Efficient product discovery is supported through advanced search and filtering functionality. Users can search for books using various criteria such as title,

		category, price range, or available discounts. In addition, filtering options help narrow down search results to better match individual user preferences.
04	User's Cart and Checkout Process	Users are provided with a shopping cart feature that allows them to add books, review selected items, adjust quantities, or remove products as needed. The checkout process is designed to be straightforward and secure, offering multiple payment options to ensure convenience and safe transaction handling.
05	Order Management	Access to order-related information is available to users, including order history, current order status, and return or exchange options. On the administrative side, staff members are able to process orders, update order statuses, and respond to customer inquiries related to purchases.
06	Reviews, comment and ratings from customers	To promote trust and transparency, customers can submit ratings and reviews for books they have purchased. These reviews provide valuable insights for other users. An administrative backend supports moderation activities, allowing submitted feedback to be reviewed, edited, or removed when necessary.
07	User Interaction Features	Additional interaction features such as wishlists and product trends contribute to increased user engagement. These features encourage users to interact more frequently with the platform and enhance the overall shopping experience.

Table 1: Functional requirements of Book Store

5.1.2. Nonfunctional Requirements

Req.ID	Requirement Name	Detailed Description
01	Performance	The online book store is expected to operate with minimal

		downtime, fast page loading, and high responsiveness. Achieving a smooth user experience requires optimized front-end components, efficient database queries, and reliable server performance to ensure users can browse, search, and purchase books without delays.
02	Scalability	As the number of users and transactions grows, the book store must be capable of handling increased traffic and data volume. A scalable system architecture is necessary to support higher workloads while maintaining consistent performance and service quality.
03	Security	Protecting user data is a critical requirement of the book store platform. This includes implementing strong encryption for data transmission and storage, secure authentication mechanisms, and regular security checks to identify and address potential vulnerabilities.
04	Usability	The platform should feature a clear, visually appealing interface that is easy to understand and navigate. To improve customer satisfaction, usability testing plays an important role in collecting user feedback and refining the interface to better match user expectations and behavior.
05	Accessibility	The book store should be accessible to all users, including individuals with disabilities. Compliance with web accessibility standards such as WCAG helps ensure that users with different abilities can effectively browse books, interact with features, and complete purchases on the platform.

Table 2: Nonfunctional requirements of Book Store

5.2. User Stories

- As a customer

  - To quickly find books that match my interests, I want to use a powerful search feature that allows me to look up books by title, category, author, or price range.
  - To narrow down my choices and focus on the most relevant results, I would like to apply filters such as price range, discounts, popularity, and customer ratings.
  - Before completing a purchase, I want to add books to my shopping cart and easily manage them by increasing quantities, removing items, or reviewing selected products.
  - To complete my order with confidence, I expect a simple and secure checkout process that supports multiple payment options, including COD or VNPay (ATM, Credit Card, E-Wallet).
  - To stay informed about my purchases, I would like to receive email notifications and order tracking updates after placing an order.
  - After purchasing books, I want to leave ratings and reviews so I can share my experience and help other readers make informed decisions.
  - I would like to browse a curated collection of popular and recommended books across different genres, such as fiction, business, education, and self-development.
  - To make well-informed purchasing decisions, I expect to see detailed book information, including descriptions, authors, publication details, pricing, and available discounts.
  - To evaluate the quality and relevance of a book, I want to read feedback and ratings provided by other customers.
  - I would like the ability to publicly submit my own reviews and comments on books so that other users can view and benefit from my opinions.
- As a visitor
  - I would like the website to recommend books automatically based on my browsing behavior and reading interests.
  - I want to search for books using specific attributes such as genre, author, language, or format, and I would like to add or remove books from my shopping cart even when I am not logged in.
- As an administrator
  - I want to manage and maintain all user accounts, book listings, and related system data to ensure the store's information remains accurate and up to date.

  + I want to monitor and moderate customer reviews to make sure they comply with platform guidelines and do not contain inappropriate or misleading content.
  + To support informed business decisions, I would like to analyze sales data and performance metrics, especially for best-selling books and popular categories.
  + My objective is to assist customers by handling issues related to orders, payments, or book quality in a timely and effective manner.
- As a payment service
  + I need access to essential customer information, such as the customer's name and account details, to ensure accurate and secure payment processing.
  + To minimize errors and protect transactions, the system should require users to comply with established purchasing policies and security standards before completing an order.

5.3. Diagrams

5.3.1. Context Diagram

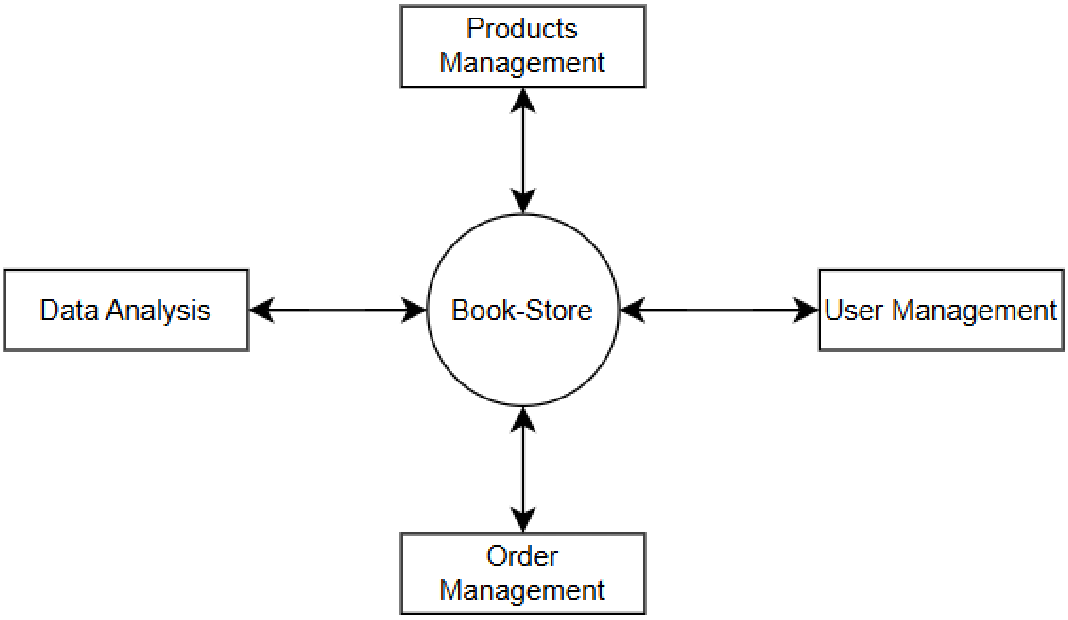


Figure 4: Context diagram of Book Store

Action	Goal
Product Management	Manages book’s information including price, book title and category, updating new price and description, adding new books, and deleting products.
User Management	Manage customer information including email, role, account status, etc. On top of that, the admin can change the customer's status from active to banned and vice versa.
Data Analysis	Analyzes store data such as sales performance, order trends, and popular products to support decision-making. The admin can view statistical summaries and visual reports to better understand customer behavior and overall business performance.
Orders Management	Manage order status (Pending, Paid, Shipped, Completed, Cancelled), payment status(Pending, Paid, Refunded), order date, and view order information of customers.

Table 3: Context diagram analysis

5.3.2. Use case Diagram

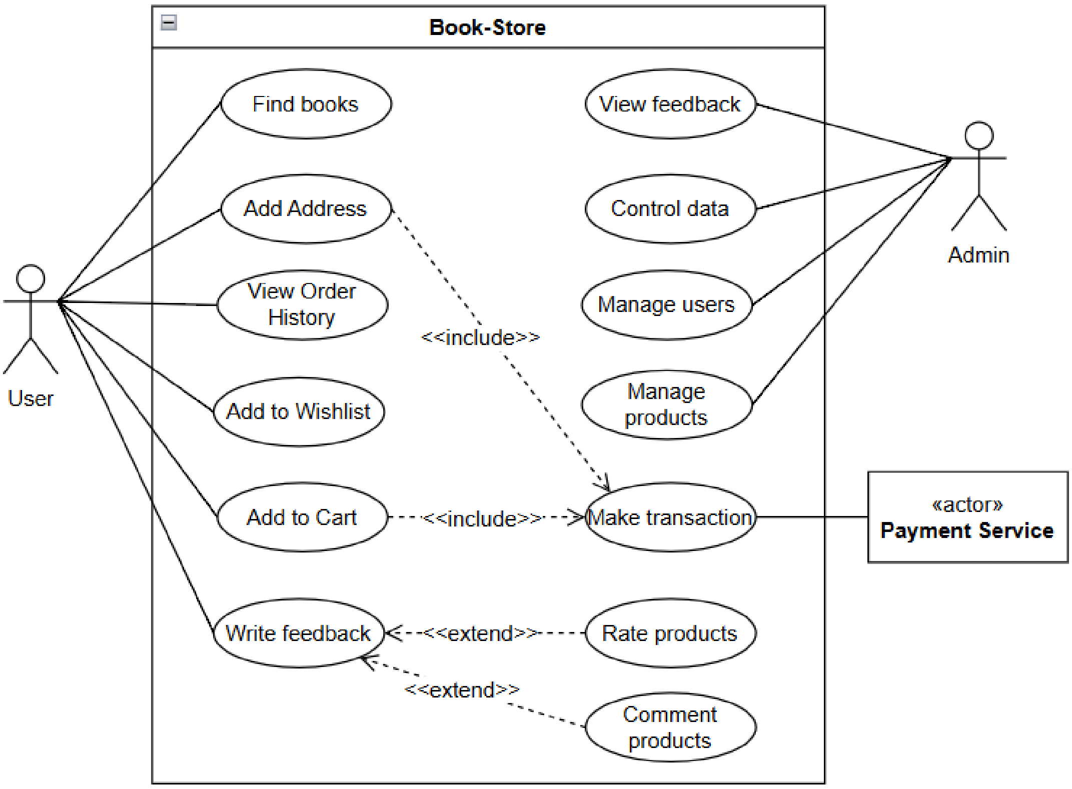


Figure 5: Use Case Diagram of Book Store

5.3.2.1. Actors Analysis

Actor	Role
User	The primary site participant who can browse available products, make purchases, and submit comments or reviews.
Admin	The user with the highest level of access, responsible for managing customers, adding or removing products, overseeing system activities, and performing administrative tasks across the platform.

Table 4: Actor Explanation

5.3.2.2. Use case Analysis

Use case	Goal
Find products	Customers can find books based on their preference or genre.
Add address	Customers can add many addresses to expand their choice when they want to receive books in another location.
View Order History	Customers can view their order's information such as order status, shipping address and customer details.
Add to Cart	Customers can add their favorite book to their Cart before making payment.
Add to Wishlist	Customers add favorite books to their Wishlist.
Comment products	Customers can write a comment to share their feelings about a specific book.
Rate products	Customers can rate products.
View feedback	Admin can view comment and rate of customers
Control data	Admin can control and receives overview data such as Total Books, Total Sales, Trending Books, Total Orders, etc
Manage users	Admin manages user information.
Manage products	Admin can perform various tasks such as adding, editing, and deleting products.

Table 5: Use cases explanation

5.3.2.3. Relationship

- The “Make Transaction” use case requires the user to have at least one product in the shopping cart; therefore, it is dependent on and builds upon the “Add to Cart” use case.
- The “Rate Products” and “Comment Products” use cases are combined under “Write feedback,” indicating that both actions contribute to providing overall product feedback within the system.

5.3.3. Entity Relationship Diagram (ERD)

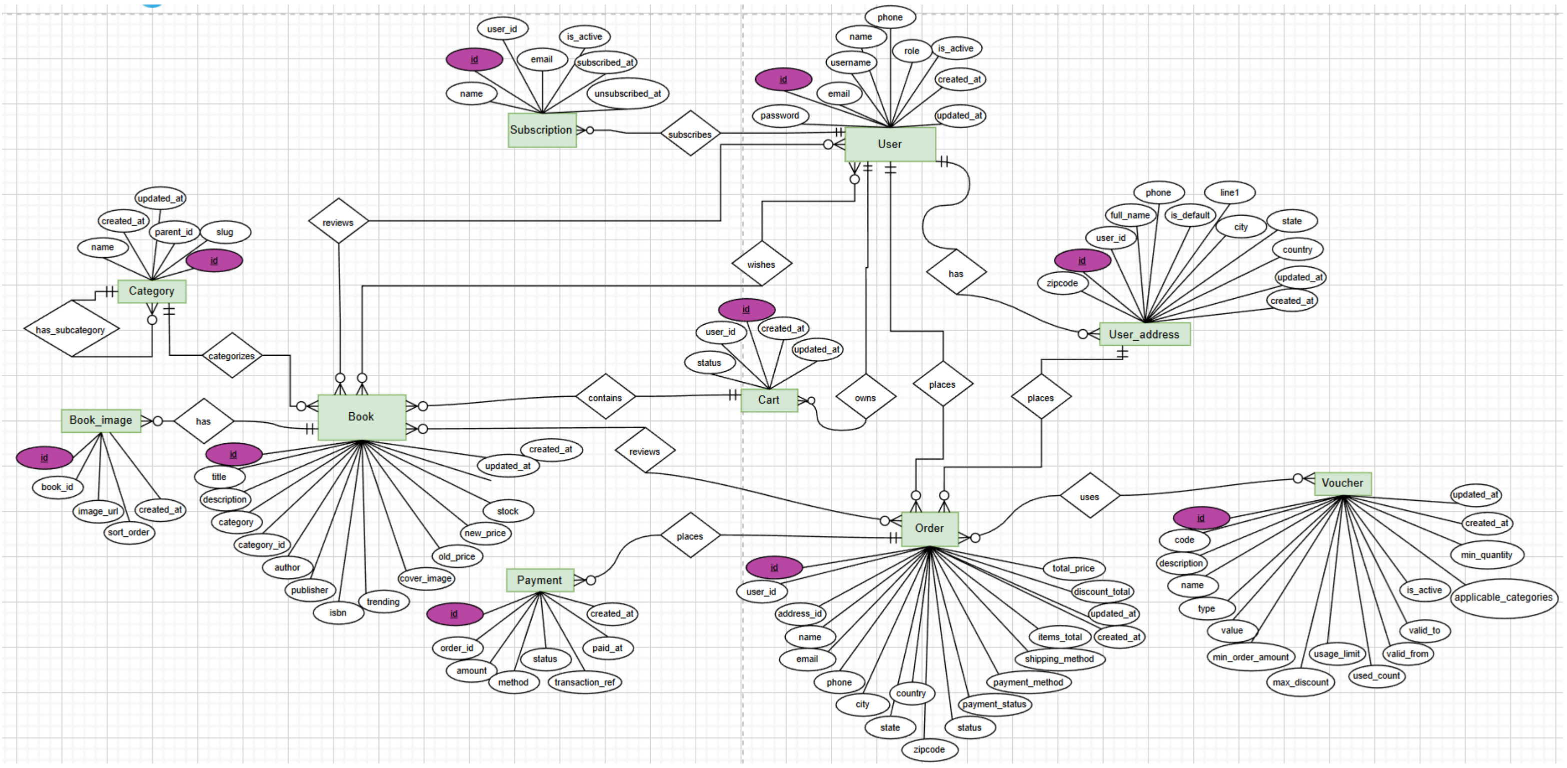
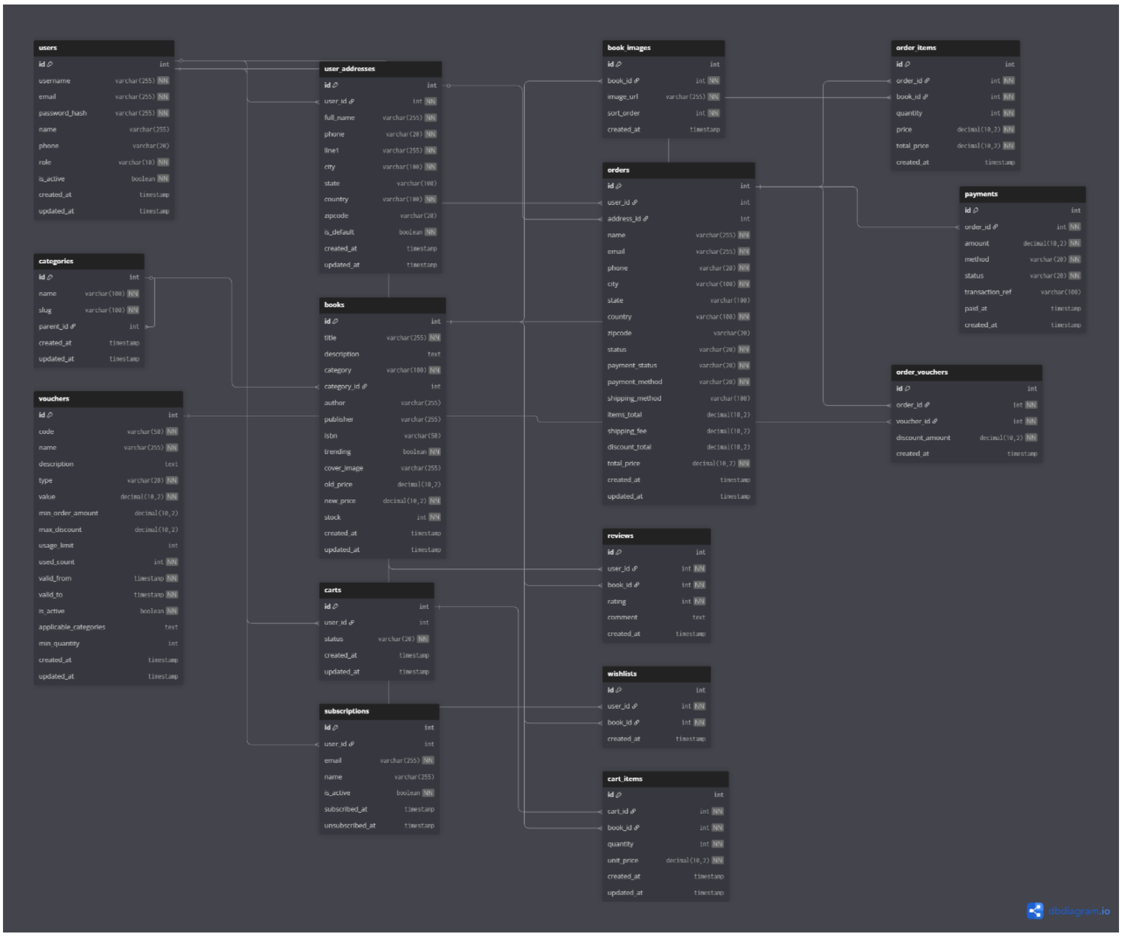


Figure 6: ERD of Book Store



users		
id	int	
username	varchar(255)	NN
email	varchar(255)	NN
password_hash	varchar(255)	NN
name	varchar(255)	
phone	varchar(20)	
role	varchar(10)	NN
is_active	boolean	NN
created_at	timestamp	
updated_at	timestamp	

user_addresses		
id	int	
user_id	int	NN
full_name	varchar(255)	NN
phone	varchar(20)	NN
line1	varchar(255)	NN
city	varchar(100)	NN
state	varchar(100)	
country	varchar(100)	NN
zipcode	varchar(20)	
is_default	boolean	NN
created_at	timestamp	
updated_at	timestamp	

categories		
id	int	
name	varchar(100)	NN
slug	varchar(100)	NN
parent_id	int	
created_at	timestamp	
updated_at	timestamp	

books		
id	int	
title	varchar(255)	NN
description	text	
category	varchar(100)	NN
category_id	int	
author	varchar(255)	
publisher	varchar(255)	
isbn	varchar(50)	
trending	boolean	NN
cover_image	varchar(255)	
old_price	decimal(10,2)	
new_price	decimal(10,2)	NN
stock	int	NN
created_at	timestamp	
updated_at	timestamp	

vouchers		
id	int	
code	varchar(50)	NN
name	varchar(255)	NN
description	text	
type	varchar(20)	NN
value	decimal(10,2)	NN
min_order_amount	decimal(10,2)	
max_discount	decimal(10,2)	
usage_limit	int	
used_count	int	NN
valid_from	timestamp	NN
valid_to	timestamp	NN
is_active	boolean	NN
applicable_categories	text	
min_quantity	int	
created_at	timestamp	
updated_at	timestamp	

carts		
id	int	
user_id	int	
status	varchar(20)	NN
created_at	timestamp	
updated_at	timestamp	

subscriptions		
id	int	
user_id	int	
email	varchar(255)	NN
name	varchar(255)	
is_active	boolean	NN
subscribed_at	timestamp	
unsubscribed_at	timestamp	

book_images		
id	int	
book_id	int	NN
image_url	varchar(255)	NN
sort_order	int	NN
created_at	timestamp	

orders		
id	int	
user_id	int	
address_id	int	
name	varchar(255)	NN
email	varchar(255)	NN
phone	varchar(20)	NN
city	varchar(100)	NN
state	varchar(100)	
country	varchar(100)	NN
zipcode	varchar(20)	
status	varchar(20)	NN
payment_status	varchar(20)	NN
payment_method	varchar(20)	NN
shipping_method	varchar(100)	
items_total	decimal(10,2)	
shipping_fee	decimal(10,2)	
discount_total	decimal(10,2)	
total_price	decimal(10,2)	NN
created_at	timestamp	
updated_at	timestamp	

reviews		
id	int	
user_id	int	NN
book_id	int	NN
rating	int	NN
comment	text	
created_at	timestamp	

wishlists		
id	int	
user_id	int	NN
book_id	int	NN
created_at	timestamp	

cart_items		
id	int	
cart_id	int	NN
book_id	int	NN
quantity	int	NN
unit_price	decimal(10,2)	NN
created_at	timestamp	
updated_at	timestamp	

order_items		
id	int	
order_id	int	NN
book_id	int	NN
quantity	int	NN
price	decimal(10,2)	NN
total_price	decimal(10,2)	NN
created_at	timestamp	

payments		
id	int	
order_id	int	NN
amount	decimal(10,2)	NN
method	varchar(20)	NN
status	varchar(20)	NN
transaction_ref	varchar(100)	
paid_at	timestamp	
created_at	timestamp	

order_vouchers		
id	int	
order_id	int	NN
voucher_id	int	NN
discount_amount	decimal(10,2)	NN
created_at	timestamp	

Figure 7: Database Design

5.3.4. Class Diagram

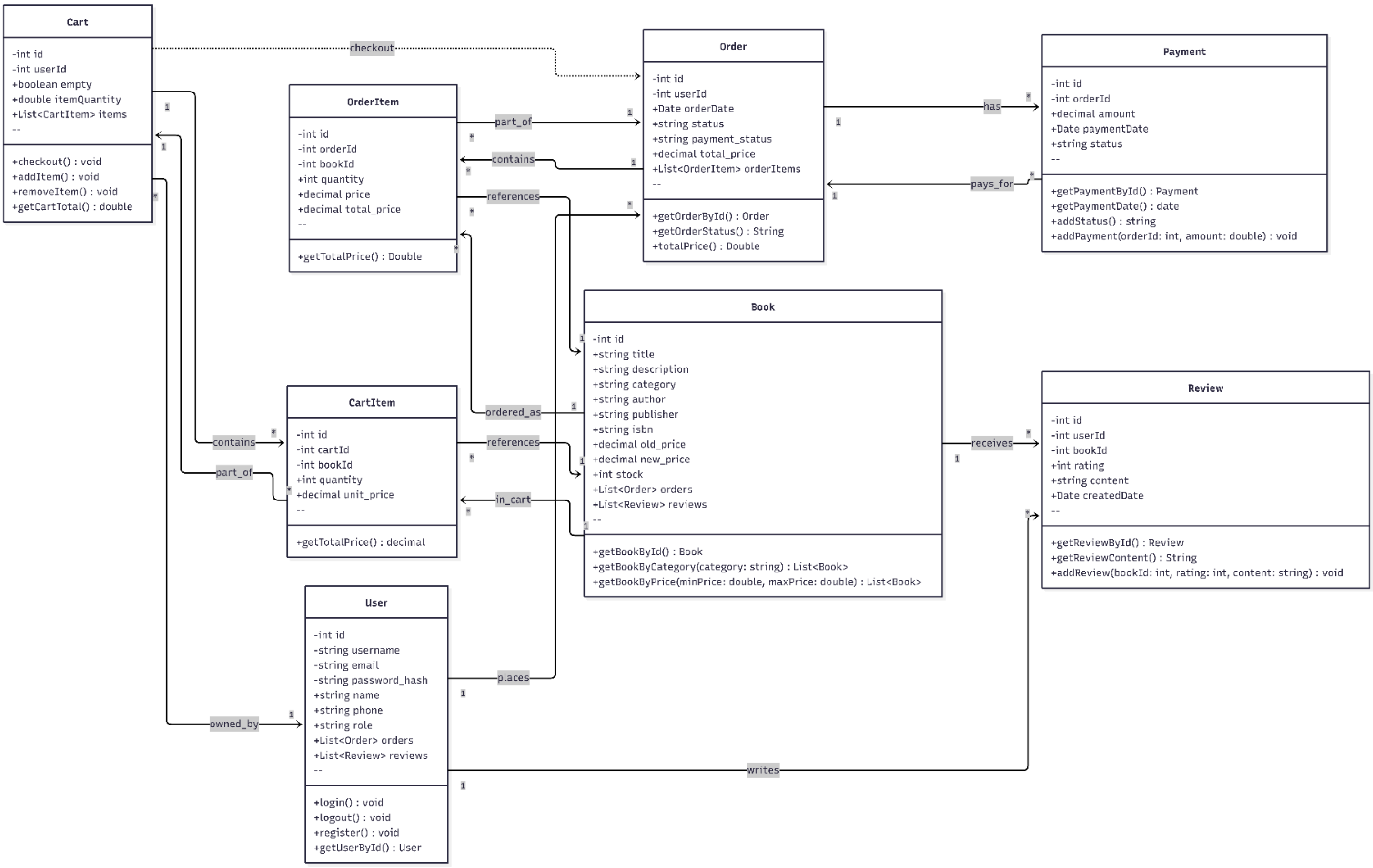


Figure 8: Class Diagram of Book Store

5.3.5. MVC Model

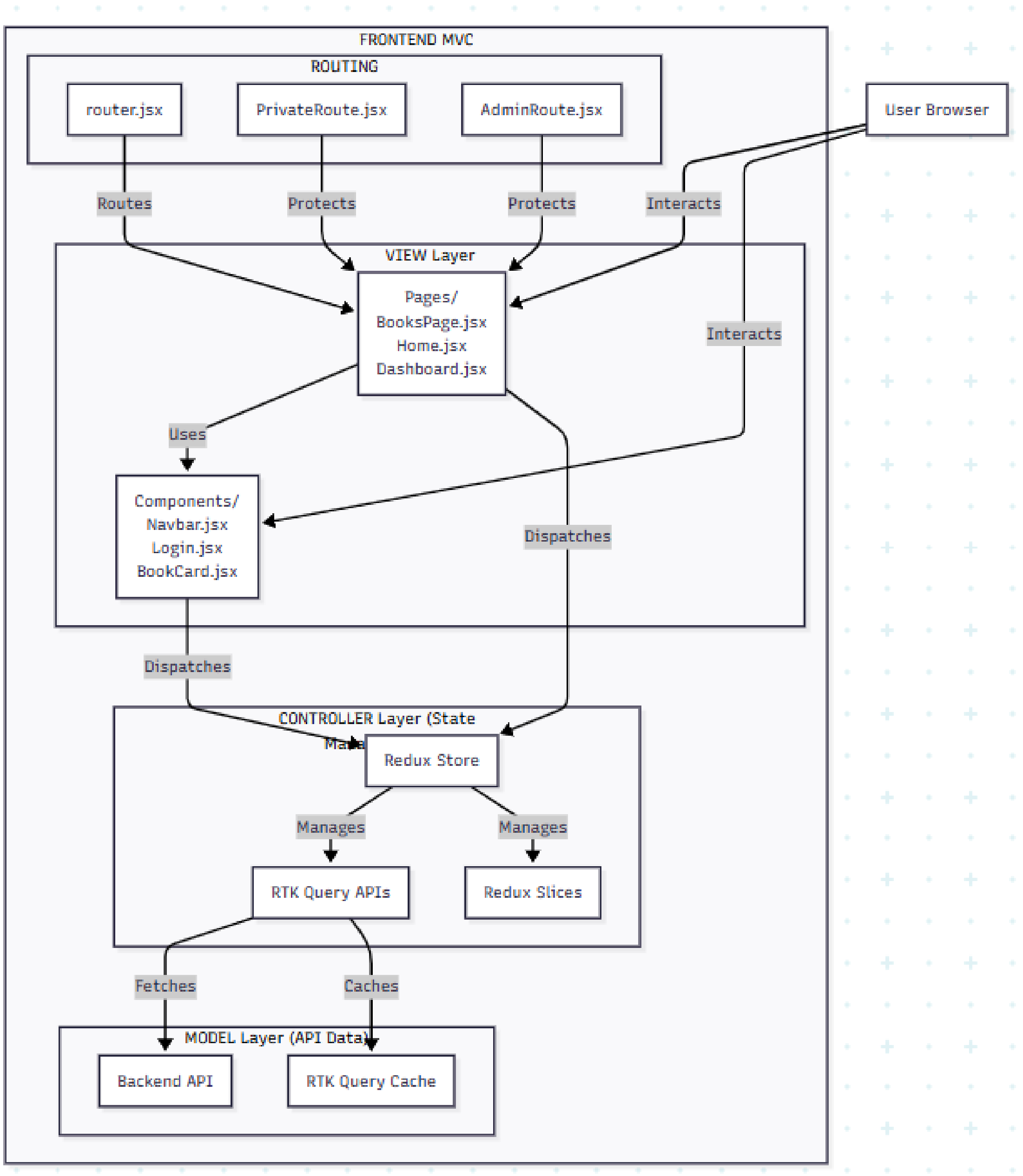


Figure 9: MVC Model of Frontend

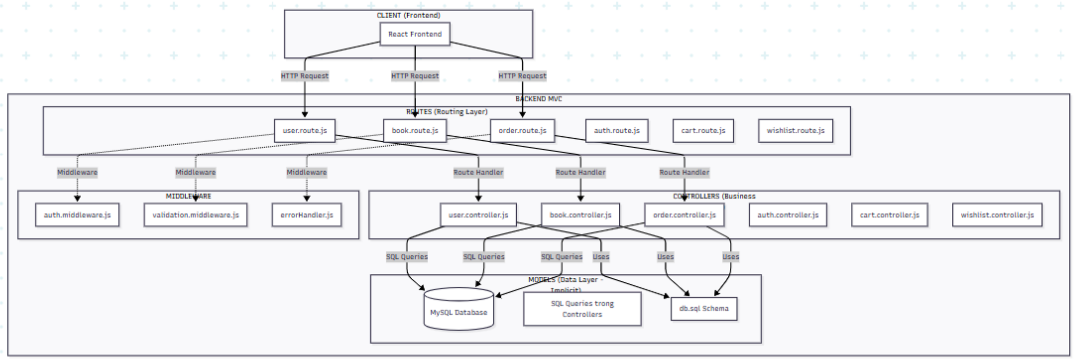


Figure 10: MVC Model of Backend

5.3.6. Sequence Diagrams

5.3.6.1. Login functionality

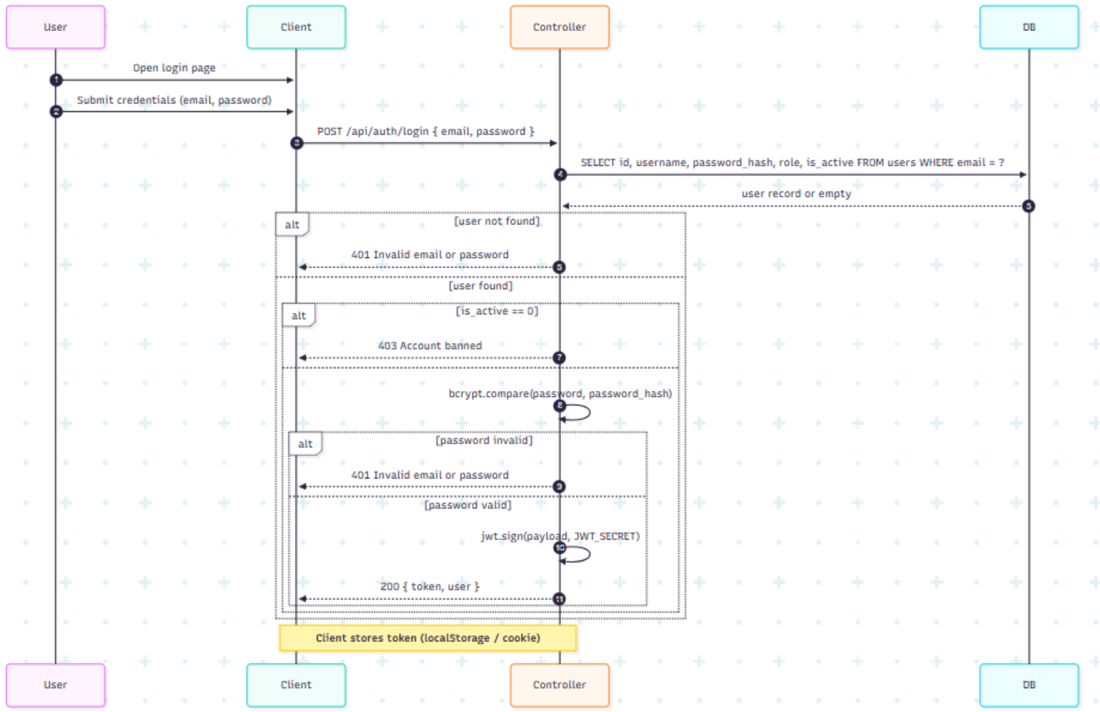

User
Client
Controller
DB
Open login page
Submit credentials (email, password)
POST /api/auth/login { email, password }
SELECT id, username, password_hash, role, is_active FROM users WHERE email = ?
user record or empty
alt
[user not found]
401 Invalid email or password
[user found]
alt
[is_active == 0]
403 Account banned
bcrypt.compare(password, password_hash)
alt
[password invalid]
401 Invalid email or password
[password valid]
jwt.sign(payload, JWT_SECRET)
200 { token, user }
Client stores token (localStorage / cookie)
User
Client
Controller
DB

Figure 11: Sequence diagram for "Login" Functionality

- Actor:
    + Customer: The end user who initiates the login process by providing an email and password.
- Object:
    + Client: The front-end application that displays the login form and sends login requests to the backend.

  - Controller: The backend component responsible for handling authentication logic and coordinating communication between the client and the database.
  - Database: Stores user records, including email, encrypted password, role, and account status.

- Control flow:
  - The customer opens the login page on the client application and enters their email and password.
  - The client sends a POST /api/auth/login request containing the login credentials to the controller.
  - The controller queries the database to retrieve the user record associated with the provided email.
  - If no matching user record is found, the system terminates the process and returns an error indicating invalid email or password.
  - If a user record is found, the controller checks whether the account is active.
  - If the account is inactive or banned, the login request is rejected and an account restriction message is returned.
  - If the account is active, the controller compares the submitted password with the stored hashed password using a secure hashing mechanism.
  - If the password comparison fails, the login process is denied and an error response is sent back to the client.
  - If the password is valid, the controller generates a JSON Web Token (JWT) containing user information.
  - The system returns a successful login response along with the authentication token.
  - The client stores the token securely (e.g., local storage or cookie) for future authenticated requests.
- Interactions:
  - Customer initiates login: The customer submits login credentials via the login form.
  - Client sends authentication request: The client forwards the credentials to the backend controller.
  - Controller queries database: The controller retrieves user data based on the provided email.
  - User not found: The system responds with an authentication failure message.

  + Account status check: The controller verifies whether the user account is active.
  + Password verification: The controller validates the password using encrypted comparison.
  + Successful authentication: A JWT token is generated and returned to the client.
  + Session persistence: The client stores the token to maintain the user's authenticated session.

5.3.6.2. Registration functionality

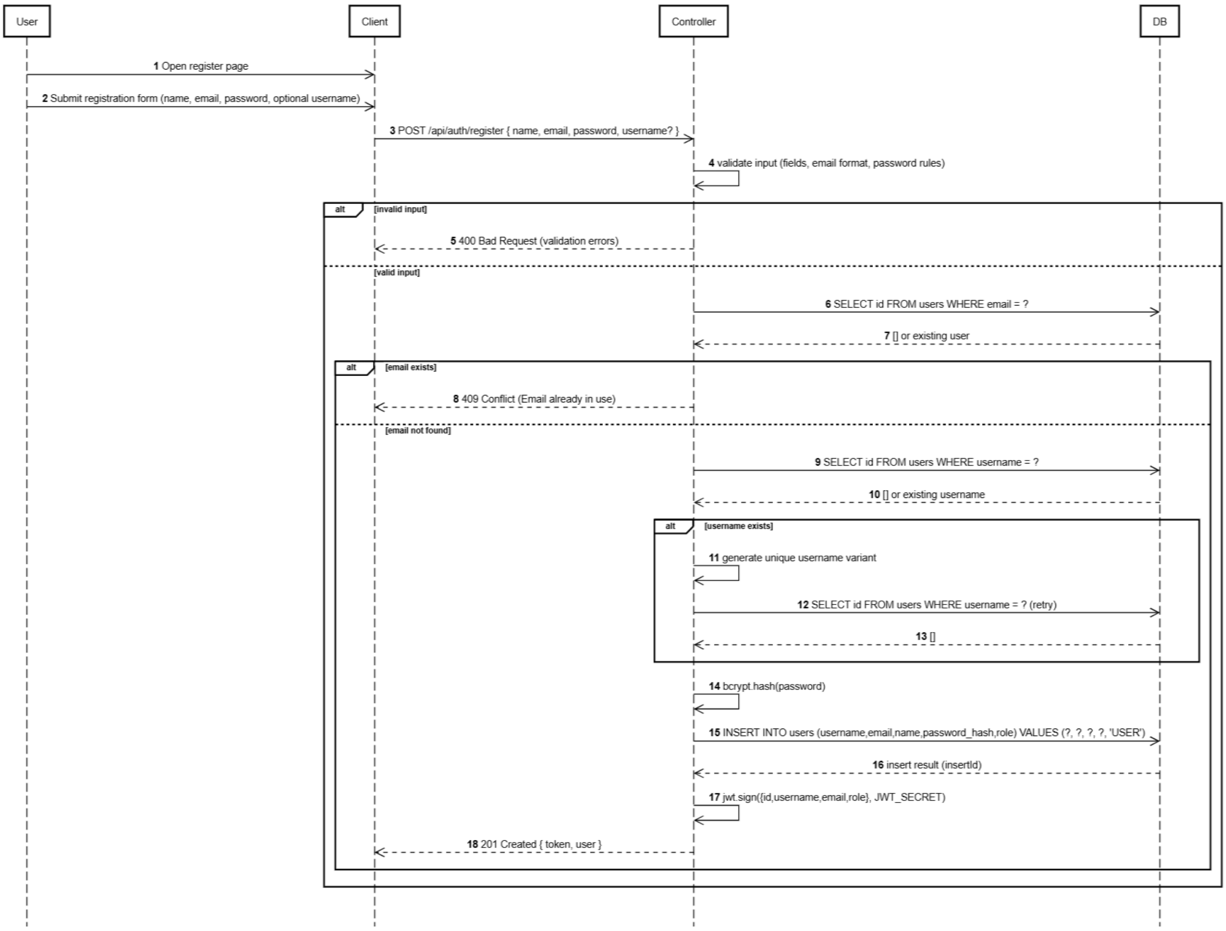

User
Client
Controller
DB
1 Open register page
2 Submit registration form (name, email, password, optional username)
3 POST /api/auth/register { name, email, password, username? }
4 validate input (fields, email format, password rules)
alt
[invalid input]
5 400 Bad Request (validation errors)
[valid input]
6 SELECT id FROM users WHERE email = ?
7 [] or existing user
alt
[email exists]
8 409 Conflict (Email already in use)
[email not found]
9 SELECT id FROM users WHERE username = ?
10 [] or existing username
alt
[username exists]
11 generate unique username variant
12 SELECT id FROM users WHERE username = ? (retry)
13 []
14 bcrypt.hash(password)
15 INSERT INTO users (username,email,name,password_hash,role) VALUES (?, ?, ?, ?, 'USER')
16 insert result (insertId)
17 jwt.sign({id,username,email,role}, JWT_SECRET)
18 201 Created { token, user }

Figure 12: Sequence diagram for "Register' functionality

- Actor:
  + Customer: The user who initiates the registration process to create a new account on the Book Store platform.
- Objects:

  - Client: The front-end application that displays the registration form and sends user data to the backend.
  - Controller: The backend component responsible for validating input data, handling business logic, and coordinating with the database.
  - Database: Stores user information such as email, username, encrypted password, role, and account status.
- Control Flow:
  - The customer opens the registration page from the client application.
  - The customer submits the registration form, including name, email, password, and an optional username.
  - The client sends a POST /api/auth/register request to the controller with the submitted data.
  - The controller validates the input fields, including required fields, email format, and password rules.
  - If the input validation fails, the process stops and a validation error response is returned to the client.
  - If the input is valid, the controller checks the database to determine whether the email already exists.
  - If the email is already registered, the system returns a conflict error indicating that the email is in use.
  - If the email is not found, the controller checks whether the provided username already exists.
  - If the username is already taken, the system generates a unique username variant and verifies its availability.
  - Once a unique username is confirmed, the controller encrypts the user's password using a secure hashing algorithm.
  - A new user record is inserted into the database with the encrypted password and default user role.
  - After successful insertion, the controller generates a JSON Web Token (JWT) for the new user.
  - The system returns a successful registration response along with the authentication token.
  - The client stores the token securely and redirects the user to the authenticated area of the application.

5.3.6.3. Adding products functionality

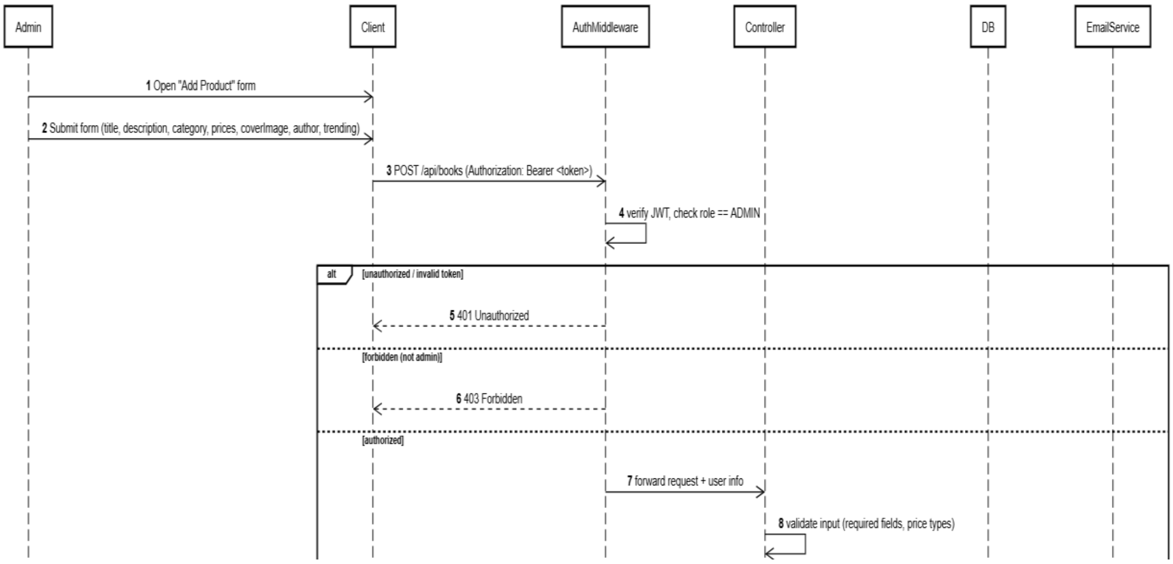

Admin
Client
AuthMiddleware
Controller
DB
EmailService
1 Open "Add Product" form
2 Submit form (title, description, category, prices, coverImage, author, trending)
3 POST /api/books (Authorization: Bearer <token>)
4 verify JWT, check role == ADMIN
alt
[unauthorized / invalid token]
5 401 Unauthorized
[forbidden (not admin)]
6 403 Forbidden
[authorized]
7 forward request + user info
8 validate input (required fields, price types)

Figure 13: Sequence diagram for "Adding Products" functionality part 1

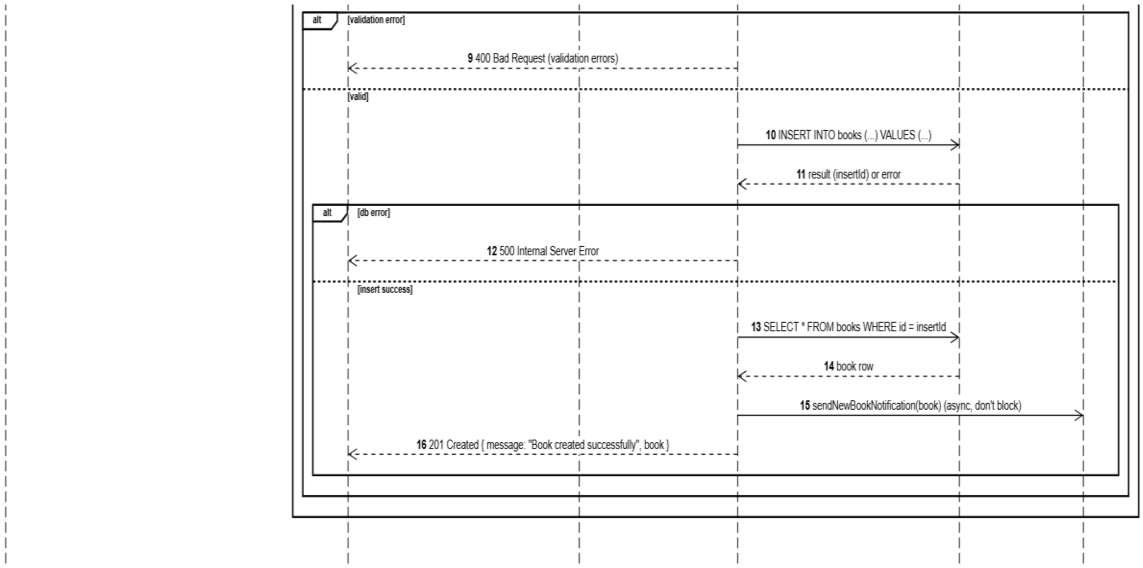

alt
[validation error]
9 400 Bad Request (validation errors)
[valid]
10 INSERT INTO books (...) VALUES (...)
11 result (insertId) or error
alt
[db error]
12 500 Internal Server Error
[insert success]
13 SELECT * FROM books WHERE id = insertId
14 book row
15 sendNewBookNotification(book) (async, don't block)
16 201 Created { message: "Book created successfully", book }

Figure 14: Sequence diagram for "Adding Products" functionality part 2

- Actor:
  - Admin: The administrator who initiates the process of adding a new book to the system.
- Objects:
  - Client: The user interface where the admin enters book information.
  - Auth Middleware: Verifies authentication token and checks admin authorization.
  - Controller: Handles business logic related to adding a new book.

  - Database: Stores book information such as title, description, price, category, and metadata.
  - Email Service: Sends notifications to subscribers after a new book is added.
- Control Flow:
  - The admin opens the “Add Product” form and enters book details including title, description, category, price, author, cover image, and trending status.
  - The client sends a request to the backend with the provided information along with an authentication token.
  - The authentication middleware verifies the token and confirms that the user has admin privileges.
  - If the token is invalid or the user is not authorized, the system stops the process and returns an error.
  - Once authorization is confirmed, the controller validates the submitted book data.
  - If validation fails, the system responds with an error indicating missing or incorrect input.
  - If validation succeeds, the controller stores the new book information in the database.
  - After successful insertion, the system retrieves the newly created book record. A background process triggers the email service to notify subscribers about the new book.
  - The system returns a success response and the client updates the user interface accordingly.
- Interactions:
  - Admin initiates product addition: The admin fills in the add-book form and submits the request.
  - Client sends add-book request: The client forwards the book data and authentication token to the backend.
  - Authentication check: The middleware validates the token and ensures the user has admin access.
  - Input validation: The controller verifies required fields such as title, price, and category.
  - Book creation: The controller inserts the new book into the database.
  - Book retrieval: The system fetches the newly created book record for confirmation.

+ Notification handling: The email service sends notifications to subscribers in the background.
+ System feedback: The admin receives confirmation that the book was added successfully and the product list is updated.

5.3.6.4. Checkout functionality

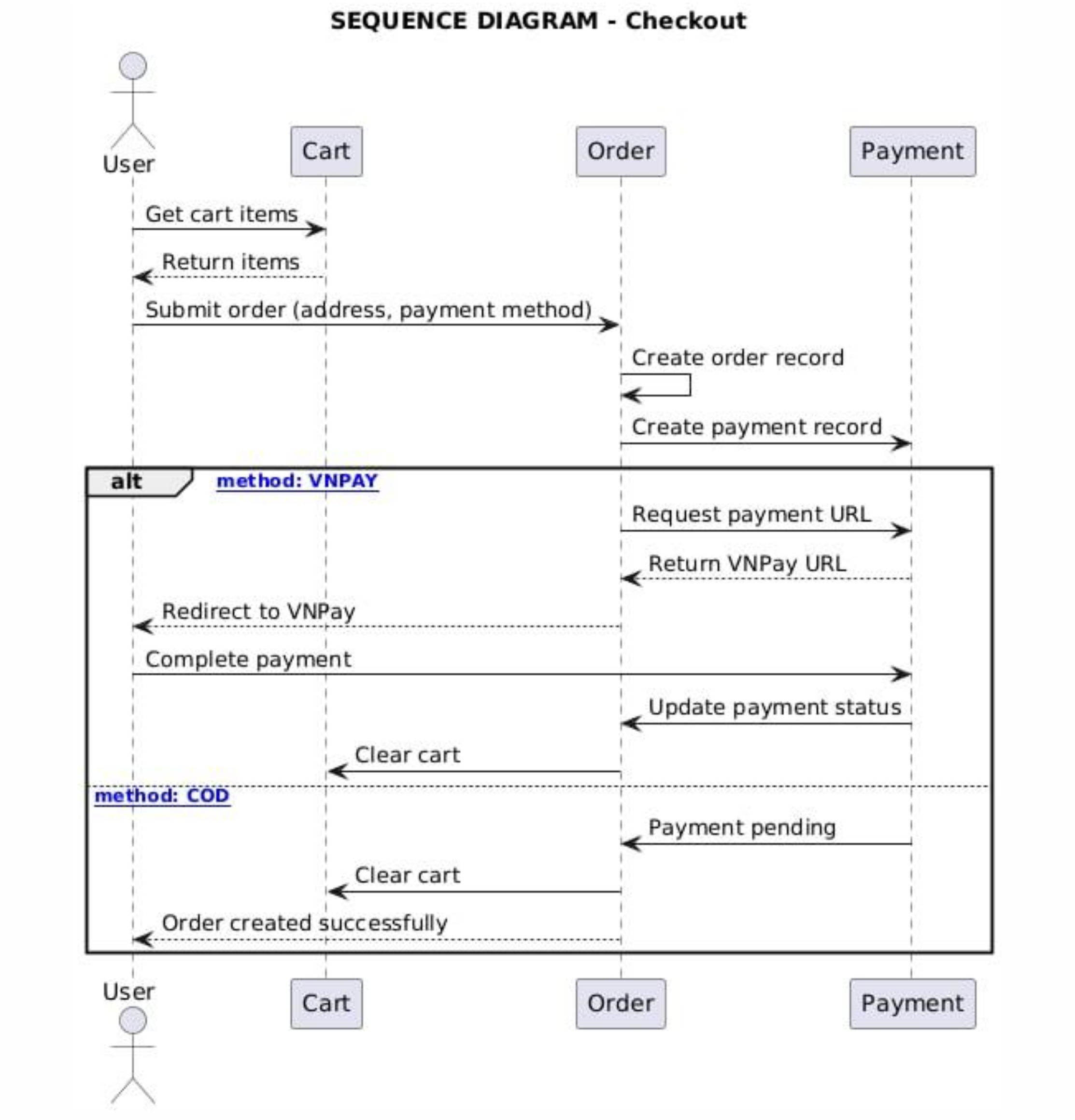


Figure 15: Sequence diagram for "Checkout" functionality

- Actor:
  + Customer: The user who proceeds to checkout in order to purchase selected books from the shopping cart.
- Objects:

  - Client: The frontend interface where the customer reviews cart items, enters shipping information, selects a payment method, and confirms the order.
  - Cart: Stores the list of selected books and their quantities before checkout.
  - Order: Responsible for creating and managing order records.
  - Payment: Handles payment processing and payment status updates based on the selected payment method.
- Control Flow:
  - The customer reviews the shopping cart and initiates the checkout process.
  - The system retrieves the current cart items and displays them to the customer.
  - The customer submits the checkout form, including delivery address and selected payment method.
  - The system creates a new order record associated with the customer.
  - A corresponding payment record is generated for the order.
  - The checkout flow then branches based on the selected payment method:
    - VNPay method:
      - The system requests a payment URL from the VNPay gateway.
      - The VNPay payment URL is returned to the client.
      - The customer is redirected to VNPay to complete the payment.

        After successful payment, the payment status is updated accordingly.
      - The shopping cart is cleared.
    - Cash on Delivery (COD) method:
      - The payment status is set to Pending.
      - The shopping cart is cleared immediately after order creation.
  - Once the checkout process is completed, the system confirms that the order has been successfully created.
  - The customer receives a confirmation message and can view the order in their order history.
- Interactions:
  - Customer initiates checkout: The customer clicks the checkout button after reviewing cart contents.
  - Cart retrieval: The system fetches and returns the current cart items to the customer.

- Order submission: The client submits checkout data, including address and payment method.
- Order creation: The system creates a new order record in the database.
- Payment record creation: A payment entry is created and linked to the order.
- VNPay processing: For online payments, the system redirects the customer to VNPay and updates payment status upon completion.
- COD handling: For cash payments, the system marks the payment as pending.
- Cart clearing: The shopping cart is cleared after successful order creation.
- System feedback: The customer is notified that the order has been created successfully and can track it in the order history.

5.3.6.5. Addresses functionality

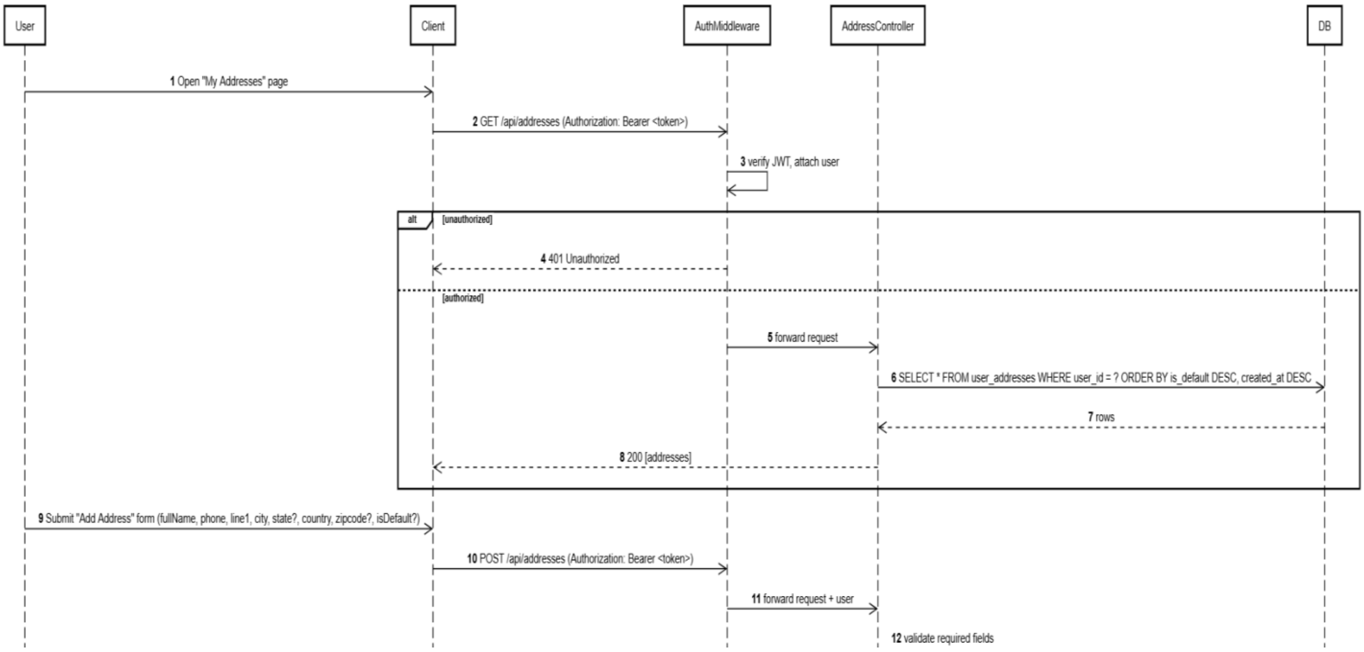

User
Client
AuthMiddleware
AddressController
DB
1 Open "My Addresses" page
2 GET /api/addresses (Authorization: Bearer <token>)
3 verify JWT, attach user
alt
[unauthorized]
4 401 Unauthorized
[authorized]
5 forward request
6 SELECT * FROM user_addresses WHERE user_id = ? ORDER BY is_default DESC, created_at DESC
7 rows
8 200 [addresses]
9 Submit "Add Address" form (fullName, phone, line1, city, state?, country, zipcode?, isDefault?)
10 POST /api/addresses (Authorization: Bearer <token>)
11 forward request + user
12 validate required fields

Figure 16: Sequence diagram for "Addresses" functionality part 1

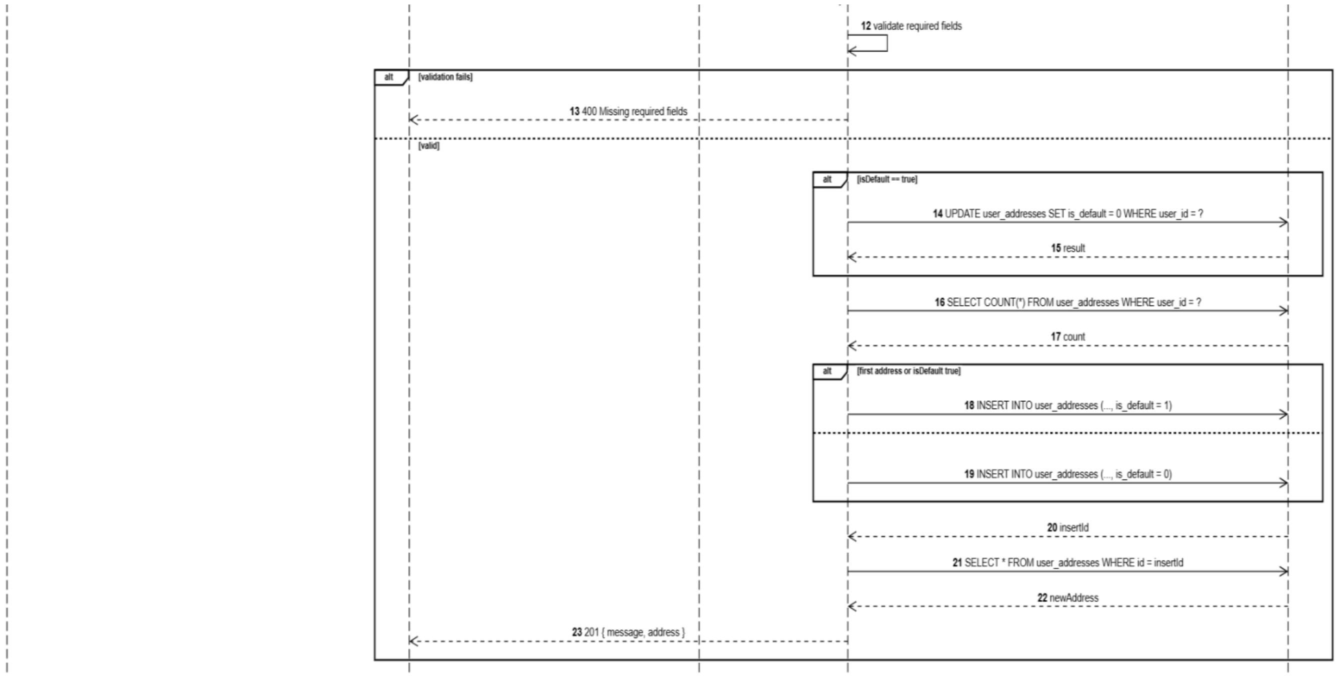

12 validate required fields
alt
[validation fails]
13 400 Missing required fields
[valid]
alt
[isDefault == true]
14 UPDATE user_addresses SET is_default = 0 WHERE user_id = ?
15 result
16 SELECT COUNT(*) FROM user_addresses WHERE user_id = ?
17 count
alt
[first address or isDefault true]
18 INSERT INTO user_addresses (..., is_default = 1)
19 INSERT INTO user_addresses (..., is_default = 0)
20 insertId
21 SELECT * FROM user_addresses WHERE id = insertId
22 newAddress
23 201 { message, address }

Figure 17: Sequence diagram for “Addresses” functionality part 2

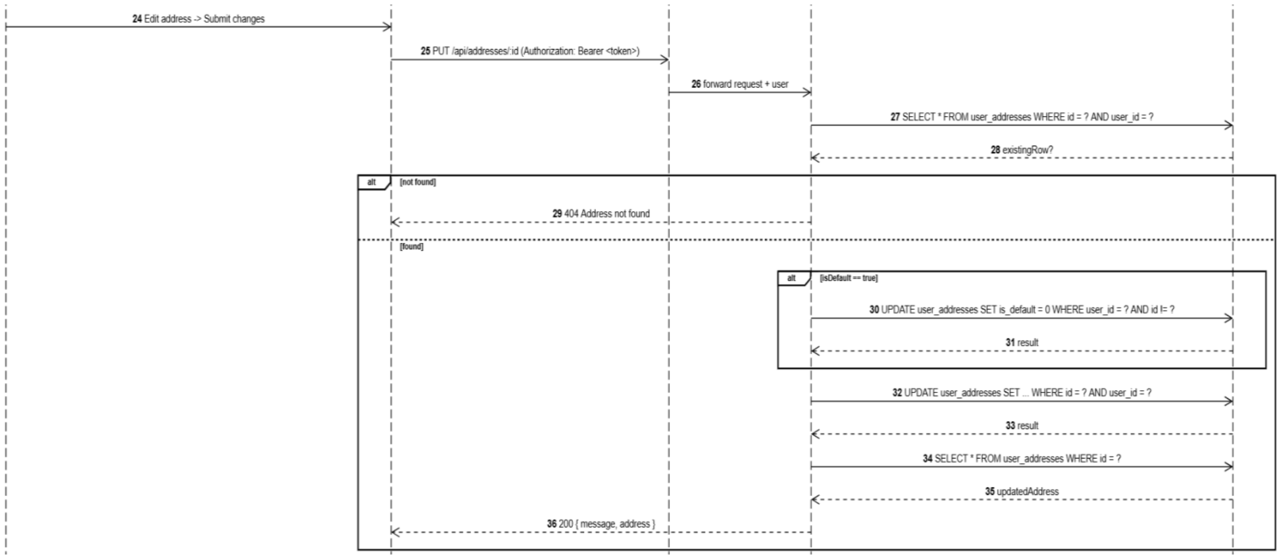

24 Edit address -> Submit changes
25 PUT /api/addresses/:id (Authorization: Bearer <token>)
26 forward request + user
27 SELECT * FROM user_addresses WHERE id = ? AND user_id = ?
28 existingRow?
alt
[not found]
29 404 Address not found
[found]
alt
[isDefault == true]
30 UPDATE user_addresses SET is_default = 0 WHERE user_id = ? AND id != ?
31 result
32 UPDATE user_addresses SET ... WHERE id = ? AND user_id = ?
33 result
34 SELECT * FROM user_addresses WHERE id = ?
35 updatedAddress
36 200 { message, address }

Figure 18: Sequence diagram for “Addresses” functionality part 3

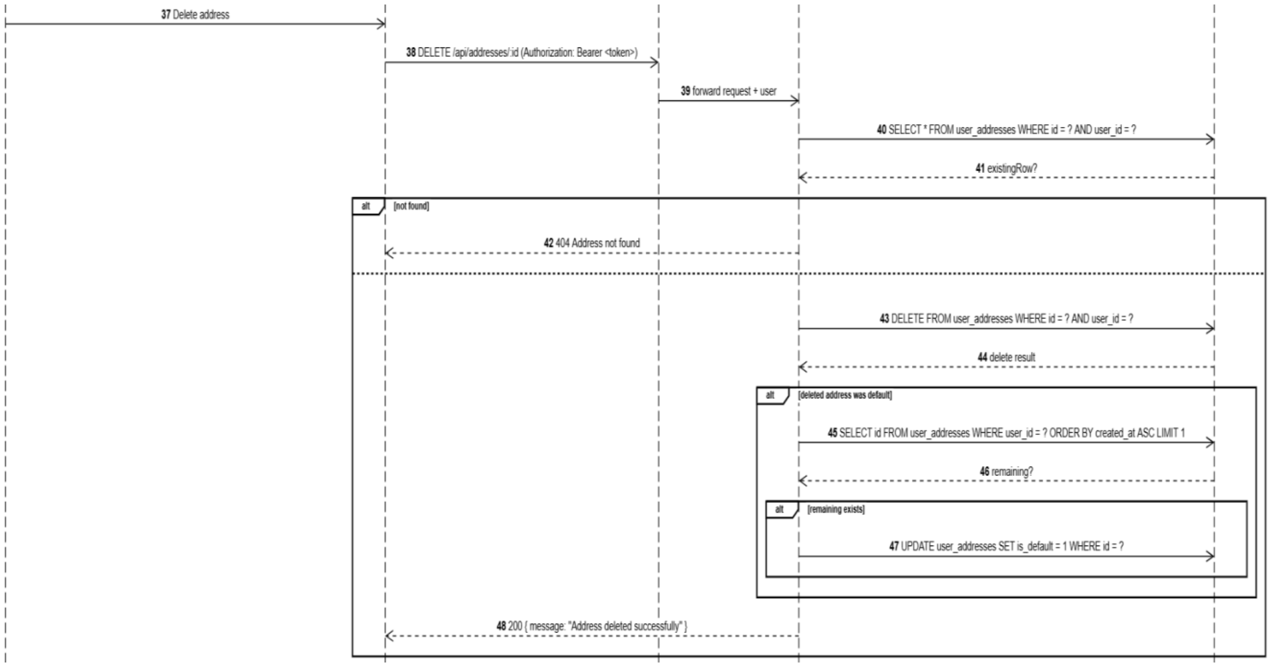

37 Delete address
38 DELETE /api/addresses/:id (Authorization: Bearer <token>)
39 forward request + user
40 SELECT * FROM user_addresses WHERE id = ? AND user_id = ?
41 existingRow?
alt
[not found]
42 404 Address not found
43 DELETE FROM user_addresses WHERE id = ? AND user_id = ?
44 delete result
alt
[deleted address was default]
45 SELECT id FROM user_addresses WHERE user_id = ? ORDER BY created_at ASC LIMIT 1
46 remaining?
alt
[remaining exists]
47 UPDATE user_addresses SET is_default = 1 WHERE id = ?
48 200 { message: "Address deleted successfully" }

Figure 19: Sequence diagram for “Addresses” functionality part 4

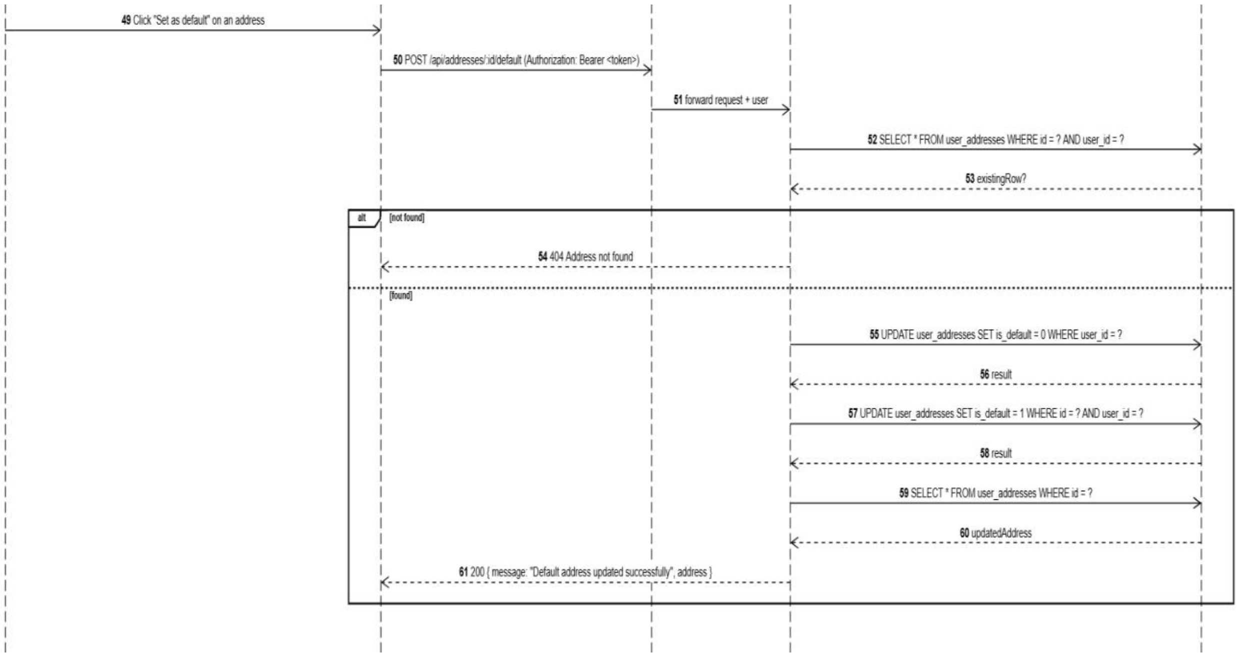

49 Click "Set as default" on an address
50 POST /api/addresses/:id/default (Authorization: Bearer <token>)
51 forward request + user
52 SELECT * FROM user_addresses WHERE id = ? AND user_id = ?
53 existingRow?
alt
[not found]
54 404 Address not found
[found]
55 UPDATE user_addresses SET is_default = 0 WHERE user_id = ?
56 result
57 UPDATE user_addresses SET is_default = 1 WHERE id = ? AND user_id = ?
58 result
59 SELECT * FROM user_addresses WHERE id = ?
60 updatedAddress
61 200 { message: "Default address updated successfully", address }

Figure 20: Sequence diagram for “Addresses” functionality part 5

- Actor:
    - User: A registered customer who manages delivery addresses for orders.
- Objects:
    - Client: The frontend interface where the user views and manages addresses.
    - Authentication Middleware: Verifies the user’s identity using an access token.
    - Address Controller: Handles address-related requests and business logic.
    - Database: Stores user address records, including default address information.

- Control Flow:
  - The user accesses the My Addresses page. The system first verifies the authentication token to ensure the user is logged in.
  - Once authenticated, the system retrieves and displays all saved addresses associated with the user.
  - When the user submits a new address, the system validates all required fields (name, phone, street, city, country, zipcode).
  - If the new address is marked as default, the system updates existing addresses to remove the default flag before saving the new one.
  - The user can edit an existing address. The system checks ownership and updates the address if it exists.

    The user may delete an address. If the deleted address is the default one, the system automatically assigns another address as the new default.
  - The user can also explicitly set an address as the default, which updates all other addresses accordingly.
- Interactions:
  - User views addresses: The client sends a request to retrieve all saved addresses after authentication is verified.
  - Add new address: The user submits address information. The controller validates input, updates default flags if needed, and stores the address in the database.
  - Edit address: The system checks whether the address exists and belongs to the user, then applies updates.
  - Delete address: The controller removes the selected address. If it was the default address, another address is automatically promoted.
  - Set default address: The system ensures only one default address exists by updating all related records.
  - System feedback: Success or error messages are returned to the user after each action (add, update, delete, or set default).

5.3.6.6. Rating functionality

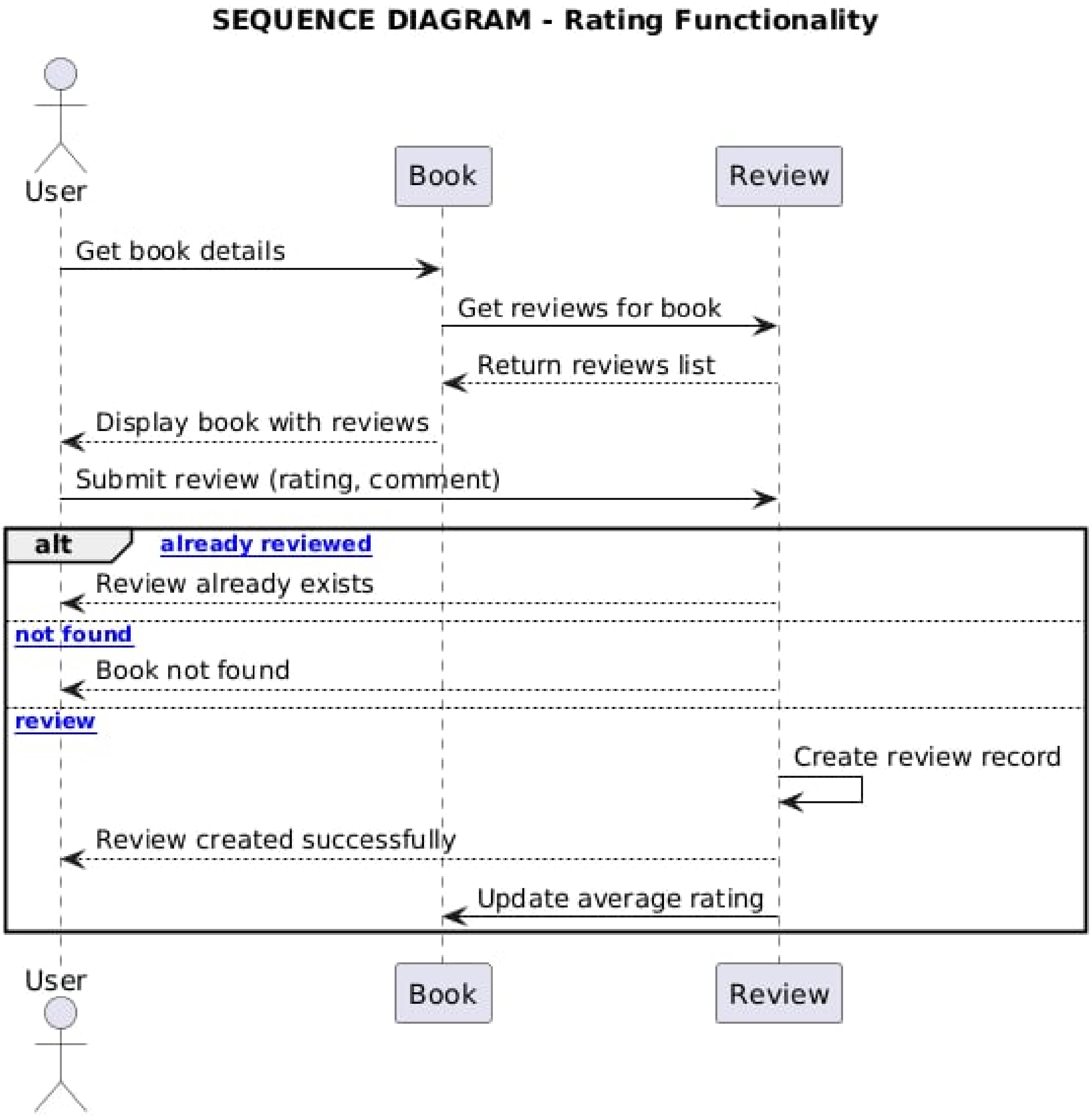


Figure 21: Sequence diagram for "Rating" functionality

- Actor:
  + Customer (User): A registered user who views a book and submits a rating and optional review comment.
- Objects:
  + Book: Provides book details and current reviews.
  + Review: Manages review creation, storage, and rating updates.
- Control Flow:
  + The user opens a book detail page.
  + The system retrieves and displays existing reviews for the selected book.
  + The user submits a rating (1–5 stars) and an optional comment.

  - The system checks whether the book exists.
  - The system checks whether the book exists.
    - If not found, an error message is returned.
  - The system checks if the user has already reviewed the book.
    - If a review already exists, duplicate submission is blocked.
  - If valid, a new review record is created.
  - The system updates the book's average rating.
  - A success message is returned to the user.
- Interactions:
  - View book reviews: The user views book details along with existing ratings.
  - Submit review: The user provides a star rating and optional comment.
  - Duplicate check: The system prevents multiple reviews from the same user.
  - Review creation: A new review is saved when conditions are met.
  - Rating update: The book's average rating is recalculated and displayed.
  - User feedback: The system confirms success or reports errors clearly.

CHAPTER 6: DESIGN AND ARCHITECTURE

6.1 Overview of The System's Architecture

The Book Store system is a complete e-commerce application designed to provide a modern and efficient online book shopping experience. The system is developed following the MERN Stack architecture, which ensures scalability, maintainability, and clear separation of concerns. The overall architecture consists of the following main components:

Customer-side(Frontend): The frontend is the user interface layer that enables customers to interact with the system visually through a web browser. It is responsible for presenting book information and supporting core user activities such as browsing products, searching for books, managing the shopping cart, placing orders, making payments, viewing order history, writing reviews, and managing personal accounts.

The user interface is built using React (version 19.2.0), a popular JavaScript framework that simplifies the development of dynamic and responsive web applications. The frontend also utilizes Vite as a build tool to improve development performance, Redux Toolkit for application state management, Tailwind CSS for styling, and React Router for page navigation.

The frontend communicates with the backend by sending and receiving data through Application Programming Interfaces (APIs). API requests are handled using the Axios library, which includes interceptors to automatically attach authentication tokens and manage error handling.

Server-side (Backend): The backend is responsible for handling business logic and processing user requests when interacting with the database. It manages critical system functionalities such as user authentication, authorization, session handling, and role-based access control for different user types (USER and ADMIN).

The server-side logic is developed using Node.js and Express.js (version 4.21.2), which facilitate efficient integration with the frontend. The backend implements JWT (JSON Web Token) for authentication, bcrypt for password encryption, Helmet and CORS for security enhancement, express-rate-limit to control request frequency, and Winston for system logging.

In addition, the backend exposes RESTful APIs to retrieve, store, and manipulate data for display on the user interface. It also integrates external services such as VNPay for online payment processing and an Email Service for sending order confirmations and notifications.

Database: The database serves as the central storage for all operational data used by the backend. The system employs MySQL (via the mysql2/promise library, version 3.15.3) as the database management system. MySQL is selected instead of MongoDB, despite the traditional MERN stack, due to the structured nature of e-commerce data and the requirement for strong data consistency through ACID transactions.

The database stores information related to users, books, categories, orders, order details, shopping carts, wishlists, reviews, delivery addresses, payments, vouchers, and system configurations. To enhance performance and scalability, the database connection is optimized using a connection pool with a maximum of 10 concurrent connections.

User Authentication: The system applies a JWT-based authentication and authorization mechanism to ensure secure data transmission and integrity between system components. Each JWT token contains essential user information such as user ID, email, name, and role, and is digitally signed using a secret key to prevent tampering.

JWT is also utilized to implement a Single Sign-On (SSO) model, allowing users to access multiple system functionalities with a single login session. Authentication tokens are stored in the browser's local storage and are automatically attached to API request headers through Axios interceptors. The system supports Role-Based Access Control (RBAC) with two main roles: USER and ADMIN, ensuring appropriate access permissions throughout the application.

6.2 Key decision and Justification

This section presents the key design decisions made during the development of the Book Store MERN E-Commerce application, along with the rationale behind each decision. These choices aim to enhance usability, scalability, security, and overall user satisfaction.

User Interface: The user interface is designed with a strong emphasis on simplicity and usability. Core functionalities are clearly presented through intuitive navigation and prominently displayed action buttons, enabling users to browse and purchase books without requiring technical expertise.
The interface follows modern design principles, featuring a clean layout, responsive design compatible with both desktop and mobile devices, and a clear visual hierarchy to guide users throughout the shopping process. Essential actions such as viewing book details, adding items to the shopping cart, and proceeding to checkout are visually distinct and easily accessible, ensuring a smooth and efficient user experience.

High-Quality Images: The system prioritizes the use of rich media content and high-quality images to enhance user engagement and purchasing confidence. Each book supports multiple images, allowing customers to examine cover designs and visual details from different perspectives.
High-resolution images provide clear visual representation of products, helping users better assess the physical appearance and quality of books. This design decision improves decision-making accuracy and contributes to a more realistic and trustworthy online shopping experience.

Search and Filter Functionality: To enable efficient product discovery, the system incorporates powerful search and filtering capabilities. Users can search for books using various criteria, including title, author, ISBN, and publisher.
Advanced filtering options allow users to refine results based on category, price range, availability status, rating, and publication date. Pagination is implemented to handle large datasets efficiently, ensuring fast response times and maintaining system performance even when navigating extensive book catalogs.

Scalable Architecture: The system adopts a modular and scalable architecture based on the MERN stack, allowing for future expansion and increased traffic without performance degradation. The architecture follows a clear separation of concerns between presentation, business logic, and data layers, enabling independent development and scaling of frontend and backend components.
Performance optimization techniques such as database connection pooling, efficient state management using Redux, and well-structured API endpoints ensure the system can support a growing user base and increasing transaction volumes while maintaining responsiveness and stability.

Secure Payment Gateway: To ensure secure financial transactions, the Book Store integrates a trusted payment gateway (**VNPay**) that safeguards users' personal and payment information. All payment-related data is transmitted using encrypted HTTPS connections and protected through token-based authentication mechanisms.

The system implements multiple security layers, including request rate limiting, strict input validation, and secure session management. These measures help prevent unauthorized access, fraud, and data breaches, reinforcing user trust in the platform's payment process.

Customized User Experience: The system provides personalized features designed to enhance user engagement and satisfaction. These include wishlists for saving favorite books and personalized recommendations based on user browsing behavior and purchase history.

By analyzing user preferences, the platform delivers relevant content such as recommended books, trending items, and category-based suggestions. This customization improves user retention and encourages repeat purchases by delivering a more tailored shopping experience.

Review and Rating System: A comprehensive review and rating system is implemented to promote transparency and informed decision-making. Customers can rate books on a scale from 1 to 5 stars and submit detailed written reviews reflecting their experiences.

Average ratings are displayed prominently on product pages, and users can filter books based on rating levels. This user-generated content builds trust among customers, provides valuable feedback for other buyers, and enhances the overall credibility of the platform.

CHAPTER 7: DEVELOPMENT

7. Description of Project Requirements

7.1. Programming Languages

7.1.1. JavaScript

JavaScript (JS) is a widely used client-side scripting language in web development that enables interaction between the client and the server, allowing the creation of dynamic and interactive web applications. With its multi-paradigm nature, single-threaded execution model, prototype-based structure, and object-oriented capabilities, JavaScript is recognized as one of the fundamental technologies supporting modern World Wide Web applications.

The Book-Store project was developed using JavaScript for several key reasons. Firstly, JavaScript offers a rich ecosystem of libraries and frameworks that significantly accelerate the development process. These frameworks allow developers to reuse prebuilt components rather than building all functionalities from scratch, reducing development time and improving productivity when constructing an online book-selling platform. Secondly, JavaScript helps optimize server performance by handling certain logical operations on the client side. In scenarios where a large number of users simultaneously browse books, place orders, or submit reviews, client-side processing reduces server workload and mitigates potential performance bottlenecks.

Another important advantage of JavaScript is its cross-platform compatibility. JavaScript applications can run smoothly on most web browsers and operating systems, enabling users to access the Book-Store platform from various devices and environments. This enhances global accessibility and ensures a consistent user experience across platforms. Finally, JavaScript provides high execution efficiency due to continuous optimization and the use of Just-In-Time (JIT) compilation. These performance enhancements allow the Book-Store web application to operate quickly, responsively, and reliably, even under increased user demand.

7.1.1.2 MySQL

MySQL, which stands for Structured Query Language, is a widely used Relational Database Management System (RDBMS) based on the client–server architecture. It organizes data into structured databases consisting of interconnected tables, where information is stored in rows and columns. This relational structure enables efficient data storage, retrieval, and management across different system components.

MySQL was selected as the primary database solution for the Book-Store project due to its proven stability and reliability. Having been continuously developed and refined over many years, MySQL has demonstrated strong performance in real-world applications and enterprise-level systems. Its ability to process queries efficiently makes it well-suited for handling large volumes of data, such as book catalogs, user accounts, orders, and customer reviews. Built-in optimization mechanisms allow MySQL to deliver fast and accurate query results, contributing to a smooth user experience.

Scalability is another key advantage of MySQL. Features such as data replication and partitioning enable the database to expand seamlessly as the number of users and transactions increases. This ensures that the Book-Store platform can grow without major architectural changes. In addition, MySQL provides robust security capabilities, including encryption, access control mechanisms, and secure communication protocols. These features help protect sensitive data and ensure that only authorized users can access or modify information, making MySQL a secure and dependable choice for the Book-Store system.

7.2. Frameworks & Library

7.2.1. ReactJS

ReactJS is a JavaScript library developed by Facebook that is widely used for building interactive and component-based user interfaces for web applications. It supports efficient data rendering primarily on the client side, with the ability to integrate server-side rendering when needed. The main goal of ReactJS is to create web applications that are responsive, scalable, and easy to maintain while delivering a smooth user experience.

ReactJS was chosen as the core front-end technology for the Book-Store project due to its performance efficiency and superior user experience. By utilizing the Virtual DOM, ReactJS minimizes direct manipulation of the browser's DOM, allowing user interface updates to occur faster and more smoothly. This approach is particularly beneficial for dynamic features such as browsing book lists, filtering results, managing shopping carts, and updating user reviews in real time. In addition, React's component-based architecture enables the application to be divided into independent, reusable components, which improves code organization and overall system performance.

Another key advantage of ReactJS is its strong support for component reusability. Reusable components reduce development time and effort by allowing consistent UI elements—such as book cards, navigation bars, and review sections—to be used across multiple pages of the Book-Store application. Each component operates independently, making it easier to maintain, update, or replace without affecting other parts of the system. Furthermore, ReactJS benefits from a large and active developer community, offering extensive documentation, libraries, and third-party tools. This rich ecosystem simplifies development, troubleshooting, and future enhancement of the Book-Store platform.

7.2.2. NodeJS

Node.js is a server-side runtime environment built on Google's V8 JavaScript engine and implemented using C++ and JavaScript. It operates based on a single-threaded, non-blocking input/output model, allowing the system to handle client requests efficiently while maintaining low memory consumption. This architecture enables Node.js to support a large number of simultaneous connections and deliver fast responses, making it well suited for modern web applications.

Node.js was selected as the backend platform for the Book-Store project primarily due to its high performance and efficiency. Its asynchronous and non-blocking I/O mechanism allows the system to process multiple concurrent requests—such as browsing books, managing carts, and handling payment transactions—without performance degradation. By utilizing an event-driven architecture and an event loop on a single thread, Node.js reduces latency and improves overall scalability, ensuring a responsive backend under increasing user demand.

Scalability is another key advantage of Node.js. Its ability to efficiently manage hundreds of concurrent connections makes it ideal for building scalable e-commerce platforms. Additionally, Node.js integrates well with microservices-based architectures, enabling individual system components—such as user management, order processing, and review services—to be developed, maintained, and scaled independently. Finally, Node.js supports rapid development through its extensive npm ecosystem, which provides access to a vast collection of reusable libraries and modules. Using JavaScript consistently across both frontend and backend further simplifies development, reduces system complexity, and accelerates the overall implementation of the Book-Store application.

7.3. Tools Used

Github: Version Control, Data Import

Discord: Room voice, Follow The Project

Visual Studio Code: Website Building, Update Code, Import packages

CHAPTER 8: TESTING AND QUALITY ASSURANCE

8.1. Testing Methodologies Employed

8.1.1. Unit Testing

Unit testing is a software testing technique that focuses on validating individual components of a system in isolation, such as functions, services, controllers, and modules. In the Book Store project, unit testing is applied to ensure that each functional unit behaves correctly and consistently before being integrated into the full system.

Objective of Unit Test:

+ Verify the correctness of individual functions and modules within the Book Store system
+ Detect and resolve errors at an early stage of development
+ Ensure that each process operates according to defined requirements
+ Reduce development costs by preventing defects from propagating to later phases
+ Support easier code maintenance and refactoring
+ Improve developer understanding of the codebase and encourage code reuse

In the Book Store platform, unit testing is performed on both backend and frontend components. On the backend, critical modules such as user authentication, book management, cart handling, order creation, address management, and rating submission are tested independently. For example, unit tests verify that login functions correctly validate credentials, cart services accurately update quantities, and order controllers correctly calculate totals and update inventory.

On the frontend, unit tests focus on validating individual components such as book cards, filter controls, forms, and buttons to ensure they render correctly and respond as expected to user interactions. This helps maintain interface reliability as new features are added or existing components are modified.

Both manual and automated unit testing approaches are used. Automated tests allow developers to quickly verify expected behavior after code changes, while manual testing helps validate edge cases during development. By isolating each module, developers can confidently modify or enhance specific features—such as checkout logic or review handling—without impacting the overall system.

8.1.2. Integrating Testing

Integration testing is a testing approach that focuses on verifying the interaction and data flow between multiple software components or modules after they have been individually tested through unit testing. In the Book Store project, integration testing is conducted to ensure that interconnected modules operate correctly when combined and that data is exchanged accurately across the system.

According to the Software Development Life Cycle (SDLC) and the V-Model, integration testing is performed after unit testing and before system testing. At this stage, individual components that have already been validated—such as authentication, book management, shopping cart, checkout, and order processing—are integrated and tested together to identify potential communication or data consistency issues.

In the Book Store platform, integration testing plays a critical role in validating workflows that span multiple modules. For example, tests are performed to confirm that user authentication integrates correctly with profile management, that selected books in the cart are accurately passed to the checkout process, and that completed orders are correctly stored in the database and reflected in the order history. Similarly, integration between the rating system and product detail pages is tested to ensure that submitted reviews are properly saved and displayed.

Integration testing also enables developers to examine database interactions and data access logic. This ensures that information retrieved from the database—such as book details, user addresses, order summaries, and ratings—is accurately presented on the user interface. On the administrative side, integration tests verify that actions performed in the admin dashboard, such as adding books or updating order statuses, are correctly reflected across the system.

By validating how system components work together as a unified whole, integration testing helps identify issues early in the development process and reduces the risk of defects appearing in later stages. As a result, it contributes to improved system stability, reduced development risks, and a more reliable and consistent user experience in the Book Store application.

8.1.2. User Acceptance Testing (UAT)

User Acceptance Testing (UAT) represents the final testing phase of the Book Store project before the system is officially deployed and made available to users. This testing stage focuses on validating whether the platform meets user expectations, functional requirements, and real-world usage scenarios. User feedback plays a central role in the UAT process, as it reflects actual customer experiences when interacting with the system.

During UAT, end users interact with the Book Store platform by performing typical activities such as browsing books, applying filters, viewing product details, adding items to the cart, completing checkout, tracking orders, managing profiles, and submitting ratings or reviews. While using the system, users identify potential functional issues, usability problems, or inconsistencies in the user interface. Their feedback is then collected and analyzed by the development team to refine and improve the system.

UAT helps ensure both system quality and user acceptability. By validating that the implemented features align with user needs and expectations, the Book Store platform delivers a more satisfying and user-friendly shopping experience. This testing phase also confirms that all project requirements have been correctly implemented, reducing the risk of functional gaps or misunderstandings after deployment.

An important benefit of UAT is the early identification of defects and usability issues before the product is released to the market. Detecting and resolving problems at this stage saves time and reduces costs associated with post-release bug fixes. Additionally, UAT helps improve the overall user experience by highlighting real interaction challenges that may not be detected during unit or integration testing.

Finally, UAT significantly reduces deployment risks. By ensuring that the Book Store system is stable, complete, and aligned with industry standards and best practices, the likelihood of releasing an unfinished or unreliable product is minimized. As a result, UAT contributes to a smooth deployment process and increases confidence in the platform’s readiness for real-world use.

8.2. Test Results and Bug Tracking

Extensive Test Suite: The Book Store project is supported by a comprehensive test suite that covers a wide range of functional and non-functional scenarios. Test cases are designed to validate core user flows such as browsing and filtering books, viewing product details, managing the shopping cart, completing checkout, tracking orders, and submitting ratings and reviews. In addition, edge cases and boundary conditions—such as empty carts, invalid inputs, and concurrent user actions—are thoroughly tested. This systematic approach ensures that the platform’s functionality, security, performance, and scalability are evaluated in depth. As new features are added or existing components are refined, the test suite is continuously updated to reflect evolving system requirements.

Bug Tracking and Management: An effective bug tracking process is essential to maintaining the quality of the Book Store system. Each identified defect is logged with a unique identifier, detailed description, severity level, and reproduction steps. This structured documentation facilitates clear communication between the development and testing teams. Regular bug review and triage sessions are conducted to prioritize critical issues, assign responsibilities, and monitor resolution progress. This approach ensures that high-impact bugs—such as checkout failures or data inconsistencies—are addressed promptly, contributing to overall system stability.

Regression Testing: Regression testing is performed after bug fixes or feature updates to confirm that recent changes do not negatively affect existing functionality. For the Book Store project, regression tests verify that updates to one module—such as payment processing or order management—do not disrupt other components like cart handling or user authentication. Automated regression testing is used where possible to provide faster feedback and improve testing efficiency. This iterative testing process helps stabilize the software development lifecycle and ensures consistent system behavior as the platform evolves.

8.3. Quality Assurance Measures Taken to Ensure Software Reliability

Code Reviews and Audits: Maintaining high code quality is a key priority in the Book Store project. Regular peer code reviews are conducted to ensure that the source code follows established coding standards, best practices, and architectural guidelines. These reviews help identify potential logic errors, performance issues, and security vulnerabilities at an early stage. In addition to improving code quality, this process promotes knowledge sharing among team members and supports a collaborative development environment.

Automated Testing: Automation plays an important role in ensuring consistent and reliable testing outcomes. Automated tests are applied to repetitive and time-consuming scenarios such as authentication validation, cart operations, checkout flows, and order processing. This approach increases test coverage, accelerates the testing cycle, and enables faster detection of defects throughout the development lifecycle. Automated testing also ensures that critical functionalities remain stable after system updates.

Security Assurance and Audits: Security is a fundamental concern for the Book Store platform due to the handling of user accounts, personal data, and payment information. Regular security assessments are performed to identify potential vulnerabilities related to authentication, authorization, data storage, and API access. Best practices such as encrypted credentials, secure token handling, and role-based access control are enforced. These proactive measures help protect user data, prevent unauthorized access, and maintain customer trust.

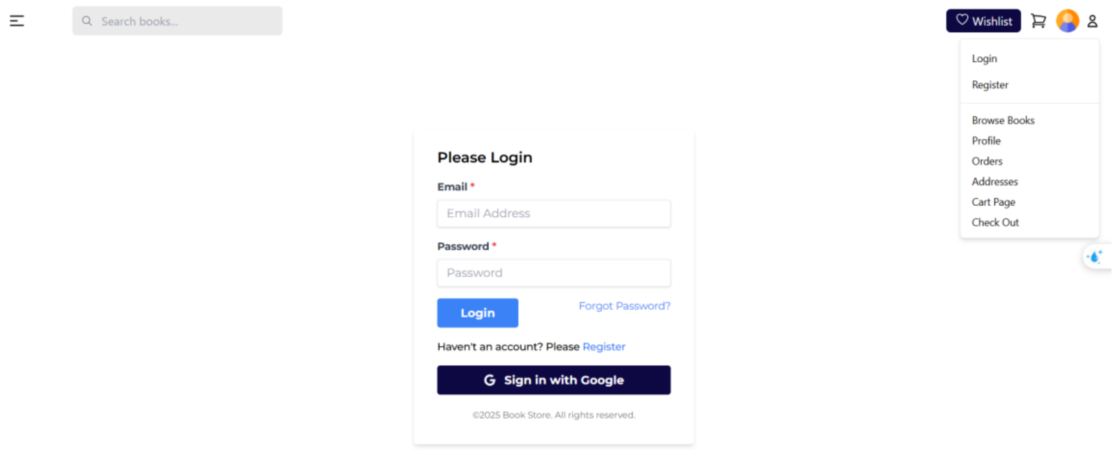
Performance and Load Testing: Performance testing is conducted to evaluate the system’s responsiveness, stability, and scalability under various load conditions. The Book Store platform is tested to ensure that key features—such as browsing books, filtering results, adding items to the cart, and completing checkout—perform reliably during peak usage periods. Results from performance testing are used to guide optimization efforts, including database query improvements and backend response tuning.

In summary, the quality assurance measures applied in the Book Store project—including code reviews, automated testing, security audits, and performance evaluation—are essential to delivering a stable, secure, and reliable e-commerce platform. These practices not only ensure that the system meets functional and non-functional requirements but also support long-term maintainability and continuous improvement of the software.

CHAPTER 9: USER INTERFACE & API

9.1. User Interface

9.1.1. "Login" page



Please Login

Email *

Email Address

Password *

Password

Haven't an account? Please Register

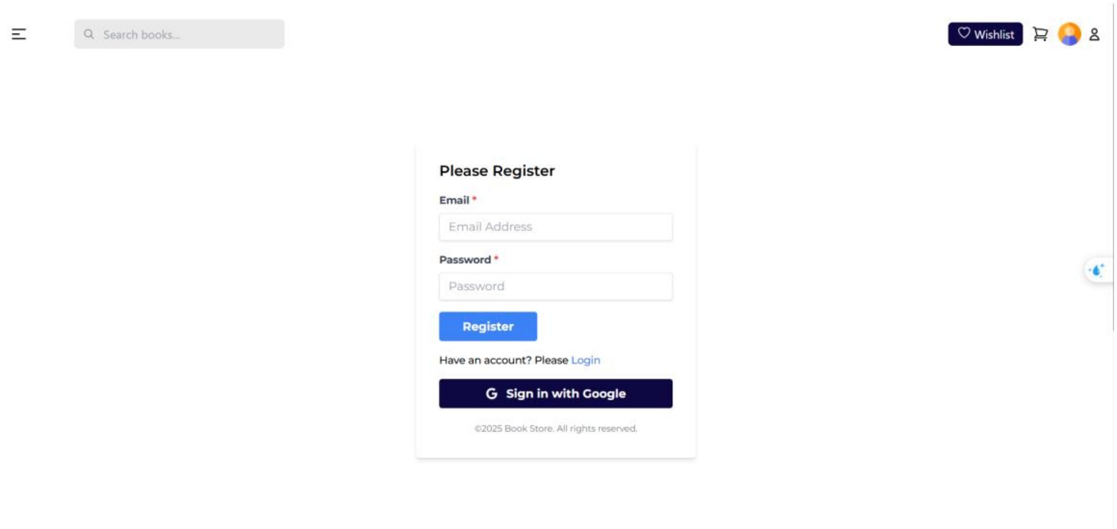
©2025 Book Store. All rights reserved.

Figure 22: Illustrator of "Login" page

The Book Store login page is designed to be basic yet comprehensive, allowing users to obtain necessary information. For increased convenience, social networking sign-up options via Google are provided below. Additionally, users must complete the form in the middle of the page, including their email and password. Click "Login" to complete the login process on the Book Store website.

On top of that, the top right corner includes the Icons section, the checkout cart, and the Wishlist. Users who are not logged in can access the Browse Books section in the icons to view books on the website. Other sections such as Orders, Addresses, Cart Page, and Checkout require a user account.

9.1.2. “Register” page



Search books...

Please Register

Email *

Email Address

Password *

Password

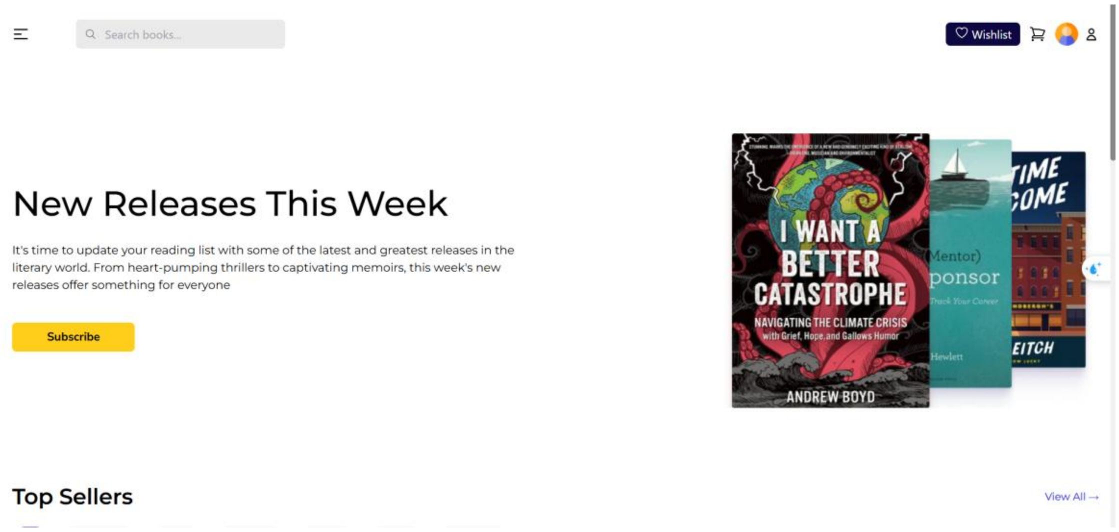
Have an account? Please Login

©2025 Book Store. All rights reserved.

Figure 23: Illustrator of “Register” page

The registration interface of the Book-Store website follows a clean, minimal, and intuitive design that makes account creation simple and user-friendly. At the center of the page, a clearly structured registration form allows new users to enter their email address and password, with mandatory fields clearly marked. A prominent “Register” button guides users through the account creation process, while a visible login link is provided for users who already have an existing account. The option to “Sign in with Google” offers a fast and convenient alternative registration method, improving accessibility and user convenience. The top navigation bar includes a search field, wishlist, shopping cart, and user menu, ensuring consistent access to core features across the platform.

9.1.3. “Home” page



Search books...

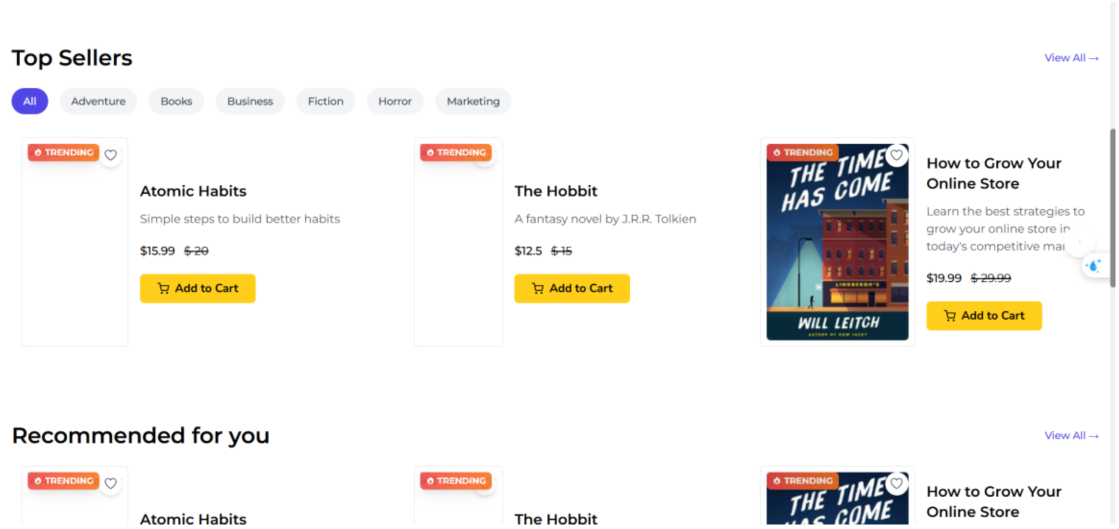
New Releases This Week

It's time to update your reading list with some of the latest and greatest releases in the literary world. From heart-pumping thrillers to captivating memoirs, this week's new releases offer something for everyone

Subscribe
I WANT A BETTER CATASTROPHE
NAVIGATING THE CLIMATE CRISIS with Grief, Hope, and Gallows Humor
ANDREW BOYD
Mentor) ponsor
Hewlett
TIME COME
EITCH

Top Sellers

Figure 24: Illustrator of “Home” page 1



Top Sellers

Atomic Habits

Simple steps to build better habits

$15.99 ~~$20~~

The Hobbit

A fantasy novel by J.R.R. Tolkien

$12.5 ~~$15~~

How to Grow Your Online Store

Learn the best strategies to grow your online store in today's competitive ma

$19.99 ~~$29.99~~

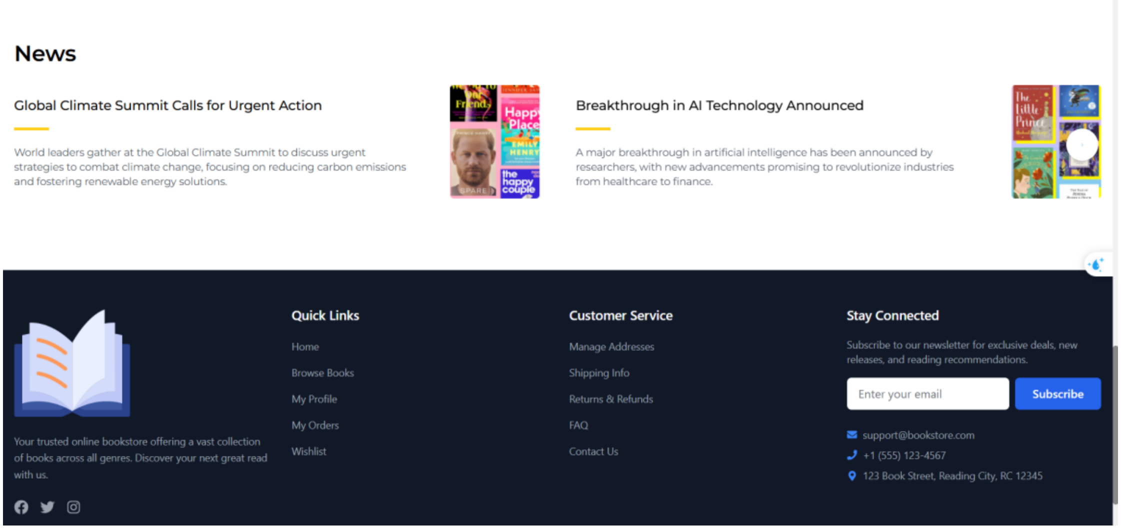
Recommended for you

Atomic Habits

The Hobbit

How to Grow Your Online Store

Figure 25: Illustrator of “Home” page 2



News

Global Climate Summit Calls for Urgent Action

World leaders gather at the Global Climate Summit to discuss urgent strategies to combat climate change, focusing on reducing carbon emissions and fostering renewable energy solutions.

Breakthrough in AI Technology Announced

A major breakthrough in artificial intelligence has been announced by researchers, with new advancements promising to revolutionize industries from healthcare to finance.

Your trusted online bookstore offering a vast collection of books across all genres. Discover your next great read with us.

Quick Links

- Home
- Browse Books
- My Profile
- My Orders
- Wishlist

Customer Service

- Manage Addresses
- Shipping Info
- Returns & Refunds
- FAQ
- Contact Us

Stay Connected

Subscribe to our newsletter for exclusive deals, new releases, and reading recommendations.

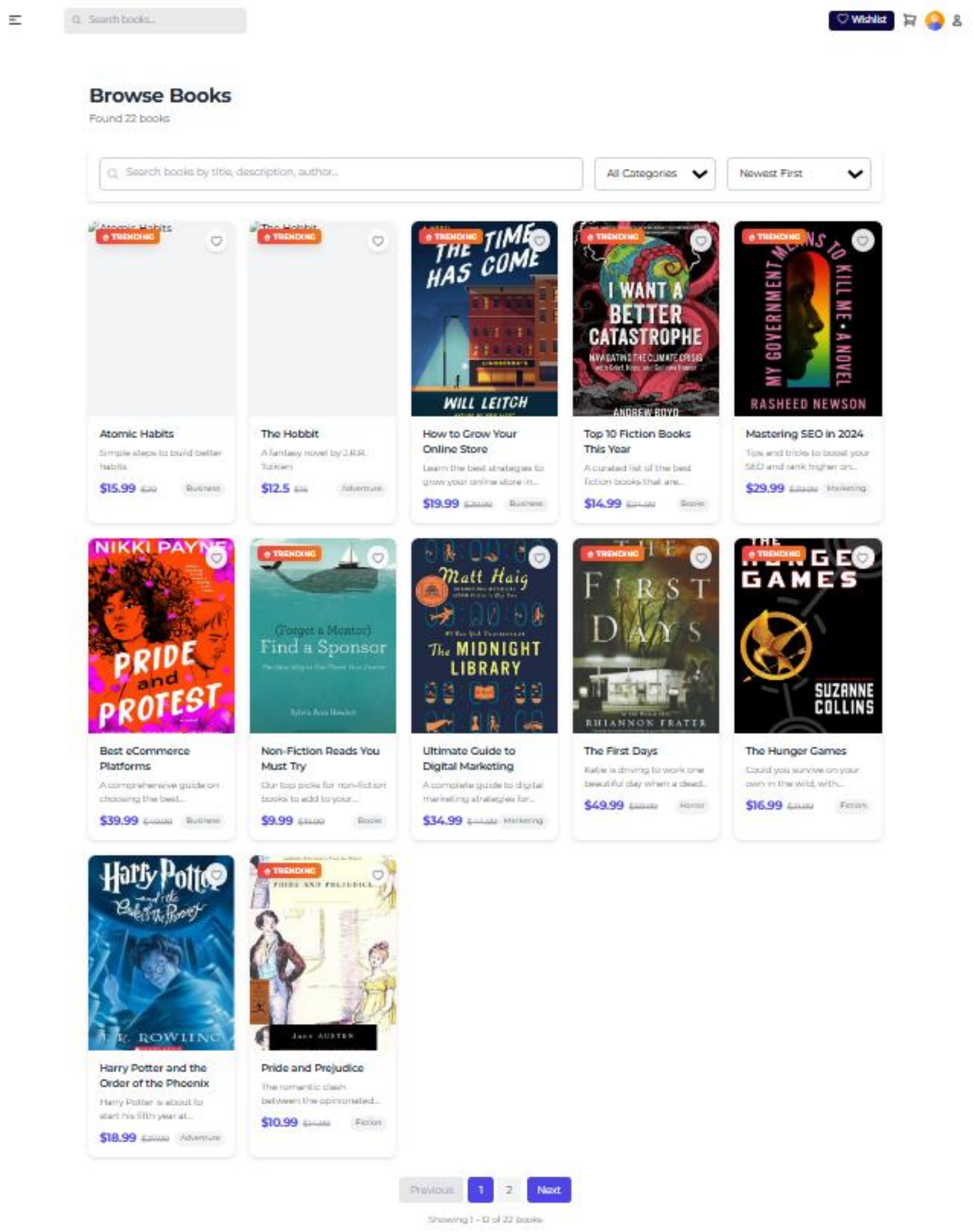
Enter your email | Subscribe

support@bookstore.com
+1 (555) 123-4567
123 Book Street, Reading City, RC 12345

Figure 26: Illustrator of "Home" page 3

The home page of the Book-Store website presents a modern, well-structured, and visually appealing layout designed to enhance user engagement and browsing efficiency. At the top, a clean navigation bar provides quick access to key features such as book search, wishlist, shopping cart, and user account options. The hero section highlights "New Releases This Week", combining promotional text with featured book covers to attract readers' attention and encourage exploration. Below this, the Top Sellers section showcases popular books with category filters, clear pricing, discount indicators, and "Add to Cart" buttons, enabling users to browse and purchase efficiently. A Recommended for You section personalizes the experience by suggesting relevant titles based on user interests. The News area adds value by presenting informative and engaging content related to literature and global topics. Finally, the footer consolidates essential information, including quick links, customer service options, newsletter subscription, and contact details. Overall, the homepage delivers a smooth, intuitive, and professional user experience suitable for an e-commerce book platform.

9.1.4. “Browse Books” page



Search books...

Wishlist

Browse Books

Found 22 books

Search books by title, description, author...

All Categories

Newest First

Title	Description	Price	Old price	Category
Atomic Habits	Simple steps to build better habits	$15.99	~~$20~~	Business
The Hobbit	A fantasy novel by J.R.R. Tolkien	$12.5	~~$16~~	Adventure
How to Grow Your Online Store	Learn the best strategies to grow your online store in...	$19.99	[illegible]	Business
Top 10 Fiction Books This Year	A curated list of the best fiction books that are...	$14.99	[illegible]	Books
Mastering SEO in 2024	Tips and tricks to boost your SEO and rank higher on...	$29.99	[illegible]	Marketing
Best eCommerce Platforms	A comprehensive guide on choosing the best...	$39.99	[illegible]	Business
Non-Fiction Reads You Must Try	Our top picks for non-fiction books to add to your...	$9.99	[illegible]	Books
Ultimate Guide to Digital Marketing	A complete guide to digital marketing strategies for...	$34.99	[illegible]	Marketing
The First Days	Katie is driving to work one beautiful day when a dead...	$49.99	[illegible]	Horror
The Hunger Games	Could you survive on your own in the wild, with...	$16.99	[illegible]	Fiction
Harry Potter and the Order of the Phoenix	Harry Potter is about to start his fifth year at...	$18.99	[illegible]	Adventure
Pride and Prejudice	The romantic clash between the opinionated...	$10.99	[illegible]	Fiction

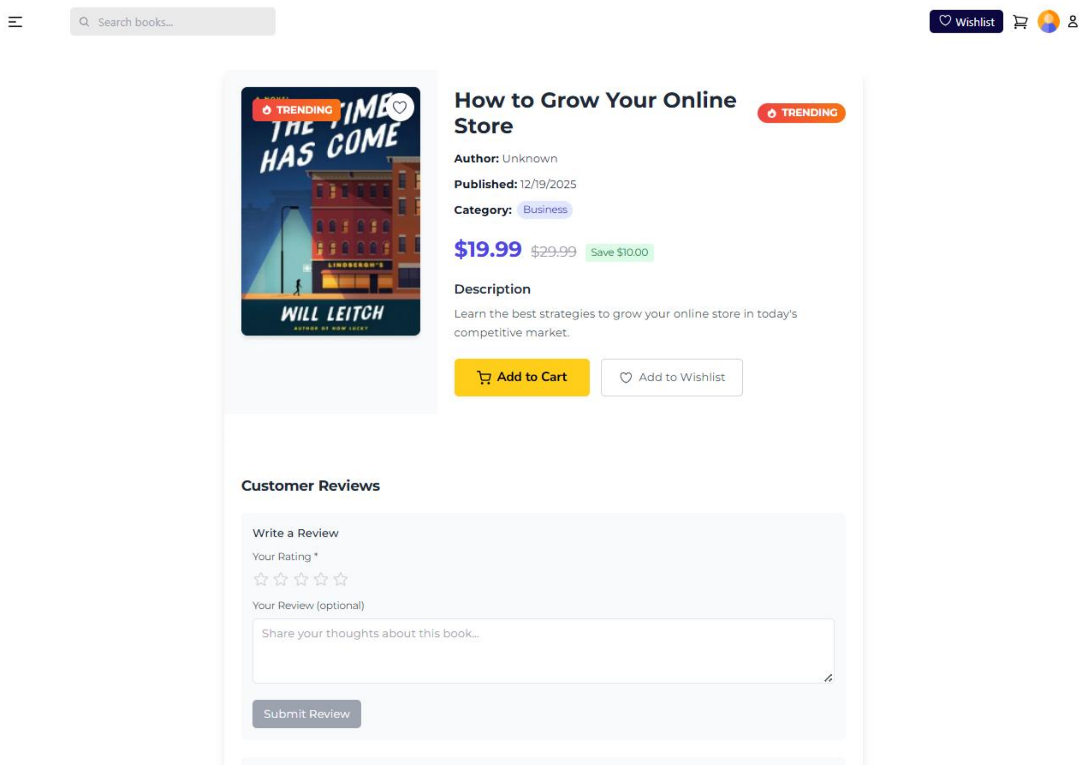
Previous 1 2 Next

Showing 1 - 12 of 22 books

Figure 27: Illustrator of “Browse Books” page

The Browse Books page provides a clean, well-organized, and user-friendly interface that allows users to explore the book catalog efficiently. At the top of the page, a search bar enables users to search for books by title, description, or author, while dropdown menus for category selection and sorting options (such as “Newest First”) help refine search results. Below this, books are displayed in a grid-based layout with clear cover images, titles, brief descriptions, prices, and category labels. Trending tags and wishlist icons enhance interactivity and highlight popular items. Each book card includes clear pricing information and visual consistency, making comparison easy for users. Pagination controls at the bottom allow smooth navigation through multiple pages of results.

9.1.5. "Product details" page



How to Grow Your Online Store

Author: Unknown

Published: 12/19/2025

Category: Business

$19.99 ~~$29.99~~ Save $10.00

Description

Learn the best strategies to grow your online store in today's competitive market.

Customer Reviews

Write a Review

Your Rating *

☆☆☆☆☆

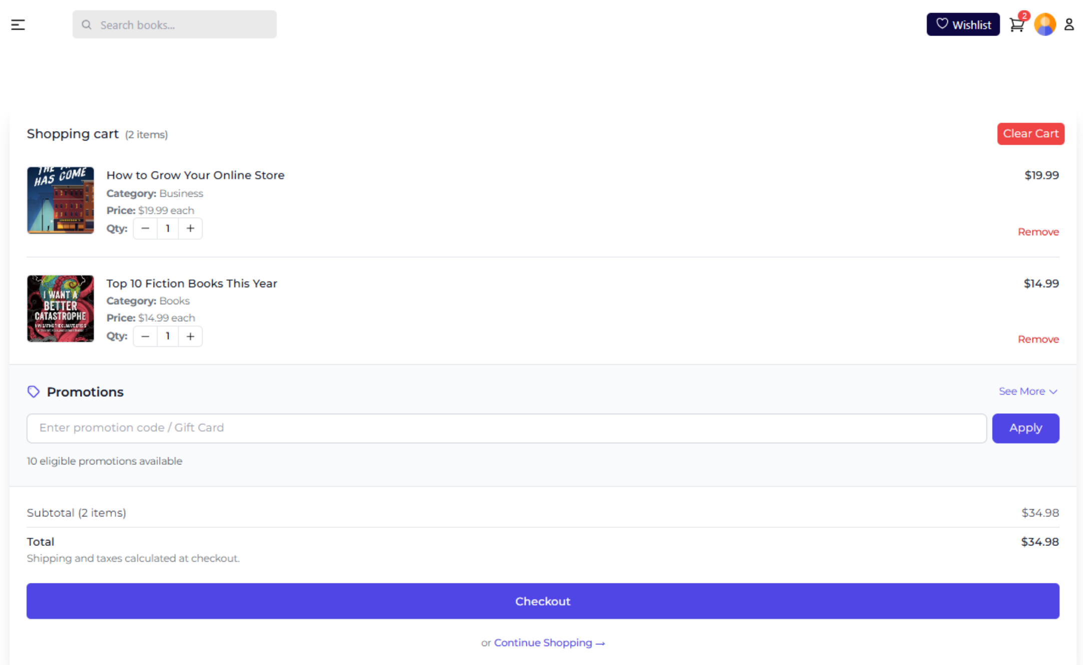
Your Review (optional)

Share your thoughts about this book...

Figure 28: Illustrator of "Product Details" page

The Product Detail page of the Book-Store website is designed to provide comprehensive and clearly structured information about a selected book, supporting informed purchasing decisions. The page prominently displays the book cover alongside essential details such as the title, author, publication date, category, and a "Trending" label to highlight popularity. Pricing information is clearly presented, including the original price, discounted price, and savings, enhancing transparency and promotional appeal. A concise description section summarizes the book's content and value to readers. Action buttons such as "Add to Cart" and "Add to Wishlist" are positioned for easy access, encouraging user interaction. Below the product information, a Customer Reviews section allows users to submit ratings and written feedback, fostering engagement and building trust through user-generated content. Overall, the Product Detail page offers a clean, intuitive, and informative layout that enhances user experience and supports effective e-commerce functionality.

9.1.6. “Cart” page



Shopping cart (2 items)

How to Grow Your Online Store
Category: Business
Price: $19.99 each
Qty: − 1 +
$19.99
Remove

Top 10 Fiction Books This Year
Category: Books
Price: $14.99 each
Qty: − 1 +
$14.99
Remove

Promotions

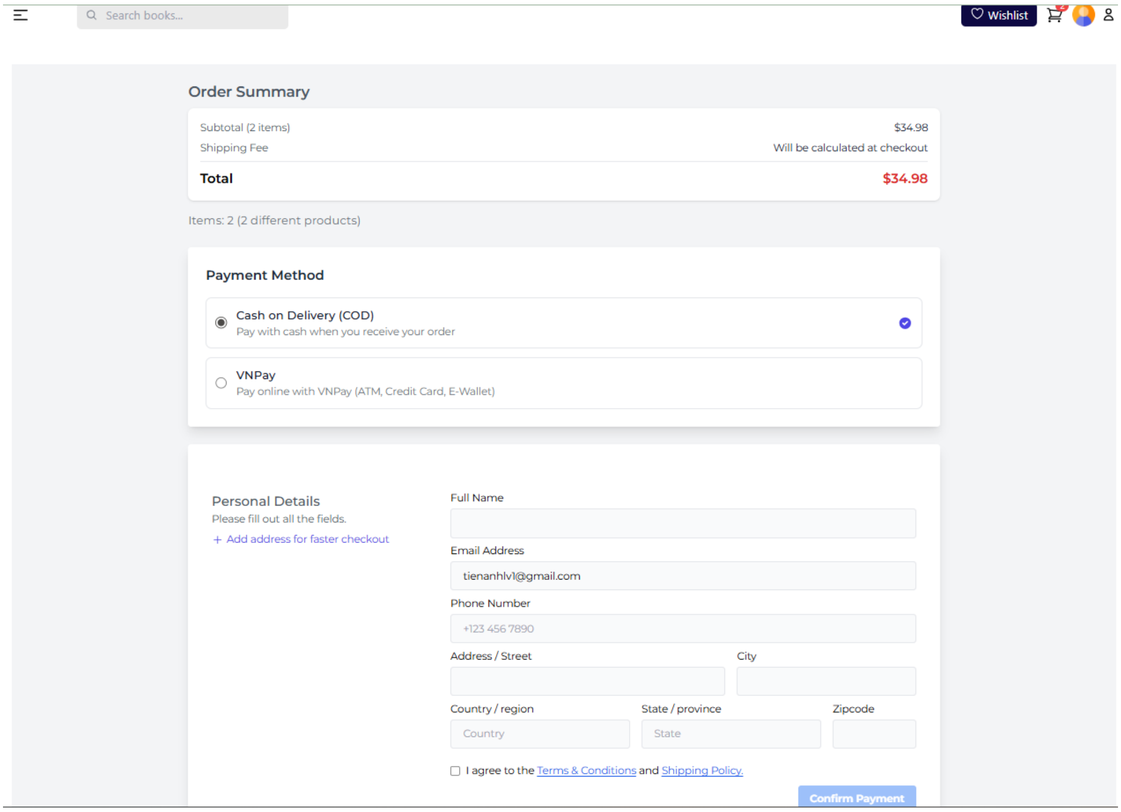
Enter promotion code / Gift Card

10 eligible promotions available

Subtotal (2 items)	$34.98
Total	$34.98

Shipping and taxes calculated at checkout.

Figure 29: Illustrator of “Cart” page



Search books...

Wishlist

Order Summary

Subtotal (2 items)	$34.98
Shipping Fee	Will be calculated at checkout
Total	$34.98

Items: 2 (2 different products)

Payment Method

● Cash on Delivery (COD)
Pay with cash when you receive your order

○ VNPay
Pay online with VNPay (ATM, Credit Card, E-Wallet)

Personal Details
Please fill out all the fields.
+ Add address for faster checkout

Full Name

Email Address: tienanhlv1@gmail.com

Phone Number: +123 456 7890

Address / Street

City

Country / region: Country

State / province: State

Zipcode

□ I agree to the Terms & Conditions and Shipping Policy.

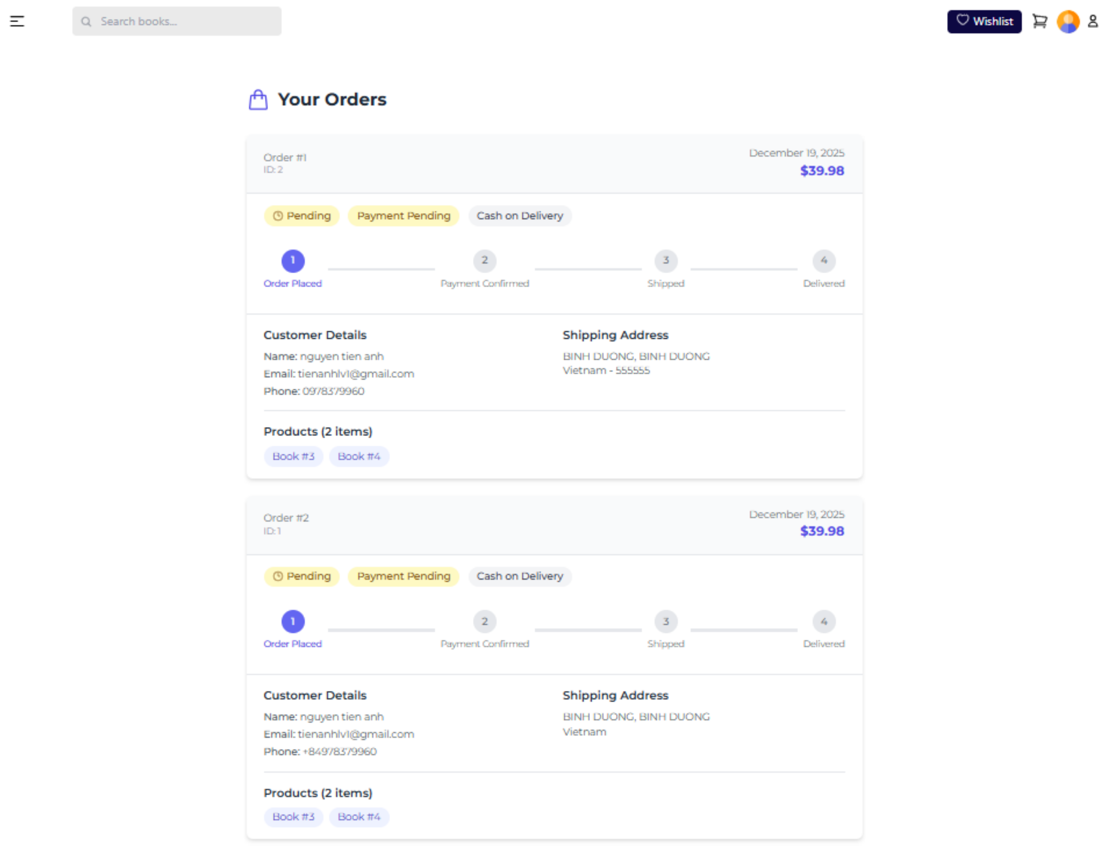
Confirm Payment

Figure 30: Illustrator of “Checkout” page

The Book-Store Cart Page provides users with a comprehensive overview of all books added to their shopping cart. Each item displays essential information including the book cover image, title, category, unit price, selected quantity, and total price. The cart summary section clearly presents the subtotal and overall total amount, allowing users to review their purchase before proceeding. Quantity adjustment buttons enable users to increase or decrease the number of items directly within the cart, while a remove option allows unwanted books to be deleted easily. Once users have reviewed and updated their cart, they can proceed by clicking the Checkout button to continue the payment process.

On the Checkout Page, users are presented with an order summary that includes the total number of items and final payment amount. Multiple payment methods are supported, such as Cash on Delivery and online payment options, ensuring flexibility and convenience. Users are required to enter personal and shipping information, including contact details and delivery address, before confirming the order. This structured and user-friendly checkout flow ensures a smooth, secure, and efficient purchasing experience.

9.1.7. “My Orders” page



Your Orders

Order #1
ID: 2
December 19, 2025
$39.98

Pending | Payment Pending | Cash on Delivery

1 Order Placed — 2 Payment Confirmed — 3 Shipped — 4 Delivered

Customer Details
Name: nguyen tien anh
Email: tienanhlvl@gmail.com
Phone: 0978379960

Shipping Address
BINH DUONG, BINH DUONG
Vietnam - 555555

Products (2 items)
Book #3 | Book #4

Order #2
ID: 1
December 19, 2025
$39.98

Pending | Payment Pending | Cash on Delivery

1 Order Placed — 2 Payment Confirmed — 3 Shipped — 4 Delivered

Customer Details
Name: nguyen tien anh
Email: tienanhlvl@gmail.com
Phone: +84978379960

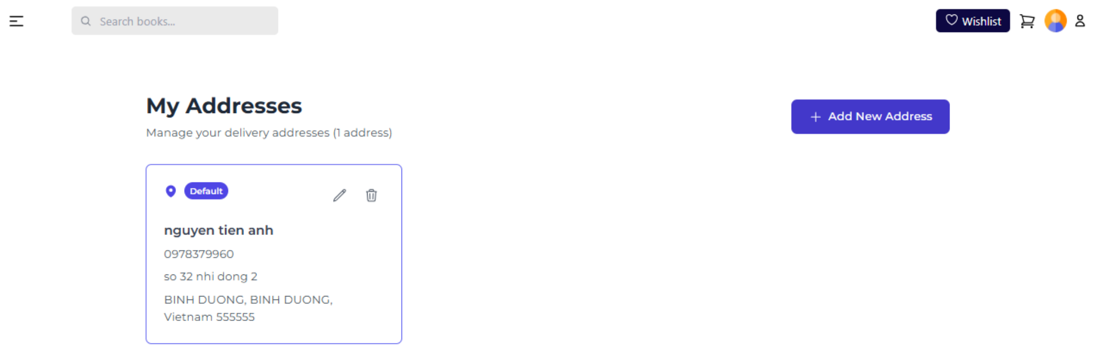
Shipping Address
BINH DUONG, BINH DUONG
Vietnam

Products (2 items)
Book #3 | Book #4

Figure 31: Illustrator of “Orders” page

The Orders page allows users to review and track all orders placed after completing the checkout process. Each order card displays essential information such as order ID, order date, total amount, payment method, and current order status. A visual progress tracker illustrates the order lifecycle, including stages such as Order Placed, Payment Confirmed, Shipped, and Delivered, helping users easily monitor order progress. In addition, the page provides detailed customer information, shipping address, and a list of purchased books for each order. Status labels such as Pending and Payment Pending offer clear insights into order processing.

9.1.8. “Addresses” page



My Addresses

Manage your delivery addresses (1 address)

nguyen tien anh

0978379960

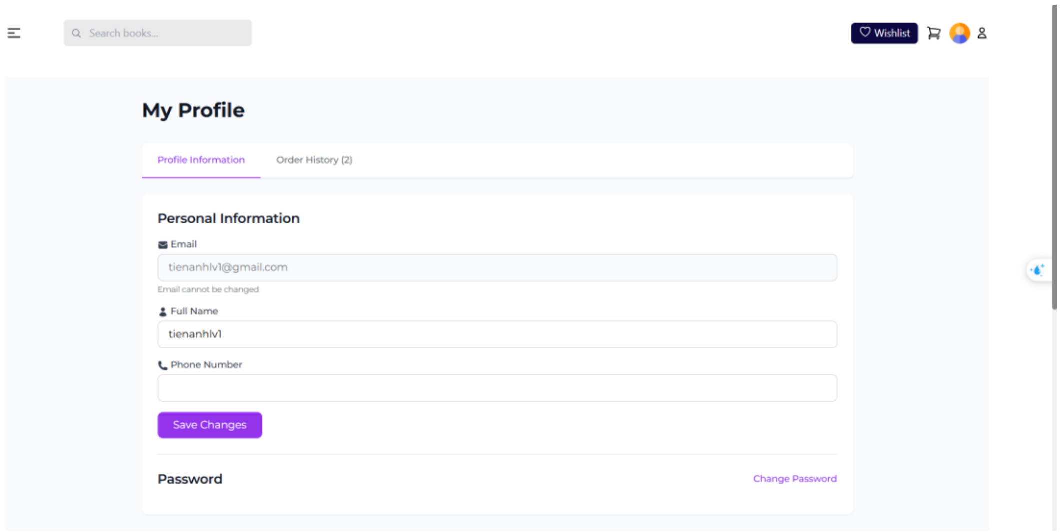
so 32 nhi dong 2

BINH DUONG, BINH DUONG, Vietnam 555555

Figure 32: Illustrator of “Addresses” page

The Address Page of the Book-Store website allows users to conveniently manage their delivery information in a clear and organized interface. The page displays a list of saved addresses, with the default address clearly highlighted for easy identification. Each address card presents essential details such as the recipient’s name, phone number, street address, city, and country. Users can edit or delete existing addresses using intuitive action icons, ensuring flexibility in managing delivery preferences. In addition, a prominent “Add New Address” button enables users to quickly add additional delivery locations.

9.1.9. “Profile” page



My Profile

Personal Information

Email: tienanhlv1@gmail.com

Email cannot be changed

Full Name: tienanhlv1

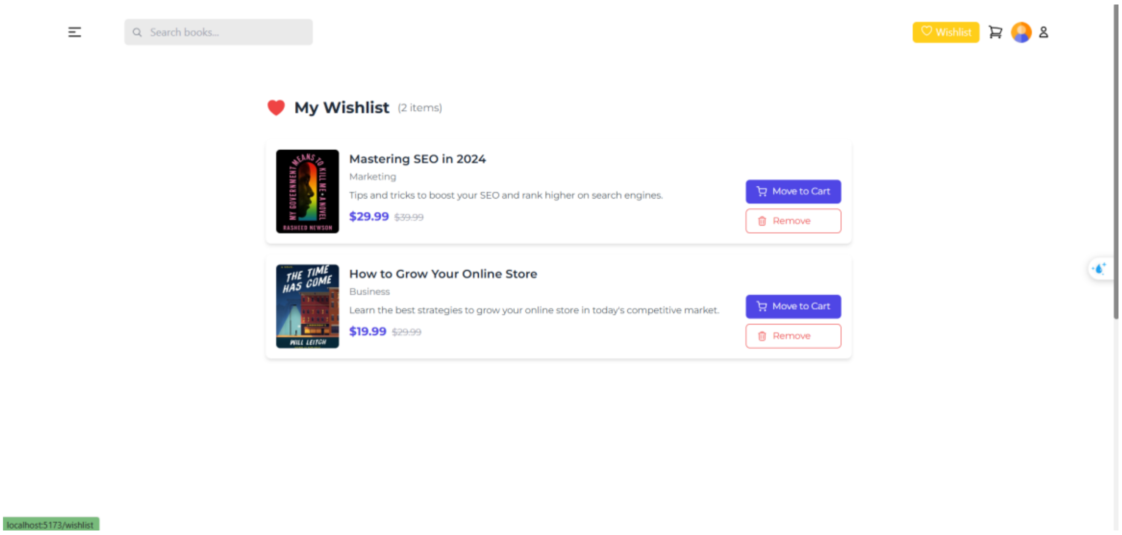
Phone Number:

Password

Figure 33: Illustrator of “Profile” page

The Profile Page of the Book-Store website allows users to view and manage their personal account information in a secure and user-friendly interface. The page is organized into tabs, enabling easy navigation between Profile Information and Order History. In the profile section, users can review and update their full name and phone number, while the email address is displayed as read-only to maintain account security. A clearly visible Save Changes button allows users to confirm updates efficiently. Additionally, the page provides a dedicated option to change the account password, enhancing user control over account security. Overall, the Profile Page supports effective account management and contributes to a personalized and trustworthy user experience.

9.1.10. "Wishlist" page



My Wishlist (2 items)

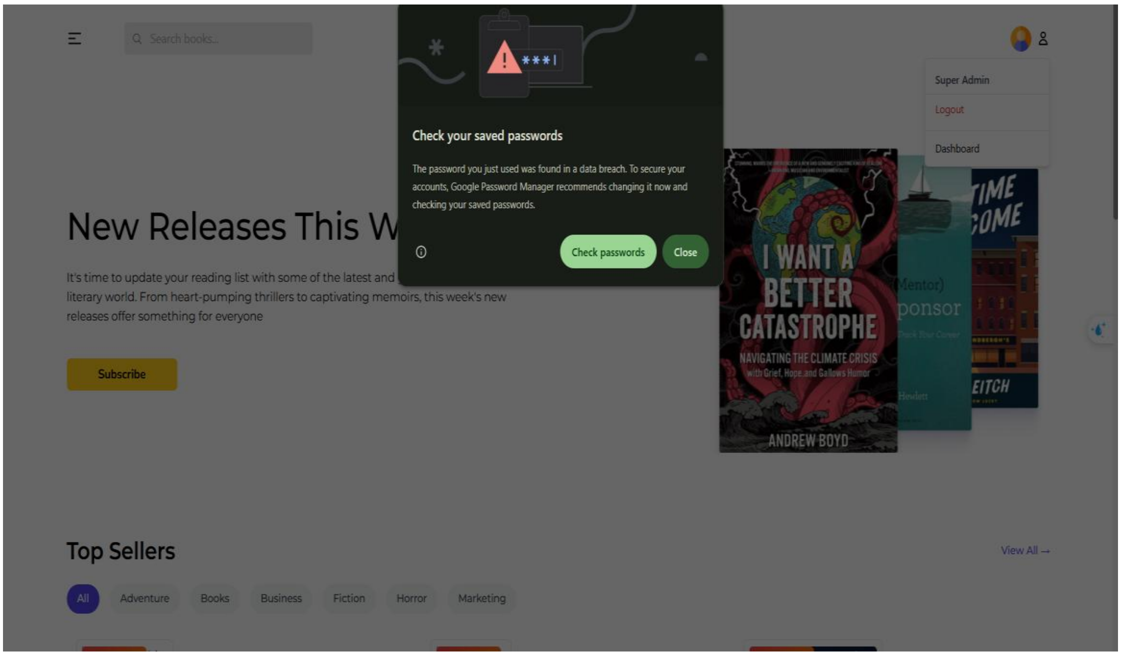
Mastering SEO in 2024
Marketing
Tips and tricks to boost your SEO and rank higher on search engines.
$29.99 ~~$39.99~~

How to Grow Your Online Store
Business
Learn the best strategies to grow your online store in today's competitive market.
$19.99 ~~$29.99~~

Figure 34: Illustrator of "Wishlist" page

The Wishlist Page of the Book-Store website allows users to save and manage books they are interested in purchasing at a later time. The page presents a clear list of saved items, displaying each book's cover image, title, category, brief description, and pricing information, including any discounted prices. Action buttons such as "Move to Cart" enable users to quickly add selected books to their shopping cart, while the "Remove" option allows items to be deleted from the wishlist easily. The total number of saved items is shown at the top of the page, helping users track their preferences. Overall, the Wishlist page enhances user engagement and supports a more personalized shopping experience by allowing users to organize and revisit their favorite books conveniently.

9.1.10. “Home” page for Admin



Check your saved passwords

The password you just used was found in a data breach. To secure your accounts, Google Password Manager recommends changing it now and checking your saved passwords.

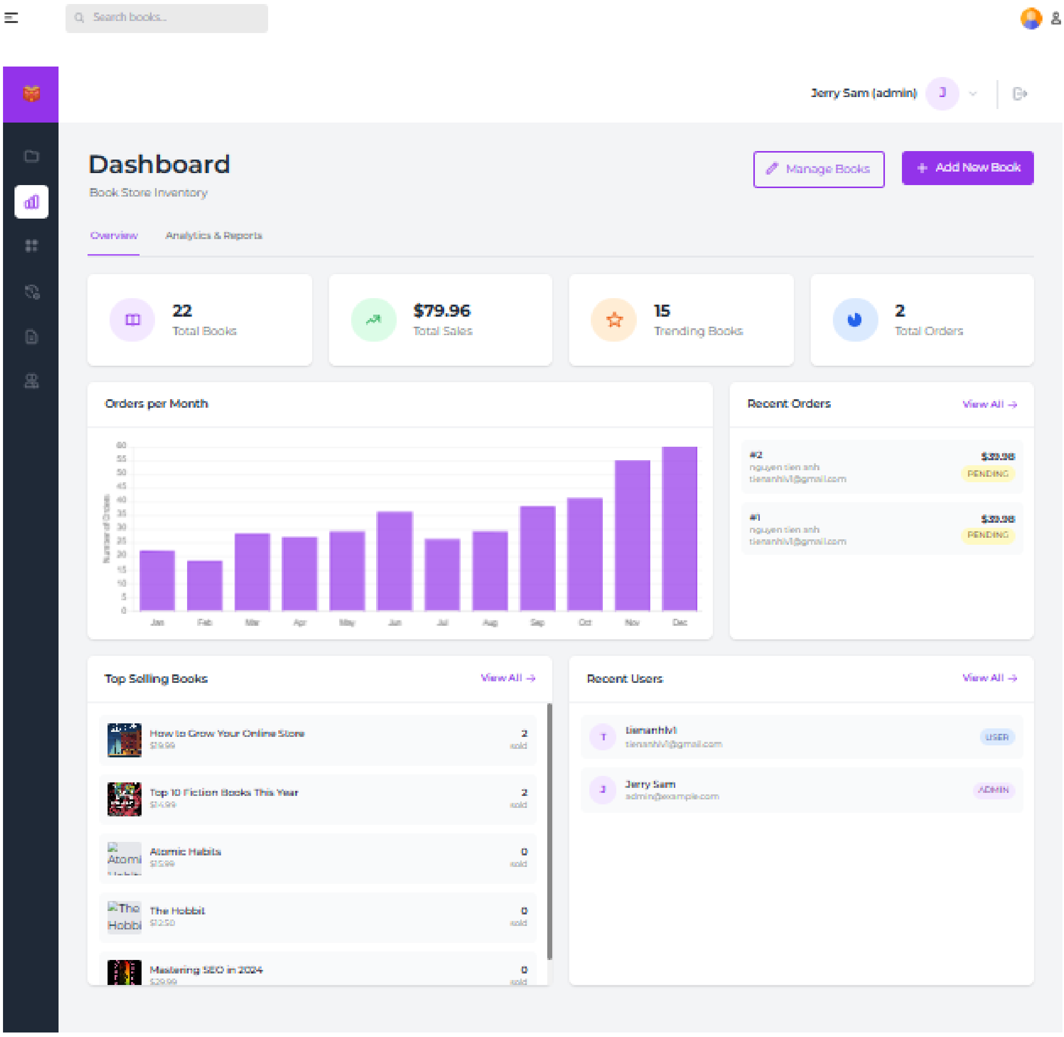
New Releases This W

It's time to update your reading list with some of the latest and literary world. From heart-pumping thrillers to captivating memoirs, this week's new releases offer something for everyone

Figure 35: Illustrator of “Home” page for Admin

The Admin Home Page serves as the main entry point for administrators after logging into the Book-Store system. It is built on the same homepage layout as the user-facing site, ensuring visual consistency, while providing additional administrative controls through the user menu. From the top-right account dropdown, administrators can access Dashboard and Logout options, enabling quick navigation to management functions. This page allows administrators to monitor the platform in a real environment, review displayed content such as featured books and promotions, and verify that the system is operating correctly from an end-user perspective.

9.1.11. “Dashboard” page for Admin



Dashboard

Book Store Inventory

22 Total Books

$79.96 Total Sales

15 Trending Books

2 Total Orders

Recent Orders

#2 nguyen tien anh tienanhlvl@gmail.com $39.98 PENDING

#1 nguyen tien anh tienanhlvl@gmail.com $39.98 PENDING

Top Selling Books

How to Grow Your Online Store $19.99	2 sold
Top 10 Fiction Books This Year $14.99	2 sold
Atomic Habits $15.99	0 sold
The Hobbit $12.50	0 sold
Mastering SEO in 2024 $29.99	0 sold

Recent Users

T	tienanhlvl tienanhlvl@gmail.com	USER
J	Jerry Sam admin@example.com	ADMIN

Figure 36: Illustrator of “Dashboard” page for Admin

The Admin Dashboard Page provides a comprehensive overview of the Book-Store’s operational performance through a clean and data-driven interface. At the top, summary cards display key metrics such as the total number of books, total sales revenue, trending books, and total orders, allowing administrators to quickly assess system status. Visual analytics, including an Orders per Month chart, help track sales trends and performance over time. The dashboard also features sections for Recent Orders, Top Selling Books, and Recent Users, enabling administrators to monitor customer activity, order status, and popular products at a glance. Action buttons such as Manage Books and Add New Book support efficient content and inventory management. Overall, the Admin Dashboard enhances decision-making and operational control by presenting critical information in a clear, organized, and accessible manner.

9.1.12. “Add new Book” page for Admin

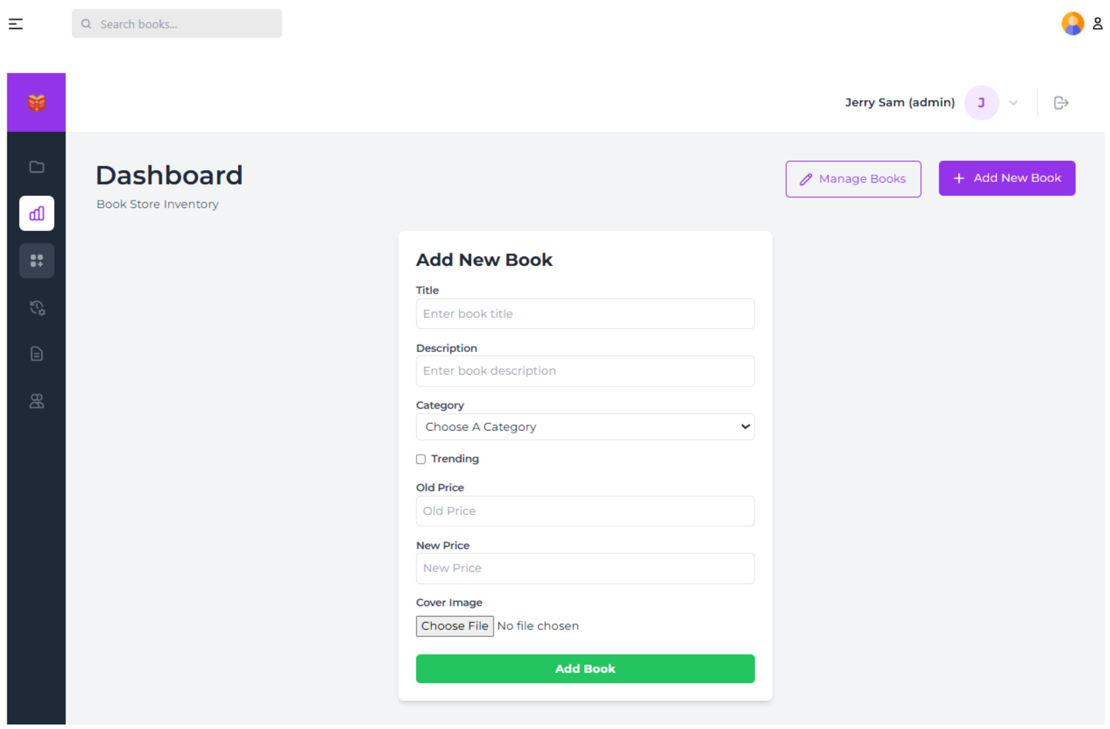
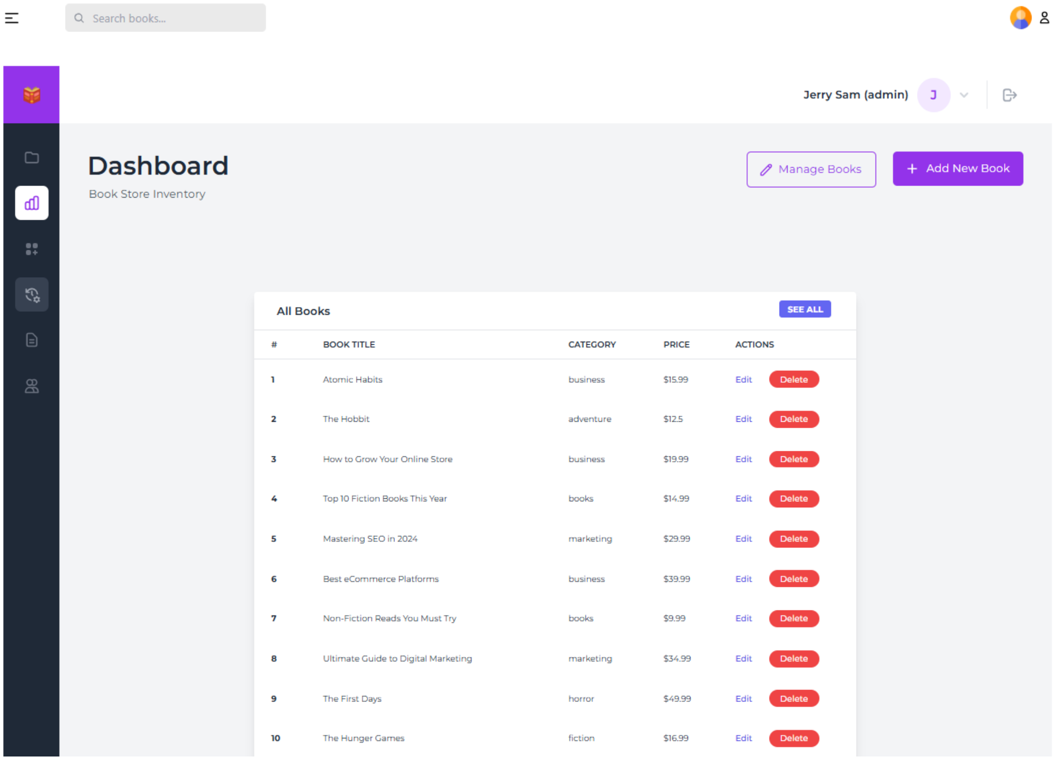


Figure 37: Illustrator of “Add New Book” page for Admin

The Add New Book Page is an administrative interface designed to allow administrators to efficiently add new books to the Book-Store system. The page presents a structured form where admins can enter essential book information, including the title, description, category, pricing details, and cover image. A “Trending” option enables administrators to highlight selected books for promotional display on the storefront. Input fields are clearly labeled, and a file upload feature supports adding book cover images. Action buttons such as Add Book and Manage Books provide quick navigation and streamlined content management.

9.1.13. “Manage Book” page for Admin



All Books

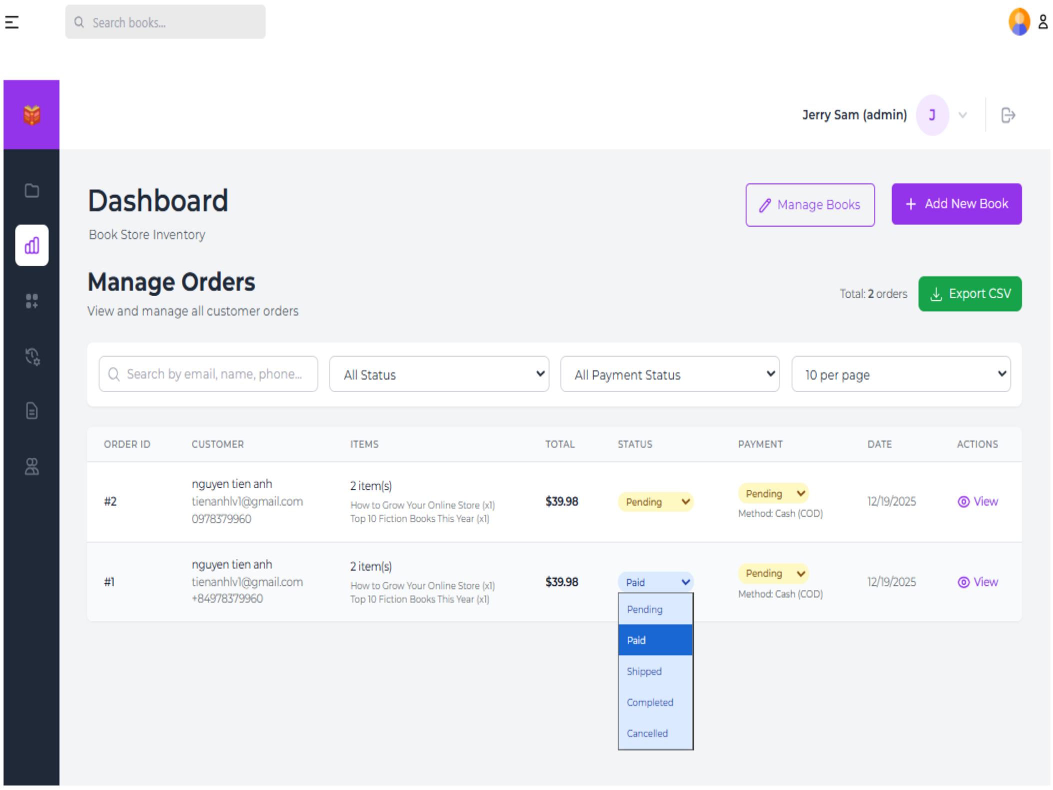
SEE ALL

#	BOOK TITLE	CATEGORY	PRICE	ACTIONS
1	Atomic Habits	business	$15.99	Edit Delete
2	The Hobbit	adventure	$12.5	Edit Delete
3	How to Grow Your Online Store	business	$19.99	Edit Delete
4	Top 10 Fiction Books This Year	books	$14.99	Edit Delete
5	Mastering SEO in 2024	marketing	$29.99	Edit Delete
6	Best eCommerce Platforms	business	$39.99	Edit Delete
7	Non-Fiction Reads You Must Try	books	$9.99	Edit Delete
8	Ultimate Guide to Digital Marketing	marketing	$34.99	Edit Delete
9	The First Days	horror	$49.99	Edit Delete
10	The Hunger Games	fiction	$16.99	Edit Delete

Figure 38: Illustrator of “Manage Book” page for Admin

The Manage Books Page is an administrative interface that enables administrators to efficiently oversee and control the book inventory of the Book-Store system. The page displays a structured table listing all available books, including key details such as book ID, title, category, and price. Action buttons next to each entry allow administrators to edit book information or delete books from the system, ensuring accurate and up-to-date inventory management. A See All option supports quick navigation when managing a large catalog, while shortcut buttons such as Add New Book and Manage Books enhance workflow efficiency.

9.1.14. “Manage Book” page for Admin



Manage Orders

View and manage all customer orders

Total: 2 orders

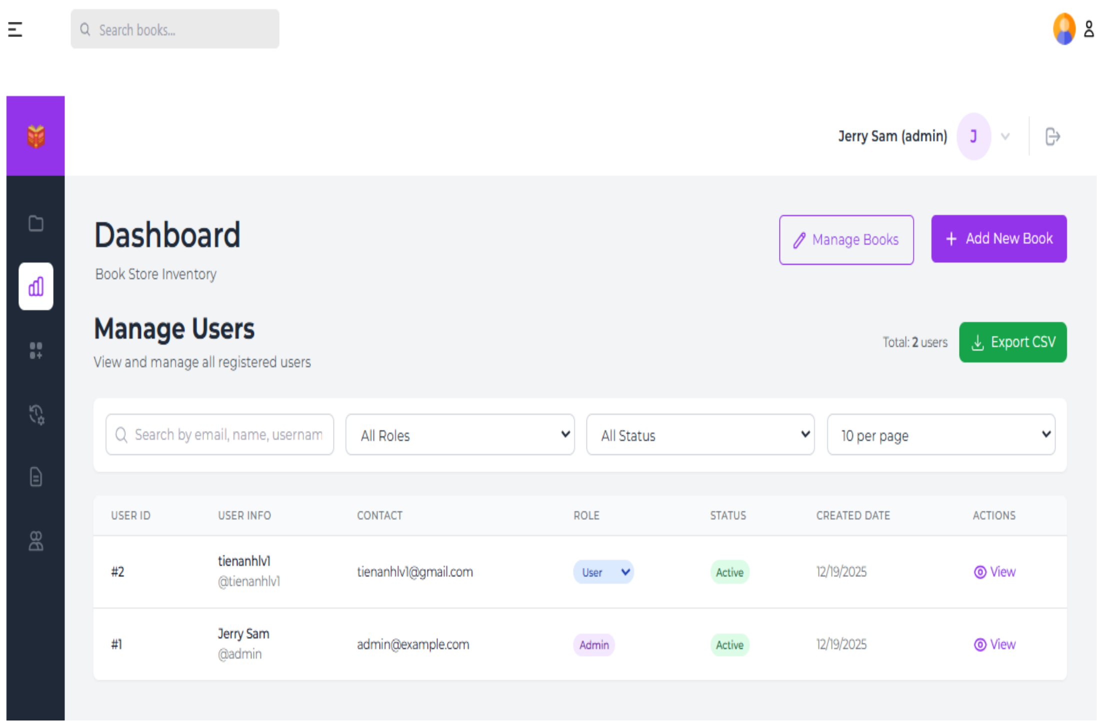
Search by email, name, phone... | All Status | All Payment Status | 10 per page

ORDER ID	CUSTOMER	ITEMS	TOTAL	STATUS	PAYMENT	DATE	ACTIONS
#2	nguyen tien anh tienanhlv1@gmail.com 0978379960	2 item(s) How to Grow Your Online Store (x1) Top 10 Fiction Books This Year (x1)	$39.98	Pending	Pending Method: Cash (COD)	12/19/2025	View
#1	nguyen tien anh tienanhlv1@gmail.com +84978379960	2 item(s) How to Grow Your Online Store (x1) Top 10 Fiction Books This Year (x1)	$39.98	Paid Pending Paid Shipped Completed Cancelled	Pending Method: Cash (COD)	12/19/2025	View

Figure 39: Illustrator of “Manage Orders” page for Admin

The Manage Orders Page is an administrative interface designed to help administrators efficiently monitor and control all customer orders within the Book-Store system. The page displays a detailed table listing each order with essential information, including order ID, customer details, purchased items, total amount, order status, payment status, and order date. Advanced filtering and search options allow administrators to quickly locate orders by customer information, order status, or payment status. Administrators can update order progress through selectable status options such as pending, paid, shipped, completed, or cancelled, ensuring accurate order tracking. An Export CSV feature supports reporting and record management, while the View action provides access to detailed order information.

9.1.15. “Manage Users” page for Admin



Dashboard

Book Store Inventory

Manage Users

View and manage all registered users

Total: 2 users

Search by email, name, usernam

All Roles

All Status

10 per page

USER ID	USER INFO	CONTACT	ROLE	STATUS	CREATED DATE	ACTIONS
#2	tienanhlv1 @tienanhlv1	tienanhlv1@gmail.com	User	Active	12/19/2025	View
#1	Jerry Sam @admin	admin@example.com	Admin	Active	12/19/2025	View

Figure 40: Illustrator of “Manage Users” page for Admin

The Manage Users Page is an administrative interface that enables administrators to view and control all registered users within the Book-Store system. The page presents a structured table displaying key user information, including user ID, username, email address, role, account status, and registration date. Advanced search and filtering options allow administrators to quickly locate users based on email, username, role, or status. Role management features enable administrators to assign or update user roles, such as user or admin, while account status indicators help monitor active accounts. An Export CSV function supports data management and reporting, and the View action provides access to detailed user profiles.

CHAPTER 10: PROJECT EVALUATION

10.1. Evaluation Criteria Used to Measure Project Success

The success of the Book Store project is evaluated based on five key criteria: performance in relation to the business case, schedule adherence, quality assurance, cost management, and stakeholder satisfaction. These criteria are assessed continuously throughout the project lifecycle and formally reviewed at the end of each major development phase. Such evaluations provide a clear understanding of project progress and support informed decision-making regarding continuation, adjustment, or enhancement of subsequent phases.

Schedule Baseline: Adherence to the planned project schedule is a critical indicator of the Book Store project’s success. The development process was divided into distinct phases, including requirement analysis, system design, core feature implementation (such as browsing, filtering, cart, checkout, and user authentication), admin functionalities, testing, and final deployment.

Progress was monitored on a regular basis to ensure that major milestones—such as completing the product browsing interface, implementing secure authentication, and deploying the admin dashboard—were achieved within the expected timeframe. Minor adjustments to the schedule were made when necessary, particularly during the testing and refinement stages, without significantly affecting the overall delivery timeline. This flexible yet controlled approach ensured timely project completion while maintaining development quality.

Quality Assurance: Quality assurance played a central role in evaluating the Book Store project. At the end of each development phase, functionality testing and validation were conducted to ensure that the system met predefined requirements. Core features such as book searching and filtering, shopping cart management, checkout processing, wishlist handling, rating and review submission, and order tracking were carefully tested to confirm accuracy, usability, and reliability.

From an administrative perspective, quality checks were also applied to features such as book management, order management, user management, and role-based access control. Any defects or inconsistencies identified during testing were addressed early to prevent issues from propagating into later stages. This systematic approach ensured that both user-facing and admin-facing components met acceptable quality standards.

Project Budget: Budget evaluation focused on ensuring that the Book Store project was developed within the allocated resources. As the project primarily involved software development, cost considerations centered on development time, tool usage, and infrastructure requirements rather than physical assets.

By maintaining a clear scope and prioritizing essential e-commerce features, the project avoided unnecessary expenditures. Continuous monitoring of development efforts helped prevent scope creep and ensured that the final system aligned with the original budget expectations. The project remained financially controlled while still delivering a fully functional and scalable Book Store platform.

Stakeholder Satisfaction: Stakeholder satisfaction is an important qualitative measure of project success. In the Book Store project, stakeholders include end users (customers), administrators, and the project development team. User satisfaction was evaluated based on the system's ease of use, clarity of interface, responsiveness, and overall shopping experience.

Features such as intuitive navigation, advanced filtering, secure checkout, and order history tracking contributed positively to user feedback. Administrators also benefited from a clear and efficient dashboard that simplifies inventory, order, and user management tasks. Feedback gathered during testing and review phases was used to refine features and improve usability, ensuring stakeholder expectations were met.

Performance to Business Case: Evaluating performance against the original business case ensures that the Book Store project continues to serve its intended purpose. The primary objective of the project was to create a secure, user-friendly, and efficient online book-selling platform that supports both customer purchases and administrative operations.

The completed system successfully addresses this objective by providing comprehensive browsing and filtering capabilities, seamless transaction handling, customer engagement features such as wishlists and reviews, and robust admin controls. The platform remains aligned with current market needs and demonstrates clear potential for future expansion, such as additional payment methods, personalized recommendations, or analytics enhancements.

Regularly revisiting the business case throughout development helped ensure that the project remained relevant and valuable, preventing misalignment with user needs or organizational goals.

10.2. Assessment of Whether Project Goals Were Achieved

The Book Store project has been successfully completed, achieving 100% implementation of the core requirements of an e-commerce platform for selling books. The system fully satisfies essential user needs, including browsing books, searching and filtering by title, category, and price, viewing detailed product information, adding items to the shopping cart, managing quantities, completing the checkout process, and tracking orders.

Most of the project objectives have been accomplished at a level of 95% to 100%, particularly in areas such as intuitive user interface design, smooth navigation, secure authentication, wishlist functionality, customer reviews and ratings, and a structured order management process. The checkout flow supports multiple payment options and integrates saved delivery addresses, contributing to a convenient and reliable purchasing experience.

On the administrative side, the project goals were also effectively achieved. Administrators are able to manage books, users, and orders through a centralized dashboard, update order statuses, control user roles, and monitor platform performance. These features ensure operational efficiency and support the long-term scalability of the system.

Some advanced features, such as further optimization of customer feedback handling and extended security monitoring, are still undergoing testing and refinement. However, these do not affect the overall stability or usability of the system. With continued improvement, the Book Store platform is well-positioned to fully realize all planned functionalities.

10.3. User Feedback and Satisfaction

Wide Selection of Books: Users appreciate the diverse collection of books available on the platform, covering multiple genres such as fiction, non-fiction, business, marketing, and adventure. The Book Store effectively serves as a one-stop destination for readers with different interests and preferences.

User-Friendly Interface: Feedback indicates that the interface is clean, intuitive, and easy to navigate. Users can quickly browse categories, apply filters, access product details, and complete purchases without confusion. The clear layout significantly enhances the overall shopping experience.

Detailed Book Information: Customers value the comprehensive product detail pages, which include book descriptions, prices, discounts, categories, publication information, and customer reviews. This level of detail helps users make informed purchasing decisions and increases confidence in the platform.

Smooth Checkout and Order Tracking: The checkout process is perceived as straightforward and secure. Features such as address management, multiple payment options, and order confirmation messages contribute to user trust. The order history and status tracking functionality allows users to stay informed about their purchases.

Reviews and Ratings System: Users find the rating and review feature especially useful, as it allows them to share their experiences and learn from other customers. This interactive element fosters trust within the community and positively influences purchasing behavior.

Responsive System and Reliability: Overall satisfaction is also driven by the system's responsiveness and reliability. Pages load quickly, actions are processed accurately, and errors are handled clearly, resulting in a stable and pleasant user experience.

In summary, user feedback reflects a high level of satisfaction with the Book Store platform. The system successfully meets user expectations in terms of functionality, usability, and reliability, while also providing a solid foundation for future enhancements and feature expansion.

CHAPTER 11: FUTURE WORK

11.1. Responsive Interface Design

The Book Store platform is designed with a fully responsive user interface to ensure a consistent and seamless experience across different devices and screen sizes. The system adapts smoothly to desktops, laptops, tablets, and smartphones, allowing users to browse, search, and purchase books comfortably regardless of the device they use.

A mobile-first design approach was applied during development. Core features such as book browsing, search and filtering, product details, cart management, and checkout were first optimized for small screens. The interface was then progressively enhanced for larger displays, ensuring that essential content and functionality remain easily accessible on mobile devices while taking advantage of additional screen space on tablets and desktops.

Responsive layout techniques, including flexible grids and adaptive components, are used throughout the platform. Book cards, product images, and banners automatically scale to fit different viewports. Images are constrained using flexible sizing rules (such as maximum width limits) to prevent overflow and maintain visual consistency. Typography relies on relative units, enabling text to resize appropriately across devices and improve readability.

Special attention is given to navigation and user interaction on smaller screens. The Book Store employs compact navigation patterns such as collapsible menus and icon-based actions to reduce clutter while preserving functionality. Buttons, form inputs, and interactive elements are designed to be touch-friendly, with adequate spacing to minimize accidental taps during actions like adding books to the cart, submitting reviews, or completing checkout.

User testing was conducted across multiple devices to identify and resolve usability issues related to responsiveness. Adjustments to spacing, alignment, and interaction feedback were made to ensure that all pages—including Browse Books, Product Detail, Cart, Checkout, Profile, and Order History—deliver a smooth and intuitive experience.

11.2. System Optimization

System optimization plays a vital role in improving the overall performance, speed, and reliability of the Book Store e-commerce platform. In a highly competitive online book retail environment, maintaining a responsive, secure, and scalable system is essential to ensure a smooth user experience and long-term sustainability. The Book Store project incorporates several optimization strategies, including server deployment, performance tuning, security enhancement, and automation of user interactions.

Deploying the Book Store platform on a dedicated server environment significantly improves system availability and reliability. Server-side hosting enables efficient resource management, faster response times, and reduced downtime during peak usage periods such as promotions or high-traffic browsing sessions. In addition, server deployment supports seamless integration with databases, payment services, and third-party tools, contributing to a stable and consistent shopping experience.

Security optimization is another critical aspect of the system. To protect user accounts and transaction data, the platform integrates secure authentication mechanisms, including encrypted credentials and token-based access control. Where applicable, additional verification methods such as One-Time Password (OTP) can be applied during sensitive operations like login, password recovery, or payment confirmation. These measures help reduce the risk of unauthorized access and increase user trust in the platform.

Automation is used to enhance efficiency and improve customer interaction. Features such as automated order confirmation messages, payment status updates, and system notifications help keep users informed without requiring manual intervention. The rating and review system further contributes to system optimization by automatically collecting and organizing customer feedback, which can be analyzed to improve product offerings and user experience.

The star rating mechanism provides valuable quantitative insights into customer satisfaction. By allowing users to rate books and share reviews, the platform gathers structured feedback that supports data-driven decision-making. This information benefits both administrators—by highlighting popular or low-performing books—and customers, who can rely on ratings to make informed purchasing decisions.

11.3. AI and Machine Learning Integration

The integration of Artificial Intelligence (AI) into the Book Store platform is primarily aimed at enhancing customer support through an intelligent chatbot system. In an online book retail environment, timely and accurate assistance plays a crucial role in improving user satisfaction and maintaining customer trust. An AI-powered chatbot serves as an effective solution to provide continuous and scalable customer support.

The AI chatbot is designed to assist customers by responding to common inquiries related to book information, order status, payment methods, delivery policies, and account management. By utilizing natural language processing (NLP) techniques, the chatbot can understand user queries written in everyday language and deliver relevant, context-aware responses. This reduces the need for manual customer service intervention while ensuring users receive immediate support.

In addition, the chatbot can guide users through the shopping process by recommending books based on user queries, browsing behavior, or selected categories. For example, when a customer asks for suggestions within a specific genre or price range, the chatbot can provide suitable book recommendations, enhancing the overall shopping experience.

From an operational perspective, AI integration supports system efficiency by automating routine interactions such as order tracking inquiries, refund policies, and frequently asked questions. This allows human support staff to focus on more complex issues that require personalized attention. Over time, machine learning techniques can be applied to improve chatbot accuracy by learning from previous conversations and customer feedback.

Overall, the integration of AI-driven chatbot functionality into the Book Store platform contributes to a more responsive, personalized, and efficient customer support system. This approach not only improves user experience but also supports the long-term scalability and competitiveness of the Book Store in the evolving e-commerce landscape.

11.4. Continuous User Feedback and Testing

Continuous user feedback and systematic testing are essential to maintaining and improving the quality of the Book Store platform. An iterative development approach is adopted to ensure that the system evolves in line with user needs, expectations, and real-world usage patterns.

The Book Store integrates multiple feedback mechanisms to collect user opinions and experiences. Features such as ratings and reviews allow customers to directly express their satisfaction with purchased books, while order-related feedback helps identify issues in checkout, delivery, or payment processes. These user-generated insights provide valuable qualitative and quantitative data for evaluating system performance and usability.

In addition to passive feedback collection, active evaluation methods such as user surveys and structured feedback forms can be employed to gather targeted insights regarding interface design, navigation flow, and feature usefulness. This information supports informed decision-making when prioritizing improvements or introducing new functionalities.

Testing plays a critical role in validating enhancements and ensuring system stability. Functional testing is regularly conducted on core features such as browsing and filtering books, cart management, checkout, order tracking, and profile management. Where applicable, A/B testing can be used to compare alternative interface designs or interaction flows, such as different layouts for product listings or checkout steps, to determine which version provides better usability and conversion rates.

Based on collected feedback and testing results, iterative refinements are applied to both frontend and backend components. This continuous improvement cycle helps reduce usability issues, enhance performance, and align the platform more closely with user preferences. Overall, the combination of ongoing user feedback and structured testing ensures that the Book Store platform remains user-centered, reliable, and adaptable to future growth.

CHAPTER 12: TESTING

12.1. Test case 1: Login

Step	Action	Expected System Response	Pass/Fail
1	Access the website	Show the user the page in guest mode	Pass
2	Enter email and password		Pass
3	Press “Login” button on login page	Check email and password into the database	Pass
4	Check the account’s validity	The supplied account is compared with the	Pass

		database. Send the user to the login page with an error message if the account is invalid. Otherwise, send a successful message if the account is valid.	
5	Examine the account type	Check the database with the provided account: If the account has a client role, go back to the client home page. Otherwise, if the account has an admin role, go back to the admin page.	Pass

Table 6: Table of Test Cases of Login function

12.2. Test case 2: Register

Step	Action	Expected System Response	Pass/Fail
1	Access the website	Show the user the page in guest mode	Pass
2	Press “Register” button	Display the “Register” page	Pass
3	Enter email and password		Pass

4	Press the “Register” button in “Register” page	Load this information into the database	Pass
5	Save user information into database	Upload user data including email and password first.	Pass
6	Navigate to “Login” page for checking		Pass
7	The user enters the email and password for the account after registering.	Show the “Successful” message and navigate to Homepage	Pass

Table 7: Table of Test Cases of Register function

12.3. Test case 3: Add to Cart

Step	Action	Expected System Response	Pass/Fail
1	Access the website	Display the homepage in guest or user mode	Pass
2	Browse books or search for a book	Show the list of available books	Pass
3	Select a book	Display the Product Detail page with book information	Pass
4	Click the “Add to Cart” button	Add the selected book to the shopping cart	Pass

5	System updates cart icon	Increase cart item count by 1	Pass
6	Navigate to Cart Page	Display cart page with selected book	Pass
7	Verify cart item details	Show correct book title, price, quantity, and subtotal	Pass
8	Change quantity using + / − buttons	Update quantity and recalculate subtotal automatically	Pass
9	Remove the book from cart (optional)	Remove item and update cart total	Pass
10	Add multiple books to cart	Display all selected books correctly	Pass
11	Refresh the page	Preserve cart items (session/local storage)	Pass
12	Click Checkout	Navigate to Checkout page with cart data	Pass

Table 8: Table of Test Cases of Add to Cart function

12.4. Test case 4: Checkout

Step	Action	Expected System Response	Pass/Fail
1	Navigate to Cart Page with items added	Display cart with selected books and correct subtotal	Pass

2	Click Checkout button	Navigate to the Checkout page	Pass
3	View Order Summary section	Display subtotal, shipping not and total amount.	Pass
4	Select a payment method (COD or VNPay)	System highlights the selected payment method	Pass
5	Choose Saved Address	Automatically load saved address into delivery form	Pass
6	Verify delivery information	Display correct name, email, phone, and address	Pass
7	Edit delivery details (if needed)	Allow user to update address fields	Pass
8	Check Terms & Conditions checkbox	Enable the Confirm Payment button	Pass
9	Click Confirm Payment	Validate input data and payment method	Pass
10	System processes the order	Save order details into database	Pass
11	Display order confirmation	Show successful order message	Pass
12	Redirect to My Orders page	Display newly created order with correct status	Pass

Table 9: Table of Test Cases of Checkout function

12.5. Test case 5: Wishlist

Step	Action	Expected System Response	Pass/Fail
1	Access the website	Display the homepage successfully	Pass
2	Login with a valid user account	Navigate to homepage in user mode	Pass
3	Browse books or search for a book	Display list of available books.	Pass
4	Click the Wishlist (heart) icon on a book	Add the selected book to Wishlist	Pass
5	System updates wishlist icon/count	Increase wishlist item count by 1	Pass
6	Navigate to Wishlist Page	Show correct book image, title, category, and price	Pass
7	Verify wishlist item details	Allow user to update address fields	Pass
8	Click Move to Cart	Add selected book to Cart and keep/remove from Wishlist	Pass
9	Navigate to Cart	Display the moved	Pass

	Page	book correctly in cart	
10	Click Remove on a wishlist item	Remove the selected book from Wishlist	Pass
11	Refresh the page	Persist wishlist items correctly	Pass
12	Logout and login again	Wishlist items remain saved for the user	Pass

Table 10: Table of Test Cases of Wishlist function

12.6. Test case 6: Addresses

Step	**Action**	**Expected System Response**	**Pass/Fail**
1	Access the website	Display the homepage successfully	Pass
2	Login with a valid user account	Navigate to homepage in user mode	Pass
3	Click on user avatar / Profile menu	Display My Profile page	Pass
4	View Profile information tab	Show personal information including email, full name, and phone number	Pass
5	Verify email field	Display email as read-only with	Pass

		"Email cannot be changed" message	
6	Edit Full Name field	Allow user to update full name	Pass
7	Edit Phone Number field	Allow user to input a valid phone number	Pass
8	Click Save Changes	Update profile information in the database	Pass
9	Display confirmation	Show success message after saving changes	Pass
10	Refresh the page	Persist updated profile information correctly	Pass
11	Click Change Password	Navigate to Change Password page / modal	Pass
12	Enter valid old and new passwords	Validate password rules and inputs	Pass
13	Submit password change	Update password securely and show success message	Pass
14	Switch to Order History tab	Display saved addresses for selection	Pass

Table 11: Table of Test Cases of Addresses function

12.7. Test case 7: Profiles

Step	Action	Expected System Response	Pass/Fail
1	Access the website	Display the homepage successfully	Pass
2	Login with a valid user account	Navigate to homepage in user mode	Pass
3	Navigate to Addresses page	Show all saved delivery addresses with correct details	Pass
4	View saved addresses	Add the selected book to Wishlist	Pass
5	Verify default address	Display the default address with “Default” label	Pass
6	Click Add New Address	Display the Add Address form	Pass
7	Enter valid address information	Allow user to input name, phone, street, city, region, and zip code	Pass
8	Save the new address	Store the address in the database	Pass
9	System updates	Display the newly	Pass

	address list	added address in the list	
10	Click Edit on an existing address	Update address data successfully	Pass
11	Update address information and save	Update address data successfully	Pass
12	Click Delete on an address	Remove the selected address after confirmation	Pass
13	Refresh the page	Persist address data correctly	Pass
14	Navigate to Checkout page	Display saved addresses for selection	Pass

Table 12: Table of Test Cases of Profiles function

12.8. Test case 8: Orders

Step	**Action**	**Expected System Response**	**Pass/Fail**
1	Access the website	Display the homepage successfully	Pass
2	Login with a valid user account	Navigate to homepage in user mode	Pass
3	Open user menu / profile dropdown	Display available account options	Pass
4	Click Orders	Navigate to the	Pass

		Orders page	
5	View the order list	Display all orders of the logged-in user	Pass
6	Verify order summary information	Show order ID, date, total price, payment method, and status	Pass
7	Check order progress tracker	Display correct steps (Order Placed → Payment Confirmed → Shipped → Delivered)	Pass
8	Verify customer details section	Show correct name, email, and phone number	Pass
9	Verify shipping address section	Show correct address information linked to the order	Pass
10	Verify products list	Display ordered books/items correctly for each order	Pass
11	Refresh the page	Persist and reload the order history correctly	Pass
12	Logout and login again	Orders remain accessible and unchanged for the user	Pass

Table 13: Table of Test Cases of Orders function

12.9. Test case 9: Filter Products

Step	Action	Expected System Response	Pass/Fail
1	Access the website	Display the homepage successfully	Pass
2	Navigate to Browse Books page	Display the list of all available books	Pass
3	View search bar, category filter, and sort dropdown	Display search input, category selector, and sorting options correctly	Pass
4	Enter a keyword in the search bar	Filter and display books matching the entered keyword	Pass
5	Clear the search input	Restore the full list of books	Pass
6	Select a category from All Categories dropdown	Display only books belonging to the selected category	Pass
7	Change category back to All Categories	Display all books again	Pass
8	Select a sorting option (e.g., *Newest First*)	Reorder books according to the selected sorting rule	Pass

9	Combine search keyword and category filter	Display books that satisfy both conditions	Pass
10	Change sorting while filters are active	Update the displayed list without losing applied filters	Pass
11	Refresh the page	Reset filters and display default book list	Pass
12	LNavigate to another page and return to Browse Books	Reload Browse Books page with default filter settings	Pass

Table 14: Table of Test Cases of Filter function

12.10. Test case 10: Rating

Step	Action	Expected System Response	Pass/Fail
1	Access the website	Display the homepage successfully	Pass
2	Login with a valid user account	User is authenticated and redirected to homepage	Pass
3	Navigate to a product details page	Display product information and Customer Reviews section	Pass
4	Scroll to Customer Reviews section	Display rating starts and review input	Pass

		form	
5	Select a star rating (e.g., 4 stars)	Highlight selected stars correctly	Pass
6	Enter text into the review textarea	Display the entered review content correctly	Pass
7	Click Submit Review button	Validate rating input before submission	Pass
8	Submit review with a valid rating	Save the rating and review to the database	Pass
9	Display submitted review	Show the new review under Customer Reviews section	Pass
10	Refresh the page	Persist and reload the submitted rating and review	Pass
11	Submit review without selecting a rating	Display validation error message	Pass
12	Submit review with empty text (optional review)	Allow submission and save rating only one per user	Pass

Table 15: Table of Test Cases of Rating function

CHAPTER 13: CONCLUSION

13.1. Reiteration of Key Achievements

The Book Store e-commerce platform has successfully achieved its core objectives and reached several important milestones throughout its development. The system provides users with a clean, intuitive, and user-friendly interface that allows customers to easily browse, search, and filter books by category, price, popularity, and other relevant attributes. This streamlined shopping experience significantly enhances user satisfaction and encourages repeated purchases.

One of the platform's notable achievements is the implementation of essential e-commerce functionalities such as the shopping cart, wishlist, and checkout process. Customers can conveniently add books to their cart, manage quantities, save favorite titles for later purchase, and complete transactions through a secure and straightforward checkout flow. These features help reduce purchase friction and improve overall engagement.

Security and reliability are also key accomplishments of the Book Store project. Secure authentication, role-based access control, and protected payment handling ensure that user data and transactions remain safe. The platform has demonstrated stable performance without major security issues, fostering trust and confidence among users.

In addition, the review and rating system plays a crucial role in strengthening transparency and credibility. By allowing customers to share ratings and feedback on purchased books, the platform supports informed decision-making for future buyers. This continuous feedback loop contributes to improved content quality, increased trust, and higher customer satisfaction.

To outline, the Book Store project has successfully prioritized user needs by delivering practical features, ensuring secure transactions, encouraging customer interaction, and maintaining a scalable and maintainable system architecture.

13.2. Closing Remarks

In conclusion, the Book Store project has proven to be a well-designed and functional e-commerce platform within the online book retail domain. By focusing on user experience, feature completeness, security, and administrative efficiency, the system establishes a solid foundation for sustainable growth. Key components such as advanced search and filtering, order tracking, user profile management, and an admin dashboard for managing books, users, and orders contribute to the platform's robustness.

The project also reflects effective teamwork and thoughtful system planning, resulting in a reliable application that meets both customer and administrator requirements. Despite being a relatively small and newly developed platform, the Book Store demonstrates strong potential to evolve and compete in the rapidly expanding e-commerce market.

With continuous improvements, feature expansion, and responsiveness to user feedback, the Book Store platform is well-positioned to deliver long-term value, enhance customer satisfaction, and achieve greater success in the online book-selling industry.

CHAPTER 14: REFERENCES

[1] What is mysql? (no date) Oracle. Available at: https://www.oracle.com/mysql/what-is-mysql/ (Accessed: 30 May

[2] GeeksforGeeks (2024) Integration testing - software engineering, GeeksforGeeks. Available at: https://www.geeksforgeeks.org/software-engineering-integration-testing/ (Accessed: 7 June 2024)

[3] What is unit testing? (no date) smartbear.com. Available at: https://smartbear.com/learn/automated-testing/what-is-unit-testing/ (Accessed: 6 June 2024).

[4] What is Node.js? Understanding Its Applications. Available at: https://nodejs.org/en/about

[5] Build Book Store Application Full Stack using React, Node, MongoDB. Available at: https://www.youtube.com/watch?v=pgw2KPfgK1E&t=2418s

[6] FURNISCAPE - WORLD OF FURNITURE : Final Report Web Application.

[7] MVC Framework Introduction. Available at: https://www.geeksforgeeks.org/software-engineering/mvc-framework-introduction/

[8] JavaScript Tutorial. Available at: https://www.w3schools.com/Js/

[9] Learning NodeJS and ExpressJS. Available at: https://www.youtube.com/watch?v=z2f7RHgvddc&list=PL_-VfJajZj0VatBpaXkEHK_UPHL7dW6I3