

**FPT UNIVERSITY**  
**SUBJECT NAME: JAVA WEB APPLICATION DEVELOPMENT**

**ASSIGNMENT REPORT  
ONLINE BOOKSTORE WEBSITE**

**Submitted By:  
Nguyen Tram Anh - Roll Number: HE182498**

**March 2024**

## TABLE OF CONTENTS

TABLE OF CONTENTS.....	2
PREFACE.....	2
INTRODUCTION.....	5
CHAPTER 1: THEORETICAL BASIS.....	6
1.1. Understanding of HTML.....	6
1.1.1. Overview of HTML.....	6
1.1.2. Basic Concepts in HTML.....	6
1.2. Understanding of CSS.....	7
1.2.1. Overview of CSS.....	7
1.2.2. Basic Concepts in CSS.....	9
1.3. Overview of Java.....	10
1.3.1. Introduction.....	10
1.3.2. Java Installation.....	10
1.3.3. Java Basics.....	11
1.3.4. Basic Data Types.....	12
1.3.5. Variables in Java.....	13
1.3.6. Constants in Java.....	14
1.3.7. Operators in Java.....	14
1.3.8. Java Servlets.....	12
1.3.9. JSP (Java Server Page).....	12
1.3.10. Introduction to Microsoft SQL Server Database Management System.....	16
1.4. Introduction to Bootstrap.....	19
1.4.1. What is Bootstrap?.....	19
1.4.2. Why use Bootstrap?.....	19
1.4.3. History of Bootstrap.....	20
1.4.4. Bootstrap Versions.....	20
CHAPTER 2: OVERVIEW, ANALYSIS AND SYSTEM DESIGN.....	21
2.1. Overview: Bookabook - Online Bookstore.....	21
2.1.1. Introduction.....	21
2.1.2. Current Situation Assessment.....	21

2.1.3. Website's Purpose.....	22
2.1.4. System Design Solutions.....	23
2.2. System Analysis and Design.....	23
2.2.1. Detailed Website Requirements.....	23
2.2.2. Diagrams.....	24
2.2.3. Database Design.....	31
CHAPTER 3: BUILDING WEBSITE.....	33
3.1. Home Interface.....	33
3.2. login Interface.....	34
3.3. Register Interface.....	34
3.4. Shopping Interface.....	35
3.5. Admin Interface.....	39
3.6. Header & Footerr.....	41
CONCLUSION.....	44
REFERENCES.....	45

## PREFACE

This project represents the culmination of my work in PRJ301 - Java Web Application Development at FPT University during the Spring Semester of 2024. It was undertaken to gain a deeper understanding of Web Development Fundamentals, Database Management, and a strong foundation in Java Programming. This final project allowed me to apply these learnings to write an online bookstore website which helps customers display and search through a product catalog, making it easier to find what they need.

This document details the design and implementation of a Java web application, emphasizing its user interface, interactivity, and adherence to the Model-View-Controller (MVC) pattern. It is intended for my instructor and fellow PRJ301 students.

I would like to express my sincere gratitude to my instructor, Mrs. Do Thi Loan for her guidance and support throughout this project. Her valuable insights and suggestions were instrumental in shaping the direction and success of this work.

Finally, I encourage any feedback or questions you may have regarding this project. It has been a rewarding experience, and I welcome the opportunity to discuss my learnings and insights further.

## INTRODUCTION

The global and Vietnamese e-commerce landscapes are experiencing a surge in growth. Digital technologies offer significant cost savings by reducing intermediary transportation and transaction expenses, while also offering the invaluable benefit of time saved. This shift allows individuals to dedicate more time to other pursuits.

Furthermore, e-commerce empowers consumers with the ability to conduct their own automated searches for various needs, receiving information curated specifically to their interests. The convenience of online shopping has transformed the way individuals interact with retail, allowing them to purchase almost anything from the comfort of their homes.

Inspired by this evolving landscape, I chose the topic "Online Bookstore Website" for this project. This choice reflects my desire to contribute to the expanding world of e-commerce and explore the potential of online bookstores.

## CHAPTER 1: THEORETICAL BASIS

### 1.1. Understanding of HTML

#### 1.1.1. Overview of HTML

HTML (HyperText Markup Language) is a markup language designed to create web pages with structured information presented on the World Wide Web. HTML is defined as a simple application of SGML and is used in organizations that require complex publishing requirements.

The name "HyperText Markup Language" has the following meanings:

- Markup: HTML is a language of markup tags. These tags define how the corresponding text is displayed on the screen.

- Language: HTML is a language similar to programming languages, but simpler. It has a strict syntax for writing commands to render text. Recognized and reused keywords are determined by the Internet community. For example, "b" represents bold, "ul" represents an unordered list, etc.
- Text: HTML primarily serves to format and present text. Other components such as images, audio, and animations are embedded within text.
- HyperText: HTML allows for linking multiple text documents scattered throughout the Internet. It serves to conceal the complexity of the Internet from the user. Internet users can read text without needing to know where the text is located or how the complex system is built.

### **1.1.2. Basic Concepts in HTML**

#### *Definition of Basic HTML Tags:*

- The `<head>...</head>` tag: Creates the header section of a web page.
- The `<title>...</title>` tag: Defines the title of the web page that appears in the browser's title bar. This tag is mandatory and provides a descriptive string for the web page.
- The `<body>...</body>` tag: Contains all the information that appears on the web page. The content within this tag is visible on the web page.

#### *Other formatting tags:*

- The `<p>...</p>` tag: Creates a new paragraph.
- The `<font>...</font>` tag: Changes the font, size, and color of the text.
- The `<table>...</table>` tag: Formats a table on the web page. After declaring this tag, you need to declare `<tr>` tags for rows and `<td>` tags for columns along with their attributes.
- The `<img>` tag: Allows you to insert images into the web page. This tag is a self-closing tag.

- The `<a>...</a>` tag: Used for hyperlinking between web pages or linking to internet addresses, emails, or local file addresses (UNC).
- The Input tags: The Input tags allow users to input data or perform certain actions. They include various types such as text, password, submit, button, reset, checkbox, radio, and image.
- The `<textarea>...</textarea>` tag: Allows users to input multiple lines of text. There is no limit to the maximum length of the text within this tag.
- The `<select>...</select>` tag: Allows users to select an item from a predefined list. If users can select only one item, it functions like a combobox. If users can select multiple items at once, it functions like a listbox.
- The Form tag: When you want to submit user-entered data from the web page's client-side to the server-side, you can use the POST and GET methods within the form tag. A web page can have multiple form tags, but these form tags should not be nested. Each form tag will have an action attribute pointing to a different page.

## **1.2. Understanding of CSS**

### **1.2.1. Overview of CSS**

CSS (Cascading Style Sheets) is used to describe the presentation of documents written in HTML and XHTML. CSS stands for Cascading Style Sheets, and it is essentially a way to add display styles (such as font, size, color, etc.) to a web document.

Benefits:

- Minimizes the cluttering of HTML code by using style rules (e.g., bold, italic, underline, color) to separate the content of a web page from its presentation, resulting in cleaner source code and easier content updates.
- Allows for the creation of styles that can be applied to multiple web pages, avoiding the need to repeat formatting for similar web pages.

- Provides convenience by placing all the attributes of a tag in a separate file with the extension ".css". CSS breaks the limitations of web design by allowing you to manage formatting and layout across multiple different pages with just one CSS file.
- Web developers can define attribute styles for certain HTML tags and then reuse them on multiple pages.

CSS can be declared in various ways. You can place your CSS code within the `<head>...</head>` tags, or you can write it in a separate file with the ".css" extension. Additionally, you can also place CSS code within individual HTML tags. However, the priority of CSS depends on the placement method. The priority order of CSS is as follows (The priority decreases from top to bottom):

1. Style defined within individual HTML tags.
2. Style defined within the `<head>` section.
3. Style defined in an external ".css" file.
4. Default browser styles.

### **1.2.2. Basic Concepts in CSS**

Text Properties:

- **font-size: 12px** - This defines the size of the font. In this case, the font size is set to 12 pixels.
- **font-family: Arial, Helvetica, sans-serif** - This specifies the font family that the text should be displayed in. The order of the fonts listed indicates the browser's preference. If "Arial" is not available, the browser will try "Helvetica" and then a generic sans-serif font.
- **font-weight: bold** - This makes the font appear bold.
- **font-style: italic** - This slants the font characters to appear italic.
- **text-decoration: underline** - This draws a line under the text, effectively underlining it.
- **text-transform: uppercase** - This converts all the letters in the text to uppercase.

- **color: red** - This sets the color of the text to red.

Properties to style an object's appearance and position on a webpage:

- **Dimensions:** Properties like width and height control the size of the object.
- **Background:** Define a background image using background: url(image.jpg) or a solid color using background: #CCC (hex color code).
- **Borders:** The border property allows you to specify the thickness, style (e.g., solid, dashed), and color of the object's border.
- **Spacing:** margin defines the spacing between the object and its surrounding elements, while padding sets the spacing between the object's border and its content. The values for both properties typically follow a format like top right bottom left, defining the spacing for each side of the object.

### 1.3. Overview of Java

#### 1.3.1. *Introduction*

The Java programming language is a general-purpose, class-based, object-oriented programming language designed for having lesser implementation dependencies. It is a compiled language and not an interpreted language. It is a concurrent language and not a sequential language.

Java is a statically typed language. It is a strongly typed language. It is a multi-threaded language. It is a platform-independent language.

Java is a popular programming language for building web applications, mobile applications, and desktop applications. It is also used for developing enterprise applications, scientific applications, and games.

#### 1.3.2. *Java Installation*

There are two main ways to install Java:

##### 1. Installing Individual Components:

- **Java Development Kit (JDK):** This core development kit provides the tools

necessary to create Java applications, including the Java compiler (javac), the Java runtime (java), and various other utilities. Download the JDK from the official Oracle website: <https://jdk.java.net/18/>.

- **Java Runtime Environment (JRE):** This allows you to run existing Java applications without any development capabilities. Download the JRE from the official Oracle website: <https://java.com/en/download/>.

## 2. Installing an Integrated Development Environment (IDE):

Many IDEs come pre-packaged with both the JDK and JRE, offering a convenient solution for development. Popular options include:

- **NetBeans:** <https://netbeans.org/>
- **IntelliJ IDEA:** <https://www.jetbrains.com/idea/>
- **Eclipse:** <https://www.eclipse.org/>

### General Installation Steps:

1. Download the desired installer (JDK, JRE, or IDE) from a trusted source.
2. Run the installer and follow the on-screen instructions.
3. Verify the installation by opening a command prompt and typing `java -version`.  
You should see information about the installed Java version.

### 1.3.3. Java Basics

#### Declaring Java Code

##### *Opening tag:*

```
// Java code starts here
```

##### *Closing tag:*

```
// Java code ends here
```

#### *Writing comments:*

```
// This is a single line comment
```

```
/* This is a multi-line comment */
```

## Printing Java Values

### Syntax:

```
System.out.println(information);
```

### Information:

- String:

```
System.out.println("Hello, Java!");
```

- Variable:

```
int age = 20;
```

```
System.out.println("Age: " + age);
```

- Combining strings and variables:

```
String name = "Alice";
```

```
System.out.println("Name: " + name + ", Age: " + age);
```

## 1.3.4. Basic Data Types

### 1. Boolean:

- True or False

### 2. Integer:

- Decimal, octal, hexadecimal

- Examples:

```
int number1 = 123; // Decimal
```

```
int number2 = -123; // Negative number
```

```
int number3 = 0123; // Octal  
int number4 = 0x1A; // Hexadecimal
```

### 3. **Float/Double:**

- Examples:

```
double number5 = 1.234;  
  
double number6 = 1.2e3; // 1.2 * 1000 = 1200
```

### 4. **String:**

- Each character takes 1 byte
- No size limit
- Example:

```
String str1 = "Hello, world!";
```

### 5. **Array:**

- Stores a collection of elements of the same data type
- Example:

```
int[] numbers = {1, 2, 3, 4};
```

### 6. **Object:**

- Create objects from a defined class
- Example:

```
class Car {  
  
    // ...  
  
}
```

```
Car car = new Car();
```

## Data Type Conversion

### Syntax:

```
(data_type) variable;
```

### Example:

```
int price = 50000;  
  
int quantity = 100000;  
  
double total = (double) (price * quantity);
```

## 1.3.5. Variables in Java

### Declaration:

```
// Declare a variable with a name and an initial value  
  
int age = 20;
```

### Naming conventions:

- Starts with a letter, underscore (\_), or dollar sign
- Cannot be a keyword
- Should start with a capital letter

### Example:

```
int _age; // Valid  
  
int 123age; // Invalid  
  
int $age; // Valid
```

### String interpolation:

```
String name = "Alice";
```

```
System.out.println("Hello, " + name + "!");
```

### 1.3.6. Constants in Java

- Values that do not change during program execution
- Syntax:

```
final int MAX_SIZE = 100;
```

#### Naming conventions:

- Same as variables but in uppercase
- Cannot be interpolated in strings

#### Example:

```
final int MAX_SIZE = 100;  
  
System.out.println("Max size: " + MAX_SIZE);
```

### 1.3.7 Operators in Java

Operators are symbols that perform specific operations on values or variables in Java. They play a crucial role in manipulating data, controlling program flow, and performing various computations. Here's an overview of some key operators in Java:

#### Arithmetic Operators:

- Used for basic mathematical operations like addition (+), subtraction (-), multiplication (\*), division (/), and modulo (%) which gives the remainder.

#### Example:

```
int sum = 10 + 20;  
int difference = 30 - 5;  
int product = 4 * 6;  
int quotient = 12 / 3;  
int remainder = 17 % 5;
```

## Assignment Operators:

- Used to assign values to variables. The most common is the simple assignment (=), but there are others like += (add and assign), -= (subtract and assign), \*= (multiply and assign), etc.

### Example:

```
int age = 25;  
age += 5; // Equivalent to age = age + 5;
```

## Comparison Operators:

- Used to compare values and return boolean results (true or false). These include == (equal to), != (not equal to), < (less than), > (greater than), <= (less than or equal to), and >= (greater than or equal to).

### Example:

```
int num1 = 10;  
int num2 = 15;  
boolean isEqual = num1 == num2; // false  
boolean isLessThan = num1 < num2; // true
```

## Logical Operators:

- Used to combine boolean expressions. These include && (and), || (or), and ! (not).

### Example:

```
boolean isAdult = age >= 18;  
boolean hasId = true;  
boolean canVote = isAdult && hasId; // true (both conditions must  
be true)  
boolean canTravel = hasId || isAdult; // true (at least one  
condition must be true)
```

## **Other Operators:**

- Java also has operators like increment (++) , decrement (--) , conditional (?:) , bitwise (used for bit-level operations), and others, each with specific functionalities.

Understanding operators is fundamental to writing effective and efficient Java programs. They allow you to manipulate data, make decisions based on conditions, and control the flow of your program.

### **1.3.8. Java Servlets**

- **Definition:** Servlets are Java programs that extend the capabilities of a web server. They handle HTTP requests and responses, dynamically generate web content, and interact with databases and other resources.
- **Key features:**
  - **Request handling:** Servlets can access information from the user's request, such as form data, headers, and cookies.
  - **Response generation:** Servlets can dynamically create HTML content, including text, images, and links.
  - **Scalability:** Servlets are designed to handle multiple requests concurrently, making them suitable for high-traffic websites.
  - **Integration with other Java technologies:** Servlets can interact with databases, session management APIs, and other Java libraries.

### **1.3.9. JSP (Java Server Pages):**

- **Definition:** JSPs are HTML pages with embedded Java code. They provide a way to separate the presentation layer (HTML) from the business logic (Java code).
- **Key features:**
  - **Separation of concerns:** JSPs keep HTML code clean and organized, while Java code handles complex logic.
  - **Easy to learn:** Developers familiar with HTML can pick up JSPs quickly.
  - **JSP tags:** JSPs provide special tags for various functionalities like including dynamic content, interacting with beans (Java objects), and

controlling program flow.

- **Compilation:** When a JSP page is accessed, the server translates it into a servlet class behind the scenes.

#### **1.3.10. Introduction to Microsoft SQL Server**

Microsoft SQL Server is a relational database management system (RDBMS) developed by Microsoft. It is one of the most popular RDBMSs in the world, used by businesses of all sizes to store and manage data.

##### **Features and benefits:**

- **High performance:** SQL Server is known for its speed and ability to handle large volumes of data.
- **Easy to use:** SQL Server provides a graphical user interface (GUI) and simplified management tools.
- **Scalability:** SQL Server can be scaled to meet the needs of growing businesses.
- **High security:** SQL Server provides strong security features to protect your data.
- **Integration with other Microsoft products:** SQL Server integrates easily with other Microsoft products such as Windows, Office, and Visual Studio.

##### **Applications:**

- **Storing business data:** SQL Server can be used to store business data such as customer information, orders, products, etc.
- **Web application development:** SQL Server can be used to store data for web applications.
- **Data analysis:** SQL Server provides powerful tools for analyzing data and making informed decisions.
- **Reporting:** SQL Server can be used to create detailed reports on your data.

#### **UML in Microsoft SQL Server**

UML (Unified Modeling Language) is a standardized modeling language used to model software systems. UML can be used to model SQL Server systems, helping developers and designers better understand the system and improve the development process.

## **Benefits of using UML in SQL Server:**

- **Improved communication:** UML helps developers and designers communicate more effectively about the system.
- **Increased understanding:** UML helps developers and designers better understand the system.
- **Improved quality:** UML helps detect and fix errors earlier in the development process.
- **Increased maintainability:** UML makes the system easier to maintain.

## **Types of UML diagrams commonly used in SQL Server:**

- **Use case diagram:** This diagram describes the system's functionality and how users interact with the system.
- **Class diagram:** This diagram describes the classes in the system and the relationships between the classes.
- **State diagram:** This diagram describes the different states of an object in the system.
- **Activity diagram:** This diagram describes the flow of data in the system.

## **1.4. Introduction to Bootstrap**

### **1.4.1. What is Bootstrap?**

Bootstrap is a framework consisting of HTML, CSS, and JavaScript that simplifies website development by providing pre-built components and responsive design features. It allows you to create websites that are user-friendly and adapt seamlessly to different screen sizes, including mobile devices, tablets, and desktops.

### **1.4.2. Why use Bootstrap?**

- **Ease of use:** Built upon fundamental web development languages (HTML, CSS, and JavaScript), Bootstrap requires minimal knowledge to implement. This makes it accessible to both beginners and experienced developers alike.
- **Responsive design:** Bootstrap comes with built-in responsive functionalities, eliminating the need to write separate code for different devices. This saves time and ensures your website displays optimally on various screen sizes.

- **Browser compatibility:** Bootstrap offers compatibility with all major browsers, including Chrome, Firefox, Internet Explorer (from version 9 onwards), Safari, and Opera.

#### **1.4.3. History of Bootstrap**

- Originally named **Twitter Blueprint**, Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter to promote design consistency across internal tools.
- After a few months of development, it was renamed Bootstrap and released as an open-source project on August 19, 2011.
- Since its release, Bootstrap has been maintained by Mark Otto, Jacob Thornton, and a core team of developers, alongside contributions from a large and active user community.

#### **1.4.4. Bootstrap Versions**

- **Version v1.0:** Released on August 19, 2011, this initial version lacked mobile support.
- **Version v2.0:** Released on January 31, 2012, it introduced a 12-column grid layout system, added new components, and revamped existing ones, but still lacked mobile support.
- **Version v3.0:** Released on August 19, 2013, it offered redesigned components following the flat design trend and incorporated mobile-first design principles.
- **Version v4.0:** Released on January 19, 2018, it was a complete rewrite of Bootstrap 3 and was widely praised for its improved ease of use.

## **CHAPTER 2: OVERVIEW, ANALYSIS AND SYSTEM DESIGN**

### **2.1. Overview: Bookabook - Online Bookstore**

#### **2.1.1. Introduction**

Bookabook is an aspiring online bookstore offering a wide selection of high-quality and trendy books. Our website aims to provide a youthful and dynamic experience for book lovers of all ages.

### **2.1.2. Current Situation Assessment**

#### **Advantages of an Online Bookstore:**

- **Profit Potential:** Increased customer reach translates to higher sales potential. By focusing on product quality, excellent customer service, competitive pricing, and informative product descriptions, Bookabook can ensure customer satisfaction and achieve higher profits.
- **Flexibility:** Bookabook operates anywhere, anytime. Customers can browse our collection, find recommendations, and make purchases at their convenience, without limitations of traditional store hours.
- **Enhanced Reputation:** A well-designed and informative website strengthens Bookabook's reputation as a reliable and trustworthy bookstore.
- **Convenience for Customers:** Busy schedules often make in-person shopping a challenge. Bookabook's online platform allows customers to save valuable time by shopping from the comfort of their homes.

#### **Disadvantages:**

- **Internet Security Concerns:** Vietnam's online security landscape is evolving. Malicious actors may attempt to infiltrate online systems, posing a threat to Bookabook's website and customer data.
- **Challenges with Trust:** Building customer trust in online product quality can be difficult. While product descriptions and images might showcase exceptional books, some customers might hesitate due to concerns about product misrepresentation upon delivery.
- **Seller's Limited Online Payment Knowledge:** Understanding and implementing secure online payment systems is crucial for Bookabook's success.

### **2.1.3. Website's Purpose**

Bookabook's goal is to establish a user-friendly online sales platform. This platform will allow customers to browse our book collection, place orders online, and enable administrators to manage product information and customer accounts.

Our research on successful online bookstores like Nha Nam, Tiki, and Fahasa has provided valuable insights into the functionalities and goals Bookabook's website

needs to achieve.

#### **2.1.4. System Design Solutions**

##### **Functionalities:**

- View Books
- Register as a User
- Login to Existing Account
- Search for Books
- Add Books to Cart and Order

## **2.2. System Analysis and Design**

### **2.2.1. Detailed Website Requirements**

- **Book Categories:** Display a clear and organized categorization of books (e.g., Fiction, Non-Fiction, Children's Books, etc.)
- **Detailed Product Pages:** Provide comprehensive information about each book, including title, author, description, reviews, and high-quality images.

##### **Functionalities for Users:**

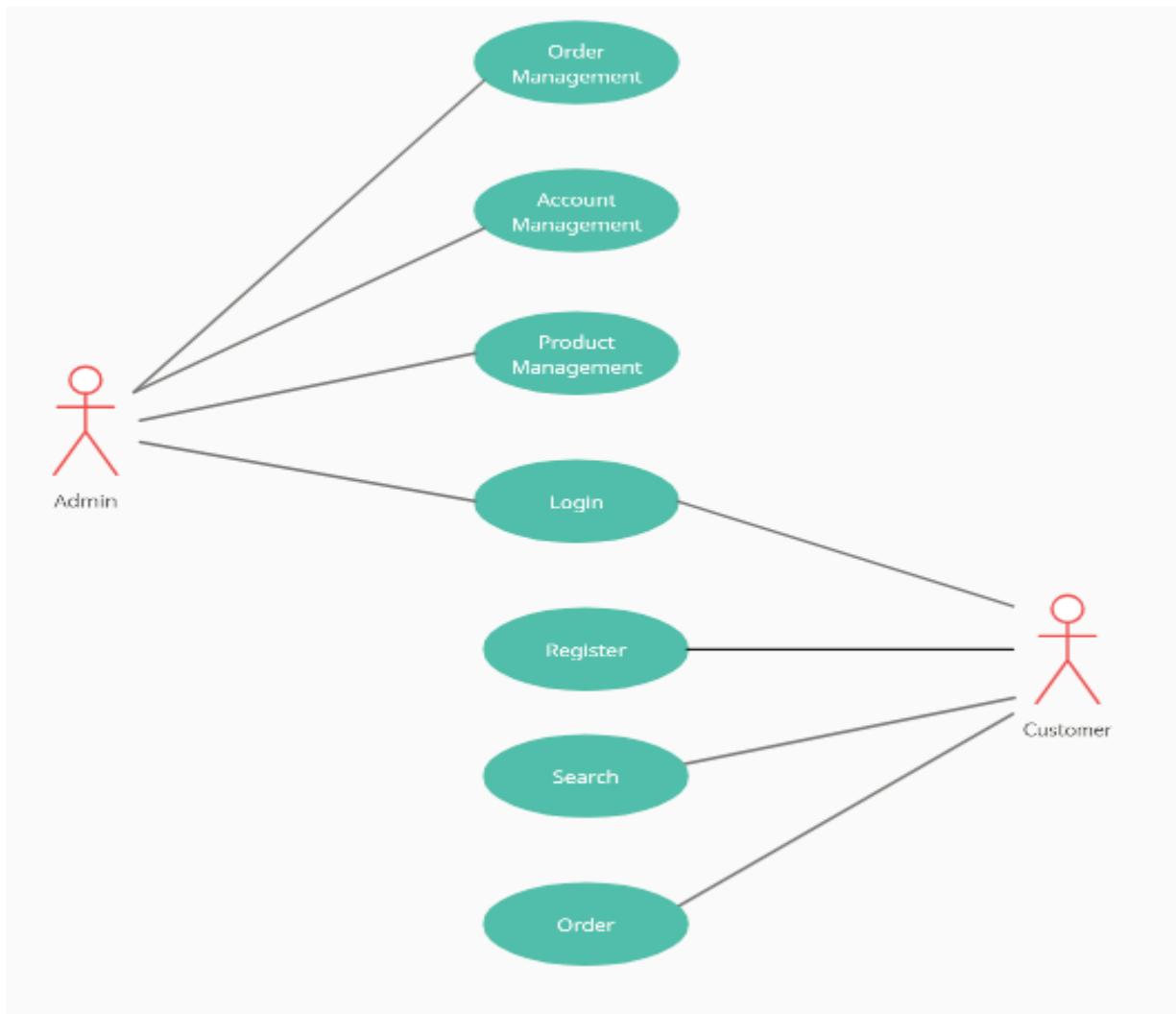
- Browse book information and descriptions freely.
- Utilize the search function to find specific books.
- Register for a user account to streamline the checkout process and track past purchases.

##### **Functionalities for Administrators:**

- Manage user accounts: Add, edit, and delete user information.
- Manage book listings: Add new books, edit existing book information, and remove books from the store.
- Process customer orders and manage order fulfillment.

### **2.2.2. Diagrams**

#### **Use Case Diagrams**



*Figure 2.1. General Use Case Diagram*

**Description:** Customers can register, login, search, and order. Admin can login, manage products, and manage accounts.

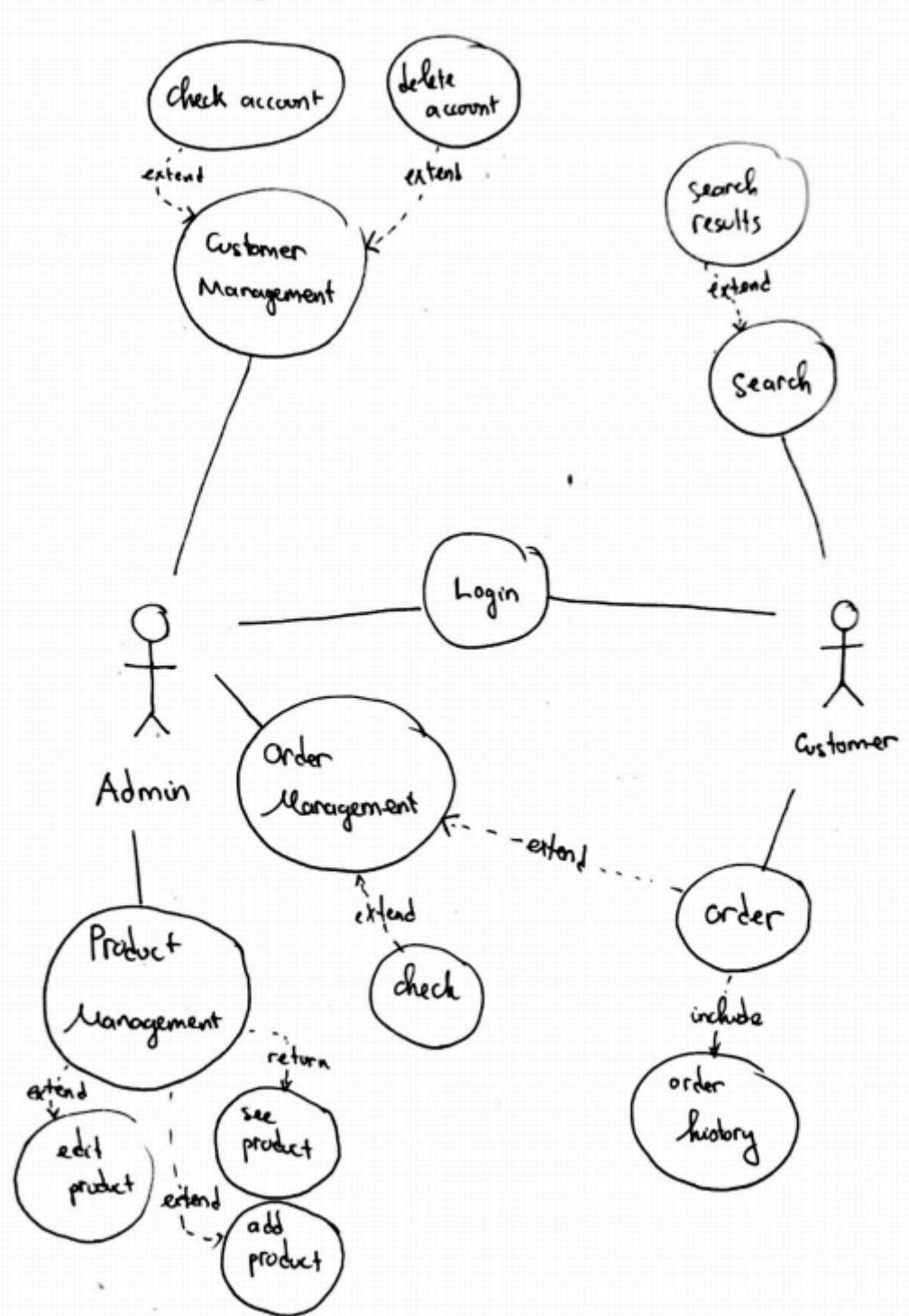
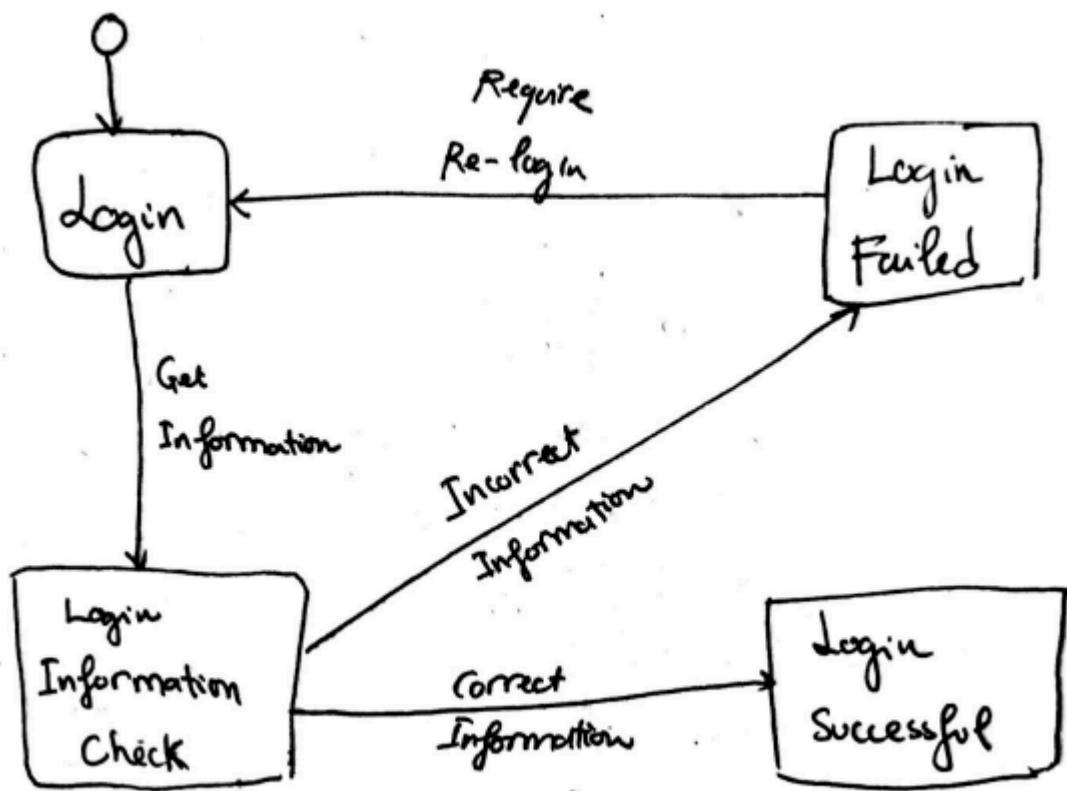


Figure 2.2. Decomposition Use Case Diagram



*Figure 2.3. Login function state transition diagram*

**Description:** The user requests to log in and enter information into the input form. If the system verifies that the input information is correct, it will indicate a successful login. If the login information is incorrect, it will indicate a login failure and request to log in again.

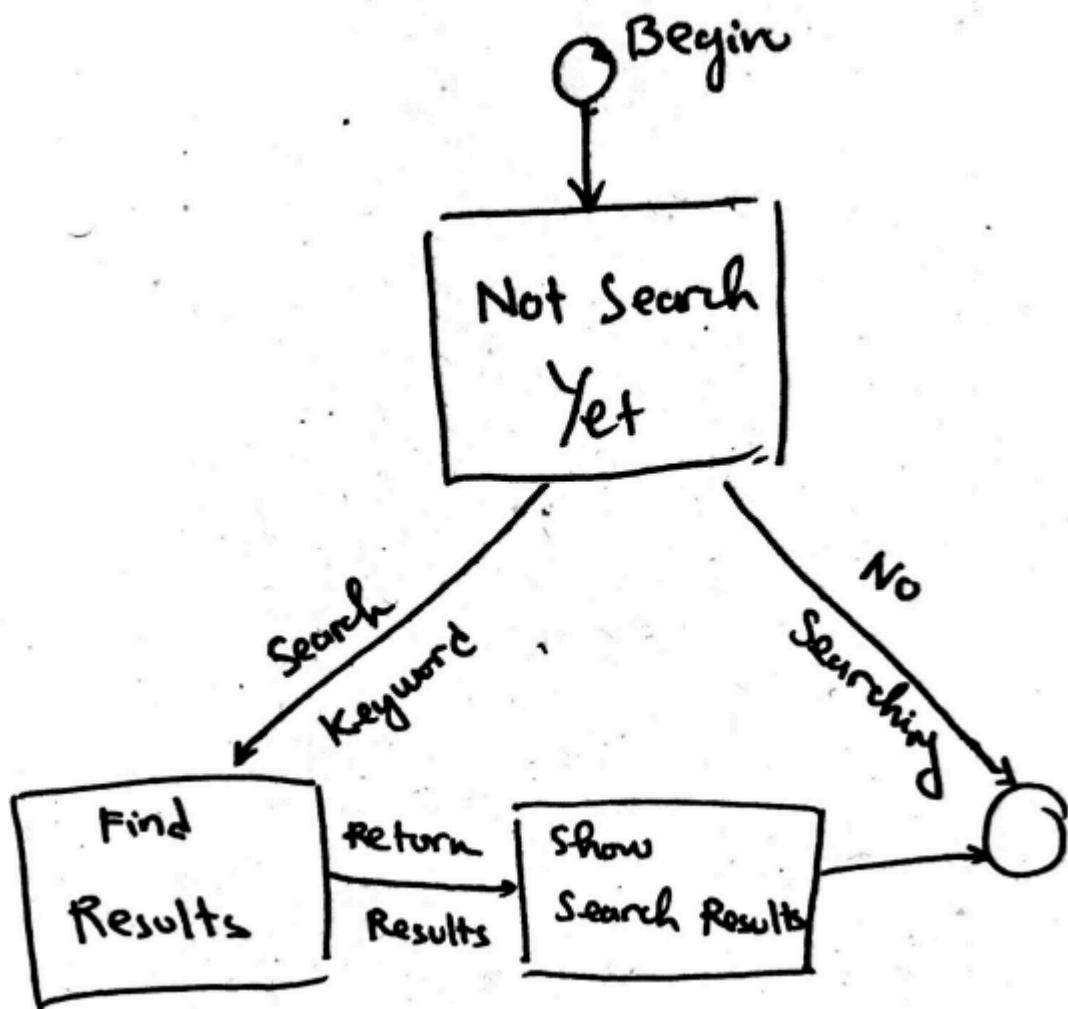


Figure 2.4. State transition diagram for the search use case

## Class Diagram

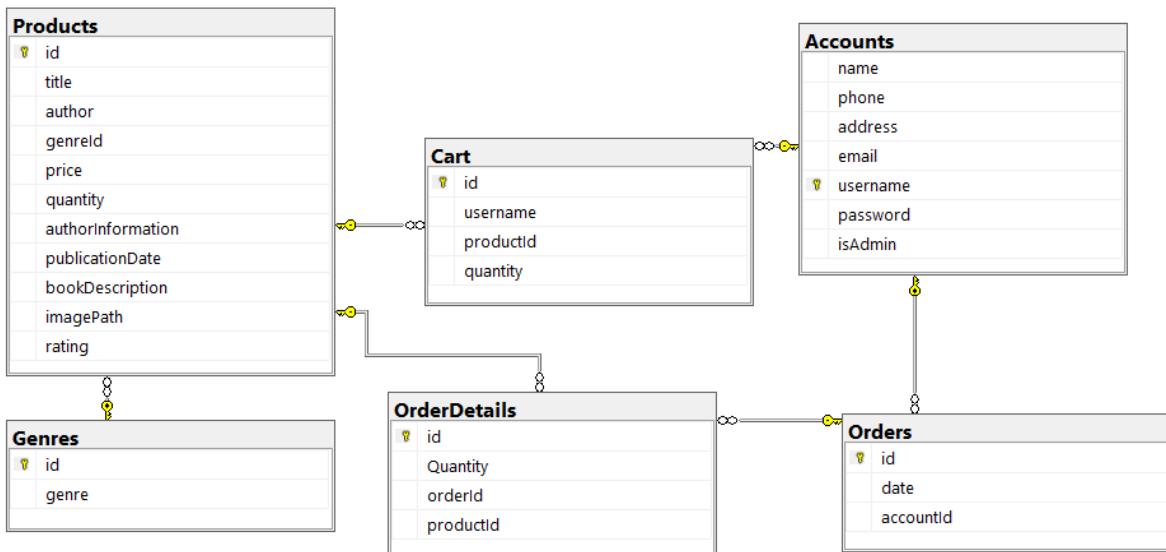


Figure 2.5. Class Diagram

### 2.2.3. Database Design

#### a. Accounts Table

Name	Data type	Note
username	varchar(255)	Primary key
password	varchar(max)	
name	varchar(max)	
phone	varchar(10)	
address	varchar(max)	
email	varchar(max)	
isAdmin	bit	

#### b. Genre table

Name	Data type	Note

<b>id</b>	varchar(255)	Primary key
<b>genre</b>	varchar(255)	

### c. Products Table

<b>Name</b>	<b>Data type</b>	<b>Note</b>
<b>id</b>	varchar(255)	Primary key
<b>title</b>	varchar(max)	
<b>author</b>	varchar(max)	
<b>genreId</b>	varchar(255)	Foreign key Genre(id)
<b>price</b>	float	
<b>quantity</b>	float	
<b>authorInformation</b>	varchar(max)	
<b>publicationDate</b>	varchar(max)	
<b>bookDescription</b>	varchar(max)	
<b>imagePath</b>	varchar(max)	
<b>rating</b>	float	

### d. Cart Table

<b>Name</b>	<b>Data type</b>	<b>Note</b>
<b>id</b>	int	Primary key
<b>username</b>	varchar(255)	Foreign key Account(username)
<b>productId</b>	varchar(255)	Foreign key Account(username)
<b>quantity</b>	int	

### e. Orders Table

<b>Name</b>	<b>Data type</b>	<b>Note</b>
<b>id</b>	varchar(255)	Primary key

date	date	
accountId	varchar(255)	Foreign key Account(username)

#### f. OrderDetails Table

Name	Data type	Note
id	varchar(255)	Primary key
Quantity	int	
orderId	varchar(255)	Foreign key Order(id)
productId	varchar(255)	Foreign key Product(id)

## CHAPTER 3: BUILDING WEBSITE

### 3.1. Home Interface

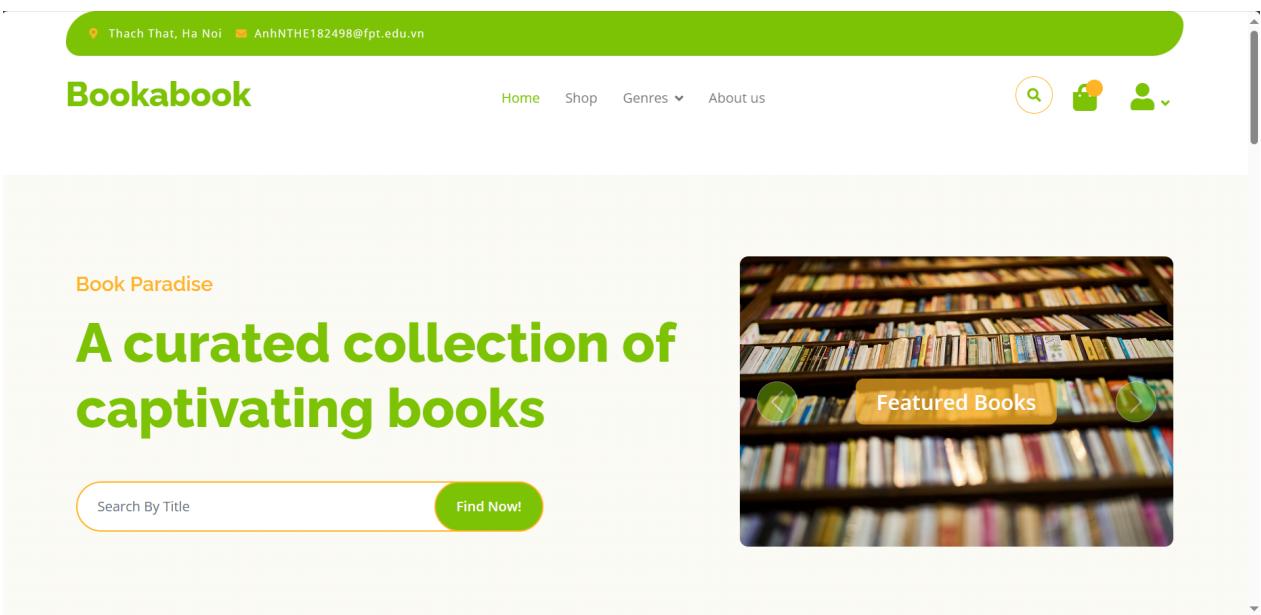


Image 3.1. Giao diện trang chủ

## My Favorite Books

[Featured](#)[Latest](#)

*Image 3.2. Giao diện trang chủ*

### 3.2. Login Interface



Please log in

 Username  
tramanh Password Remember me**Log in**[Forgot Password?](#)[Not registered? Create an account](#)

*Image 3.2. Login Interface*

### 3.3. Register Interface

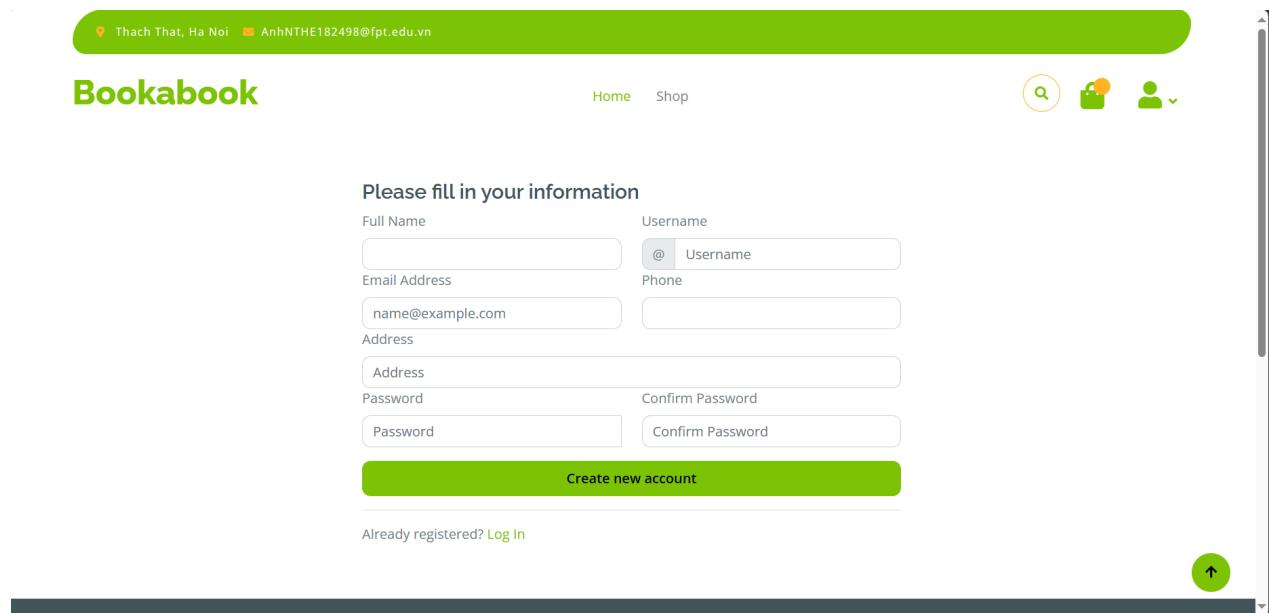


Image 3.3. Register Interface

### 3.4 Shopping Interface

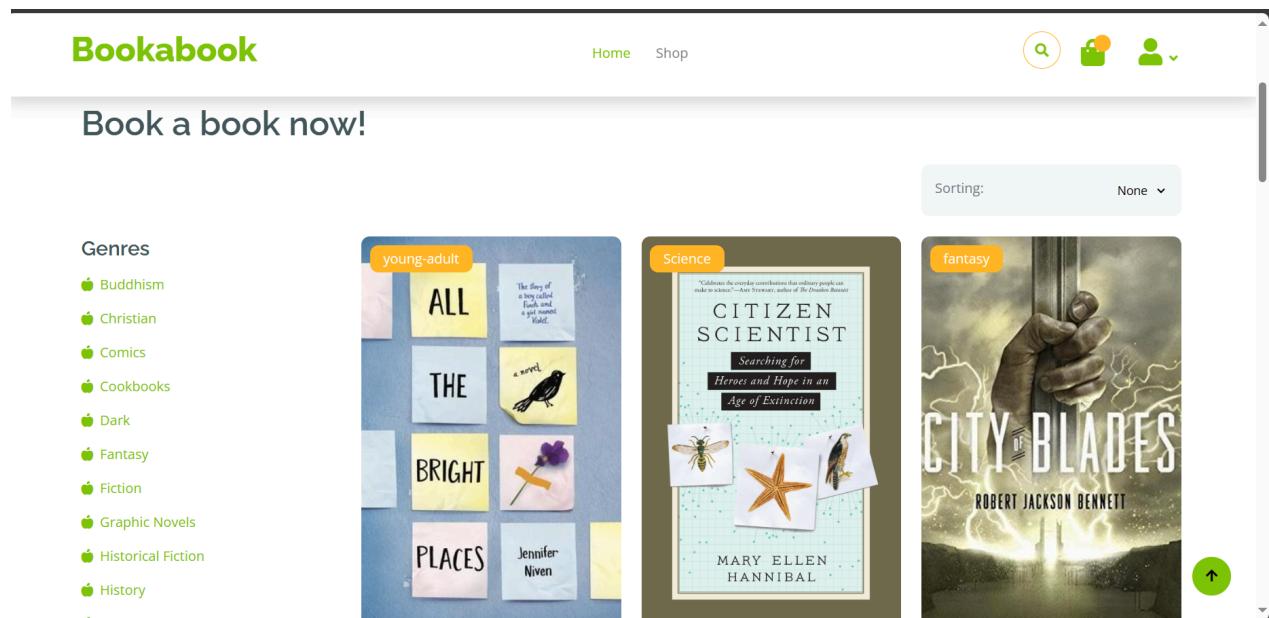


Image 3.4. Shopping Interface

### 3.5 Admin Interface

The screenshot shows the Bookabook Admin interface. At the top, there's a green header bar with user information: 'Thach That, Ha Noi' and 'AnhNTHE182498@fpt.edu.vn'. On the right, it says 'Hi tramanh'. Below the header is the Bookabook logo and a navigation menu with links to 'Home', 'Shop', 'Manage Products', 'Genres', and 'About us'. To the right of the menu are icons for search, shopping cart, and user profile. The main content area has a background image of bookshelves. The title 'Manage Products' is centered at the top of the content area. Below it, there's a breadcrumb navigation: 'Home / Manage Products'. A button labeled 'Add New Product' is visible. The table below lists a single product: 'A Show for Two' by 'A. Slobin', with a price of '\$13.99' and a quantity of '20'. There are edit and delete icons next to the product row.

Image 3.5. Product Management

### 3.6. Header & Footer

The screenshot shows the Bookabook Header and Footer sections. The header is identical to the one in Image 3.5, featuring the Bookabook logo, navigation links, and user information. The footer is a dark blue section with white text. It starts with the heading 'Book a book' and the tagline 'Home of the bookworms'. Below this is a 'Why People Like us!' section containing a paragraph about the website's mission to provide a literary haven for book lovers. To the right are sections for 'Account' (links to 'My Account', 'Shopping Cart', and 'Order History') and 'Contact' (links to address, email, and phone number). At the bottom of the footer, there's a copyright notice: '© Bookabook, All right reserved.' and credits for 'Designed By [HTML Codex](#) Distributed By [ThemeWagon](#)'. There's also a small upward arrow icon.

Image 3.6. Header

## **CONCLUSION**

### **Results A**

- Hệ thống đáp ứng được phần nào nhu cầu tin học hóa khâu quảng bá sản phẩm cho cửa hàng và xây dựng giỏ hàng của khách, tiện lợi hơn, tiết kiệm thời gian của cả hai bên mà hiệu quả không giảm.
- Hệ thống có giao diện đơn giản theo đúng xu hướng, gần gũi, thân thiện và dễ sử dụng cho cả người quản trị và khách hàng.

### **Hạn chế:**

Hệ thống vẫn còn nhiều thiếu sót cả về chức năng và giao diện.

### **Khắc phục và phát triển:**

- Hoàn thiện các chức năng còn thiếu và chưa hoàn chỉnh, các chức năng còn khá cơ bản.
- Tối ưu hóa code để cải thiện hiệu năng của website.
- Phát triển giỏ hàng để có thể thanh toán trực tuyến.
- Tăng khả năng bảo mật và chịu lỗi của website.
- Mở rộng thêm các module/plugin ứng dụng, thực hiện triển khai trên host, gán tên miền.

## **TÀI LIỆU THAM KHẢO**

[1] Trang web <http://w3schools.com>

[2] Trang web <http://getbootstrap.com>

[3] Các website bán hàng như: lazada.vn, sendo.vn, tiki.vn, capvirgo.com

Oracle Java SE Documentation:

<https://docs.oracle.com/javase/8/docs/technotes/guides/install/>

Oracle Java Tutorials: <https://docs.oracle.com/javase/tutorial/>

createme

themewagon