

VIETNAM NATIONAL UNIVERSITY OF HOCHIMINH CITY
THE INTERNATIONAL UNIVERSITY
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING



A SMART WEB APP FOR BOOKSTORE MANAGEMENT

By
Nguyen Thi Kim Thu

A thesis submitted to the School of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
Bachelor of Engineering in Computer Science

Ho Chi Minh City, Vietnam

2025

A SMART WEB APP FOR BOOKSTORE MANAGEMENT

APPROVED BY:

Assoc. Prof. Nguyen Van Sinh

Assoc. Prof. Nguyen Thi Thuy Loan

Dr. Nguyen Trung Ky

THESIS COMMITTEE

(Whichever applies)

ACKNOWLEDGMENTS

I would like to express my sincere and deepest gratitude to Associate Professor Nguyen Van Sinh, my thesis supervisor, for his professional guidance throughout the process. Without his support and insightful understanding, I would have encountered many difficulties in completing this thesis. His guidance has provided me with invaluable insights and I am truly grateful to him for accompanying me on this journey.

I would like to take this opportunity to express my sincere gratitude to the lecturers of the Faculty of Computer Science and Engineering at International University - VNU-HCM for providing me with knowledge, a solid academic foundation, and a conducive learning environment over the past four years.

In addition to the learning environment, I would like to thank my parents for their love, encouragement, and support in completing this thesis. Their strong belief in me has always been a great motivation for me.

To my friends, I truly appreciate the companionship and sharing that everyone has given me throughout this time, making this journey interesting and meaningful.

This thesis is supported by The central Interdisciplinary Laboratory in Electronics and Information Technology (AI and Cooperation Robot), International University - VNU-HCM. We would like to thank for supporting machines in experiments

TABLE OF CONTENTS

ACKNOWLEDGMENTS	3
TABLE OF CONTENTS	4
LIST OF FIGURES	6
LIST OF TABLE.....	7
ABSTRACT	8
CHAPTER 1	9
INTRODUCTION	9
1.1. Background	9
1.2. Problem Statement.....	9
1.3. Scope and Objectives.....	10
1.4. Assumption and Solution.....	12
1.5. Structure of thesis	13
CHAPTER 2	15
RELATED WORK.....	15
2.1. Amazon Platform	15
2.1.1. Key Funtionalities	15
2.1.2. Strength and weakness	17
2.2. Tiki.....	21
2.2.1. Key functionalities.....	21
2.2.2. Strength.....	23
2.3. Recommendation System	24
2.4. Price Suggestion System.....	26
2.4.1. Overview	26
2.4.2. Theoretical basis.....	26
2.4.3. Actual data.....	27
CHAPTER 3	28
METHODOLOGY	28
3.1. User requirement analysis.....	28
3.1.1. Functional requirements	29
3.1.2. Non-Functional requirements	31
3.2. System Design	31
3.2.1. Datadase design	32
3.2.2. Sequence Diagram.....	34
3.2.3. User Interface design	37
3.3. Algorithms and Technologies Used.....	38
3.3.1. Recommendation Systems in Next.js:	38

3.3.2. Smart Inventory system:	41
3.3.3. Advanced Search Function	46
3.3.4. Backend Development with Node.js:	47
3.4. Class diagram.....	48
CHAPTER 4	50
IMPLEMENT AND RESULTS	50
4.1. Implement	50
4.1.1. Backend Development.....	50
4.1.2. Frontend Development	50
4.2. Results.....	50
4.2.1. Authentication	51
4.2.2. Products	53
CHAPTER 5	65
DISCUSSION AND EVALUATION	65
5.1. Discussion	65
5.2. Comparison.....	66
Table 5.2. Comparison table.....	66
5.3. Evaluation	66
CHAPTER 6	68
CONCLUSION AND FUTURE WORK	68
6.1. Conclusion	68
6.2. Future work.....	68
REFERENCES	70
APPENDIX	71

LIST OF FIGURES

Figure 1.5. Report Structure	13
Figure 2.1.1. Amazon search function with caterogies	16
Figure 2.1.1. Rating in Amazon	16
Figure 2.1.2. Recommendation system in Amazon.....	18
Figure 2.1.2. Betseller Recommend in Amazon.....	19
Figure 2.2.1. Tiki's Reviews	22
Figure 2.2.1. Tiki's Chatbox.....	23
Figure 2.2.2. Tiki's Recommendation	24
Figure 2.3. Shopee's Recommendation.....	25
Figure 2.3. Youtube's Recommendation.....	26
Figure 3.1. Use case diagram	29
Figure 3.3.1. Table Relationship diagram	32
Figure 3.3.2. Authentication and Update sequence diagram.....	34
Figure 3.3.2. Product sequence diagram	36
Figure 3.5. Class diagram.....	48
Figure 4.2.1. Signin page.....	52
Figure 4.2.1. Manage page	52
Figure 4.2.1. Register page.....	53
Figure 4.2.1. Login database	53
Figure 4.2.2. Add products page.....	54
Figure 4.2.2. Caterogy dropdown.....	54
Figure 4.2.2. Book field insert.....	55
Figure 4.2.2. Products table	55
Figure 4.2.2. Show products page (Store owner side).....	56
Figure 4.2.2. Editing product.....	56
Figure 4.2.2. All products page after filter	57
Figure 4.2.2. Cart page	57
Figure 4.2.2. Product detail page with tags	58
Figure 4.2.3. Intersting Component.....	58
Figure 4.2.3. Customer-tag-score table in database.....	59
Figure 4.2.3. Book of the day component	59
Figure 4.2.3. Details page with Related Items.....	60
Figure 4.2.2. Product table with Clicked change.....	60
Figure 4.2.2. Hot product and Best seller component	61
Figure 4.2.4. After clicking History tag.....	61
Figure 4.2.4. Search result for products that do not exist in database	62
Figure 4.2.5. Check out page.....	62
Figure 4.2.5. Orders page [Customer]	63
Figure 4.2.5. Orders details page with review [Customer].....	63
Figure 4.2.5. Notification page	64
Figure 4.2.5. Orders page [Store Owners].....	64

LIST OF TABLE

Table 5.2. Comparison table.....	66
----------------------------------	----

ABSTRACT

The transformation in shopping habits brought about by rapid the development of the Internet and e-commerce platforms is a common phenomenon in today's society. This project focuses on creating an intelligent and automated web application for bookstore management. The developed system incorporates intelligent systems that improve customer experience as well as store management.

On the customer's end, the application includes book recommendation systems that encourage purchases tailored to user behavior. Moreover, the platform is provided with a simple and clear layout making sure that even people who are not technologically savvy are able to use it.

With regard to the store owner's point of view, the system is directed toward automation and real-time data analysis. It has an advanced inventory control system that manages stock balance information, alerting when there is too much or too little stock and helping avoid supply disruptions. In addition, the system provides additional automated reports for business analysis to improve clarity and understanding of sales performance indicators, revenue earned in the last days/weeks/months, and the most demanded goods. All of these help store owners act promptly and make data-driven business decisions.

The target of the system is to increase user satisfaction as well as business performance. Overall, this project is an example of how smart web applications.

CHAPTER 1

INTRODUCTION

1.1. Background

Today, in our modern life, e-commerce become more and more popular all over the world. That led to a change in book sales. In fact, online bookstores bring a large number of benefits and convenience for both sellers and customers. Customers can browse, purchase, and receive books from home instead of jostling and time-consuming searching in traditional bookstores. Besides that, the system can automatically manage inventory, recognize revenue, and notify the sellers. However, that also brings out many challenges for the owner, for example, processing orders efficiently, and reaching customers needed by personalized recommendations system because of the lack of human factors to make important decisions and advise customers.

In recent years, there have been more and more individuals who are concerned about the development of intelligent systems to improve the efficiency of bookstore management. Now, people can take advantage of advanced technologies such as artificial intelligence, machine learning, and data analysis to solve and enhance their systems. They can used to analyze customer behavior, streamline the inventory management system, and order tracking.

The need for a smart system, that can be extended and affordable to manage the bookstore is clear. By developing a smart and automatic web that integrates modern technologies, many bookstores can overcome the challenges and bring many benefits that can raise consumer satisfaction and business performance.

1.2. Problem Statement

Traditional bookstores face constraints such as limited space, difficulty, and inconvenience in finding products. These constraints often lead to a poor shopping experience, especially when customers cannot easily find or receive recommendations for suitable books.

Although online bookstores have improved accessibility, many bookstores still suffer from weak recommendation systems, poor inventory management, and a lack of real-time analytics. Customers often receive irrelevant recommendations, missing out on new or trending books.

This project addresses these constraints by developing an intelligent, automated bookstore system. The system has the following features:

- Personalized recommendation algorithm based on user behavior, quick search by name, author, genre, and integration of Google book sources to improve user experience when searching for products that are not available,
- Real-time inventory tracking and alerts,
- Automatic sales reports by day, week, and month.

By combining intelligence and automation, the system improves customer satisfaction and optimizes store operations.

1.3. Scope and Objectives

The scope of this project is to develop a smart web application for small and medium bookstores. It solves the challenges that businesses face such as inventory management, personalized recommendations, and interacting with customers,... This application is designed for a specific market of book sales, mainly for bookstores with a moderate amount of books and orders. This website will be built to be easily customized to suit each seller and has plenty of room for future expansion. In terms of functionality, the system will automate basic functions such as real-time inventory management, order management, and sales status. The system will also focus on the user interface so that the website can operate easily and navigate smoothly for both buyers and sellers, ensuring customer reach and ease of use. Although the initial scope is bookstores, the website can still be expanded and changed to accommodate other shopping platforms in the future.

The main objective of this thesis is to develop a smart, automatic, and user-friendly website with essential functionalities and expandable features, capable of integrating advanced capabilities like artificial intelligence, machine learning, and data analysis. Additionally, the project aims to meet several specific objectives, including creating a seamless user interface, implementing smart inventory management, offering personalized recommendations, and providing sales analytics and third-party system integration.

1.3.1. User Interface

The system can provide a simple, intuitive, and attractive interface that works seamlessly across different devices and platforms.

The system can guide users efficiently through key processes, such as product browsing and order management, ensuring smooth user interaction.

1.3.2. Smart Inventory Management System

The system can ensure inventory balanced by automatically updating stock in real-time for all transactions.

In detail, when customers decide to buy products and completely order them, the database will call a function to update the quantity of these products immediately and notify the seller about it. If the quantity of that product is below 20, the system will warn of out-of-stock risk and encourage restocking. In addition, if the quantity of products in stock is less than the total quantity of products added to the cart, the system will issue a warning.

On the other hand, after a day, if there are any products that are over 100, the system will remind that the owner should not import goods.

The system can prevent out-of-stock or overstock issues by monitoring and reflecting inventory changes promptly.

1.3.3. Automatic report system

The system can auto-generate the report about revenue after a day, a week, or a month and send it to the owner's email as well as update it continuously on the Revenue page.

Collect data and compile into tables such as Hot products (based on the number of searches and clicks on the product's detail page), Top sellers (top 10 most sold products) and report to the store owner.

1.3.4. Personalized Recommendation Function

The system can analyze customer preferences and behavior to provide tailored product recommendations based on browsing history, previous orders, and products in the cart,

When the customers buy, and search for products, the system will store their attributes in the database, then find the other products with related attributes and display them on the customer's homepage.

The system also recommends products that are best-sellers, and hot items which collected daily to stimulate shopping.

1.3.5. Search and Filtering Function

The system can allow users to quickly search products by name, keyword, or category and filter them based on specific criteria such as price or rating. The system also has suggestions from Google Books API to display when users search for products, not in the system's data.

The system can provide related products to encourage and suggest purchases. If the user goes to a product's detail page, the system will collect keywords (tags, type, name,

categories,...) of a product and find others in the database to match with similar data, then show in Related Component in this page.

1.4. Assumption and Solution

The assumption in this thesis is that the construction of a smart web application for an online bookstore will target individual users, schools, and large libraries, providing them with an easy, convenient, and personalized shopping experience. The application will also assume that users do not have much time to research books in depth but need to buy quickly and get accurate advice based on their personal preferences and search history. In addition, the management of book inventory, customer information, and online payment are also seamlessly integrated to optimize the operational process for both the bookstore and the customer.

The proposed solution in the thesis is to build a smart web application, using modern technologies such as NextJs/ReactJS, NodeJS, and MySQL. The system will develop recommendation algorithms based on users' search and shopping history to give exact recommendations. In addition, It also has a rating system and review features, helping users make smarter purchasing decisions. To improve the user experience, the application will have an advanced search system with detailed filters, helping users search for books based on many criteria such as author name, genre, year of publication, or reviews from other users. Furthermore, the book warehouse management system will be optimized, helping managers easily track inventory status, and orders, and manage customer information effectively. The solution does not stop at building an online sales system but also optimizes personalized experiences, and improves user interaction through smart suggestions and effective management capabilities for bookstores.

1.5. Structure of thesis

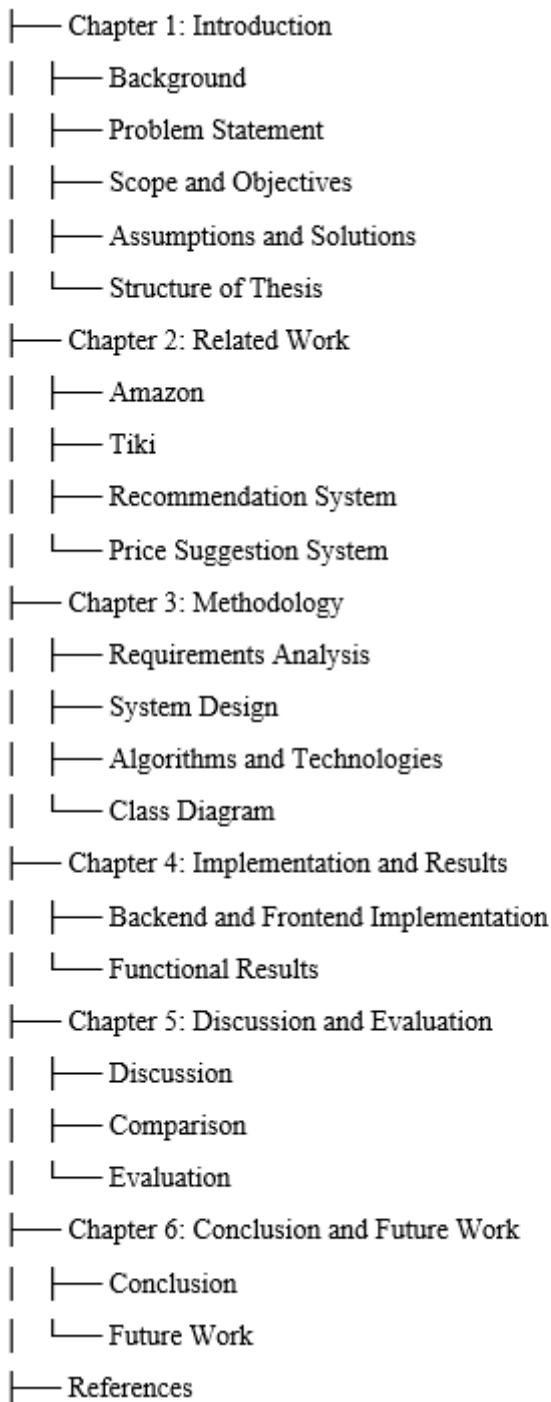


Figure 1.5. Report Structure

The structure of this thesis is illustrated in Figure 1.5, providing a visual overview of the organization of the report. The thesis is divided into six main chapters, each serving a distinct purpose in presenting the research, development, and evaluation of the proposed smart web application for bookstore management.

Chapter 1 – Introduction: This chapter introduces the background and motivation behind the project. It identifies real-world problems in both traditional and online bookstores and states the objectives, scope, assumptions, and proposed solution. The chapter also explains the importance of building an intelligent system to improve user experience and optimize store operations. It sets the foundation for understanding the rationale of the proposed solution.

Chapter 2 – Literature Review (Related Work): This chapter reviews existing platforms and studies relevant to the topic. It analyzes applications such as Amazon and Tiki, focusing on their key features, strengths, and limitations. It also explores recommendation systems in e-commerce. The goal is to identify gaps in current solutions and justify the need for a smarter approach.

Chapter 3 – Methodology: This chapter details the system design and development methodology. It includes user requirement analysis, functional and non-functional requirements, system architecture, and algorithms. Diagrams such as use case, ERD, and sequence diagrams are used to illustrate system logic and interactions. Technologies, programming languages, and tools used (e.g., Next.js, Node.js, MySQL) are also described.

Chapter 4 – Implementation and Results: This chapter describes how the proposed system was developed and deployed. It outlines configuration details, libraries used, and the steps of implementation. Screenshots are provided to demonstrate the main features of the application, such as product management, recommendation system, and automated reporting.

Chapter 5 – Discussion and Evaluation: This chapter provides a critical analysis of the implemented system. It compares the proposed solution with other systems reviewed in Chapter 2 and evaluates how effectively the objectives were achieved. Strengths, limitations, and areas for improvement are discussed.

Chapter 6 – Conclusion and Future Work: The final chapter summarizes the main contributions of the project. It also suggests possible future enhancements, including scalability, feature expansion, and integration with more advanced technologies such as machine learning models for smarter recommendations.

Each chapter builds on the previous one to create a complete, coherent presentation of the research and development process, ensuring both technical depth and practical relevance.

CHAPTER 2

RELATED WORK

This chapter provides a comprehensive review of existing studies, applications, and commercial platforms related to smart bookstore systems and e-commerce management solutions. The aim is to analyze current technologies and identify the strengths, limitations, and applicability of these systems in the context of building an intelligent, automated bookstore web application.

2.1. Amazon Platform

Amazon stands out as the biggest online e-commerce website; its comprehensive online bookstore has revolutionized how people can buy books. From its humble beginnings as an online bookseller, Amazon grew to develop the capacity for using state-of-the-art technology to enhance its online shopping experience. This e-commerce platform succeeded with a strong recommendation system, flawless inventory management, and a very user-friendly interface with a range of features for buyers and sellers.

2.1.1. Key Functionalities

- Recommendation System

Personalization system: Amazon's recommendation system is probably advanced. It uses a hybrid of collaborative filtering, content-based filtering, and machine-learning algorithms that suggest books depending on the user's last purchases and browsing history. That is one of the major success factors of Amazon; increasing customer interest will drive more sales.⁽¹⁾

“Customers who bought this product also bought”: This feature, based on collaborative filtering, recommends books that are often purchased together, providing customers with relevant recommendations in real-time.⁽²⁾

- User Interface and User Experience

Advanced Search and Filter: Amazon's search functionality is highly advanced, allowing users to search for books by title, author, genre, publication date, language, and customer reviews. Additionally, Amazon's filtering system lets customers refine searches by price range, publication date, and even whether a book is available in physical or digital format.

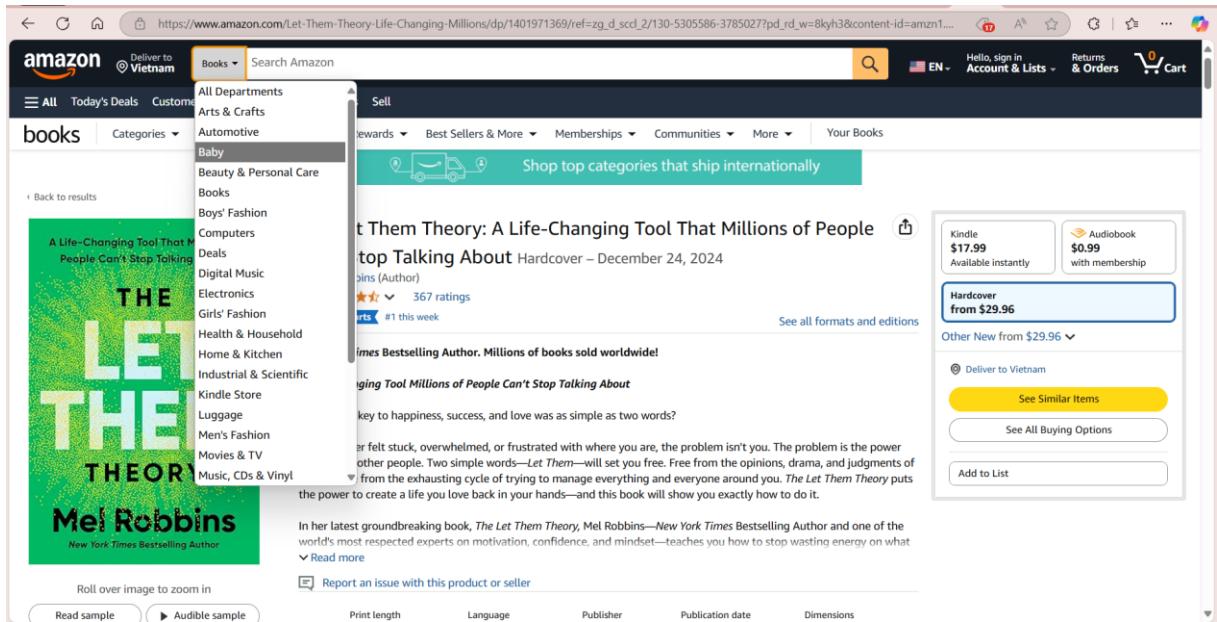


Figure 2.1.1. Amazon search function with categories

User Reviews and Ratings: Amazon's review system provides intuitive reviews from users who have experienced the product, helping customers make informed decisions. These reviews are integrated into the recommendation system, further improving the relevance of product recommendations.

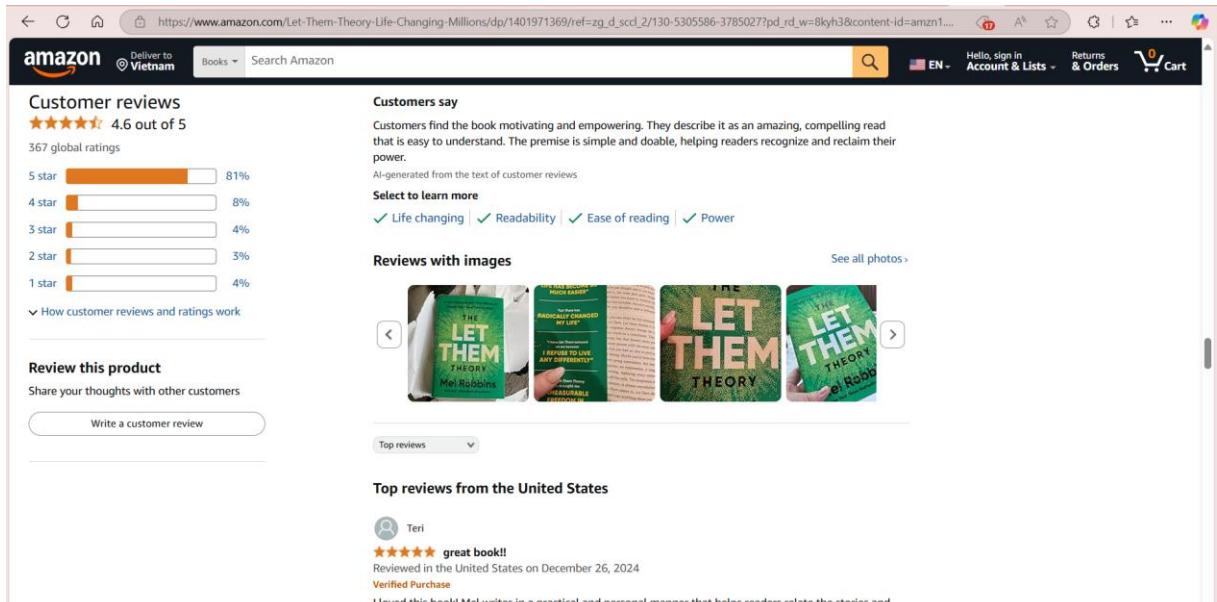


Figure 2.1.1. Rating in Amazon

2.1.2. Strength and weakness

2.1.2.1. Strength

- **Highly Advanced Recommendation System**

Amazon's ability to provide personalized recommendations based on its vast amount of user data gives it an edge over other e-commerce companies. This system has contributed significantly to higher sales rates.⁽²⁾

- Personalized User Experience⁽³⁾:

Amazon uses a relatively sophisticated recommendation system that employs machine learning algorithms for statistical analysis of large-scale data sets. This system allows to offer customers an extremely personalized shopping experience by providing product recommendations based on various factors like their past transactions or searching history.

Behavioral data on users - it includes data on purchasing preferences, the history of products they have viewed on the platform, and what is currently on their shopping list, all being part of the data collected. It ensures that the customer is satisfied and therefore improves their shopping experience overall.

- Data-driven insights:

The system creates a lot of data including user information, product attributes, and contextual information like time and location. For example, through conducting an extensive analysis of such information, Amazon can notice trends and patterns that are valuable for its recommendation program.

Big data analytics allows Amazon to dynamically update its recommendations in real time so that users get the most relevant suggestions as their preferences change.

- Boost sales conversions⁽⁴⁾:

Amazon's personalized recommendations boost sales conversions. Research indicates a large percentage of Amazon's revenues are generated through its recommendation engine, proof of its efficacy in driving purchases. By introducing to users products that closely match their interests, Amazon encourages impulse purchases and increases the likelihood of completing a transaction, leading to higher revenue.

- Increased customer interaction:

Recommendation systems drive deeper customer engagement by making the shopping experience more interactive. Features like "Customers who bought this item also bought" and "Recommended for you" help users actively explore the platform.

This engagement not only enhances customer satisfaction but also builds brand loyalty, where customers are likely to come back for the next purchase from Amazon.

- Adaptive learning and continuous improvement:

The recommendation algorithms of Amazon learn continuously from interactions by users and from feedback, which refines the system for making better recommendations over time. The adaptability of the learning feature ensures that the recommendations stay relevant and effective with the changes in market trends and consumer preferences.

It uses the feedback from the users in its models in order to improve the accuracy and effectiveness of the recommendations, making shopping more satisfying.

- Competitive advantage:

Amazon's advanced recommendation system gives it a strong competitive advantage in the e-commerce space. Many retailers can't even come close to personalization, yet Amazon sets a high bar for them.

This advantage not only helps retain existing customers but also attracts new ones, further solidifying Amazon's leadership position in the industry.

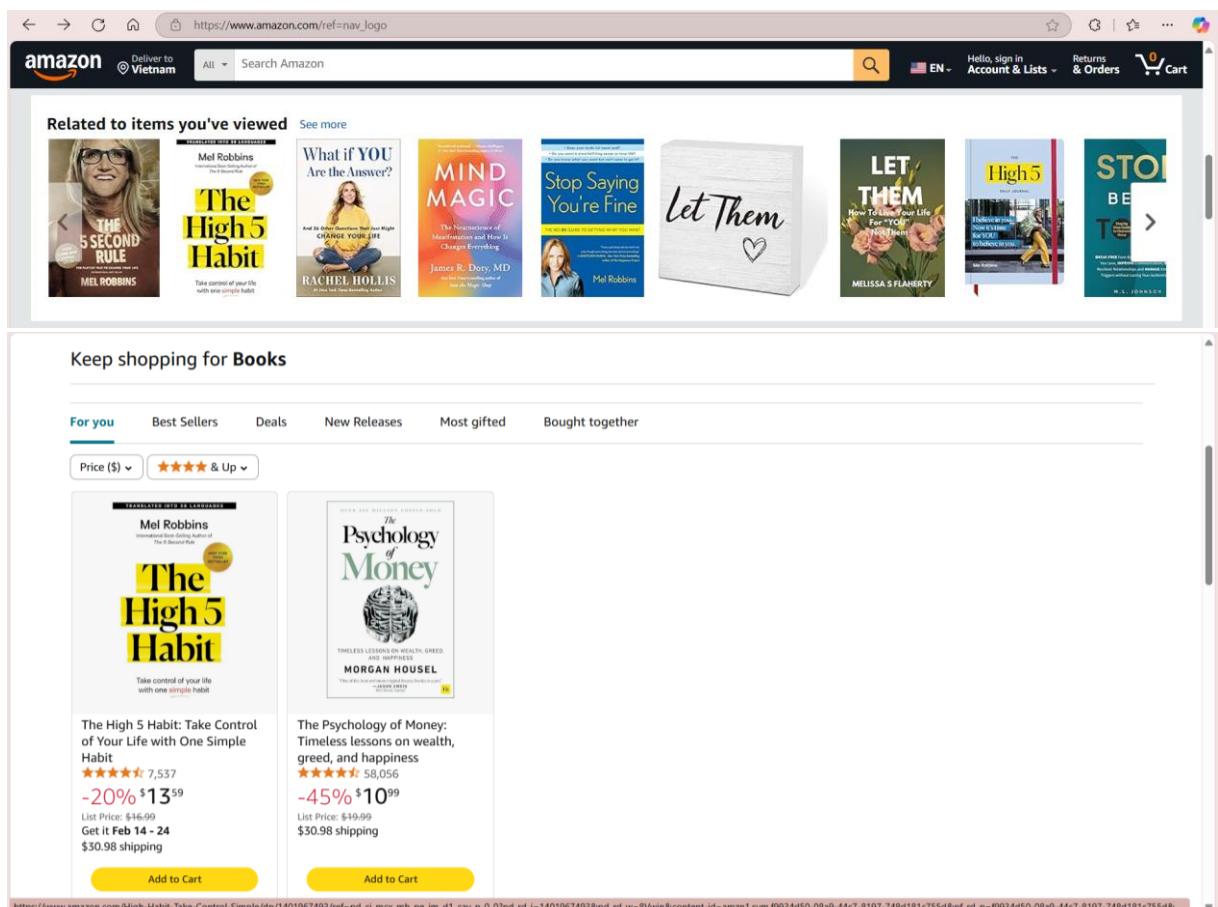


Figure 2.1.2. Recommendation system in Amazon

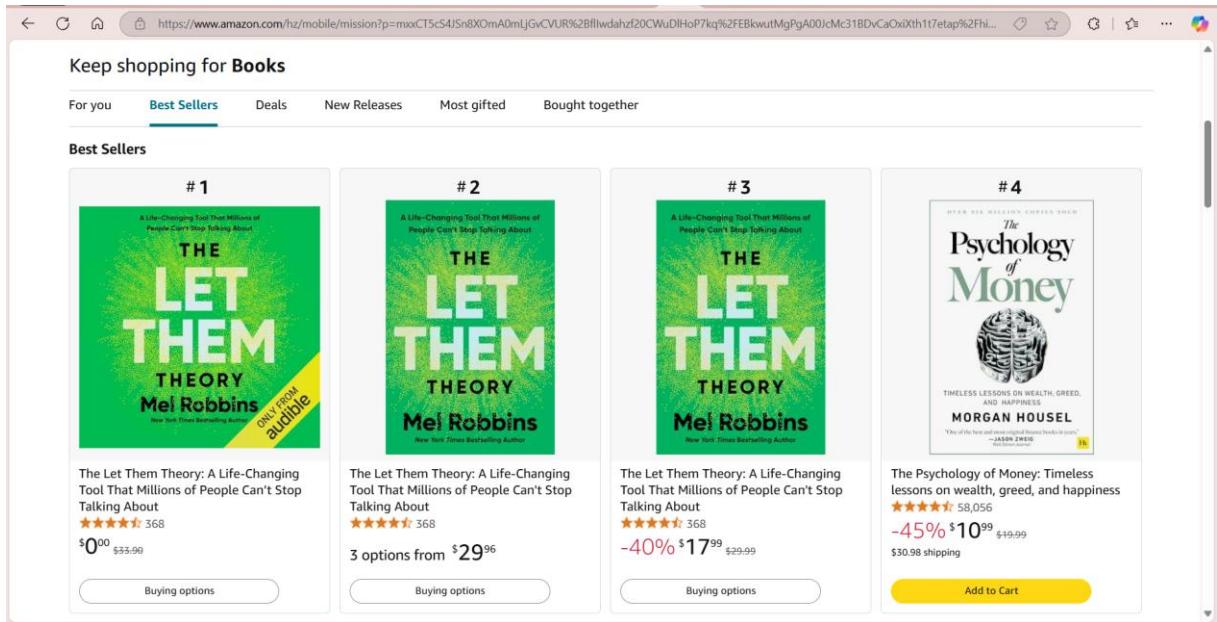


Figure 2.1.2. Bestseller Recommend in Amazon

- Efficiency in Inventory Management

Amazon has a competitive edge because of its sophisticated inventory management techniques, which contribute to enhancing effectiveness and competition in the e-commerce market. E-commerce companies have to deal with high competition and other challenges such as maintaining inventory usage during the production supply chain. This practice reduces storage expenditures and prevents surplus volume, which in turn guards the earnings.⁽⁵⁾

As a result, Amazon can maintain a negligible margin of error related to any given supply chain operation. Thanks to the emergence of this economic tool, it is necessary to adjust the volume of the inventory level due to the trends to ensure that the market shall neither experience excess nor partial supply. As an illustration, Amazon fasts its lead time in the event of selling towards special seasons such as holiday seasons by increasing the number of stocks sold.

Additionally, Amazon uses a decentralized distribution model, locating fulfillment centers in different sectors. This network ensures faster delivery times, allowing services such as same-day or next-day delivery. That not only improves customer satisfaction but also make Amazon become market leadership. The company continues to enhance this system through real-time inventory tracking and advanced automation, streamlining operations while maintaining a competitive edge.

Furthermore, Amazon's integration of robotics and automation into its warehouses has significantly reduced order fulfillment times. This technology improves operational accuracy, allowing Amazon to deliver competitive performance while maintaining a fast and reliable

delivery protocol. In short, Amazon's advanced management strategies, starting with JIT management, play a key role in operational efficiency, cost effectiveness, and customer satisfaction. The ability to quickly respond to market changes and use technology ensures the company's continued value in the e-commerce landscape.⁽⁵⁾

- **Strong Brand Recognition and Reputation**

Amazon is a brand widely known all over the world. Amazon has strong brand equity, everyone can recognize it by a logo, name. Amazon is a brand that profited from trust from the customer. This made possible thanks to company's customer service over decades, great customer policies, and intriguing ad campaigns which are why Amazon has become a brand name that is associated with reliability and trust in consumer eyes.⁽⁶⁾

The importance of brand recognition in e-commerce cannot be overstated. Consumers have a tendency to trust popular brands when they have to make a purchase decision in a crowded environment. Amazon has become the most trusted brand. Amazon's approach to customer satisfaction is showcased by its delivery options, which are not only cost-effective and fast but also easy to interact with.⁽⁶⁾

- **Diverse Product Offerings**

One of Amazon's key advantages is its vast product range. Customers can find almost anything, from electronics and books to clothing and household goods, on a single platform. This diversity appeals to a wide range of customers with varying needs and preferences, helping Amazon capture a significant market share.

Amazon's partnerships with manufacturers and businesses further expand its product catalog. Its ability to respond to shifting consumer trends, using advanced data analytics, ensures that it continues to offer relevant products, contributing to the company's long-term success.

- **Wide Range of Services**

Beyond e-commerce, Amazon's services extend to cloud computing (AWS), streaming media (Amazon Prime), and more. AWS, in particular, has become a market leader in cloud infrastructure, contributing significantly to Amazon's diversification beyond retail. Amazon Prime's membership model enhances customer loyalty while adding a steady revenue stream through subscriptions.

- **Strong Partnerships and Collaborations**

Amazon's ability to collaborate with other companies—be it in logistics, content creation, or technology—has amplified its success. Strategic partnerships allow Amazon to offer a broader range of products and services while improving operational efficiency.⁽⁶⁾

2.1.2.2. Weakness

- **Dependence on Third-Party Sellers**

While third-party sellers significantly contribute to Amazon's extensive product range, they also pose a challenge. The company has limited control over the quality and delivery of products from these sellers, which can result in fraud, counterfeit goods, or poor customer experiences. Negative experiences with third-party products can damage Amazon's reputation.⁽⁶⁾

- **Data Security Concerns**

Amazon's huge repository of customer data becomes a target for cyber-attacks. All the security measures notwithstanding, cybersecurity threats are a major concern. A significant data breach would cause a loss of customer trust and major legal and financial consequences.⁽⁷⁾

Such a huge data collection by the recommendation system does give rise to the concern of users' privacy, especially under the tightening regulations of data protection like GDPR.⁽⁷⁾

2.2. Tiki

Tiki is one of the first e-commerce platforms in Vietnam, established in 2010. Started as an online bookstore, Tiki has expanded to include electronics, household goods, fashion, and more. The platform is known for customer-centricity, efficiency in logistics, and a strong marketplace connecting buyers and sellers across the country.

2.2.1. Key functionalities

- **Comprehensive Product Categories**

Tiki provides a wide array of products, ranging from books and electronics to household items and fashion. This extensive selection allows customers to find virtually everything they need in one place, making it a convenient shopping destination.

- **User-Friendly Interface**

The Tiki website and mobile app are designed for easy navigation. Users can quickly search for products, filter results by category or price, and access detailed product descriptions and customer reviews. This streamlined interface enhances the overall shopping experience.

- **Secure Payment Options**

Tiki offers various secure payment methods, including credit/debit cards, bank transfers, and cash on delivery (COD). This flexibility caters to different consumer preferences and enhances trust in the platform.

- Customer Reviews and Ratings

Tiki enables customers to leave reviews and ratings for products they purchase. This feature helps potential buyers make informed decisions based on the experiences of others and fosters a sense of community on the platform.

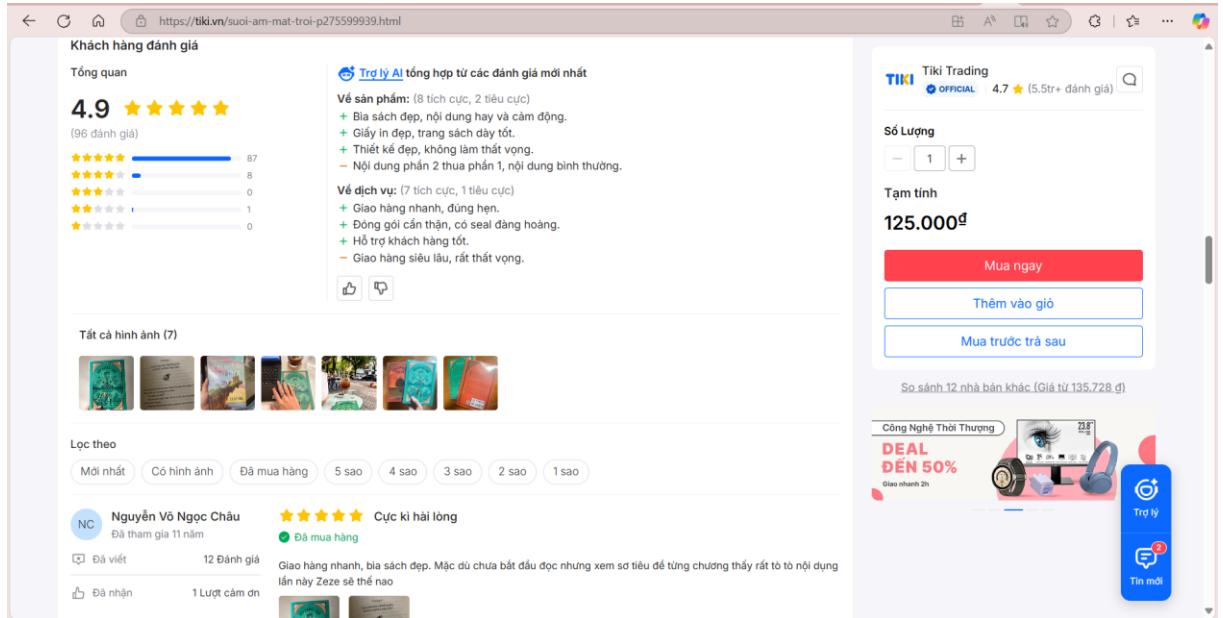


Figure 2.2.1. Tiki's Reviews

- Customer Support

Tiki provides various customer support options, including a dedicated help center and chat support, to assist users with their inquiries and issues. This commitment to customer service is critical in building trust and loyalty among users.

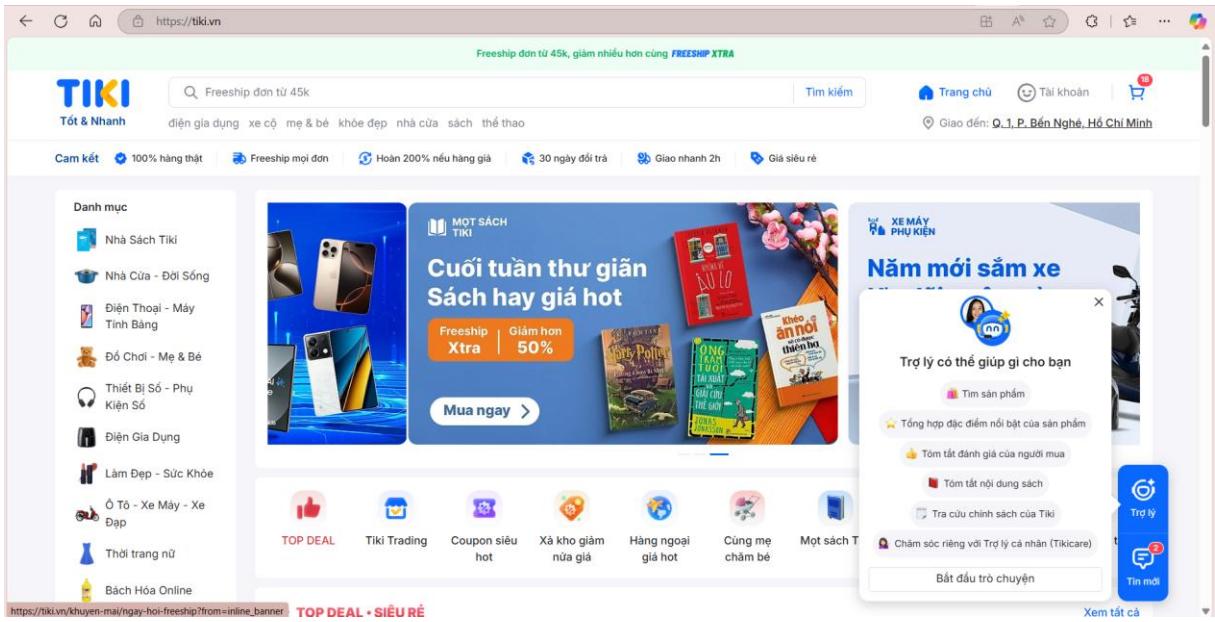


Figure 2.2.1. Tiki's Chatbox

2.2.2. Strength

Recommendation system

Tiki uses a sophisticated recommendation system to personalize the shopping experience for users, using the following algorithms⁽⁸⁾

- Nearest Neighbor Search (ANN): To improve the retrieval time in its recommendation engine, Tiki specifically uses Annoy, to efficiently search for nearest neighbors in a high-dimensional space.
- Matrix Analysis: This technique is applied to analyze the interaction matrix between users and products, making it easier to discover hidden factors that influence purchasing behavior.

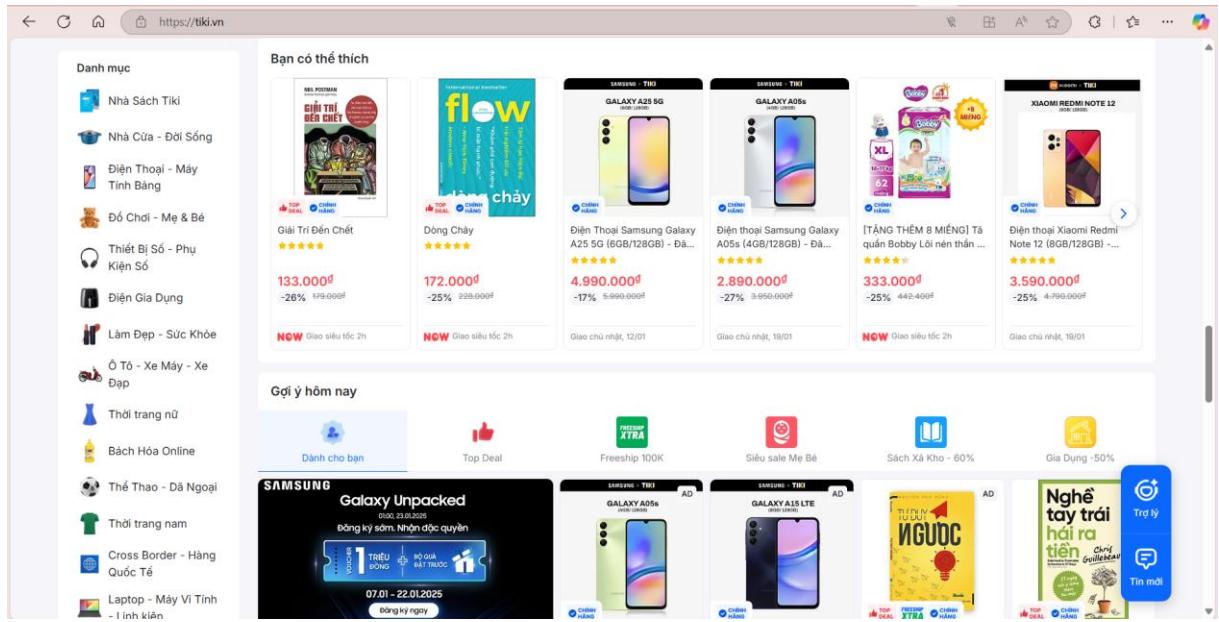


Figure 2.2.2. Tiki's Recommendation

User-interface

Tiki uses user-centric design to ensure an intuitive and responsive interface:

- Tini UI Design System: Tiki has further developed the Tini UI Design System to standardize the user interface, enhance consistency, and improve user experience on all its platforms.
- Bilingual Interface: Tiki caters to the customers of other countries through its bilingual interface in Vietnamese and English.

Security

Security of transactions and protection of data matters a lot for Tiki.⁽⁹⁾

- Data encryption and secure transactions: Tiki provides strong security for users' data and ensures transactions are secured, following all industry standards for e-commerce platforms.
- Compliance with security standards: Tiki follows the best practices of web application security to prevent possible threats and vulnerabilities.

2.3. Recommendation System

A recommendation system is an important part of machine learning algorithms that predict user preferences based on historical data, enabling platforms to provide personalized suggestions. Those systems have been used across industries and especially e-commerce, streaming, and social media. Their main task is to improve and enhance user experience, interaction, and revenue.

- Content-Based Filtering: The goal of this approach is to recommend items that are similar to those that a user has previously interacted with. This approach analyzes the features of the item and the user's preferences, such as the product description, category, or rating. For example, users who buy a thriller might receive recommendations for other thriller books.
- Collaborative Filtering: This method relies on the preferences of other users with similar behavior to suggest products that the user might like. For example, Amazon's "Customers who bought this item also bought" feature uses collaborative filtering to recommend products that customers often buy together.
- Hybrid Systems: Hybrid systems combine the advantages of both content-based filtering and collaborative filtering. For example, Netflix uses hybrid recommendation models when recommending TV shows and movies to watch, based on a user's viewing history and ratings.

Implement

This system is widely used in e-commerce sites to boost sales and customer product approval.

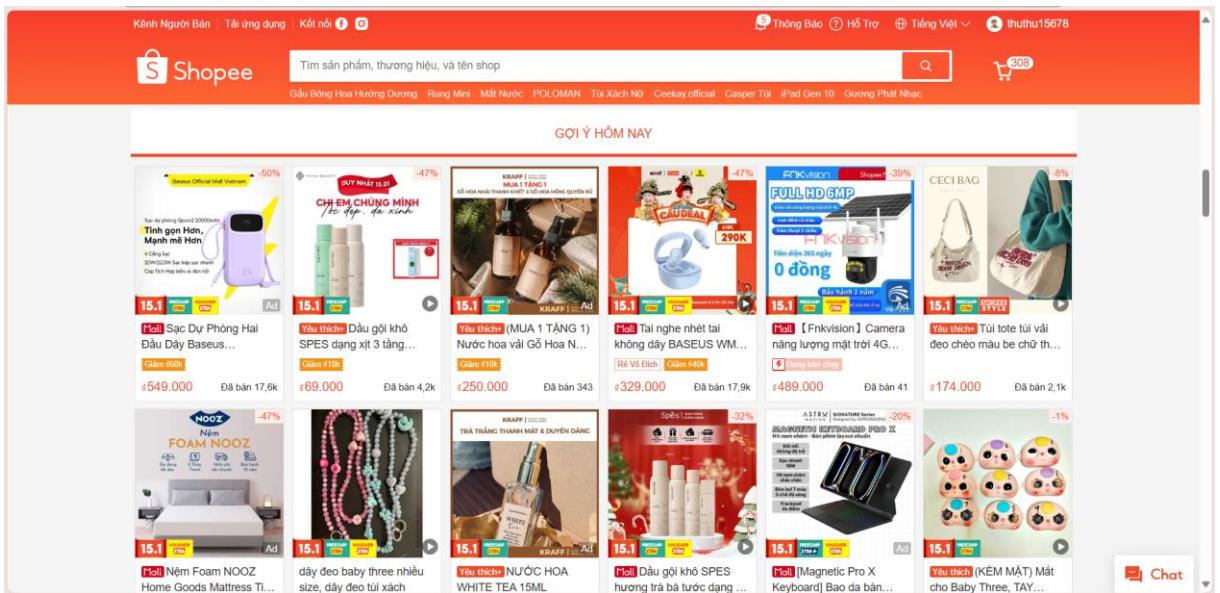


Figure 2.3. Shopee's Recommendation

In Shopee, the number of times a user views a product will be recorded and the system will automatically suggest similar products directly to the homepage. When a user views a product, the system will immediately update the homepage.

Similarly, this system is also applied to entertainment sites such as Youtube or Nhaccuatui. The system will also record the videos that have been watched and based on that,

recommend similar videos to the homepage. For example, when users watch videos about studying Ielts, the homepage will be flooded with videos with similar topics.

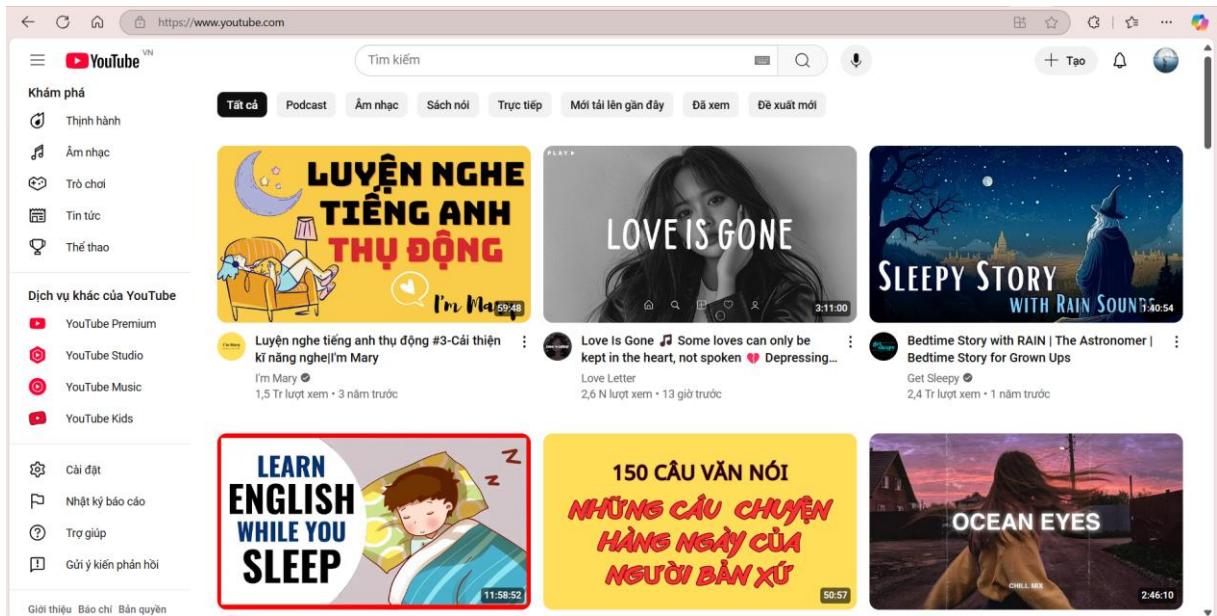


Figure 2.3. Youtube's Recommendation

With social networking sites like Meta or Tiktok, the system will use previous user interactions such as dwell time, likes, and shares to push posts with similar content and groups. Based on interactions between users, the system can suggest people you may know.

2.4. Price Suggestion System

2.4.1. Overview

In today's business, flexible product pricing is key to optimizing revenue and profit. Selling price is no longer a fixed factor, but instead needs to be adjusted based on consumer behavior and market fluctuations.

2.4.2. Theoretical basis

To build the Price Suggestion system, the project has applied many important principles in modern revenue management and marketing. One of the core theories used is Revenue Management. This is a selling price management strategy to optimize revenue based on the analysis of customer behavior data, market demand and supply capacity. In the context of bookstores, the main factors considered are the conversion rate from visits to purchases and the change in revenue over time.

Specifically, the conversion rate is calculated by dividing the number of products sold in a week by the number of users who clicked on the product details. This is an important

indicator that reflects the attractiveness and effectiveness of the current selling price. In addition, the change in revenue between this week and the previous week (Revenue Difference) helps to assess purchasing trends, thereby making appropriate decisions about price adjustments.

An add-on strategy for Revenue Management is Demand-Based Pricing. This method adjusts the selling price based on the willingness of consumers to pay and the actual market situation. Large retail corporations such as Amazon, Walmart or Shopee are all using on-demand pricing to increase competitiveness and optimize profits. In particular, Amazon is recorded to change the prices of millions of products every day based on an algorithm that assesses customer needs and behavior.

In addition, consumer psychology is also considered through the Psychological Pricing strategy. For example, the price of 99,000 VND is often perceived by consumers as "cheaper" than 100,000 VND, even though the difference is not significant. This is a factor that can further improve the price recommendation algorithm in the next phase of the project.

2.4.3. Actual data

According to a report from Vietnam Report in 2024, book industry revenue in Vietnam has grown by 15% compared to the previous year, showing that reading demand is recovering strongly after the pandemic. The Edge by Ascential report also found that large retail platforms that adopt dynamic pricing can increase revenue by an average of 8 to 12%. At the same time, Nielsen Vietnam's survey in 2023 shows that 47% of consumers are directly affected by changes in book prices when making purchasing decisions.

These data not only reinforce the efficiency of the system, but also clarify the urgency of adopting a flexible pricing model in the book retail industry.

CHAPTER 3

METHODOLOGY

This chapter describes the methodology used to develop the smart automatic bookstore. It covers the analysis of user requirements, system design, and the technologies and algorithms used throughout the development process. Key tools include NextJS/ReactJS for the frontend, Node.js for backend development, MySQL for database management, and Git for version control. Additionally, various algorithms, including collaborative filtering and content-based filtering, support personalized recommendations, while Chart.js enables effective data visualization.

3.1. User requirement analysis

This section captures the functional and non-functional requirements of the system to ensure it meets user expectations, including customers, store owners, and administrators.

- Customers: Individuals browsing and purchasing products. They expect easy search, comparison, and personalized product recommendations.
- Store Owners: They need tools for inventory management, order tracking, and sales analytics.

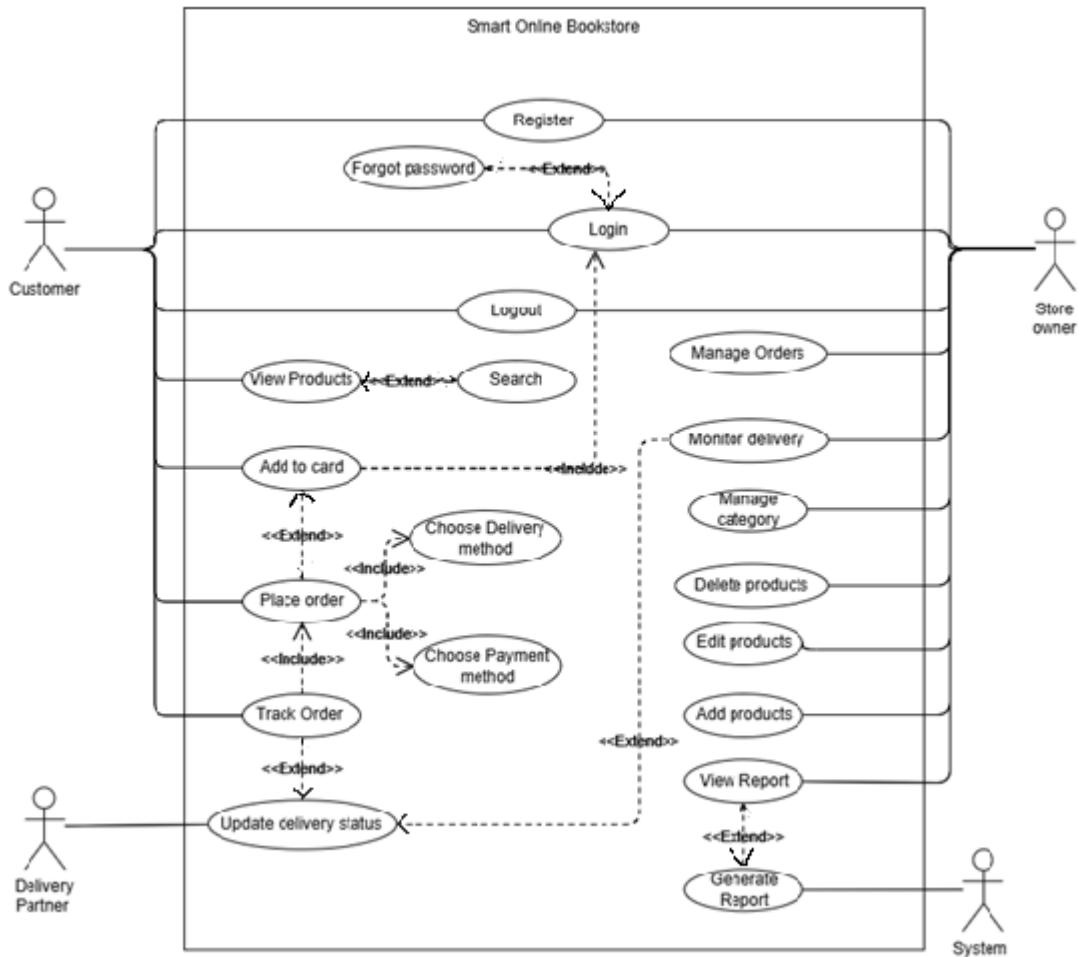


Figure 3.1. Use case diagram

3.1.1. Functional requirements

The intended smart online bookstore system shall provide a frustration-free and feature-laden shopping experience to customers, as well as full management facilities for store owners and administrators. The system incorporates major e-commerce features, along with smart features such as personalized recommendations, AI-powered assistance, and real-time analytics.

Customer Side Functionalities

The system provides functionality to customers to login, register, search, filter, compare, and purchase products effortlessly. When the user is redirected to the login page, he/she is requested to enter their password and email address. The login credentials are verified by decrypting the stored password and comparing the login credentials. When the credentials are invalid, an error message is displayed; otherwise, the session of the user is stored securely

through JSON Web Tokens (JWT), and the user is redirected to the home page if he/she has the "Customer" role.

The registration process is straightforward, whereby users are asked to provide their email and password. The system checks for duplicate emails—if the email already exists, the user is sent to the login page with an appropriate alert. Otherwise, the newly created account is added to the database, followed by success message and homepage redirection.

An offered search field at the top of the site enables customers to search for products by keywords, categories, or product names. Search queries are fetched from the database and displayed on a dedicated results page. When no products match the query, a courteous message is shown. For enhanced usability, search suggestions and autocompletion are provided.

On the Products page, customers have the ability to filter products by means of a dropdown menu. Filters allow users to sort products based on price bracket, customer ratings, category, and availability. Dynamic update of the list of products occurs based on the filters applied.

Side-by-side comparisons of products are also available from product detail pages. Users can compare up to three products side by side when they click on a "Compare" button. The comparison table highlights such important features as price, specifications, and user reviews to support making an informed decision.

The cart functionality allows users to add items from the product page of any product. The system keeps a record of items selected with respective quantities and prices in the database. Users can access their cart at any time via a cart icon in the header, where they can alter quantities or remove items. All changes are automatically synchronized in the database to ensure data consistency.

For the process of buying, users proceed to checkout from their cart. The system supports integration with secure payment gateways and shipping companies. Upon successful payment, the system logs an entry of the order into the database, updates stock levels, and confirms the order both via email and on-screen. The process of checkout is ensured to be secure, easy to use, and rapid.

Store Owner Side Functionalities

The owners of the store are given a special login page where they enter with a username and password. The process is identical to customers as far as security and validation are

concerned. Upon successful login, if the role of the user is "Owner", they get redirected to the management interface.

Product owners are able to view all products, update product details (e.g., stock availability, quantity, name), delete unwanted products, or add new ones via the Product Management interface. This interface provides full control over the product list in an intuitive format.

The Order Tracking page lists customer orders in date order. Owners can filter orders by status of completion and sort by price total. Orders can be canceled by the system, and a customer notification is also sent. One order can be expanded to view detailed information like items ordered, prices, and customer information.

Optional manager accounts are added by the store managers by setting email and password in a particular format. These accounts are stored with the "Manager" role and might be useful for day-to-day operations.

In the Reporting Dashboard, owners can monitor performance indicators such as best-selling products, and revenue trends daily, weekly, or monthly. This page is a powerful decision-supporting page, utilized to maximize business strategy. The system also generates a report per day and emails it to the owner's account.

3.1.2. Non-Functional requirements

A smart online bookstore system needs to meet a series of important non-functional requirements to ensure the optimum user experience, acceptable performance and data safety. First, the website interface needs to be fully responsive, adaptable and display well on different devices such as desktops, tablets and mobiles, and operate steadily on current mainstream browsers. From the performance perspective, the system must ensure fast and instant page loading time, simple navigation and support simultaneous access by many users without causing a decline in the overall performance. Security is one key area, wherein customer data and transactions must be encrypted based on the latest encryption standards. The system should be able to provide secure storage and processing of all sensitive data like passwords and payment information, thus making it more reliable and keeping users' privacy intact.

3.2. System Design

A This section outlines the technical structure of the system, including the database and user interface design, as well as the algorithms and technologies used for backend processes.

3.2.1. Datadase design

The database is built using MySQL, which ensures efficient data storage and retrieval. It follows normalization rules to prevent redundancy and maintain data integrity.

Table Relationship diagram:

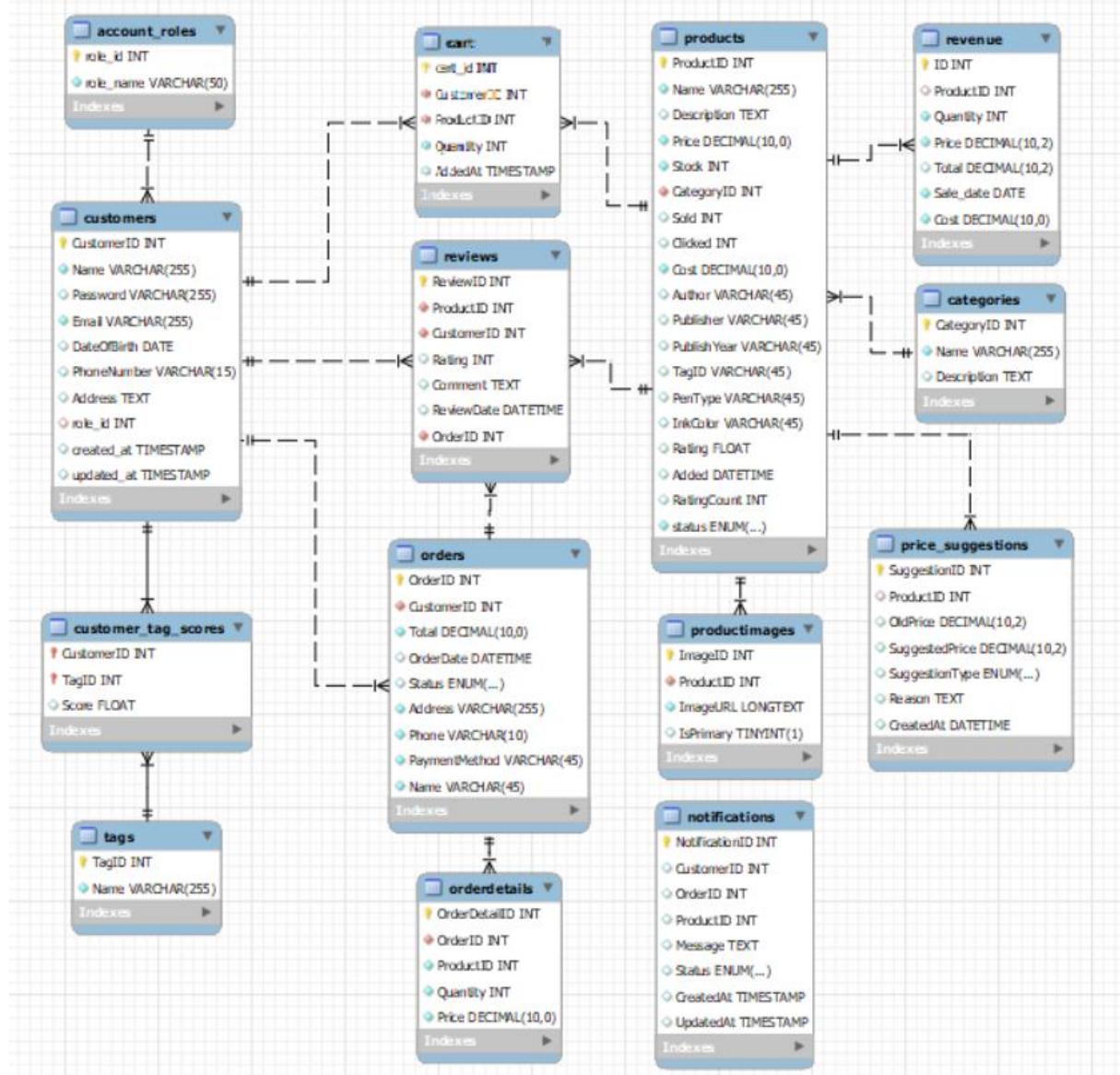


Figure 3.3.1. Table Relationship diagram

Database description:

The schema of the database illustrates the structure and relationships of an intelligent online bookstore system. At the center of the schema is the customer's table that stores user information such as name, password, email, phone number, address, and date of birth. It also has a foreign key **role_id** that references the **account_roles** table, which defines user roles. Each customer is able to talk to the system by placing products in their basket, placing orders,

posting reviews, receiving notifications, and suggesting price changes through the price_suggestions table.

The products table contains detailed information for every product, including name, description, price, author, publisher, and category (CategoryID). It also tracks performance measures such as the number of times a product has been sold or clicked. Products can be associated with tags (TagID) for analysis and recommendation. The interaction between the customer and the tag is kept in the customer_tag_scores table, which contains a score indicating the interest of the user in a tag.

Orders are handled by the orders table, which is related to the customer and includes information such as total amount, order date, status, delivery address, and payment method. Each order may have multiple entries in the orderdetails table, including products purchased, quantity, and prices at the order time. Daily revenue is also tracked by the system in the revenue table, including total quantity, total price, and cost.

User-generated content is also reflected in the reviews table, where customers are able to comment and rate products that they have purchased. A review is associated with a specific product and order, along with containing the date of the review. The productImages table allows for multiple images per product, with a flag to designate one image as the primary display.

To enhance user experience, the notifications table stores account or product notifications and their read status. The price_suggestions table, however, enables customers to give feedback on product prices, including the old price, new suggested price, reason for the suggestion, and type of suggestion.

Product categorization is managed by the categories table, and tags are kept in the tags table. The schema in general is nicely normalized, with neat relationships among tables, to enable good query performance, sophisticated behavior analysis, personalized recommendations, and future system scalability.

3.2.2. Sequence Diagram

The authentication flow begins when the user enters credentials (email and password) into the login form. After submitting the form, the client sends a request to the server with the provided credentials. On the server side, the system first checks if the submitted email exists in the database. If the account is found, the server verifies the password by comparing it with a stored hashed version. If the credentials are valid, the server generates a JWT authentication token and sends it back to the client. If the credentials are invalid, the server reports an error. After successful authentication, the client stores the token for session management. If the user wants to log out, the system deletes the token after receiving the request.

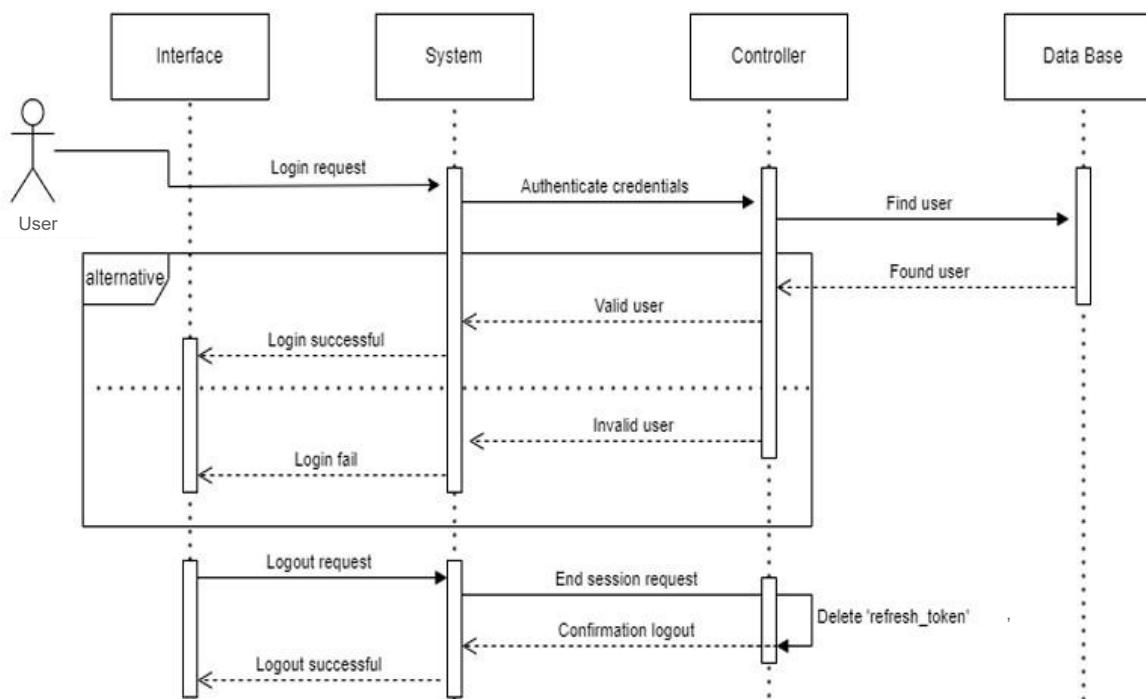


Figure 3.3.2. Authentication and Update sequence diagram

If customer want to interact with a product, they can starts by clicking on a product, the client sends a request and the server accesses the database and retrieves the product corresponding to the request, returning it to the client to display the desired product. When deciding to add a product to the cart, the customer clicks the "Add to cart" button on the product page. The client requests the server to add the product to the cart, along with product information and quantity. The server processes the request, updates the user's cart, and returns a successful response. After the product is added to the cart, the user can go to the cart page to check the selected products. When entering the cart, the client requests the server to get a list of current products in the user's cart. The server returns a list of products, including information about name, price, and quantity. If the user wants to change the amount of a product in the cart, they can change the quantity on the user interface. The client sends a request to update the product quantity to the server, and the server updates the cart accordingly. After updating, the server returns new cart information, including quantity and total value, allowing users to review their cart before checking out.

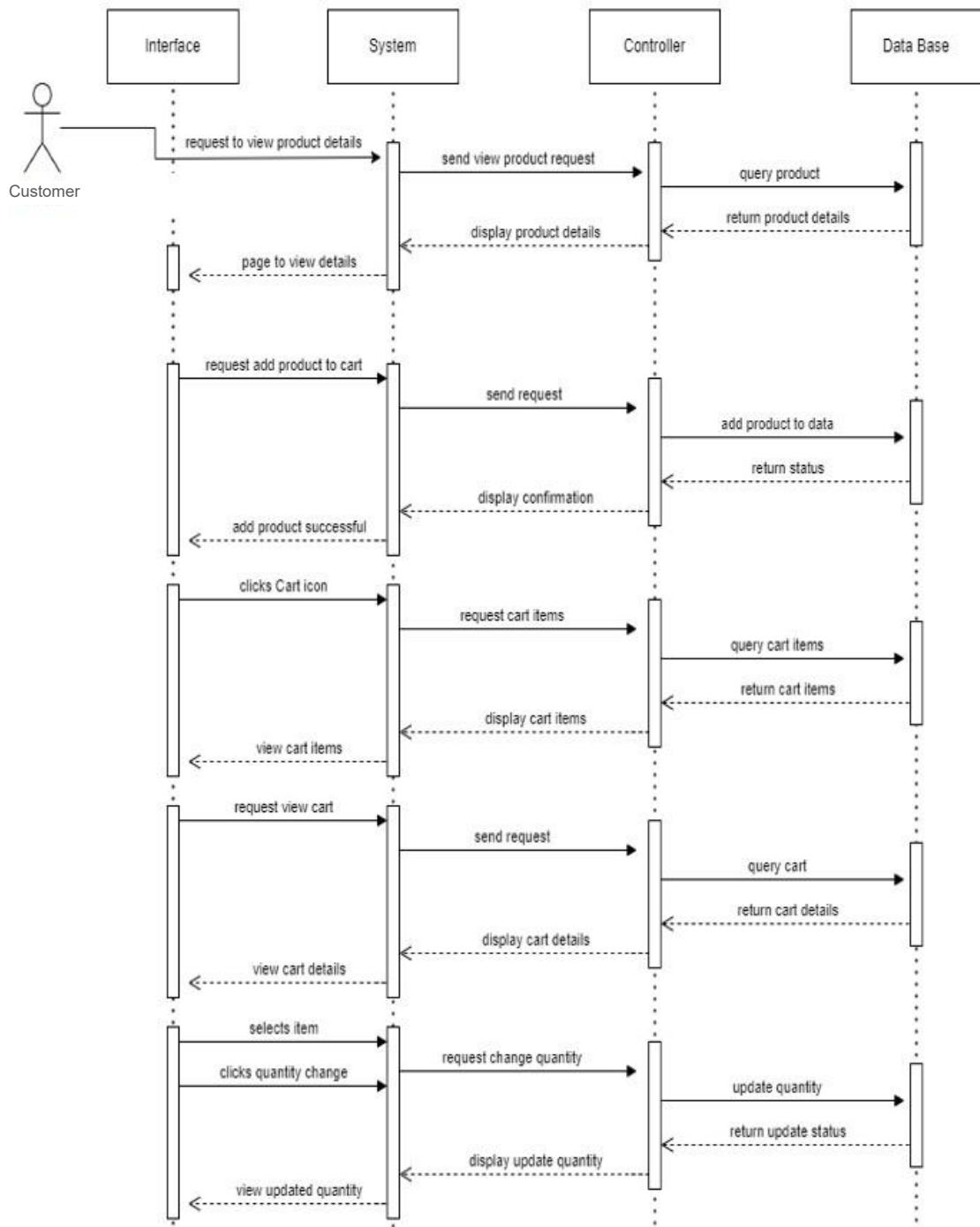


Figure 3.3.2. Product sequence diagram

3.2.3. User Interface design

The frontend of the system is developed using Next.js and ReactJS, leveraging their features for building highly responsive and dynamic user interfaces. These technologies ensure an optimal user experience across different devices.

Design Features

The interface is fully responsive, adapting seamlessly to various screen sizes, including desktops, tablets, and mobile devices, providing a consistent and intuitive user experience.

The design incorporates intuitive navigation paths that guide users through key features like product browsing, checkout, and order tracking.

All UI components follow a unified styling approach, ensuring consistency in typography, color schemes, and layouts across the platform.

UI Components

The interface and functionality of the online bookstore system are designed so that they can provide a smooth, friendly and efficient experience to administrators and users alike. The entry point is the homepage with major product categories for browsing and promotions to facilitate effective marketing. A search box that is global is placed at the top of the page so that users can conveniently search for the product they need.

Product page displays the whole product of the store and allows users to select categories such as Books, Pens, or Other Products from a dropdown list. Users can also filter products by many criteria such as price or customer ratings. When users click on a specific product, the product detail page will display the full information including images, descriptions, prices, stock status and user reviews. Functions such as "Add to Cart" or "Compare" are also added to facilitate the shopping experience.

The cart page allows customers to view products selected along with quantity and price and provides the option to refresh or remove products in the cart. The check-out process is streamlined, involving a secured payment gateway and simple shipping options.

On the admin side, the system provides a dashboard page that presents an overview of important indicators such as sales performance, revenue and orders over time. On the admin product page, new products can be created, products can be edited or removed from the stock. The system also includes an order management feature, where the admin can see, filter and process customer orders. Sales analysis is also shown in pictorial charts that reflect up-to-date data on consumer trends, inventory levels, and business performance.

In addition, there is also a focus on personalized customer experience in the system. Each customer's taste and shopping habits will be recommended with products that are appropriate to them. An intelligent chat box with the system makes use of artificial intelligence to give automatic replies to frequent queries and interact with sellers if needed. The overall design of the system is consistently towards simplicity, consistency, and flexibility for buyers and managers..

3.3. Algorithms and Technologies Used

3.3.1. Recommendation Systems in Next.js:

Recommendation systems are one of the most important in improving the level of user experience and engagement within an e-commerce and content-driven application. It is possible to create an effective system of recommendations in a Next.js application with personalized suggestions that will drive users to interact. There are generally two types of recommendation algorithms: collaborative and content-based filtering.

Collaborative Filtering: This is a technique that analyzes the behavior of several users to make recommendations. It finds patterns in user interactions—whether purchases, browsing history, or ratings—to recommend products frequently bought or liked by users with similar tastes. There are two approaches to collaborative filtering:

- *User-based Collaborative Filtering:* This involves comparing the preferences of a target user with those of other users to promote products liked by similar users. For instance, if User A and User B have similar buying patterns, then the products bought by User B would be recommended to User A.

```
Function UserBasedCollaborativeFiltering(userId):
    userInteractions ← Get interactions of user with ID userId
    allUsers ← Get interactions of all users
    recommendedItems ← empty list
    For each otherUser in allUsers:
        If otherUser.id ≠ userId Then
            similarity ← Calculate similarity between userInteractions and
            otherUser.interactions
            If similarity > 0.7 Then
                For each item in otherUser.interactions:
                    If item not in userInteractions Then
                        Add item to recommendedItems
```

```

        End If
    End For
    End If
    End If
End For
Return recommendedItems
End Function

```

- *'Item-based Collaborative Filtering:* In this method, the emphasis is on item similarity, not user similarity. For example, if Item X and Item Y are purchased together very often by many users, then Item Y will be recommended to every user who has ever bought or engaged with Item X. It will generally work well with many users relative to fewer items being compared.

```

Function ItemBasedCollaborativeFiltering(userId):
    userInteractions ← Get interactions of user with ID userId
    recommendedItems ← empty list
    For each item in userInteractions:
        similarItems ← Find items similar to item
        Add all similarItems to recommendedItems
    End For
    Return recommendedItems
End Function

```

Next.js can be integrated with APIs or services like TensorFlow.js for machine learning or external services like AWS Personalize for scalable, real-time recommendations. These systems track user interactions through data stored in a MySQL database and process them to identify trends and preferences.

Content-Based Filtering: This approach does not depend on other users' activities; it is focused on the user's previous interactions with some specific items. Capable of making recommendations based on genre, authors, price points, and even descriptions of any given item with which a user has shown associated past preferences. For instance, if a user has purchased a few fantasy novels from an author, it might recommend other books from the same author or in the same genre. Content-based filtering can be implemented easily in a Next.js app with data-driven models analyzing item attributes stored in a database. For example, it recommends similar novels to a user who has browsed or bought fantasy novels

before by the attribute of genre. This could be implemented in Next.js by creating a database query with matching tags or attributes and displaying them in a dynamic, personalized UI.

```
Function ContentBasedFiltering(userId):
    userInteractions ← Get the list of items the user has interacted with
    recommendedItems ← empty list
    userPreferences ← Extract user preferences based on the attributes of
    interacted items
    allProducts ← Get all products in the system
    For each product in allProducts:
        similarity ← Calculate the similarity between the product's attributes
        and userPreferences
        If similarity > 0.7 Then:
            Add product to recommendedItems
        End If
    End For
    Return recommendedItems
End Function
```

Similarly, in collaborative filtering and content-based filtering algorithms, key roles are played in offering a personalized experience that aims to enhance user engagement and satisfaction. Adopting a hybrid approach, which combines both methods, could yield even more accurate recommendations by leveraging their strengths. For instance, a system may first use collaborative filtering to identify a set of potential recommendations and then apply content-based filtering to fine-tune those suggestions based on a user's preferences.

Integration in a Next.js Application: Implementing recommendation systems in Next.js involves several steps:

- *Data Collection*: User data, such as browsing history, purchases, and interactions, needs to be stored in a relational database like MySQL. This data forms the basis for the recommendation algorithm.
- *Backend API*: A backend API, often built with Next.js API routes, can handle the processing of data for generating recommendations. These routes would interact with the recommendation engine, which could be an external service or a custom-built model in Node.js or Python.

```

Function APIHandler(request, response):
    userId ← Get userId from request query
    Try:
        recommendations ← Get recommendations based on userId
        Send response with status 200 and recommendations as JSON
    Catch error:
        Send response with status 500 and error message: "Failed to generate
        recommendations"
    End Try
End Function

```

- *Real-time Updates:* Recommendation systems should adapt to user behavior in real time. For instance, if a user adds a new item to their cart, the system can immediately update the list of suggested products.
- *Frontend Integration:* On the frontend, Next.js can dynamically render personalized recommendations based on user data. Using React components, you can create recommendation widgets, displaying products that align with the user's past actions or similar users' interests. These components can update asynchronously to reflect new suggestions based on user interactions.

3.3.2. Smart Inventory system:

The bookstore system integrates real-time reporting and smart inventory management to empower store owners with actionable insights and tools for efficient management. By leveraging technologies like Chart.js for data visualization and Next.js for a seamless user interface, the system ensures that owners can monitor key metrics, automate tasks, and boost customer engagement.

```

// Function to automate inventory notifications

Function AutomateNotifications():
    inventoryData ← Fetch current inventory data
    lowStockItems ← Check for items with low stock in inventoryData
    overstockItems ← Check for items with excess stock in inventoryData
    Generate alerts based on lowStockItems and overstockItems
End Function

// Function to periodically update sales and inventory information

Function UpdateData():

```

```

salesData ← Collect latest sales data
inventoryData ← Fetch updated inventory data
Manage inventory based on current data
Display sales report using salesData
End Function

```

Real-Time Sales Performance Reports

The system continuously tracks and analyzes sales data to provide up-to-date insights on:

Top-Selling Products: Identifies books that are currently in high demand, helping store owners focus on restocking and promoting these items.

Revenue Trends: Displays revenue performance over specific periods (e.g., daily, weekly, monthly) to help owners assess business growth and adjust strategies accordingly.

Category Performance: Breaks down sales by categories (e.g., fiction, non-fiction, children's books) to identify which segments are driving revenue.

Data is visualized using Chart.js, enabling the creation of interactive bar charts, line graphs, and pie charts. These visualizations make complex metrics easier to interpret at a glance.

Function collectSalesData():

```

salesData ← Fetch sales data from database
Return salesData
End Function

```

Function generateReports(salesData):

```

topSellingProducts ← Extract top-selling products from salesData
revenueTrends ← Calculate revenue trends over time from salesData
categoryPerformance ← Analyze sales performance by category from salesData
Return topSellingProducts, revenueTrends, categoryPerformance
End Function

```

Function displaySalesReport(topSellingProducts, revenueTrends, categoryPerformance):

```

Display bar chart for topSellingProducts
Display line chart for revenueTrends
Display pie chart for categoryPerformance
End Function

```

```

// Main process
salesData ← collectSalesData()
topSellingProducts,      revenueTrends,      categoryPerformance ←
generateReports(salesData)
displaySalesReport(topSellingProducts, revenueTrends, categoryPerformance)

```

Smart Inventory Management

The system includes a smart inventory management module that automates stock monitoring and provides actionable alerts. Features include:

- Stock Notifications:
 - o Low-stock alerts for books that are about to run out, ensuring timely restocking.
 - o Overstock warnings to help reduce storage costs and prevent waste.
- Bestseller and Top-Seller Tracking:
 - o Automatically identifies and tracks bestselling books within defined time periods.
 - o Recommends restocking and promotional strategies for high-demand items.
- Revenue Analytics Integration:
 - o Links sales data with inventory levels to highlight products contributing most to revenue.

Function manageInventory():

```

inventoryData ← fetchDataFromDatabase("inventory")
lowStockAlerts ← checkLowStock(inventoryData)
overstockAlerts ← checkOverstock(inventoryData)
bestsellers ← trackBestsellers(inventoryData)
revenueAnalytics ← integrateRevenueAnalytics(inventoryData, salesData)
Return lowStockAlerts, overstockAlerts, bestsellers, revenueAnalytics

```

End Function

Function checkLowStock(inventoryData):

```
lowStockItems ← empty list
```

For each book in inventoryData:

If book.stock <= book.lowStockThreshold Then:

 Add book to lowStockItems

End If

```

        End For
        Return lowStockItems
    End Function

    Function checkOverstock(inventoryData):
        overstockItems ← empty list
        For each book in inventoryData:
            If book.stock >= book.maxStockThreshold Then:
                Add book to overstockItems
            End If
        End For
        Return overstockItems
    End Function

    Function generateAlerts(lowStockItems, overstockItems):
        For each item in lowStockItems:
            generateLowStockAlert(item)
        End For
        For each item in overstockItems:
            generateOverstockAlert(item)
        End For
    End Function

    Function integrateRevenueAnalytics(inventoryData, salesData):
        revenueAnalytics ← empty list
        For each category in inventoryData.categories:
            categorySales ← calculateCategoryRevenue(category, salesData)
            Add { category: category.name, revenuePercentage:
categorySales.revenuePercentage } to revenueAnalytics
        End For
        Return revenueAnalytics
    End Function

```

Email Promotion for Customers

The system sends personalized email campaigns to customers:

- Top Sellers: Promote high-demand books.
- New Arrivals: Notify customers about recently added titles.

Features:

- Email Personalization: Each email includes the customer's name and a list of recommended books.
- Batch Email Sending: The system sends emails in bulk to a predefined customer list.
- Scheduling: Emails are scheduled to be sent during peak engagement times.

Work flow

- Data Collection: Sales and inventory data are updated in real time via the database.
- Analytics Dashboard: Store owners access interactive charts and tables for performance insights.
- Automation: Notifications for low-stock items and top-seller reports are automatically generated.
- Customer Engagement: Scheduled promotional emails keep customers informed and engage

```

Function preparePromotionEmails():

    customerList ← fetchCustomerList()
    topSellers ← getTopSellers()
    newArrivals ← getNewArrivals()

    For each customer in customerList:
        emailContent ← createPersonalizedEmail(customer, topSellers,
                                                newArrivals)
        scheduleEmail(customer.email, emailContent)

    End For

End Function

Function createPersonalizedEmail(customer, topSellers, newArrivals):

    emailSubject ← "Your Personalized Book Recommendations"
    emailBody ← "Hello " + customer.name + ",\n\n"
                + "Check out our latest top sellers and new arrivals!\n\n"
                + "Top Sellers:\n" + joinBooks(topSellers) + "\n\n"
                + "New Arrivals:\n" + joinBooks(newArrivals)

    Return { subject: emailSubject, body: emailBody }

End Function

Function scheduleEmail(emailAddress, emailContent):

    scheduledTime ← getOptimalSendTime()
    scheduleEmailToSendAt(emailAddress, emailContent, scheduledTime)

End Function

```

In the current system, the price recommendation algorithm is implemented based on the actual database collected from the 'revenue' and 'products' tables. Key parameters include the number of products sold this week and last week, the number of clicks on the product, and the current price of the product. Once the data is available, the system calculates the conversion rate and the change in revenue between the two weeks.

If a product has a conversion rate greater than 20% and revenue increases compared to the previous week, the system will suggest a 10% increase in the price of that product. Conversely, if the product has a conversion rate of less than 5% and revenue decreases, the system recommends a 10% discount. These recommendations are then saved in the 'price_suggestions' table with fields such as: product ID, old price, suggested price, recommendation type (increase or decrease), reason and time for creating the proposal.

The dashboard display pulls data from the 'price_suggestions' and 'products' tables, combined with images from the 'productimages' table to provide managers with a complete and visual view of the latest price proposals.

3.3.3. Advanced Search Function

To improve user experience and product accessibility, the system integrates a flexible and intelligent search functionality that allows users to search for products using various attributes. These include the product name, description, publish year, author, ink color, pen type, and associated tags. This multi-attribute search supports complex queries using multiple keywords separated by the operator "and", thereby enabling more precise filtering of results.

When a search request is initiated, the system first parses and processes the provided search parameters. Each attribute is used to construct SQL LIKE queries that compare the input keywords against relevant fields in the local database. The query supports partial matches using wildcard operators to ensure that even incomplete or approximate search inputs return meaningful results.

The following SQL structure is dynamically generated for each search input:

```
SELECT DISTINCT
    p.*,
    pi.ImageURL AS image
FROM products p
LEFT JOIN productimages pi ON p.ProductID = pi.ProductID AND pi.IsPrimary = 1
LEFT JOIN tags t ON p.TagID = t.TagID
```

```
WHERE (...) -- dynamically built conditions  
ORDER BY p.Name ASC  
LIMIT 10
```

If matching results are found in the local database, they are returned to the user immediately along with the product's primary image. This approach ensures quick access to existing store data without unnecessary API calls.

However, in cases where no relevant products are found locally, the system automatically extends the search to the Google Books API. This ensures that the user is not left without results and enhances the completeness of the platform by integrating external resources. The retrieved Google Books data includes essential product information such as title, author, thumbnail, and publication details.

```
const url =  
'https://www.googleapis.com/books/v1/volumes?q=${encodeURIComponent(query)}&m  
axResults=5';
```

This dual-source search system (local database first, then Google Books API as fallback) not only increases the success rate of product searches but also enriches the catalog dynamically without manual data entry. Moreover, all search operations are handled through asynchronous API requests to ensure non-blocking user experience.

3.3.4. Backend Development with Node.js:

The backend of the bookstore system is developed using Node.js, providing a fast, efficient, and scalable environment to handle server-side operations. It ensures seamless communication between the client-side (React components in Next.js) and the database (MySQL). By managing API requests and other backend logic, Node.js acts as the backbone of the system, supporting various functionalities and ensuring a smooth user experience.

Key Features and Responsibilities:

Key features and responsibilities include comprehensive API management where Node.js exposes RESTful API endpoints to handle frontend requests such as updating inventory after purchases, processing customer orders and recording them in the database, and retrieving user data for personalized interactions or authentication. The system integrates seamlessly with a MySQL database using libraries like mysql2 or knex.js, executing SQL queries for CRUD operations and supporting transactions for critical processes like order processing to maintain data integrity. Security is a priority, with sensitive information such as database credentials

securely managed through environment variables (dotenv), alongside input sanitization measures to prevent SQL injection and other vulnerabilities. Notifications, including alerts about low-stock items or order confirmations, are managed through API endpoints that the frontend periodically calls to fetch the latest status updates. Additionally, the architecture is designed with a modular structure to ensure scalability, allowing for the easy addition of new features such as promotional emails or advanced analytics.

3.4. Class diagram

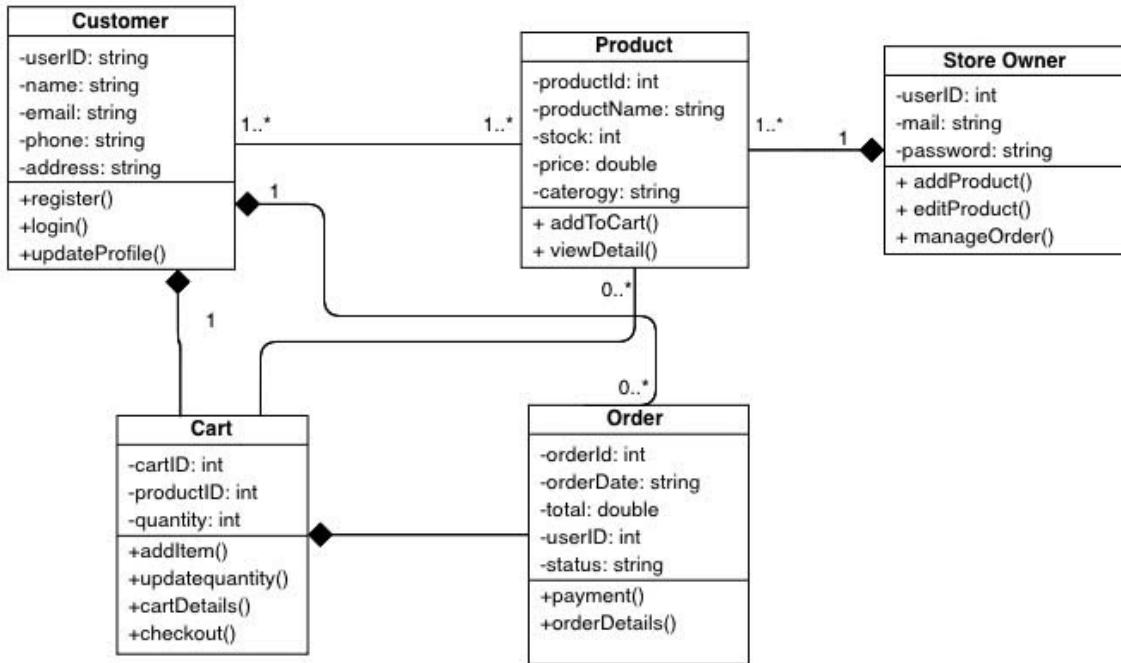


Figure 3.5. Class diagram

In the bookstore system, the Customer and Cart classes are closely linked, with each customer having a shopping cart. The shopping cart is owned by the customer and is linked through the customerId attribute, ensuring that each shopping cart is uniquely mapped to each customer. The Cart and Product classes are also linked, as a shopping cart can contain multiple products. Each item in the shopping cart is identified and managed by its product.

The Customer and Order classes are linked in that a customer can create multiple orders; each order references only the customer ID to maintain a clear link to the customer who placed the order. Furthermore, the Order class is linked to the Product in such a way that an order contains multiple products, each identified by their product ID.

Finally, the StoreOwner and Product classes are linked through a management link. The store owner will manage multiple products by adding, updating, or removing them from

inventory. This structure ensures smooth management and interaction between the different components of the bookstore system.

CHAPTER 4

IMPLEMENT AND RESULTS

4.1. Implement

This section outlines the process of implementing the system, detailing the steps taken to develop and integrate its components.

4.1.1. Backend Development

Database Setup

- Created a relational database schema to manage entities like users, products, orders, and carts using MySQL.
- Established relationships between tables to enforce data integrity.
- Utilized indexing to optimize search and filter operations.

API Development

- Developed APIs using Node.js and Express to handle operations like login, product management, cart updates, and order processing.
- Integrated middleware for user authentication and authorization using jsonwebtoken.

4.1.2. Frontend Development

Framework and Libraries

- Built the frontend using Next.js/ReactJS for its performance and server-side rendering capabilities.
- Used libraries like Tailwind CSS or Bootstrap for styling and component consistency.

UI Implementation

- Designed responsive UI components for the homepage, product pages, cart, and admin dashboard.

4.2. Results

- The results section highlights the outcomes of the implementation phase, presenting the functionality and performance of the system.

4.2.1. Authentication

Authentication plays a crucial role in ensuring the security and functionality of the system by restricting access to authorized users and providing a personalized experience. This section explains the technologies and implementation details of the authentication process used in the bookstore system.

- **Overview:** The authentication system ensures that only valid users can access restricted features of the bookstore. It includes user registration, login, and token-based authentication to secure communication between the client and server.
- **Technology use:** The authentication system is built using the following technologies:
 - o Next.js API Routes: Handles server-side logic for authentication-related tasks.
 - o JSON Web Token (JWT): Provides stateless authentication by generating and verifying tokens.
 - o MySQL: Stores user information, including hashed passwords.
 - o bcrypt: Hashes passwords before saving them in the database and verifies them during login.
- **Authentication flow**
 - o **Registration:**
 - Users input their details (username, email, and password) in the registration form.
 - The password is hashed using bcrypt and stored in the MySQL database.
 - o **Login:**
 - Users provide their credentials (email and password).
 - The system validates the email and password combination by comparing the hashed password in the database.
 - A JWT token is generated upon successful validation and sent to the client.
 - o **Token Verification:**
 - Each client request includes the JWT token in the headers.
 - The server verifies the token to determine access to protected resources.
 - o **Logout:** The JWT token is removed from the client-side storage (e.g., localStorage or cookies), effectively ending the session.

- **Visualization**

- After clicking Login button, user will be redirected to Login page

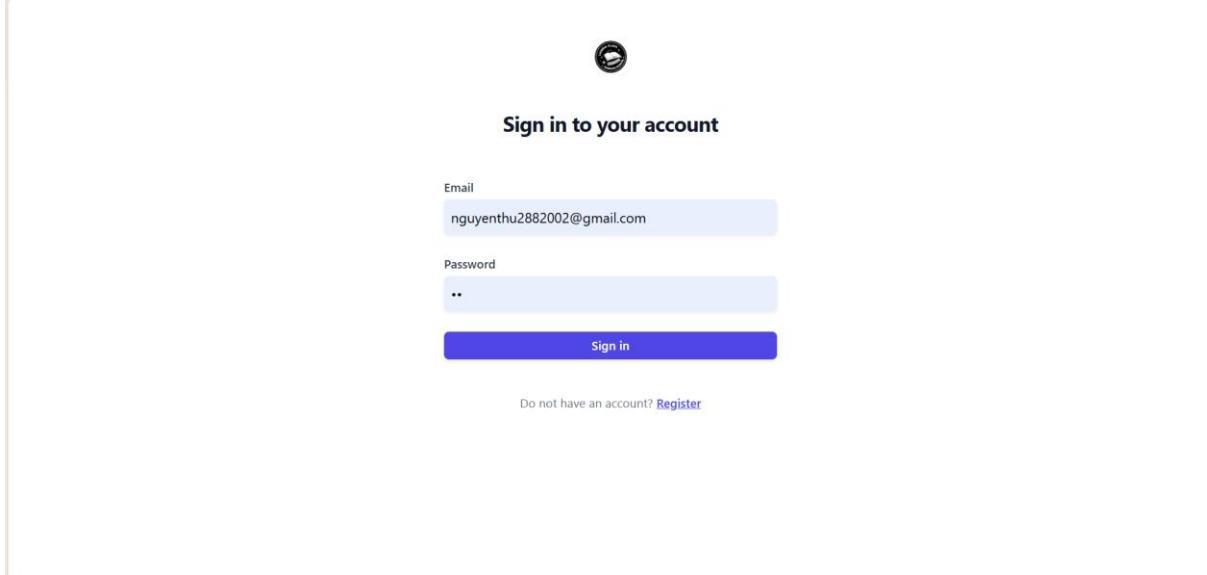


Figure 4.2.1. Signin page

- Fill information then click Sign in
 - If invalid email or password, show error message
 - If that account is for Store Owner, redirect to Manage page

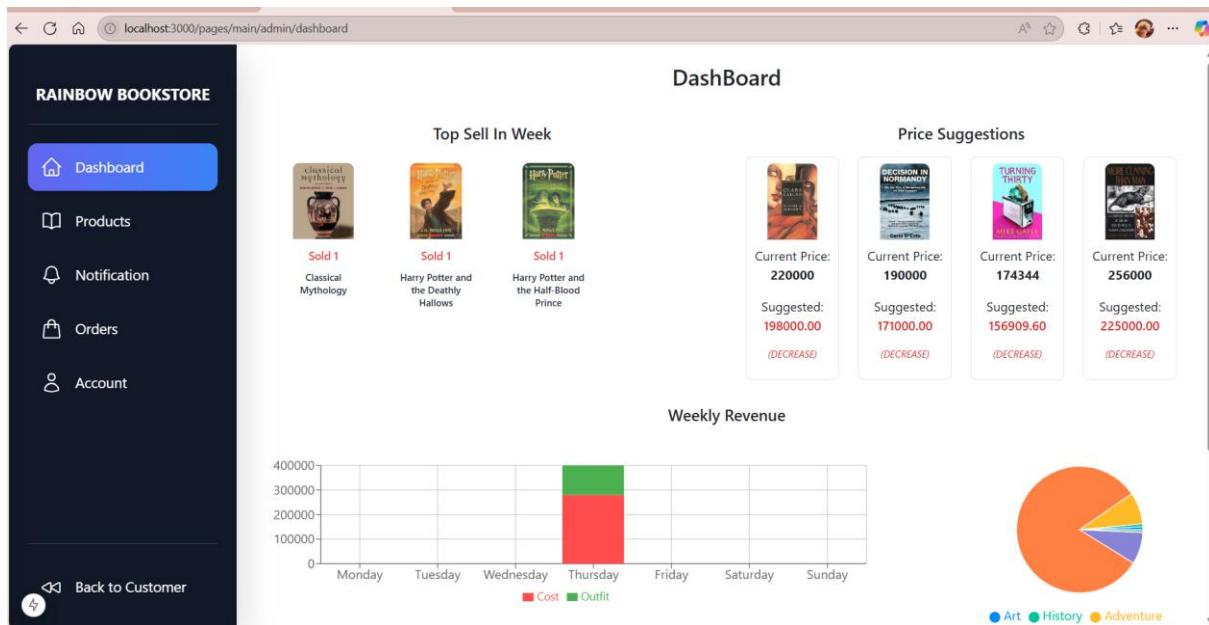


Figure 4.2.1. Manage page

- If customer, redirect to Homepage
 - Register page
 - If the entered email already exists, the system displays an error message.

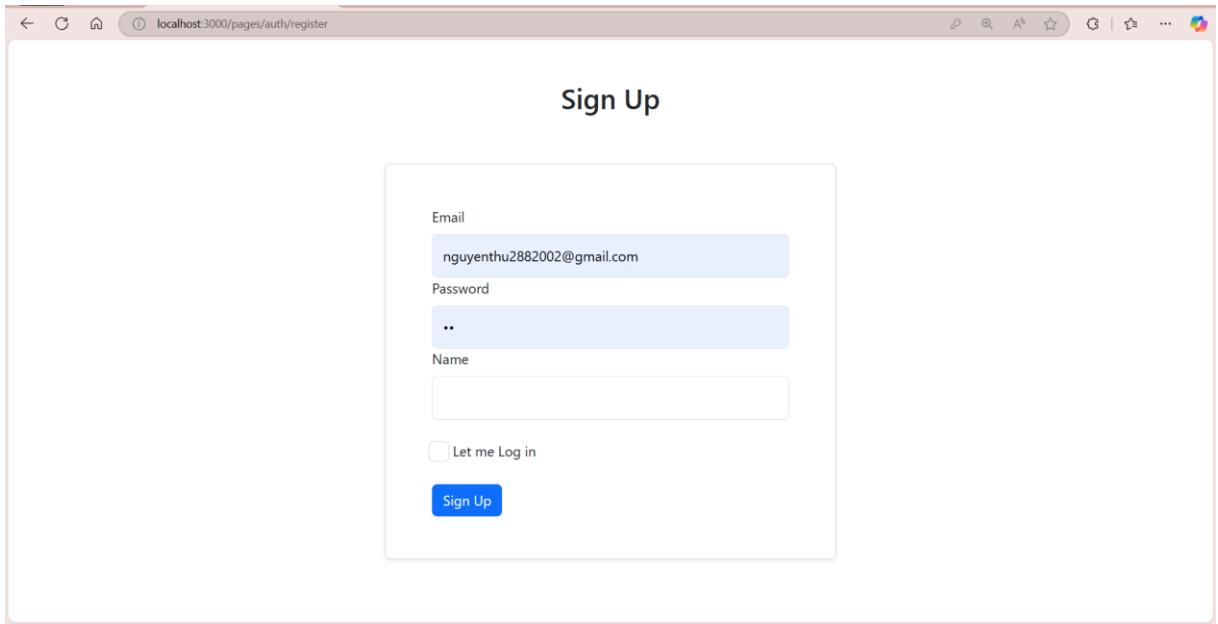


Figure 4.2.1. Register page

User table

- After Sign up, system hash the password them store into database

Result Grid		Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:	
CustomerID	Name	Password	Email	DateOfBirth	PhoneNumber	Address	role_id		
1	Nguyen Thi Kim Thu	\$2b\$10\$TDHLKthy1aPsCxA3Npb0jesk1oMYWyp...	thuthu2882002@gmail.com	2002-08-28	NULL	419/1B Vườn Lài	2		
2	Nguyen Thi Kim Thu	\$2b\$10\$TDHLKthy1aPsCxA3Npb0jesk1oMYWyp...	nguyenthu2882002@gmail.com	NULL	NULL	NULL	1		

Figure 4.2.1. Login database

4.2.2. Products

The Products module is a key component of the bookstore system, enabling the management and display of product information. This section highlights the implementation of product features, including adding, displaying, and managing products for both administrators and customers, as well as the integration of a "hot item" feature based on user interactions.

4.2.2.1. Insert Products (Admin Panel)

The Insert Products feature enables administrators to add new products dynamically based on their category. Default fields are stored in a general Products table, while additional fields, specific to the selected category, are stored in corresponding category-specific tables.

The screenshot shows a web-based administration interface for adding a new product. On the left, a dark sidebar lists navigation options: Dashboard, Products, Orders, Account, and Back to Customer. The main content area is titled "Insert New Product". It includes several input fields: "Product Name" (text input), "Description" (text area), "Price" (text input), "Stock" (text input), and "Category" (a dropdown menu currently showing "Select Category"). Below these are "Tags" represented by a grid of checkboxes. The available tags are categorized into two columns: Adventure, Fiction, Literature, Blue, Bag; and Novel, Manga, Thriller, Black, Ruler; and Romance, Comic, Red, Color, Eraser.

Figure 4.2.2. Add products page

Dynamic Category Selection

When adding a product, the administrator can select a category from the following options:

- Books: For book-related products.
- Pen: For writing tools.
- Other: For miscellaneous items.

The screenshot shows a dropdown menu with the title "Select Category". The menu is currently open, displaying three categories: "Books", "Pens", and "Others".

Figure 4.2.2. Caterogy dropdown

Dynamic Input Fields

Upon selecting a category, specific input fields tailored to the category appear:

- Books: Book title, Author, Published year

Category

Tags

<input type="checkbox"/> Adventure	<input type="checkbox"/> Novel	<input type="checkbox"/> Romance
<input type="checkbox"/> Fiction	<input type="checkbox"/> Manga	<input type="checkbox"/> Comic
<input type="checkbox"/> Literature	<input type="checkbox"/> Thriller	<input type="checkbox"/> Red
<input type="checkbox"/> Blue	<input type="checkbox"/> Black	<input type="checkbox"/> Color
<input type="checkbox"/> Bag	<input type="checkbox"/> Ruler	<input type="checkbox"/> Eraser
<input type="checkbox"/> Kid	<input type="checkbox"/> Teenager	<input type="checkbox"/> Adult
<input type="checkbox"/> 16+	<input type="checkbox"/> 18+	

Author

Publish Year

Product Images

 No file chosen

Submit Product

Figure 4.2.2. Book field insert

- Pen: Pen type (e.g., ballpoint, fountain)
- Other: Just default field

Backend:

The submitted data is processed via a Next.js API Route, which validates and formats the input.

ProductID	Name	Description	Price	Stock	Categor	Sold	Clicked	Cost	Author	Publisher
1	Classical Mythology	Provides an introduction to classical myths plac...	275000	99	1	11	12	180119	Mark P. O. Morford	Oxford University Press
2	Clara Callan	In a small town in Canada, Clara Callan reluctantly...	220000	90	1	11	10	126030	Richard Bruce Wright	HarperFlamingo Canada
3	Decision in Normandy	Here, for the first time in paperback, is an outst...	190000	97	1	10	6	105267	Carlo D'Este	HarperPerennial
4	Flu: The Story of the Great Influenza Pandemic ...	"Scientists have recently discovered shards of t...	213000	100	1	10	8	143463	Gina Kolata	Farrar Straus Giroux
5	The Mummies of Urumchi	A look at the incredibly well-preserved ancient ...	173000	100	1	10	6	113491	E. J. W. Barber	W. W. Norton & Company
6	The Kitchen God's Wife	An absorbing narrative of Winnie Louie's life.	208000	100	1	10	5	153673	Amy Tan	Putnam Pub Group
7	What If?: The World's Foremost Military Historia...	Essays by respected military historians, includin...	317000	100	1	10	6	233514	Robert Cowley	Berkley Publishing Group
8	PLEADING GUILTY	Returning to the now-renowned locale of Kindle...	203000	100	1	10	5	136440	Scott Turow	Audioworks
9	Under the Black Flag: The Romance and the Re...	Examines the popular image of pirates in moder...	279000	100	1	10	6	222173	David Cordingly	Random House
10	Where You'll Find Me: And Other Stories	Now back in print, Ann Beattie's finest short sto...	257000	100	1	10	5	175719	Ann Beattie	Scribner
11	Nights Below Station Street	Another story based in the fictional rural town i...	174000	100	1	10	5	101114	David Adams Richards	Emblem Editions
12	Hilter's Secret Bankers: The Myth of Swiss Neut...	Hilter's Secret Bankers was the first book to disc...	271000	100	1	10	5	221092	Adam Lebor	Citadel Press
13	The Middle Stories	Part Dorothy Parker, part Jose Saramago, with ...	304000	100	1	10	5	244348	Sheila Heti	House of Anansi Press

Figure 4.2.2. Products table

4.2.2.2. Show and Edit Products (Store owner Panel)

The admin panel displays a list of all products fetch from api, including:

- Product ID, name, price, stock quantity from products table, and image from productimages table.

ID	Image	Name	Price	Cost	Stock	Delete
1		Classical Mythology	275000	180119	99	-
2		Clara Callan	220000	126030	90	-
3		Decision in Normandy	190000	105267	97	-
4		Flu: The Story of the Great Influenza Pandemic of 1918 and the Search for the Virus That Caused It	213000	143463	100	-
5		The Mummies of Urumchi	173000	113491	100	-
6		The Kitchen God's Wife	208000	153673	100	-

Figure 4.2.2. Show products page (Store owner side)

Actions to edit or delete products.: Edit action can change Name, Price, Stock and update the database immediately by double click in each field.

ID	Image	Name	Price	Cost	Stock	Delete
1		Classical Mythology	275000	180119	99	-
2		Clara Callan	220000	126030	90	-
3		Decision in Normandy	190000	105267	97	-
4		Flu: The Story of the Great Influenza Pandemic of 1918 and the Search for the Virus That Caused It	213000	143463	100	-
5		The Mummies of Urumchi	173000	113491	100	-
6		The Kitchen God's Wife	208000	153673	100	-

Figure 4.2.2. Editing product

If click Delete, that products will immediately set Status to Inactive and do not display in frontend anymore, then reload page Show product.

4.2.2.3. Show List and Detail products (Customer side)

Customers can view products displayed on the homepage or within specific categories, each product card includes the product image, name, price, and an Add to Cart button.

Customers can filter the products by clicking the attributes in the left hand side of the page

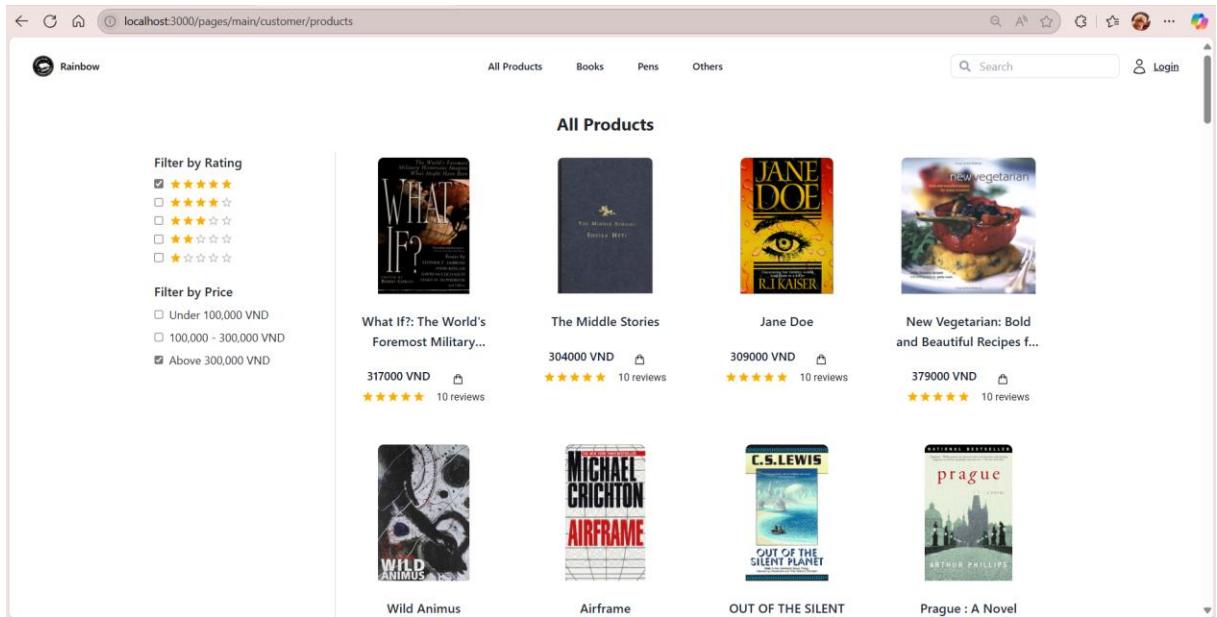


Figure 4.2.2. All products page after filter

If click cart button, that product will be added 1 quantity to customer cart by ProductID and use this ProductID to select other information to show in cart page. Then the system will calculate total by use quantity and price of each product

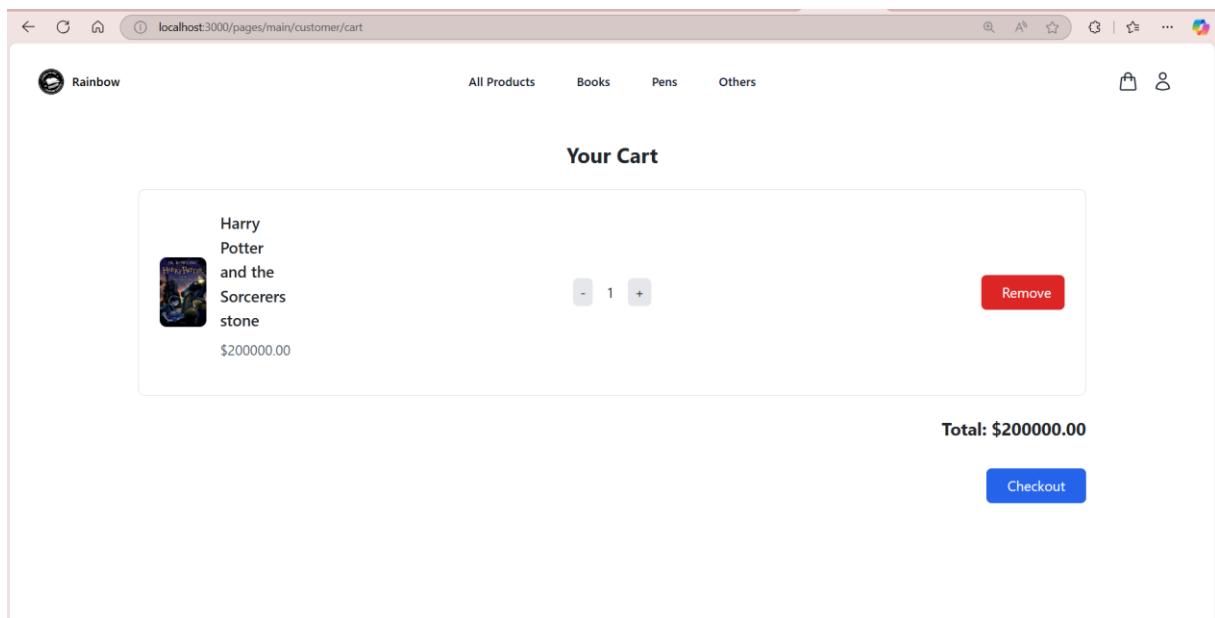


Figure 4.2.2. Cart page

Clicking a product cart navigates the user to the Detail Product Page.

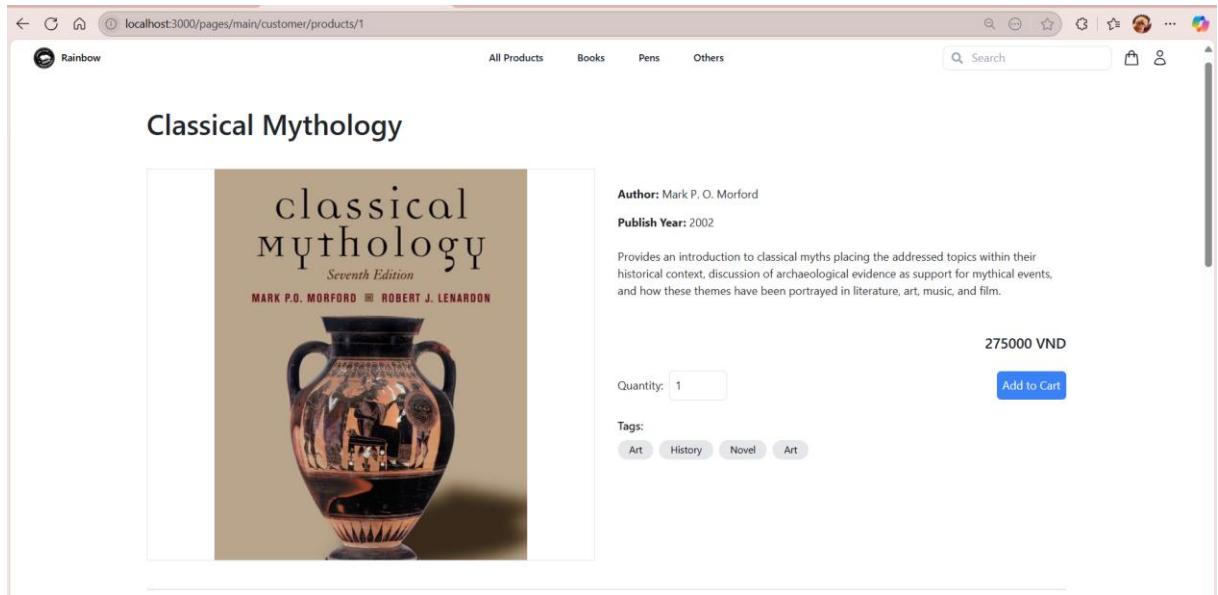


Figure 4.2.2. Product detail page with tags

In this page, customer can choose quantity then add to cart.

4.2.3. Recommendation system

4.2.3.1. Interesting products

To enhance user experience, the system will rely on the user's previous interactions with products and score them with corresponding tags, thereby selecting products with corresponding tags and displaying them in this component.

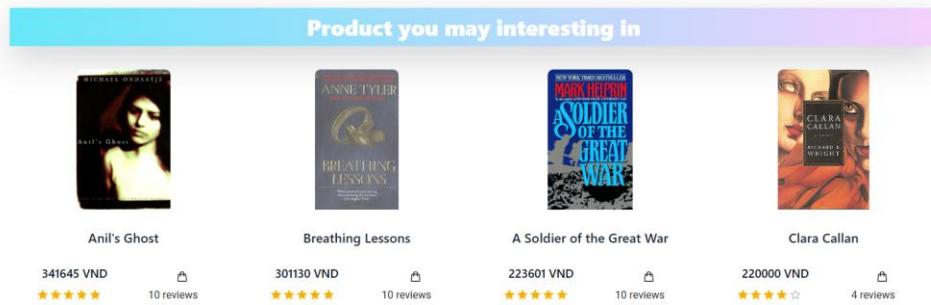


Figure 4.2.3. Intersting Component

	CustomerID	TagID	Score
▶	1	42	4
	1	44	4
	1	45	4
	2	5	22
	2	22	11
	2	24	11
	2	28	44
	2	42	112
	2	43	116
	2	44	33

Figure 4.2.3. Customer-tag-score table in database

4.2.3.2. Book of the day

To be able to recommend the best books based on high ratings, purchases and clicks, the system will render 10 books with their short descriptions, titles so that buyers can passively learn about them and if they are interested, they can click to see more details.

The screenshot shows a product card for a book titled "MORE CUNNING THAN MAN". The cover art features a black and white illustration of a rat. Below the title, it says "A COMPLETE HISTORY OF THE RAT AND ITS ROLE IN HUMAN CIVILIZATION" and "ROBERT HENDRICKSON". To the right of the book image, the title "More Cunning Than Man: A Social History of Rats and Man" is displayed in bold black text. Below the title is a brief description: "This eye-opening, well-researched examination of mankind's oldest competitor is filled with weirdly fascinating information about the history of the rat and the way it consistently outsmarts man. Illustrations." Below the description are the price "256000 VND", a "Details" button (green), and an "Add to Cart" button (orange). At the bottom of the card, there are five small circular icons representing reviews or ratings.

Figure 4.2.3. Book of the day component

4.2.3.3. Related Items

To make it easier for customers to access books of their favorite genres, the system has components that display books with the same calculated and selected attributes based on tags that are all displayed on the product detail page.

Classical Mythology

Author: Mark P. O. Morford
Publish Year: 2002
 Provides an introduction to classical myths placing the addressed topics within their historical context, discussion of archaeological evidence as support for mythical events, and how these themes have been portrayed in literature, art, music, and film.

Tags: Art, History, Novel, Art

275000 VND

Nguyen Thi Kim Thu
 ★★★★★ (2 reviews)

Good Book ever
 5/22/2025

Nguyen Thi Kim Thu
 ★★★★★

Hahaha
 5/15/2025

Related Items

Decision in Normandy	The Mummies of Urumchi	What If?: The World's Foremost Military Historians Imagine...	Nights Below Station Street

Figure 4.2.3. Details page with Related Items

4.2.3.4. Hot Products and Best Seller

To highlight popular products by increase Clicked count whenever load product detail page, each product's click count is recorded in the database.

	ProductID	Name	Description	Price	Stock	CategoryID	Sold	Clicked
▶	16	Harry Potter and the Sorcerers stone	Harry Potter is a fantasy book series by J.K. Ro...	200000	25	1	NULL	13
●	18	Thiên Long pen	Thiên Long Pen is a popular brand known for its ...	10000	100	2	NULL	3
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 4.2.2. Product table with Clicked change

A "Hot Items" section dynamically displays the 4-most-clicked products and updated automatically

This feature encourages customers to explore popular items.

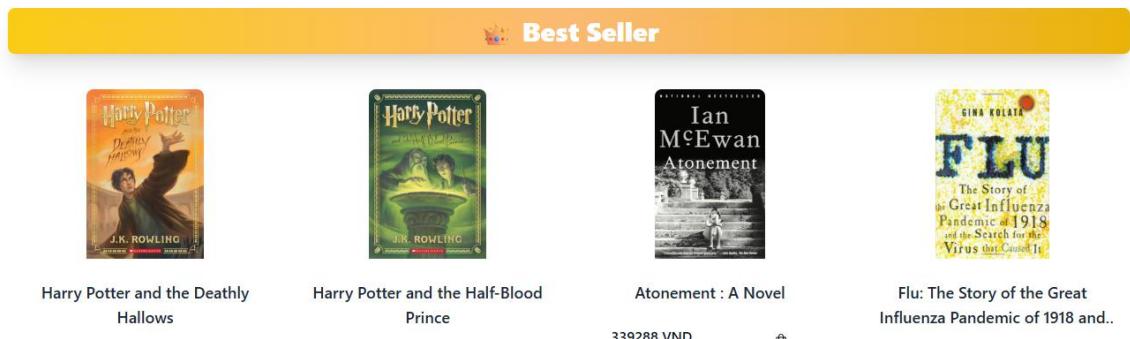
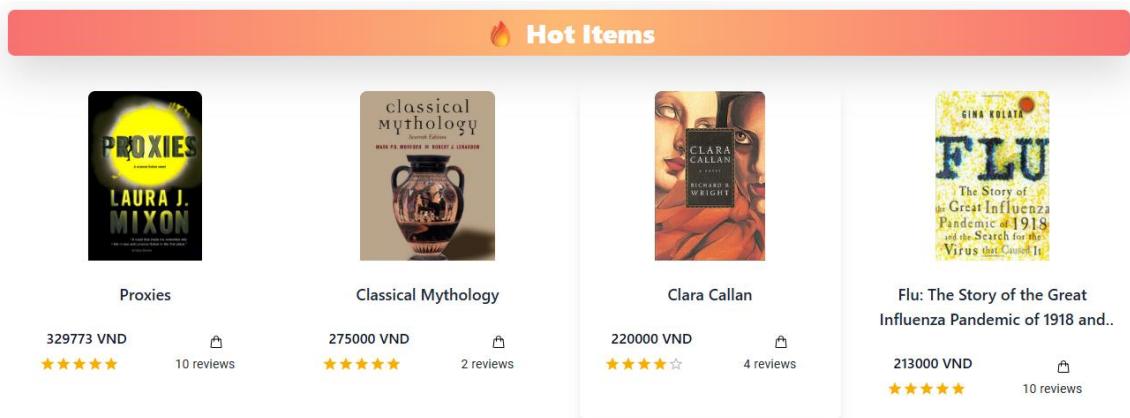


Figure 4.2.2. Hot product and Best seller component

For Best Seller components, the system will rely on the revenue table to select the best-selling products of the week.

4.2.4. Search function

Users can search for products based on title, author, category,...The system can also search quickly by clicking on related tags on the product detail page.

Filter by Rating	Under 100,000 VND	100,000 - 300,000 VND	Above 300,000 VND
<input type="checkbox"/> ★★★★★	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> ★★★★★	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> ★★★★☆	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> ★★★☆☆	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> ★★☆☆☆	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Filter by Price	Under 100,000 VND	100,000 - 300,000 VND	Above 300,000 VND
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
223601 VND	378823 VND	301061 VND	
★★★★★ 10 reviews	★★★★★ 10 reviews	★★★★★ 10 reviews	

Figure 4.2.4. After clicking History tag

To provide a better experience, when customers search for a product that is not in the system, the system will get the api from google books api and display it. If the user is interested, they can click and go to the google books page.

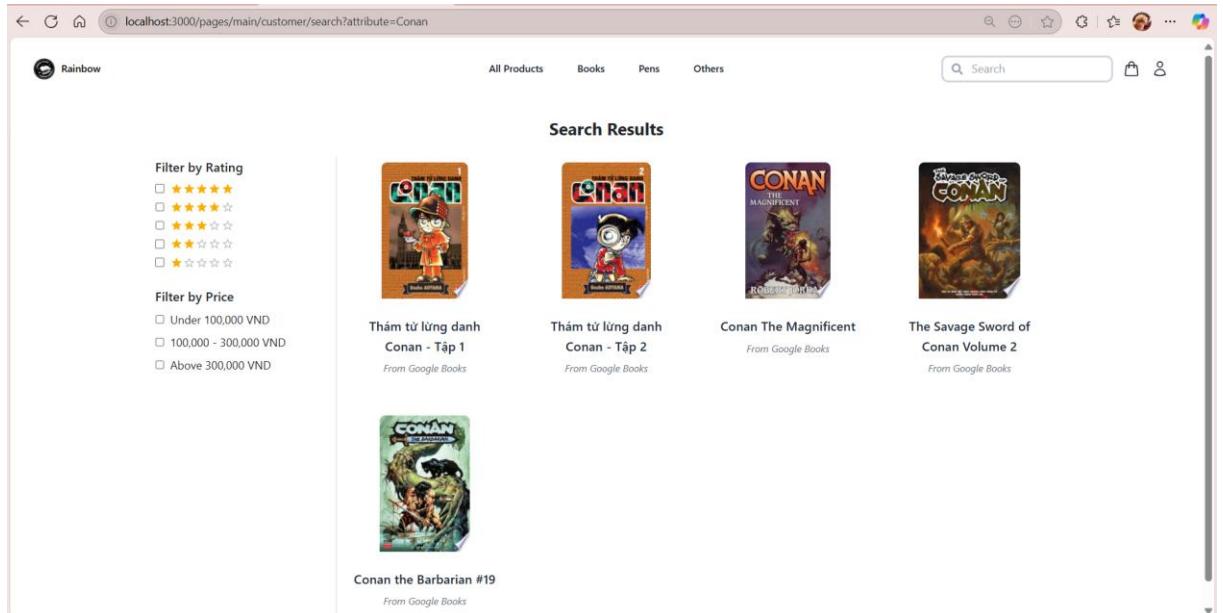


Figure 4.2.4. Search result for products that do not exist in database

4.2.5. Purchasing process

4.2.5.1. Customer Side

After adding products to the cart, users can proceed to checkout.

On the payment page, the system will display the customer's personal information if available in the database, the customer can change it if desired.

Figure 4.2.5. Check out page

After completing order, customer will be directed to Homepage

On the Orders page, buyers can view the status and details of their orders.

The screenshot shows a web browser window with the URL `localhost:3000/pages/main/customer/account/orders`. The page title is "Order History". On the left, there is a sidebar with links: "Profile", "Password", and "Orders" (which is highlighted with a blue background). The main content area displays a table titled "Order History" with columns: "#", "ORDER DATE", "TOTAL", and "STATUS". The table contains 11 rows of order data:

#	ORDER DATE	TOTAL	STATUS
1	22/05/2025	612000 VND	Completed
2	15/05/2025	295000 VND	Completed
3	12/05/2025	240751 VND	Completed
4	30/04/2025	240751 VND	Completed
5	30/04/2025	856597 VND	Completed
6	30/04/2025	1695234 VND	Completed
7	25/01/2025	1350000 VND	Shipped
8	25/01/2025	260000 VND	Pending
9	25/01/2025	1000000 VND	Shipped
10	25/01/2025	600000 VND	Pending
11	25/01/2025	35000 VND	Completed

Figure 4.2.5. Orders page [Customer]

If the order completed, this will be able to write a review

The screenshot shows a web browser window with the URL `localhost:3000/pages/main/customer/account/orders/54`. The page title is "Order Details". On the left, there is a sidebar with links: "Profile", "Password", and "Orders" (which is highlighted with a blue background). The main content area displays "Order Details" for order #54, which is "Completed" and has a total of 592000 VND. Below this, there is a section for a review of a product titled "Classical Mythology". The review says "Thank you for your review! You have already rated this product. ★★★★★ Good Book ever". At the bottom, there is a "Rate and Review This Product" section with a 5-star rating and a comment "Not bad!".

Figure 4.2.5. Orders details page with review [Customer]

4.2.5.2. Store owner Side

When there is an order, the system will send a notification to the seller, the seller can update the order status on the Orders page.

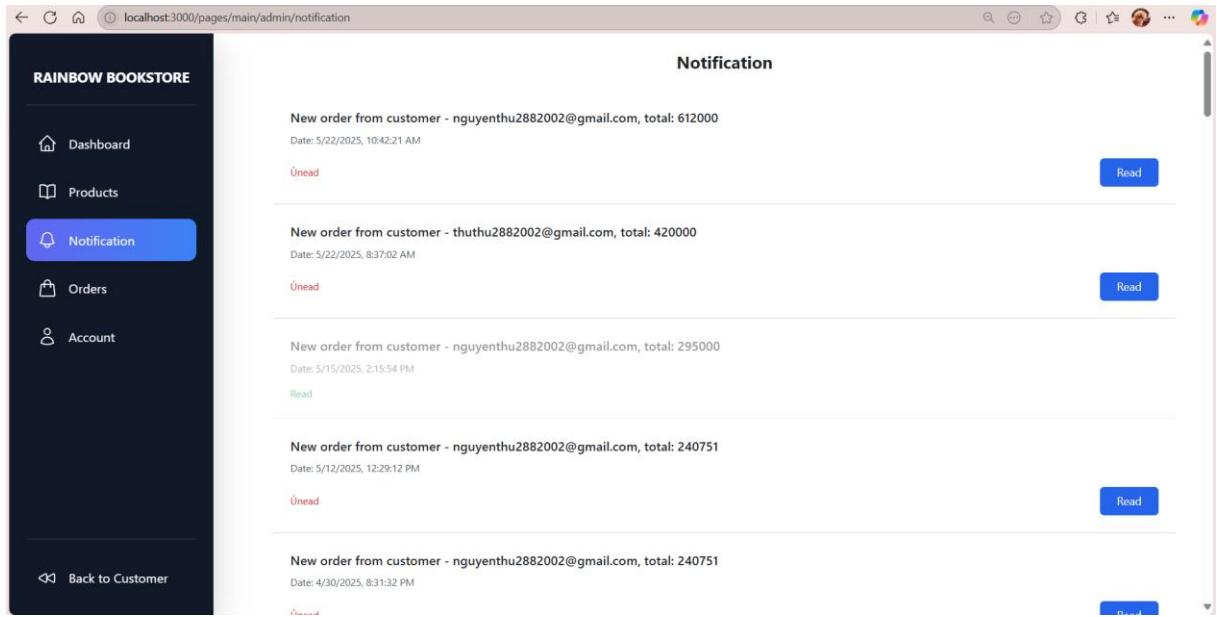


Figure 4.2.5. Notification page

The Orders page will display orders in order of oldest to newest so that sellers can arrange orders reasonably, prioritizing orders first. Here, sellers will adjust the order status to display to buyers.

Order ID	Customer ID	Date	Total Amount	Status
15	1	29/03/2025	612000 VND	Pending
16	11	29/03/2025	1170000 VND	Pending
17	1	29/03/2025	71000 VND	Pending
18	1	29/03/2025	270000 VND	Pending
19	1	31/03/2025	280000 VND	Pending
20	1	31/03/2025	60000 VND	Pending
21	1	31/03/2025	270000 VND	Pending
22	1	31/03/2025	975000 VND	Pending
23	1	31/03/2025	570000 VND	Pending
24	1	03/04/2025	50000 VND	Pending
25	1	03/04/2025	40000 VND	Pending
26	1	22/05/2025	420000 VND	Shipped
27	2	22/05/2025	612000 VND	Completed

Figure 4.2.5. Orders page [Store Owners]

CHAPTER 5

DISCUSSION AND EVALUATION

Chapter 5 presented the discussion and evaluation process of the smart bookstore system developed by the team. Through the actual implementation process, the system not only meets the basic requirements of an e-commerce website but also integrates many advanced features such as product suggestions based on user behavior, user authorization, visual statistics by chart and smart search.

5.1. Discussion

The development process of the smart bookstore system goes through many stages, from database design, user interface construction to developing advanced functions such as product suggestions or smart search. Applying Next.js on the frontend combined with MySQL and Node.js on the backend helps the system achieve high performance, easy maintenance and expansion.

However, the development process also encountered many difficulties. One of the major problems is processing large data from the user side, including product images and user information. Due to limited storage capacity and security requirements, the team had to use a third-party storage service to store product images. In addition, integrating Google Books API to expand search data also poses many challenges in terms of processing speed and data format compatibility.

Building a decentralized system and authenticating users through JWT combined with localStorage also requires to ensure information security and avoid basic security vulnerabilities.

A notable point is the development of a product recommendation feature based on user interactions (add to cart, view product details, etc.). This is a factor that helps the system create a personalized experience, different from conventional online bookstores..

5.2. Comparison

	Other Systems	My System
<i>Tiki</i>	Has a relatively good product recommendation system, but limited openness to external data sources.	Connects to external APIs like Google Books API, allowing users to search for foreign books or out-of-stock books, expanding the search scope.
<i>Open-source Systems</i>	Many only support basic CRUD operations. Often lack features such as analytics, charts, tag suggestions, or flexible authentication.	Provides advanced features including user behavior analysis, statistics dashboards, tag recommendations, and customizable authentication mechanisms.
<i>Scalability & Modernity</i>	Often limited in scalability and not designed with modern architecture in mind.	Designed as a smart, modern, and highly scalable system that can be extended with features like promotional emails and advanced analytics.

Table 5.2. Comparison table

5.3. Evaluation

Target Completion Level: The system has completed over 90% of the initial target:

- Friendly, easy-to-use interface
- Clear authorization system
- Product suggestion feature by tag and user behavior
- Product rating and comment system from users
- Clear statistical charts and reports
- Integrating open data via Google Books API
- System performance evaluation
 - o The average API response time is less than 300ms for product queries, demonstrating the backend's good processing ability.
 - o Analytical charts (bar charts, pie charts) can be updated weekly, by category, helping managers easily track purchasing trends.

- The search feature works effectively, returning approximate results even if the user does not type completely accurately.

Limitations

- No online payment gateway integration
- No detailed error handling mechanism in the user interface (clearer notification display)
- Product suggestions do not use machine learning algorithms but mainly rely on heuristics (increasing tag points)

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1. Conclusion

During the implementation of the project "Smart bookstore system", the group researched, designed and successfully built a book selling website that integrates many smart features such as product suggestions based on user behavior, access authorization, product management, shopping cart, ordering and visual statistics. The system is deployed with a separate frontend-backend architecture, using Next.js for the user interface and Node.js combined with MySQL database for the server side, ensuring scalability and maintenance.

The application of product suggestion algorithms and advanced search functions has helped improve the user experience, while supporting administrators to track sales activities more effectively. In addition, the system also allows authorization between administrators and customers, contributing to ensuring security and reasonable data management.

Although there are still some limitations, such as not integrating online payment methods and lacking product review functions, the system has achieved its initial goals in terms of functionality, performance and practical application. The project is an important premise for developing more in-depth features and bringing higher practical value in the future.

6.2. Future work

Based on the results achieved and feedback during the testing process, the team proposed some future development directions to improve and enhance the system:

- Integrate online payment: Connect with popular payment gateways such as Momo, VNPay or bank cards to support fast and convenient payment.
- Build a rating and comment system: Allow users to rate products, contributing to improving reputation and product selection.
- Improve the recommendation system: Increase accuracy by incorporating additional factors such as age, favorite genre or purchase history.
- Develop the interface on mobile platforms: Optimize the system for user experience on smartphones with a responsive interface or develop a separate Android/iOS application.

- Advanced reporting system for administrators: Add charts to analyze sales trends, traffic, user behavior over time.

- Artificial Intelligence (AI) Applications: Analyze big data from users for deeper personalization, supporting decision making such as automatically adjusting prices or warehousing.

REFERENCES

1. Baeldung. (2024). Amazon recommendation system.
<https://www.baeldung.com/cs/amazon-recommendation-system>
2. Amazon Science. (2019). The history of Amazon's recommendation algorithm.
<https://www.amazon.science/the-history-of-amazons-recommendation-algorithm>
3. Stratoflow. (2024). Amazon recommendation system.
<https://stratoflow.com/amazon-recommendation-system>
4. Evdelo. (2020). Amazon's recommendation algorithm drives 35% of its sales.
<https://evdelo.com/amazons-recommendation-algorithm-drives-35-of-its-sales>
5. Seller Mobile. (2023). Amazon inventory management guide for 2023: Manage stock like a pro.
<https://sellermobile.com/amazon-inventory-management-guide-for-2023-manage-stock-like-a-pro>
6. Business Model Analyst. (2024). Amazon SWOT analysis.
<https://businessmodelanalyst.com/amazon-swot-analysis/>
7. Wired. (2021). Amazon failed to protect your data: Investigation.
<https://www.wired.com/story/amazon-failed-to-protect-your-data-investigation/>
8. Tiki Engineering. (2021). Using approximate nearest neighbor for faster retrieval on recommendation system.
<https://engineering.tiki.vn/using-approximate-nearest-neighbor-for-faster-retrieval-on-recommendation-system/>
9. Google Cloud.Tiki: Google Cloud customers.
<https://cloud.google.com/customers/tiki-en>
10. Google Books API: Google. (n.d.). Google Books APIs.
<https://developers.google.com/books>
11. Amazon: Amazon. (n.d.). Amazon.com Official Site.
<https://www.amazon.com>
12. Kaggle Dataset - Books Dataset by Saurabh Bagchi: Bagchi, S. (2020). Books Dataset. Kaggle.
<https://www.kaggle.com/datasets/saurabhbhagchi/books-dataset>

APPENDIX

Appendix A: Tools Used

Tools / Technology :	Intended Use
Next.js :	Frontend Development
Node.js + Express :	Backend API
MySQL :	Relational Database
Tailwind CSS :	Interface Design
Chart.js / Recharts :	Statistical Charting
JWT / localStorage :	User Authentication
Postman :	API Testing

Nguyen Thi Kim Thu

ORIGINALITY REPORT

6%	5%	1%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

- 1 Submitted to International University - VNUHCM 2%
Student Paper
- 2 fastercapital.com 1%
Internet Source
- 3 file.allitebooks.com <1%
Internet Source
- 4 Submitted to Foreign Trade University <1%
Student Paper
- 5 Submitted to University of Greenwich <1%
Student Paper
- 6 etd.astu.edu.et <1%
Internet Source
- 7 openaccess.city.ac.uk <1%
Internet Source
- 8 Submitted to University College Dublin (UCD) <1%
Student Paper
- 9 askmeoffers.com <1%
Internet Source

10	dspace.cvut.cz Internet Source	<1 %
11	Submitted to Father Saturnino Urios University Student Paper	<1 %
12	Submitted to HELP UNIVERSITY Student Paper	<1 %
13	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1 %
14	arxiv.org Internet Source	<1 %
15	cashflowinventory.com Internet Source	<1 %
16	www.openpr.com Internet Source	<1 %
17	eprints.nottingham.ac.uk Internet Source	<1 %
18	www.oasis-rainharvesting.co.uk Internet Source	<1 %
19	Submitted to Technical University of Cluj-Napoca Student Paper	<1 %
20	dl.lib.uom.lk Internet Source	<1 %

21	publik.tuwien.ac.at Internet Source	<1 %
22	umpir.ump.edu.my Internet Source	<1 %
23	www.denverservers.com Internet Source	<1 %
24	repository.tudelft.nl Internet Source	<1 %
25	cdnc.heyzine.com Internet Source	<1 %
26	github.coventry.ac.uk Internet Source	<1 %
27	lintang.staff.gunadarma.ac.id Internet Source	<1 %
28	odysseymagazine.com Internet Source	<1 %
29	Submitted to Segi University College Student Paper	<1 %
30	Zerman, Oğuz. "Effect of the Number of Stories and Aspect Ratio on the Seismic Performance of Base-Isolated Buildings", Middle East Technical University (Turkey), 2024 Publication	<1 %

31	brightideas.houstontx.gov Internet Source	<1 %
32	espace.curtin.edu.au Internet Source	<1 %
33	home.simula.no Internet Source	<1 %
34	libstore.ugent.be Internet Source	<1 %
35	www.coursehero.com Internet Source	<1 %

Exclude quotes Off
Exclude bibliography On

Exclude matches Off