**Fy ỦY BAN NHÂN DÂN TP HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KÌ MÔN**

**CÔNG NGHỆ .NET**

**ĐỒ ÁN : WEBSITE BÁN SÁCH**

**Họ tên và Mã số sinh viên :**

**3120410272 - Nguyễn Tuấn Kiệt**

**3120410534 – Ngô Văn Tín**

**3120410620 – Lê Thanh Vũ**

**3120410539 – Nguỵ Đoan Toàn**

**3120410298 – Nguyễn Thành Long**

**Giảng viên hướng dẫn : ThS. Từ Lãng Phiêu**

***TP. Hồ Chí Minh, tháng 12 năm 2024***

# **LỜI CẢM ƠN**

Lời đầu tiên, chúng em xin trân thành cảm ơn quý thầy cô trong khoa Công nghệ thông tin đã hỗ trợ em trong suốt quá trình học tập, rèn luyện tại trường và khi em thực hiện đề tài này. Em xin gửi lời cảm ơn đến cô ThS**.** Từ Lãng Phiêu người đã tận tình hướng dẫn, giúp đỡ, chỉ bảo em trong suốt thời gian thực hiện đồ án tốt nghiệp. Đồng thời em xin trân trọng cảm ơn những tình cảm quý báu mà các thầy cô trong Trường Đại học Sài Gòn truyền đạt cho em những những kinh nghiệm, kỹ thuật và cách thức trong việc xây dựng đề tài này.

Thông qua đồ án tốt nghiệp này, em đã tiếp thu được vô số kiến thức bổ ích, trau dồi kỹ năng chuyên môn và củng cố nền tảng lý thuyết đã được học. Sự hướng dẫn, khuyến khích, đồng hành và hỗ trợ của giáo viên hướng dẫn đã giúp em có cơ hội nghiên cứu, áp dụng kiến thức vào thực tiễn và phát triển kỹ năng chuyên môn. Em rất biết ơn vì sự quan tâm và kiến thức mà cô đã chia sẻ với em. Thông qua đồ án tốt nghiệp lần này em mong muốn tiếp tục học hỏi và trau dồi để hoàn thiện bản thân, xứng đáng với sự tin tưởng và kỳ vọng của thầy cô.

Chúng em xin chân thành cảm ơn!

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc185410326)

[DANH MỤC CÁC HÌNH iv](#_Toc185410327)

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN vii](#_Toc185410328)

[BẢNG PHÂN CÔNG 1](#_Toc185410329)

[CHƯƠNG 1: MÔ TẢ YÊU CẦU CỦA HỆ THỐNG 2](#_Toc185410330)

[1.1. LÝ DO CHỌN ĐỀ TÀI 2](#_Toc185410331)

[1.2. KHẢO SÁT HỆ THỐNG 2](#_Toc185410332)

[1.2.1. Tổng quan về hệ thống 2](#_Toc185410333)

[1.3. CÁC HOẠT ĐỘNG CỦA HỆ THỐNG 3](#_Toc185410334)

[1.3.1. Hoạt động bán hàng 3](#_Toc185410335)

[1.3.2. Báo cáo, thống kê 4](#_Toc185410337)

[1.3.3. Cập nhật thông tin hệ thống 4](#_Toc185410338)

[1.4. Các yêu cầu của hệ thống 5](#_Toc185410339)

[1.4.1. Yêu cầu chức năng 5](#_Toc185410340)

[CHƯƠNG 2. PHÂN TÍCH CÁC YÊU CẦU CHỨC NĂNG CỦA HỆ THỐNG 6](#_Toc185410341)

[2.1. BIỂU ĐỒ USE CASE 6](#_Toc185410342)

[2.1.1. Các Use Case chính 6](#_Toc185410343)

[2.1.2. Quan hệ giữa các Use Case 7](#_Toc185410345)

[CHƯƠNG 3 : THIẾT KẾ CƠ SỞ DỮ LIỆU 13](#_Toc185410356)

[3.1. MÔ HÌNH HOÁ DỮ LIỆU 13](#_Toc185410357)

[3.3. QUAN HỆ CÁC BẢNG 25](#_Toc185410384)

[CHƯƠNG 4 : THIẾT KẾ GIAO DIỆN VÀ CÀI ĐẶT 26](#_Toc185410386)

[4.1. MÔ HÌNH KIẾN TRÚC DỰ ÁN 26](#_Toc185410387)

[4.1.1. Giới thiệu kiến trúc Onion 26](#_Toc185410389)

[4.1.2. Các lớp trong kiến trúc Onion 27](#_Toc185410390)

[4.1.3. Ưu nhược điểm kiến trúc Onion 27](#_Toc185410391)

[4.2. GIỚI THIỆU NGÔN NGỮ SỬ DỤNG 28](#_Toc185410392)

[4.2.1. Asp.Net Core 28](#_Toc185410393)

[4.2.2. Entity Framework 30](#_Toc185410395)

[4.2.3. Boostrap 31](#_Toc185410396)

[4.2.4. Jquery 32](#_Toc185410397)

[4.2.5. Hệ quản trị cơ sở dữ liệu SQL Server 33](#_Toc185410398)

[4.3. THIẾT KẾ CHI TIẾT 34](#_Toc185410399)

[4.3.1. Xem sản phẩm 34](#_Toc185410400)

[4.3.2. Đánh giá sản phẩm 34](#_Toc185410402)

[4.3.3. Giỏ hàng 35](#_Toc185410404)

[4.3.4. Danh sách yêu thích 35](#_Toc185410406)

[4.3.5. Đặt hàng 36](#_Toc185410408)

[4.3.6. Quản lý thông tin tài khoản 36](#_Toc185410410)

[4.3.7. Đơn hàng của tôi 37](#_Toc185410412)

[4.3.8. Đăng ký 38](#_Toc185410414)

[4.3.9. Đăng nhập 39](#_Toc185410416)

[4.3.10. Quản lý đơn hàng 39](#_Toc185410418)

[4.3.11. Quản lý khách hàng 40](#_Toc185410420)

[4.3.12. Quản lý sản phẩm 40](#_Toc185410422)

[4.3.13. Quản lý nhân viên 40](#_Toc185410424)

[4.3.14. Thống kê 41](#_Toc185410426)

[CHƯƠNG 5 : TỔNG KẾT VÀ ĐÁNH GIÁ 42](#_Toc185410428)

[5.1. KẾT QUẢ ĐẠT ĐƯỢC 42](#_Toc185410429)

[5.2. HƯỚNG PHÁT TRIỂN 43](#_Toc185410430)

[TÀI LIỆU THAM KHẢO 44](#_Toc185410431)

# **DANH MỤC CÁC HÌNH**

[Hình 1.1. Sản phẩm của hệ thống 4](#_Toc185410507)

[Hình 2.1 : Biểu đồ Use Case 6](#_Toc185410515)

[Hình 2.2 : Biểu đồ Use Case Đăng nhập 7](#_Toc185410517)

[Hình 2.3 : Biểu đồ Use Case Xem sản phẩm 8](#_Toc185410518)

[Hình 2.4 : Biểu đồ Use Case Quản lý sản phẩm 8](#_Toc185410519)

[Hình 2.5. Biểu đồ use case Quản lý đơn hàng 9](#_Toc185410520)

[Hình 2.6. Biểu đồ use case Đơn hàng của tôi 9](#_Toc185410521)

[Hình 2.7. Biểu đồ use case Quản lý khách hàng 10](#_Toc185410522)

[Hình 2.8. Biểu đồ use case Quản lý thông tin tài khoản 10](#_Toc185410523)

[Hình 2.9. Biểu đồ use case Quản lý nhân viên 11](#_Toc185410524)

[Hình 2.10. Biểu đồ use case Quản lý giỏ hàng 11](#_Toc185410525)

[Hình 2.11. Biểu đồ use case Quản lý danh sách yêu thích 12](#_Toc185410526)

[Hình 3.1 : Cấu trúc database của dự án 13](#_Toc185410529)

[Hình 3.2. Chi tiết bảng Authors của dự án 16](#_Toc185410531)

[Hình 3.3. Chi tiết bảng AuthorProducts của dự án 16](#_Toc185410532)

[Hình 3.4. Chi tiết bảng Banners của dự án 16](#_Toc185410533)

[Hình 3.5. Chi tiết bảng Brands của dự án. 17](#_Toc185410534)

[Hình 3-6. Chi tiết bảng Categories của dự án. 17](#_Toc185410535)

[Hình 3.7. Chi tiết bảng CategoryProducts của dự án 17](#_Toc185410536)

[Hình 3.8. Chi tiết bảng Comments của dự án. 18](#_Toc185410537)

[Hình 3.9. Chi tiết bảng Customers của dự án. 18](#_Toc185410538)

[Hình 3.10. Chi tiết bảng Employees của dự án. 18](#_Toc185410539)

[Hình 3.11. Chi tiết bảng FavouriteProducts của dự án. 19](#_Toc185410540)

[Hình 3.12. Chi tiết bảng Images của dự án 19](#_Toc185410541)

[Hình 3.13. Chi tiết bảng Orders của dự án 19](#_Toc185410542)

[Hình 3.14. Chi tiết bảng OrderDetails của dự án. 20](#_Toc185410543)

[Hình 3.15. Chi tiết bảng Products của dự án. 20](#_Toc185410544)

[Hình 3.16. Chi tiết bảng Users của dự án. 21](#_Toc185410545)

[Hình 3.17. Chi tiết bảng UserClaims của dự án. 21](#_Toc185410546)

[Hình 3.18. Chi tiết bảng Roles của dự án. 22](#_Toc185410547)

[Hình 3.19. Chi tiết bảng RoleClaims của dự án 22](#_Toc185410548)

[Hình 3-20. Chi tiết bảng UserRoles của dự án 22](#_Toc185410549)

[Hình 3.21. Chi tiết bảng UserLogins của dự án 23](#_Toc185410550)

[Hình 3.22. Chi tiết bảng UserTokens của dự án. 23](#_Toc185410551)

[Hình 3.23. Chi tiết bảng Provinces của dự án 23](#_Toc185410552)

[Hình 3.24. Chi tiết bảng Districts của dự án. 24](#_Toc185410553)

[Hình 3.25. Chi tiết bảng Wards của dự án 24](#_Toc185410554)

[Hình 3.26 : Quan hệ của các bảng trong database 25](#_Toc185410556)

[Hình 4.1 : Mô hình kiến trúc Onion 26](#_Toc185410559)

[Hình 4.2 : Giới thiệu Asp.Net Core 28](#_Toc185410565)

[Hình 4.3 : Giao diện “Xem sản phẩm” 34](#_Toc185410572)

[Hình 4.4 : Giao diện “Đánh giá sản phẩm” 34](#_Toc185410574)

[Hình 4.5 : Giao diện “Giỏ hàng” 35](#_Toc185410576)

[Hình 4.6 : Giao diện “Danh sách yêu thích” 35](#_Toc185410578)

[Hình 4.7 : Giao diện “Đặt hàng” 36](#_Toc185410580)

[Hình 4.8 : Giao diện “Quản lý thông tin tài khoản” 36](#_Toc185410582)

[Hình 4.9 : Giao diện “Đơn hàng” 37](#_Toc185410584)

[Hình 4.10 : Giao diện “Đăng ký” 38](#_Toc185410586)

[Hình 4.11 : Giao diện trang “Đăng nhập” 39](#_Toc185410588)

[Hình 4.12 : Giao diện “Quản lý đơn hàng” 39](#_Toc185410590)

[Hình 4.13 : Giao diện “Quản lý khách hàng” 40](#_Toc185410592)

[Hình 4.14 : Giao diện “Quản lý sản phẩm” 40](#_Toc185410594)

[Hình 4.15 : Giao diện “Quản lý nhân viên” 40](#_Toc185410596)

[Hình 4.16 : Giao diện “Thống kê” 41](#_Toc185410598)

# **NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................... ................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................... .....................................................................................................................................................................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

**Giảng viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

ThS. Từ Lãng Phiêu

# **BẢNG PHÂN CÔNG**

|  |  |
| --- | --- |
| MSSV – Họ Tên | Công việc |
| 3120410534 – Ngô Văn Tín (Leader) | Onion Architecture  CartController  CheckOutController  ProvincesController  Admin |
| 3120410272 – Nguyễn Tuấn Kiệt | Onion Architecture  ChatToEmployeeController  Admin |
| 3120410620 – Lê Thanh Vũ | FavouritesController  MyOrdersController |
| 3121410298 – Nguyễn Thành Long | AccountController  AccountsProfileController |
| 3120410539 – Nguỵ Đoan Toàn | HomeController  ProductsController |

**CHƯƠNG 1: MÔ TẢ YÊU CẦU CỦA HỆ THỐNG**

## **1.1. LÝ DO CHỌN ĐỀ TÀI**

Trong thời đại công nghệ số ngày nay, việc phát triển các website chất lượng cao không chỉ đòi hỏi sự hiểu biết sâu rộng về lập trình mà còn yêu cầu kiến thức chuyên sâu về các công nghệ xây dựng web. Chính vì vậy, đề tài xây dựng một ứng dụng website bán sách sẽ giúp sinh viên có cơ hội áp dụng và phát triển những kỹ năng quan trọng để chuẩn bị cho sự nghiệp sau này.

Thị trường bán lẻ trực tuyến đang ngày càng phát triển, và sách cũng là một trong những mặt hàng có nhu cầu mua sắm lớn.

Sử dụng ASP.NET Core mang lại sự ổn định, bảo mật và hiệu suất cao cho hệ thống và còn được hỗ trợ đa nền tảng. Xây dựng một website bán sách bằng ASP.NET Core đòi hỏi lập trình viên phải nắm vững các kiến thức về lập trình web, xử lý dữ liệu, bảo mật, và quản lý phiên làm việc của người dùng.

Xây dựng một ứng dụng thương mại điện tử đòi hỏi sinh viên đối mặt với những thách thức thực tế như quản lý đơn hàng, thanh toán trực tuyến, và quản lý sản phẩm. Đây là cơ hội tốt để nâng cao kỹ năng lập trình và hiểu biết về phát triển web hiện đại.

## **1.2. KHẢO SÁT HỆ THỐNG**

### **1.2.1. Tổng quan về hệ thống**

Hệ thống của nhà sách Minh Tiến chuyên bán về sách và ngoài ra còn bán thêm cả về một số mặt hàng liên quan khác như vở ghi, bút, dụng cụ học tập, đồ lưu niệm, đồ chơi giáo dục.

Hệ thống nhà sách Minh Tiến gồm các thể loại sách chính như: sách thiếu nhi, sách khoa học, sách giáo khoa và các loại sách khác.

Định hướng của hệ thống nhà sách Minh Tiến là tập trung kinh doanh các mặt hàng là sách và văn phòng phẩm và hướng tới các đối tượng khách hàng của hệ thống bao gồm: trẻ em, thanh niên, trung niên và người già.

Đến với không gian mua sắm trực tuyến của hệ thống nhà sách Minh Tiến, khách hàng có thể dễ dàng tìm thấy những cuốn sách hay, đa thể loại của nhiều nhà xuất bản, công ty sách trong và ngoài nước cùng nhiều dụng cụ học tập, văn phòng phẩm, quà lưu niệm, đồ chơi giáo dục chính hãng của những thương hiệu uy tín.

Quy mô: Diện tích của các cửa hàng thuộc hệ thống nhà sách Minh Tiến không rộng mà nhu cầu của người mua càng ngày càng lớn dẫn đến các cửa hàng thuộc hệ thống bị quá tải vào các ngày cuối tuần.

Nhân lực:

* + Nhân viên bán hàng
  + Quản lý cửa hàng

Nhược điểm: Khi khách quá đông mà diện tích của cửa hàng lại nhỏ, không đủ để phục vụ hết tất cả khách hàng.

Hướng phát triển của hệ thống hiện tại: Cần xây dưng 1 phần mềm bán các sản phẩm của hệ thống qua nền tảng trực tuyến. Phần mềm có các chức năng giúp cho khách hàng có thể mua sắm trực tuyến và thanh toán online, giúp cho nhân viên và người quản lý có thể quản lý các sản phẩm và danh mục cũng như là xem được thống kê việc bán hàng của hệ thống.

## **1.3. CÁC HOẠT ĐỘNG CỦA HỆ THỐNG**

### **1.3.1. Hoạt động bán hàng**

Hoạt động nghiệp vụ

* Khi khách hàng truy cập vào hệ thống thì hệ thống sẽ hiển thị các sản phẩm của hệ thống lên màn hình, nếu khách có nhu cầu mua mặt hàng nào thì khách có thể nhấn vào nút “Thêm vào giỏ hàng” ở mỗi sản phẩm.
* Sau khi thêm sản phẩm vào giỏ hàng thì khách nhấn vào nút “Giỏ hàng” và sau đó hệ thống sẽ hiển thị các sản phẩm mà khách vừa thêm vào giỏ hàng.
* Khách hàng tiếp tục nhấn vào nút “Đặt hàng” sau đó hệ thống sẽ hiển thị màn hình của trang Đặt hàng, khách hàng sẽ phải nhập địa chỉ giao hàng, số điện thoại người nhận, tên người nhân, ghi chú (nếu có), sau khi nhập thông tin xong thì khách hàng sẽ chọn phương thức thanh toán và sau đó nhấn nút “Mua hàng” sau đó hệ thống sẽ hiển thị tới màn hình thông báo về đơn hàng vừa đặt.
* Sau khi giao hàng tới khách hàng thành công, đơn vị vận chuyển sẽ thông báo lại với nhân viên của hệ thống về đơn hàng đã được giao, và sau đó Nhân viên hệ thống sẽ truy cập vào trang quản lý của hệ thống bán hàng và sửa trạng thái đơn hàng thành “Giao hàng thành công”.

Tài liệu khảo sát:

* A person with a white beard

  Description automatically generatedA book cover with cartoon characters

  Description automatically generatedSản phẩm

# Hình 1.1. Sản phẩm của hệ thống

### **1.3.2. Báo cáo, thống kê**

Khi có yêu cầu thống kê doanh số bán hàng Nhân viên của hệ thống sẽ vào trang quản trị của phần mềm sau đó mở phần thống kê thì hệ thống sẽ hiển thị doanh thu và số mặt hàng bán được theo từng tháng.

### **1.3.3. Cập nhật thông tin hệ thống**

Nhân viên và người quản lý có thể thêm, sửa, xóa thông tin:

* Sản phẩm.
* Thể loại.
* Thương hiệu.
* Tác giả.
* Banner.
* Nhân viên.
* Khách hàng.

## **1.4. Các yêu cầu của hệ thống**

### **1.4.1. Yêu cầu chức năng**

Hệ thống cần có các chức năng sau:

* Quản lý thông tin về: sản phẩm, khách hàng, nhân viên,… .
* Tạo đơn hàng online và thanh toán trực tuyến bằng VnPay và PayPal.
* Tìm kiếm sản phẩm qua nhiều cách thức như: qua từ khóa, qua thể loại, qua tác giả, qua thương hiệu.
* Đánh giá các sản phẩm của hệ thống.
* Thống kê doanh thu và lượng hàng bán ra hàng tháng.

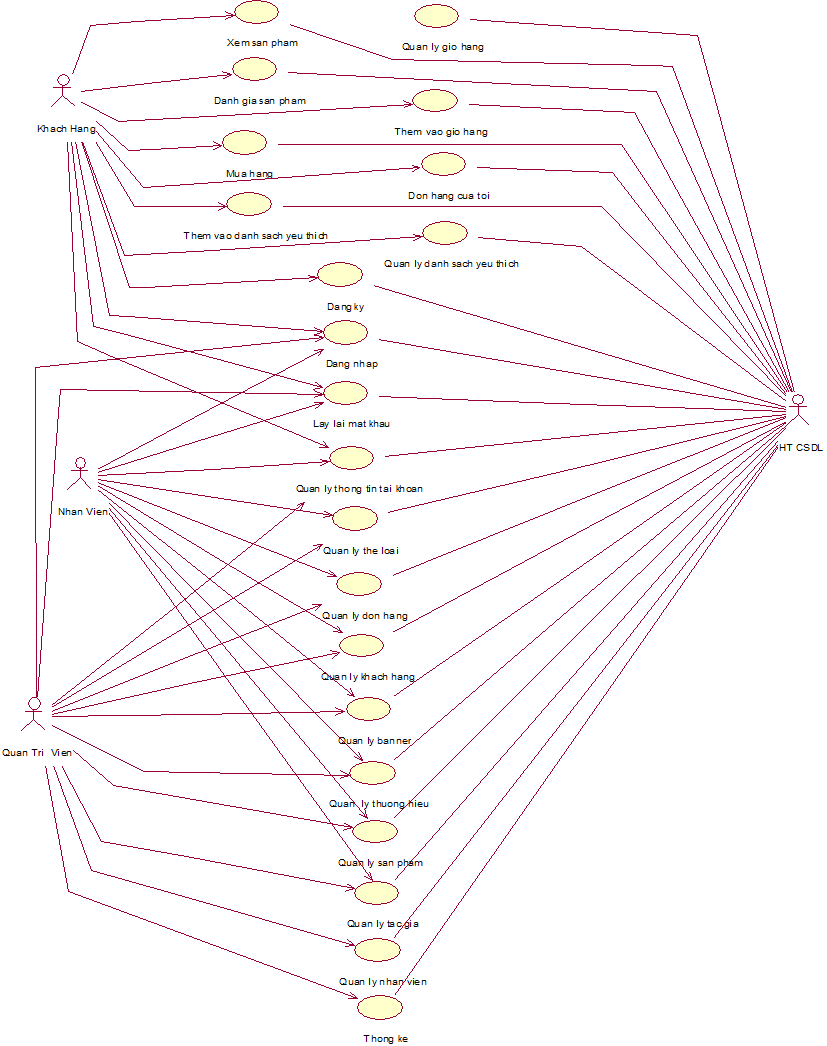
Ngoài ra hệ thống cần phải đáp ứng:

* Khả năng hoạt động ổn định.
* Hệ thống làm việc nhanh chóng và đảm bảo tin cậy.
* Giao diện dễ sử dụng, thân thiện với người dùng.
* Độ bảo mật hệ thống cao.

# **CHƯƠNG 2. PHÂN TÍCH CÁC YÊU CẦU CHỨC NĂNG CỦA HỆ THỐNG**

## **2.1. BIỂU ĐỒ USE CASE**

### **2.1.1. Các Use Case chính**



# Hình 2.1 : Biểu đồ Use Case

### **2.1.2.** **Quan hệ giữa các Use Case**

**Đăng nhập**

A diagram of a network

Description automatically generated

# Hình 2.2 : Biểu đồ Use Case Đăng nhập

**Xem sản phẩm**

A diagram of a network

Description automatically generated

# Hình 2.3 : Biểu đồ Use Case Xem sản phẩm

**Quản lý sản phẩm**

A diagram of a diagram

Description automatically generated

# Hình 2.4 : Biểu đồ Use Case Quản lý sản phẩm

Quản lý đơn hàng

A diagram of a diagram

Description automatically generated

# Hình 0.5. Biểu đồ use case Quản lý đơn hàng

Đơn hàng của tôi

A diagram of a diagram

Description automatically generated

# Hình 0.6. Biểu đồ use case Đơn hàng của tôi

Quản lý khách hàng

A diagram of a diagram

Description automatically generated

# Hình 0.7. Biểu đồ use case Quản lý khách hàng

Quản lý thông tin tài khoản

A diagram of a diagram

Description automatically generated

# Hình 0.8. Biểu đồ use case Quản lý thông tin tài khoản

Quản lý nhân viên

A diagram of a diagram

Description automatically generated

# Hình 0.9. Biểu đồ use case Quản lý nhân viên

Quản lý giỏ hàng

A diagram of a diagram

Description automatically generated

# Hình 0.10. Biểu đồ use case Quản lý giỏ hàng

Quản lý danh sách yêu thích

A diagram of a diagram

Description automatically generated

# Hình 0.11. Biểu đồ use case Quản lý danh sách yêu thích

**CHƯƠNG 3 : THIẾT KẾ CƠ SỞ DỮ LIỆU**

## **3.1. MÔ HÌNH HOÁ DỮ LIỆU**

*A diagram of a computer

Description automatically generated*

# Hình 3.1 : Cấu trúc database của dự án

**Authors**: Dùng để lưu trữ các tác giả trong hệ thống và gồm các thông tin AuthorId, AuthorCode, AuthorName, CodeNumber, AuthorSlug, Information, ImageName, UrlImage. Một tác giả có 0, 1 hoặc nhiều sản phẩm. Một sản phẩm có 0, 1 hoặc nhiều tác giả.

**AuthorProducts**: Dùng để lưu trữ chi tiết từng tác giả trong hệ thống và gồm các thông tin AuthorId, ProductId.

**Banners**: Dùng để lưu trữ banner quảng cáo cho các chiến dịch quảng cáo của hệ thống gồm có các thông tin như BannerId, Title, Content, ImageName, UrlImage.

**Brands**: Dùng để lưu trữ các thương hiệu sản phẩm trong hệ thống và gồm các thông tin BrandId, BrandCode, BrandName, BrandSlug, CodeNumber, Description, ImageName, UrlImage. Một thương hiệu gồm có 0, 1 hoặc nhiều sản phẩm, một sản phẩm thuộc 1 thương hiệu.

**Categories**: Dùng để lưu trữ các thể loại của hệ thống và gồm có CategoryId, CategoryName, CategoryCode, CategorySlug, CodeNumber, Description. Một thể loại thuộc 0, 1 hoặc nhiều sản phẩm, một sản phẩm có 1 hoặc nhiều thể loại.

**CategoryProducts**: Dùng để lưu trữ chi tiết từng thể loại trong hệ thống và gồm có các thông tin CategoryId, ProductId.

**Comments**: Dùng để lưu trữ các đánh giá của khách hàng về sản phẩm trong hệ thống và gồm có các thông tin CommentId, Vote, Message, DateCreated, CustomerId, ProductId. Một đánh giá thuộc 1 sản phẩm, một sản phẩm có 0, 1 hoặc nhiều đánh giá. Một đánh giá thuộc về một khách hàng, một khách hàng sẽ có 0, 1 hoặc nhiều đánh giá.

**Customers**: Dùng để lưu trữ các khách hàng của hệ thống và gồm có các thông tin CustomerId, FullName, DateOfBirth, Gender, CustomerCode, CodeNumber, Address. Một khách hàng có một tài khoản. Một tài khoản thuộc về 0 hoặc 1 khách hàng.

**Cart**: Dùng để lưu trữ các dòng trong giỏ hàng của khách hàng gồm có các thông tin Customers, ProductId, Quantity. Một khách hàng có 0, 1 hoặc nhiều dòng giỏ hàng, một dòng giỏ hàng thuộc về một khách hàng. Một sản phẩm có thuộc 0, 1 hoặc nhiều dòng giỏ hàng, một dòng giỏ hàng có một sản phẩm.

**Employees**: Dùng để lưu trữ các nhân viên của hệ thống gồm có các thông tin EmployeeId, EmployeeName, EmployeeCode, CodeNumber, Gender, Address, DateOfBirth. Một nhân viên có 1 tài khoản, một tài khoản thuộc về 0 hoặc một nhân viên.

**FavoriteProducts**: Dùng để lưu trữ từng dòng sản phẩm yêu thích của khách hàng và gồm có các thông tin CustomerId, ProductId. Một dòng sản phẩm yêu thích sẽ thuộc về 1 khách hàng, một khách hàng có 0, 1 hoặc nhiều dòng sản phẩm yêu thích. Một dòng sản phẩm yêu thích có 1 sản phẩm, một sản phẩm thuộc về 0, 1 hoặc nhiều dòng sản phẩm yêu thích.

**Images**: Dùng để lưu trữ các ảnh của sản phẩm và gồm các thông tin ImageId, ImageName, Url, ProductId. Một ảnh thuộc về một sản phẩm, một sản phẩm có 0, 1 hoặc nhiều ảnh.

**Orders**: Dùng để lưu trữ các đơn hàng của khách hàng trong hệ thống và gồm các thông tin OrderId, OrderCode, DateCreated, PhoneNumber, Address, Status, TransportFee, Note, DateDelivery, CodeNumber, FullName, CustomerId. Một đơn hàng thuộc về 1 khách hàng, một khách hàng có 0, 1 hoặc nhiều đơn hàng.

**OrderDetials**: Dùng để lưu trữ chi tiết đơn hàng của hệ thống và gồm các thông tin OrderId, ProductId, Price, Quantity. Một chi tiết đơn hàng thuộc về một đơn hàng, một đơn hàng có nhiều chi tiết đơn hàng. Một chi tiết đơn hàng có một sản phẩm, một sản phẩm thuộc 0, 1 hoặc nhiều chi tiết đơn hàng.

**Products**: Dùng để lưu trữ các sản phẩm trong hệ thống và gồm các thông tin ProductId, ProductCode, CodeNumber, ProductSlug, ProductName, Price, PercentDiscount, Description, Quantity, IsActive, BrandId.

**Users**: Dùng để lưu trữ các tài khoản trong hệ thống gồm các thông tin UserId, EmployeeId, CustomerId, UserName, NormalizedUserName, Email, NormalizedEmail, EmailConfirmed, SecurityStamp, ConcurrencyStamp, PhoneNumber, PhoneNumberConfirmed, TwoFactorEnabled, LockoutEnd, LockoutEnable, AccessFailedCount.

**Roles**: Dùng để lưu trữ quyền trong hệ thống và gồm các thông tin RoleId, Name, NormalizedName, ConccurrencyStamp. Một quyền thuộc về 0, 1 hoặc nhiều tài khoản, một tài khoản có 0, 1 hoặc nhiều quyền.

**UserRoles**: Dùng để lưu trữ chi tiết các quyền và gồm các thông tin RoleId, UserId.

**UseClaims**: Dùng để lưu trữ thuộc tính người dùng gồm các thông tin Id, ClaimType, ClaimValue, UseId.

**RoleClaims**: Dùng để lưu trữ thuộc tính của quyền và gồm các thông tin Id, RoleId, ClaimType, ClaimValue.

**UserLogins**: Dùng để lưu trữ thông tin về người dùng đăng nhập và gồm các thông tin LoginProvider, ProviderKey, ProviderDisplayName, UserName.

**UserTokens**: Dùng để lưu trữ token của người dùng và gồm các thông tin LoginProvider, UserId, Name, Value.

**Provinces**: Dùng để lưu trữ dữ liệu các tỉnh thành Việt Nam và gồm các thông tin: CodeProvince, Name, NameEn, FullName, FullNameEn, CodeName.

**Districts**: Dùng để lưu trữ dữ liệu các quận huyện Việt Nam và gồm các thông tin: CodeDistrict, Name, NameEn, FullName, FullNameEn, CodeName, CodeProvince.

**Wards**: Dùng để lưu trữ dữ liệu các phường xã Việt Nam và gồm các thông tin: CodeWard, Name, NameEn, FullName, FullNameEn, CodeName, CodeDistrict.

## **3.2. THIẾT KẾ CÁC BẢNG**

Bảng Authors

A screenshot of a computer

Description automatically generated

# Hình 3.2. Chi tiết bảng Authors của dự án

Bảng AuthorProducts

A screenshot of a computer

Description automatically generated

# Hình 3.3. Chi tiết bảng AuthorProducts của dự án

Bảng Banners

A screenshot of a computer

Description automatically generated

# Hình 3.4. Chi tiết bảng Banners của dự án

Bảng Brands

A screenshot of a computer

Description automatically generated

# Hình 3.5. Chi tiết bảng Brands của dự án.

Bảng Categories

A screenshot of a computer

Description automatically generated

# Hình 3-6. Chi tiết bảng Categories của dự án.

Bảng CategoryProducts

A screenshot of a computer

Description automatically generated

# Hình 3.7. Chi tiết bảng CategoryProducts của dự án

Bảng Comments

A screenshot of a computer

Description automatically generated

# Hình 3.8. Chi tiết bảng Comments của dự án.

Bảng Customers

A screenshot of a computer

Description automatically generated

# Hình 3.9. Chi tiết bảng Customers của dự án.

Bảng Employees

A screenshot of a computer

Description automatically generated

# Hình 3.10. Chi tiết bảng Employees của dự án.

Bảng FavouriteProducts

A screenshot of a computer

Description automatically generated

# Hình 3.11. Chi tiết bảng FavouriteProducts của dự án.

Bảng Images

A screenshot of a computer

Description automatically generated

# Hình 3.12. Chi tiết bảng Images của dự án

Bảng Orders

A screenshot of a computer

Description automatically generated

# Hình 3.13. Chi tiết bảng Orders của dự án

Bảng OrderDetails

A screenshot of a computer

Description automatically generated

# Hình 3.14. Chi tiết bảng OrderDetails của dự án.

Bảng Products

A screenshot of a computer

Description automatically generated

# Hình 3.15. Chi tiết bảng Products của dự án.

Bảng Users

A screenshot of a computer

Description automatically generated

# Hình 3.16. Chi tiết bảng Users của dự án.

Bảng UserClaims

A screenshot of a computer

Description automatically generated

# Hình 3.17. Chi tiết bảng UserClaims của dự án.

Bảng Roles

A screenshot of a computer

Description automatically generated

# Hình 3.18. Chi tiết bảng Roles của dự án.

Bảng RoleClaims

A screenshot of a computer

Description automatically generated

# Hình 3.19. Chi tiết bảng RoleClaims của dự án

Bảng UserRoles

A screenshot of a computer

Description automatically generated

# Hình 3-20. Chi tiết bảng UserRoles của dự án

Bảng UserLogins

A screenshot of a computer

Description automatically generated

# Hình 3.21. Chi tiết bảng UserLogins của dự án

Bảng UserTokens

A screenshot of a computer

Description automatically generated

# Hình 3.22. Chi tiết bảng UserTokens của dự án.

Bảng Provinces

A screenshot of a computer

Description automatically generated

# Hình 3.23. Chi tiết bảng Provinces của dự án

Bảng Districts

A screenshot of a computer

Description automatically generated

# Hình 3.24. Chi tiết bảng Districts của dự án.

Bảng Wards

A screenshot of a computer

Description automatically generated

# Hình 3.25. Chi tiết bảng Wards của dự án

## **3.3. QUAN HỆ CÁC BẢNG**

A computer screen shot of a computer

Description automatically generated

# Hình 3.26 : Quan hệ của các bảng trong database

# **CHƯƠNG 4 : THIẾT KẾ GIAO DIỆN VÀ CÀI ĐẶT**

## Kiến trúc củ hành (Onion Architecture) – Craftsmanship**4.1. MÔ HÌNH KIẾN TRÚC DỰ ÁN**

# Hình 4.1 : Mô hình kiến trúc Onion

### **4.1.1. Giới thiệu kiến trúc Onion**

Kiến trúc Onion (Onion Architecture) là một kiểu kiến trúc phần mềm được phát triển bởi Jeffrey Palermo nhằm giải quyết các vấn đề về độ phức tạp và sự phụ thuộc trong các ứng dụng lớn. Mục tiêu của kiến trúc này là tạo ra một hệ thống có khả năng bảo trì và mở rộng dễ dàng, đồng thời giảm thiểu sự phụ thuộc giữa các thành phần.

### **4.1.2. Các lớp trong kiến trúc Onion**

Kiến trúc Onion sử dụng khái niệm lớp, nhưng nó khác với kiến trúc N-layer và kiến trúc 3-Tier.

Domain Layer: Lớp này nằm ở trung tâm của kiến trúc nơi chúng ta có các thực thể (entity) của ứng dụng gồm các lớp application model của ứng dụng hoặc các lớp database model phụ thuộc vào cách tiếp cận về code trong quá trình phát triển ứng dụng. Trường hợp sử dụng ASP.NET, các thực thể này được sử dụng để tạo các bảng trong cơ sở dữ liệu (database model).

Repository Layer: Lớp repository hoạt động như một lớp trung gian giữa lớp service và các đối tượng mô hình, các lớp model và database context sẽ được thực hiện trong lớp này. Các lập trình viên sẽ thêm các interface bao gồm các thao tác truy cập dữ liệu cho các thao tác đọc và ghi với cơ sở dữ liệu.

Service Layer: Lớp này được sử dụng để giao tiếp với lớp presentation và repository. Lớp service chứa tất cả logic nghiệp vụ của thực thể (entity). Trong lớp này, các service interface được giữ tách biệt với việc triển khai chúng để các lớp ít phụ thuộc vào nhau

Presentation Layer: Tương tự như lớp Presentation trong kiến trúc 3 tầng. Nhiệm vụ chính của lớp này là hiển thị dữ liệu với người dùng. Trong trường hợp lớp Presentation API dữ liệu đối tượng từ cơ sở dữ liệu được truyền tải thông qua HTTP request dưới dạng json.

### **4.1.3. Ưu nhược điểm kiến trúc Onion**

Ưu điểm:

* Kiến trúc onion cung cấp cho chúng ta khả năng bảo trì code tốt hơn vì code phụ thuộc vào các lớp
* Nó cung cấp khả năng kiểm thử tốt cho các unit tests, chúng ta có thể viết các test cases riêng biệt theo lớp mà không ảnh hưởng đến mô-đun khác trong ứng dụng.
* Với kiến trúc onion, các lớp trong ứng dụng sẽ không phụ thuộc lẫn nhau vì các lớp giao tiếp với nhau một cách trừu tượng (thông qua interface)
* Các domain entity là cốt lõi và trung tâm của kiến trúc và có quyền truy cập vào cơ sở dữ liệu và Lớp UI.
* Lớp bên trong không bao giờ phụ thuộc vào lớp bên ngoài (chiều phụ thuộc từ bên ngoài vào trong)

Nhược điểm

* Việc triển khai kiến trúc này đòi hỏi đầu tư thời gian và công sức ban đầu lớn để thiết kế và xây dựng các tầng một cách đúng đắn, dẫn đến chi phí phát triển ban đầu cao hơn.
* Đối với các dự án nhỏ, lợi ích của kiến trúc Onion có thể không đáng kể so với độ phức tạp và chi phí phát sinh, làm cho nó trở thành một lựa chọn không phù hợp.

## **4.2. GIỚI THIỆU NGÔN NGỮ SỬ DỤNG**

### A blue and white logo Description automatically generated**4.2.1. Asp.Net Core**

# Hình 4.2 : Giới thiệu Asp.Net Core

ASP.NET Core là một framework phát triển ứng dụng web mã nguồn mở, hiệu suất cao, do Microsoft và cộng đồng phát triển. Nó được thiết kế để xây dựng các ứng dụng web hiện đại, kết nối IoT và các dịch vụ đám mây. Đây là phiên bản cải tiến và đa nền tảng của ASP.NET, giúp các nhà phát triển xây dựng ứng dụng trên Windows, macOS và Linux. Sau đây là một số đặc điểm của framework này:

* Đa nền tảng: Chạy trên Windows, macOS, và Linux.
* Hiệu suất cao: Thường đứng đầu trong các bảng xếp hạng hiệu suất.
* Mã nguồn mở: Phát triển trên GitHub với sự đóng góp của cộng đồng.
* Modular và Lightweight: Sử dụng kiến trúc modular, giảm thiểu footprint.
* Middleware: Cấu hình pipeline xử lý yêu cầu HTTP linh hoạt.
* Dependency Injection (DI): Tích hợp sẵn, quản lý phụ thuộc dễ dàng.
* Razor Pages: Mô hình trang đơn giản hóa giao diện người dùng.
* Blazor: Xây dựng ứng dụng web tương tác bằng C#.
* Web API: Tạo dịch vụ RESTful APIs dễ dàng.
* Công cụ phát triển: Hỗ trợ mạnh mẽ qua Visual Studio, Visual Studio Code, và CLI.
* Triển khai linh hoạt: Self-contained hoặc framework-dependent, hỗ trợ Docker.
* Tính năng nổi bật: Razor View Engine, Tag Helpers, SignalR, Identity.
* Cộng đồng và Tài liệu: Cộng đồng mạnh mẽ và tài liệu phong phú từ Microsoft.

**Ưu điểm**

* Đa nền tảng (Cross-platform): ASP.NET Core cho phép phát triển và triển khai ứng dụng trên nhiều hệ điều hành như Windows, macOS và Linux, tăng tính linh hoạt và khả năng tiếp cận đối với người dùng.
* Hiệu suất cao (High Performance): ASP.NET Core được tối ưu hóa để cung cấp hiệu suất cao hơn so với các phiên bản trước đó của ASP.NET, nhờ sử dụng Kestrel và xử lý non-blocking.
* Modular và linh hoạt (Modular and Flexible): ASP.NET Core được thiết kế theo mô hình modular, cho phép bạn chỉ cài đặt các thành phần cần thiết, giảm kích thước ứng dụng và tăng tốc độ khởi động.
* Hỗ trợ Docker (Docker Support): ASP.NET Core tích hợp tốt với Docker, giúp dễ dàng triển khai và vận hành ứng dụng trên các môi trường khác nhau một cách đồng nhất.
* Dependency Injection (DI): ASP.NET Core tích hợp sẵn Dependency Injection, giúp quản lý các phụ thuộc và cung cấp các dịch vụ một cách linh hoạt và dễ dàng kiểm soát.
* Configuration cấu hình linh hoạt: ASP.NET Core cung cấp cơ chế cấu hình mạnh mẽ, cho phép bạn cấu hình ứng dụng từ nhiều nguồn khác nhau như file JSON, XML, hoặc các biến môi trường.
* Hỗ trợ Web API mạnh mẽ: ASP.NET Core cung cấp một bộ công cụ mạnh mẽ cho việc phát triển các dịch vụ API Web, bao gồm hỗ trợ cho RESTful API và giao thức HTTP.

**Nhược điểm:**

* Thư viện và công cụ hạn chế: Mặc dù hệ sinh thái của ASP.NET Core ngày càng phát triển, nhưng vẫn có một số thư viện và công cụ hạn chế so với các nền tảng phát triển ứng dụng web khác.
* Khả năng tương thích: Một số tính năng của ASP.NET Framework không được hỗ trợ hoặc hoạt động khác biệt trên ASP.NET Core, điều này có thể gây khó khăn khi di chuyển từ phiên bản cũ sang phiên bản mới.

### **4.2.2. Entity Framework**

Entity Framework (EF) là một ORM (Object-Relational Mapping) framework được phát triển bởi Microsoft. Nó là một thành phần của công nghệ .NET và cung cấp một cách tiếp cận dễ dàng và linh hoạt để làm việc với cơ sở dữ liệu trong ứng dụng. Sau đây là một số tổng quan về Entity Framework:

1. Đối tượng - Quan hệ (Object-Relational): Entity Framework cung cấp một cách tiếp cận đối tượng - quan hệ, cho phép bạn làm việc với cơ sở dữ liệu bằng cách sử dụng các đối tượng và quan hệ giống như trong mã lập trình. Nó ánh xạ các đối tượng trong ứng dụng của bạn vào cơ sở dữ liệu tương ứng, và tự động thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên cơ sở dữ liệu.
2. Mô hình dữ liệu: Entity Framework cho phép bạn xác định mô hình dữ liệu của ứng dụng bằng cách sử dụng các lớp (classes) và thuộc tính (properties). Bằng cách xác định mối quan hệ giữa các lớp, bạn có thể tạo ra một mô hình dữ liệu logic và dễ dàng thao tác với các thực thể (entities) trong cơ sở dữ liệu.
3. Linq to Entities: Entity Framework hỗ trợ Linq (Language-Integrated Query) để truy vấn dữ liệu từ cơ sở dữ liệu. Linq to Entities cho phép bạn sử dụng các truy vấn LINQ mạnh mẽ để lấy dữ liệu từ cơ sở dữ liệu một cách linh hoạt
4. Tự động tạo câu lệnh SQL: Entity Framework tự động tạo câu lệnh SQL cần thiết để thao tác với cơ sở dữ liệu dựa trên các thao tác trên đối tượng. Bạn không cần viết câu lệnh SQL thủ công, EF sẽ tự động xử lý việc chuyển đổi giữa đối tượng và cơ sở dữ liệu.
5. Migration: Entity Framework cung cấp công cụ Migration để quản lý việc thay đổi cấu trúc cơ sở dữ liệu theo thời gian. Bạn có thể thay đổi mô hình dữ liệu và EF sẽ tự động tạo và áp dụng các tác động lên cơ sở dữ liệu mà không làm mất dữ liệu hiện có.
6. Hỗ trợ đa nền tảng: Entity Framework không chỉ hỗ trợ SQL Server mà còn hỗ trợ nhiều hệ quản trị cơ sở dữ liệu khác như MySQL, PostgreSQL, SQLite và Oracle. Điều này cho phép bạn dễ dàng chuyển đổi giữa các hệ quản trị cơ sở dữ liệu khác nhau mà không cần thay đổi mã lập trình chính.

### **4.2.3. Boostrap**

Bootstrap là một framework CSS phổ biến được sử dụng để phát triển các giao diện web thân thiện với thiết bị di động và sau đây là một số tổng quan về Bootstrap.

1. Responsive Design (Thiết kế đáp ứng): Bootstrap giúp bạn xây dựng các trang web đáp ứng tự động, có nghĩa là giao diện sẽ tự thích ứng và hiển thị phù hợp trên các thiết bị khác nhau như máy tính bàn, máy tính xách tay, điện thoại di động và máy tính bảng.
2. Grid System (Hệ thống lưới): Bootstrap cung cấp một hệ thống lưới linh hoạt và dễ sử dụng, giúp bạn tạo ra bố cục trang web có cấu trúc gọn gàng và chia cột một cách dễ dàng. Hệ thống lưới của Bootstrap cho phép bạn tạo ra các khu vực và cột có độ rộng linh hoạt, tạo điểm neo cho các phần tử trên trang.
3. CSS Components (Các thành phần CSS): Bootstrap cung cấp một loạt các thành phần CSS tiện ích như nút, bảng, biểu đồ, thanh điều hướng, hộp thoại, biểu mẫu và nhiều hơn nữa. Các thành phần này đã được thiết kế sẵn và có thể được sử dụng trực tiếp hoặc tùy chỉnh để xây dựng giao diện web.
4. Typography (Chữ viết): Bootstrap cung cấp một bộ CSS giúp điều chỉnh kiểu chữ, định dạng văn bản và các thành phần liên quan đến typography. Bạn có thể dễ dàng tùy chỉnh các thuộc tính như font-family, font-size, font-weight và màu sắc chữ viết.
5. JavaScript Plugins (Các plugin JavaScript): Bootstrap đi kèm với một số plugin JavaScript được tích hợp sẵn như carousel (trình chạy hình ảnh), modal (cửa sổ popup), dropdown (menu thả xuống), và nhiều plugin khác. Các plugin này giúp tạo ra các hiệu ứng và chức năng tương tác trên trang web của bạn một cách dễ dàng.
6. Customization (Tùy chỉnh): Bootstrap cho phép người dùng tùy chỉnh giao diện của mình bằng cách chọn và tùy chỉnh các thành phần, biểu đồ và màu sắc theo ý muốn. Người dùng có thể sử dụng Sass (một ngôn ngữ CSS mở rộng) và các công cụ biên dịch để thay đổi và tạo ra phiên bản tùy chỉnh của Bootstrap.

### **4.2.4. Jquery**

JQuery là một thư viện JavaScript phổ biến, được thiết kế để đơn giản hóa việc lập trình JavaScript trên các trang web. Được tạo ra bởi John Resig và ra mắt lần đầu vào năm 2006, jQuery đã trở thành một trong những thư viện JavaScript được sử dụng rộng rãi nhất. Dưới đây là một tổng quan về jQuery

1. JQuery cung cấp một cú pháp đơn giản và ngắn gọn để thực hiện các tác vụ JavaScript phổ biến như thao tác DOM (Document Object Model), xử lý sự kiện, và thực hiện các hiệu ứng động.
2. JQuery giúp giải quyết vấn đề tương thích giữa các trình duyệt, đảm bảo mã JavaScript hoạt động nhất quán trên các trình duyệt khác nhau như Chrome, Firefox, Safari, và Internet Explorer.
3. JQuery cung cấp các hàm tiện ích để thực hiện các yêu cầu AJAX dễ dàng, giúp tải dữ liệu không đồng bộ mà không cần làm mới trang.
4. JQuery có các hàm tích hợp để tạo hiệu ứng và hoạt hình đơn giản mà không cần phải viết nhiều mã phức tạp.
5. JQuery có một hệ sinh thái plugin phong phú, cho phép mở rộng chức năng của nó mà không cần phải viết lại từ đầu. Có hàng ngàn plugin có sẵn để giải quyết các nhu cầu khác nhau của người dùng.
6. jQuery cho phép thay đổi các thuộc tính CSS của các phần tử HTML một cách dễ dàng.

### **4.2.5. Hệ quản trị cơ sở dữ liệu SQL Server**

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft. Nó là một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất trên thế giới và được sử dụng rộng rãi trong các ứng dụng doanh nghiệp, web và di động.Dưới đây là một tổng quan về SQL Server:

1. Tính năng và khả năng: SQL Server cung cấp một loạt các tính năng và khả năng mạnh mẽ để quản lý, lưu trữ và truy vấn dữ liệu. Nó hỗ trợ ngôn ngữ truy vấn SQL, có khả năng xử lý các tác vụ phức tạp như truy vấn dữ liệu, tạo, sửa đổi và xóa cơ sở dữ liệu, và quản lý quyền truy cập. SQL Server cũng cung cấp các tính năng mở rộng như dịch vụ phân tích, khai thác dữ liệu và xử lý dữ liệu trực quan.
2. Bảo mật: SQL Server cung cấp các cơ chế bảo mật mạnh mẽ để bảo vệ dữ liệu. Nó hỗ trợ xác thực và phân quyền, cho phép quản trị viên quản lý quyền truy cập vào cơ sở dữ liệu và đảm bảo tính bảo mật của dữ liệu.
3. Quản lý hiệu suất: SQL Server cung cấp các công cụ và tính năng để quản lý hiệu suất của cơ sở dữ liệu. Nó bao gồm trình tối ưu hóa truy vấn, các công cụ giám sát và xử lý lỗi, cơ chế lập lịch và sao lưu dữ liệu.
4. Khả năng mở rộng: SQL Server cho phép mở rộng hệ thống cơ sở dữ liệu để đáp ứng nhu cầu tăng trưởng của ứng dụng. Nó hỗ trợ cụm cơ sở dữ liệu (database clustering), phân tán dữ liệu và khả năng chia sẻ tài nguyên.
5. Tích hợp: SQL Server tích hợp tốt với các công nghệ và dịch vụ khác của Microsoft như .NET Framework, Visual Studio và Azure. Điều này giúp việc phát triển ứng dụng và triển khai cơ sở dữ liệu trở nên thuận tiện và liên kết chặt chẽ với các công nghệ Microsoft khác.
6. Hỗ trợ và cộng đồng: SQL Server có một cộng đồng lớn và mạnh mẽ, với tài liệu phong phú, các diễn đàn thảo luận và các nguồn tài nguyên hữu ích khác. Microsoft cũng cung cấp hỗ trợ chính thức và cập nhật thường xuyên để đảm bảo tính ổn định và an toàn của SQL Server.

## **4.3. THIẾT KẾ CHI TIẾT**

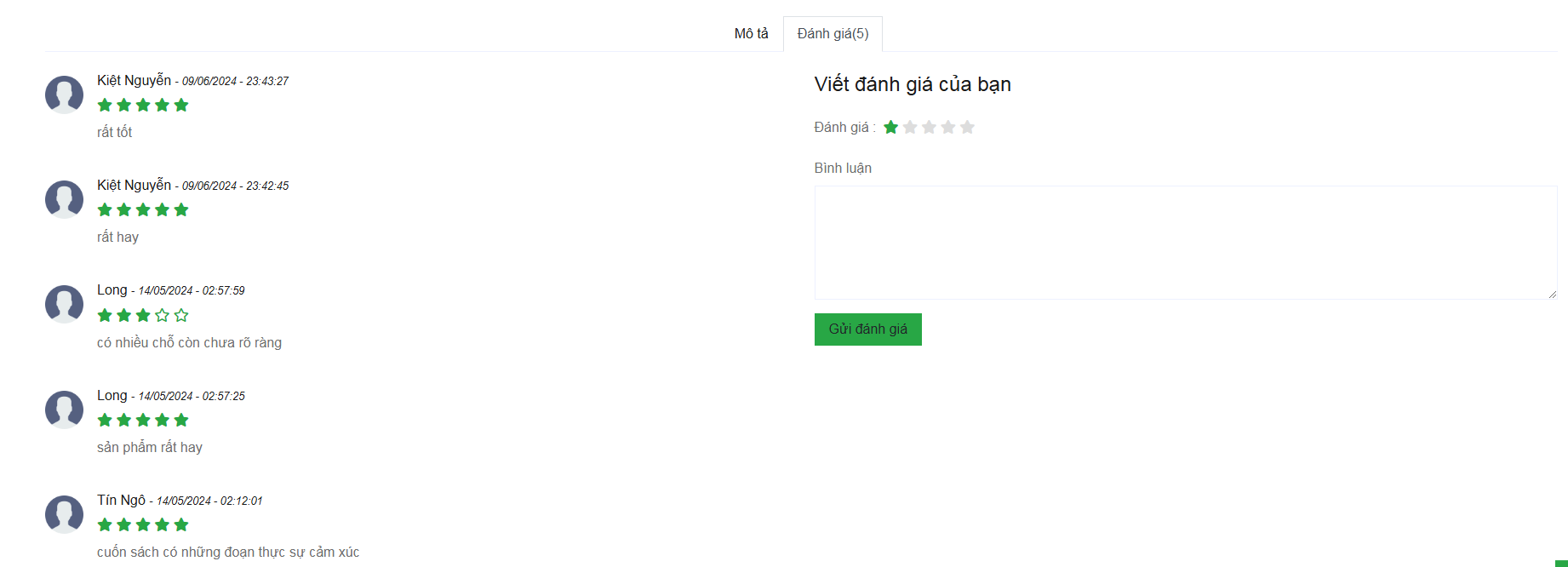
### **4.3.1. Xem sản phẩm**

A group of books on a screen

Description automatically generated

# Hình 4.3 : Giao diện “Xem sản phẩm”

### **4.3.2. Đánh giá sản phẩm**



# Hình 4.4 : Giao diện “Đánh giá sản phẩm”

### **4.3.3. Giỏ hàng**

A screenshot of a computer

Description automatically generated

# Hình 4.5 : Giao diện “Giỏ hàng”

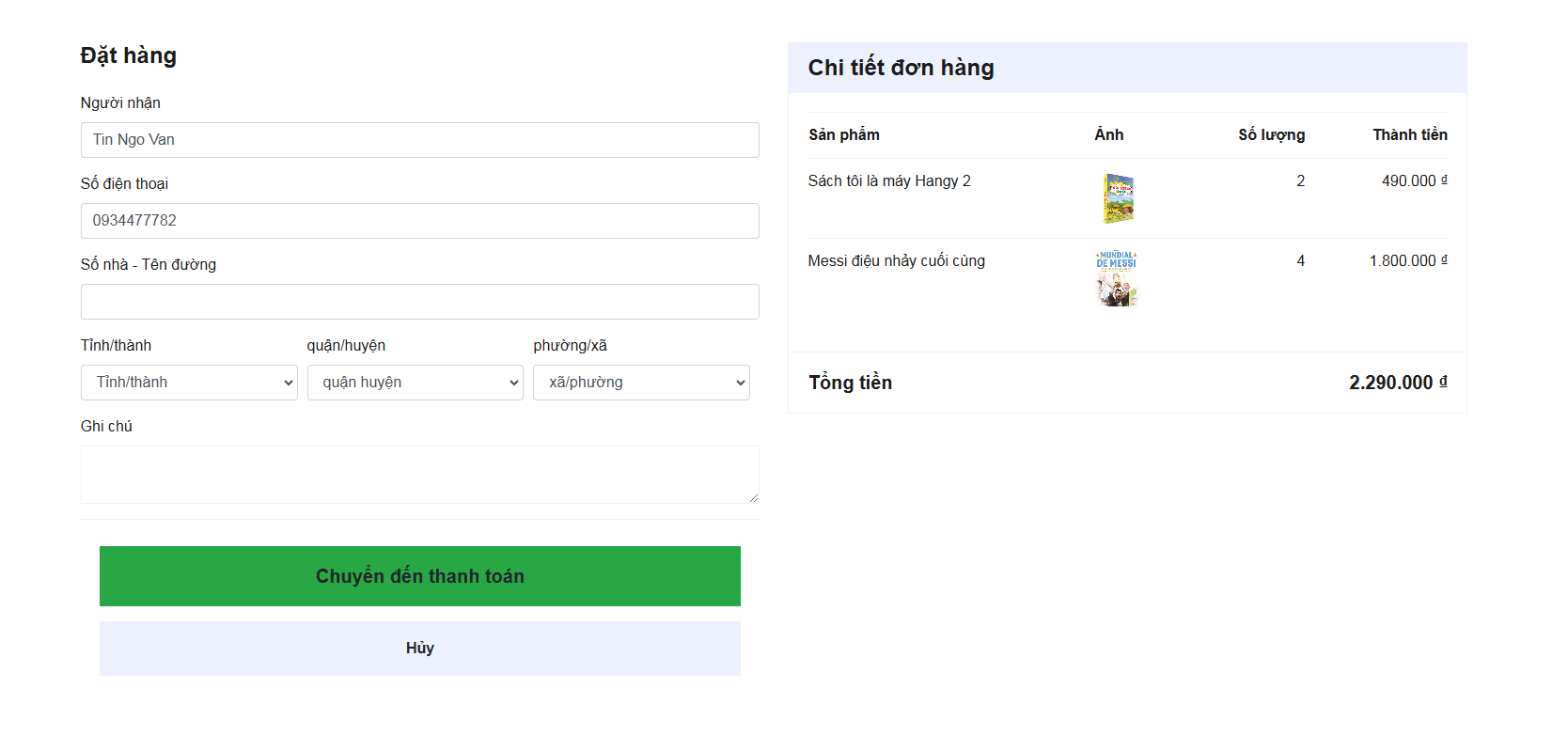
### **4.3.4. Danh sách yêu thích**

A book with a picture on it

Description automatically generated

# Hình 4.6 : Giao diện “Danh sách yêu thích”

### **4.3.5. Đặt hàng**

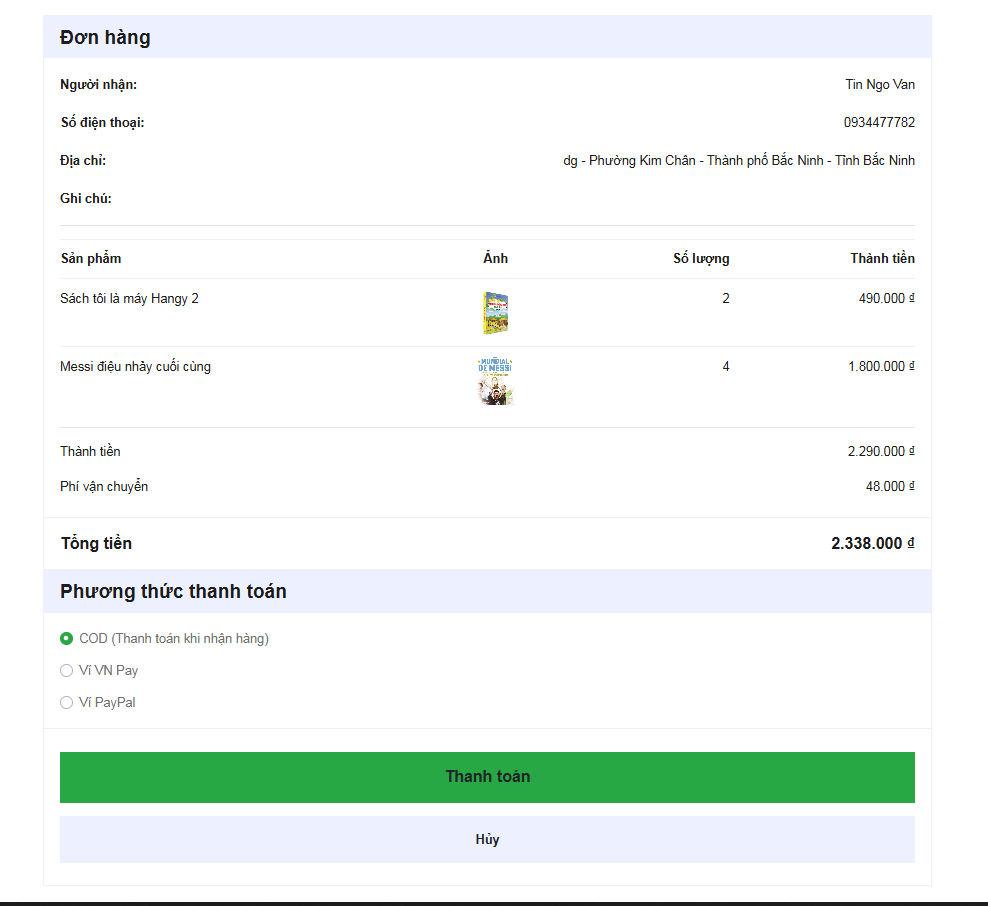


# Hình 4.7 : Giao diện “Đặt hàng”

### **4.3.6. Quản lý thông tin tài khoản**

# Hình 4.8 : Giao diện “Quản lý thông tin tài khoản”

### **4.3.7. Đơn hàng của tôi**



# Hình 4.9 : Giao diện “Đơn hàng”

### **4.3.8. Đăng ký**

A screenshot of a computer

Description automatically generated

# Hình 4.10 : Giao diện “Đăng ký”

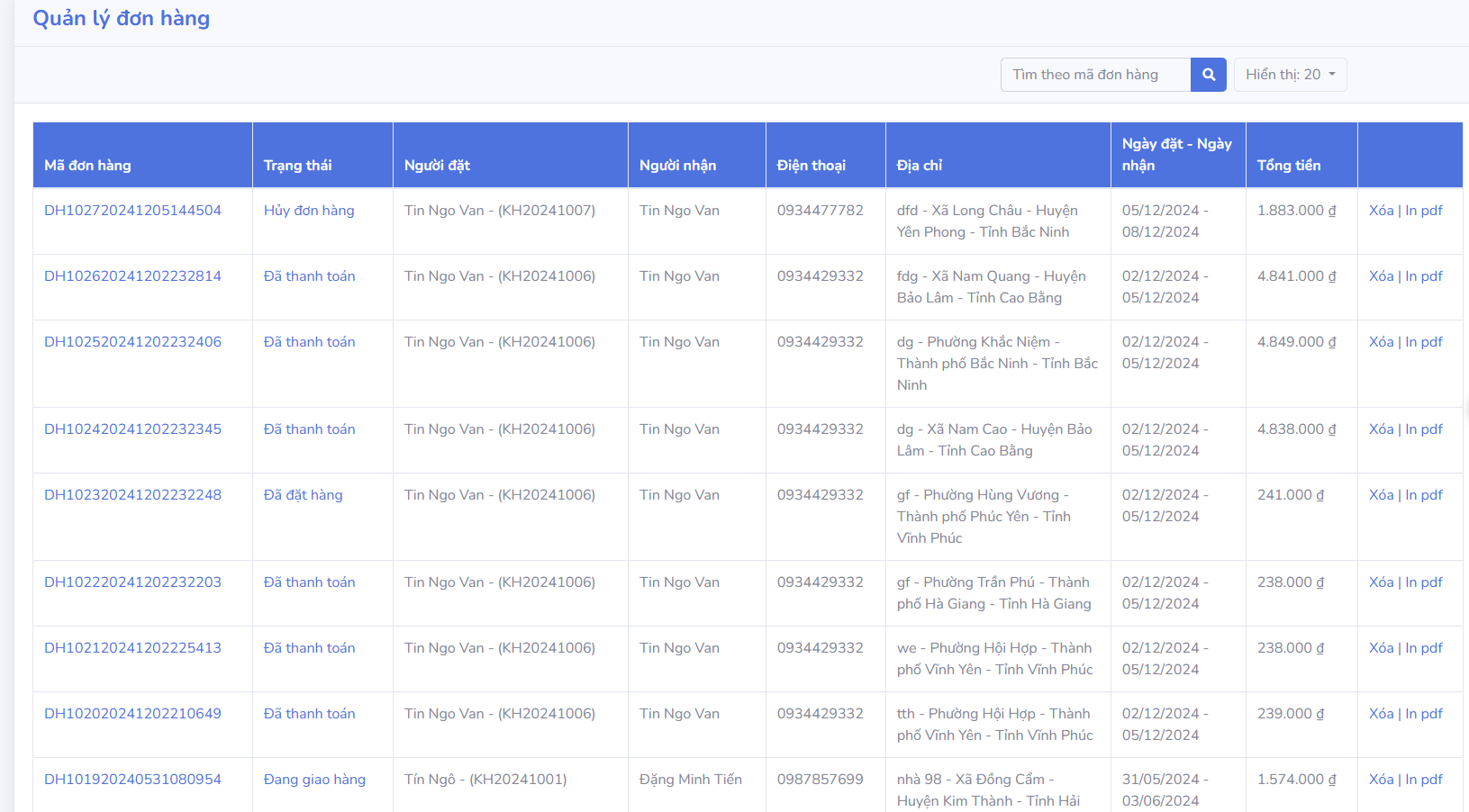
### **4.3.9. Đăng nhập**

A screenshot of a login form

Description automatically generated

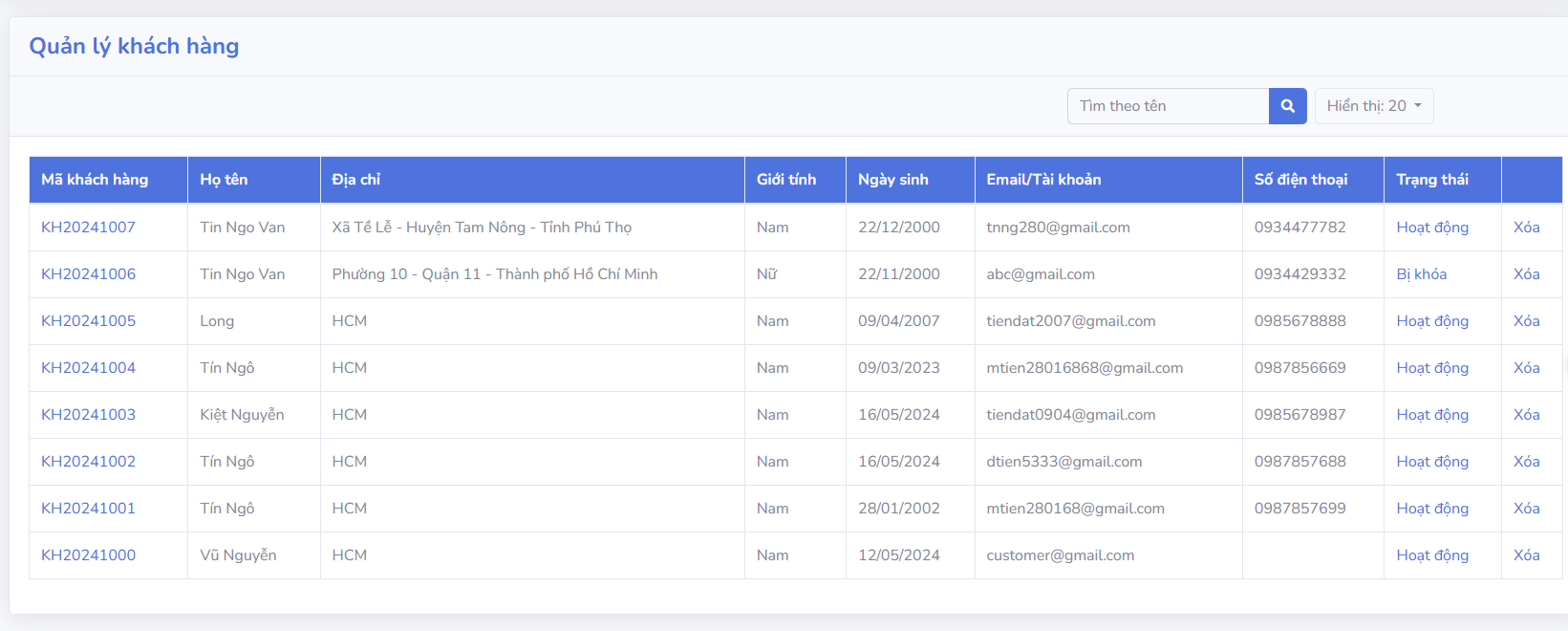
# Hình 4.11 : Giao diện trang “Đăng nhập”

### **4.3.10. Quản lý đơn hàng**



# Hình 4.12 : Giao diện “Quản lý đơn hàng”

### **4.3.11. Quản lý khách hàng**



# Hình 4.13 : Giao diện “Quản lý khách hàng”

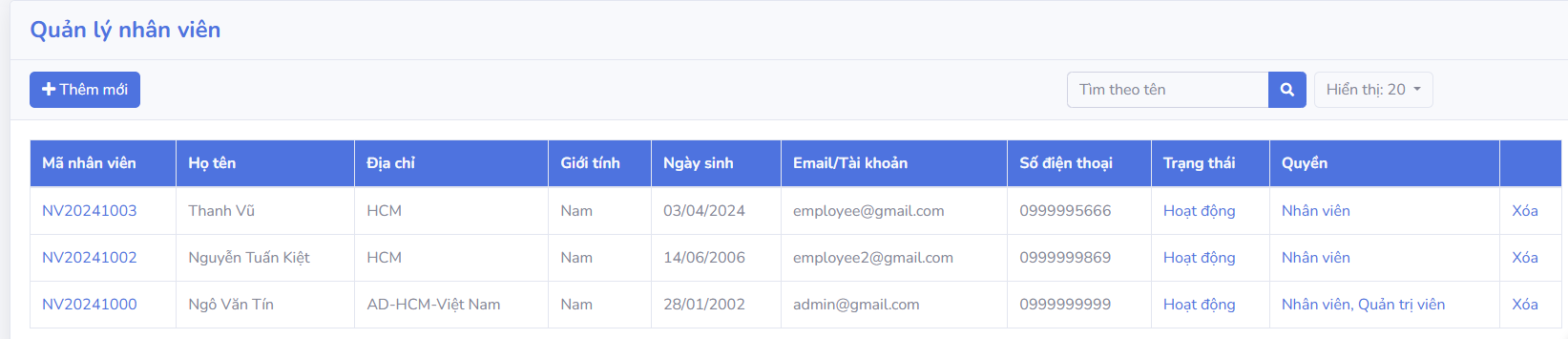
### **4.3.12. Quản lý sản phẩm**

A screenshot of a computer

Description automatically generated

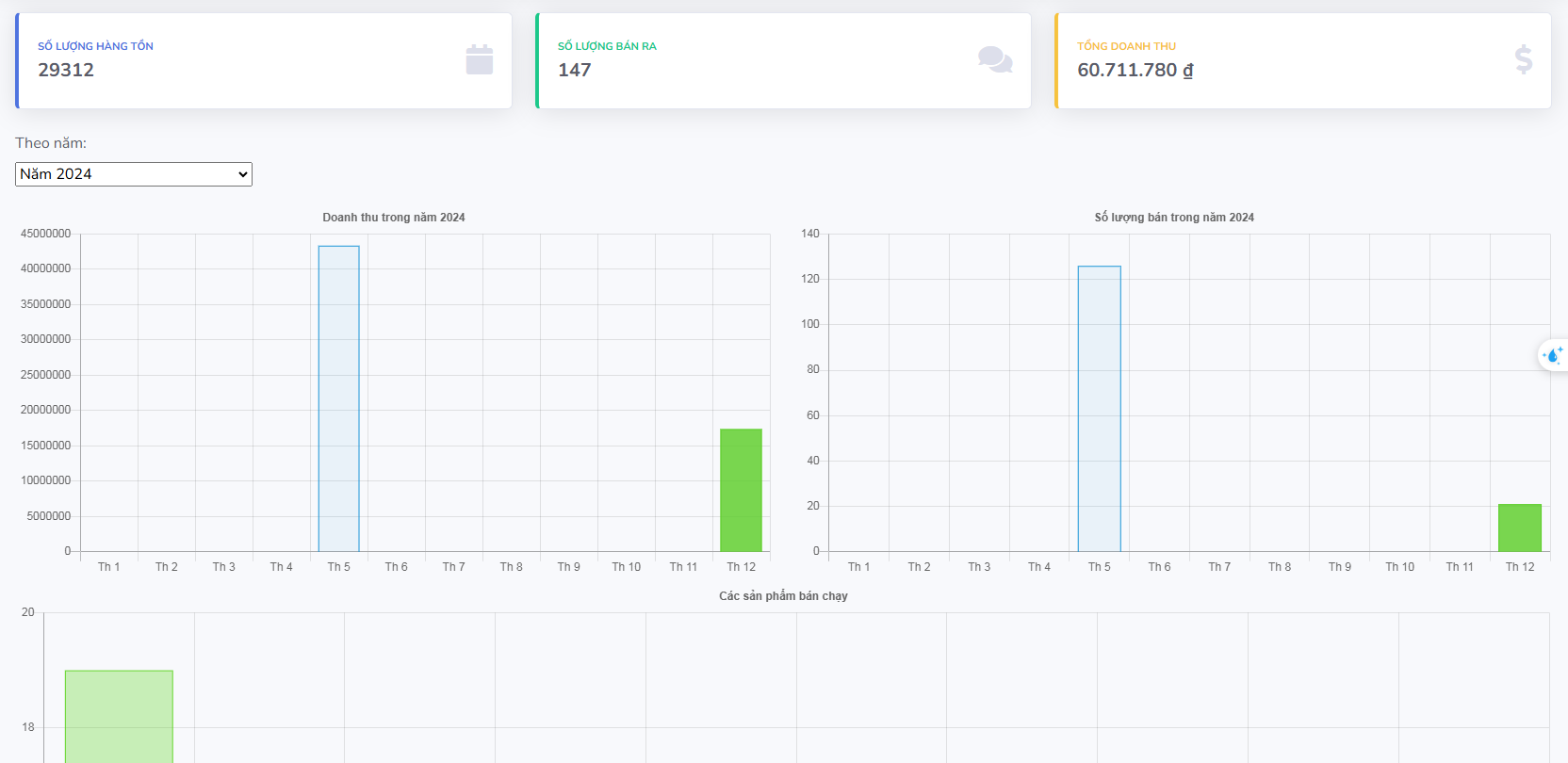
# Hình 4.14 : Giao diện “Quản lý sản phẩm”

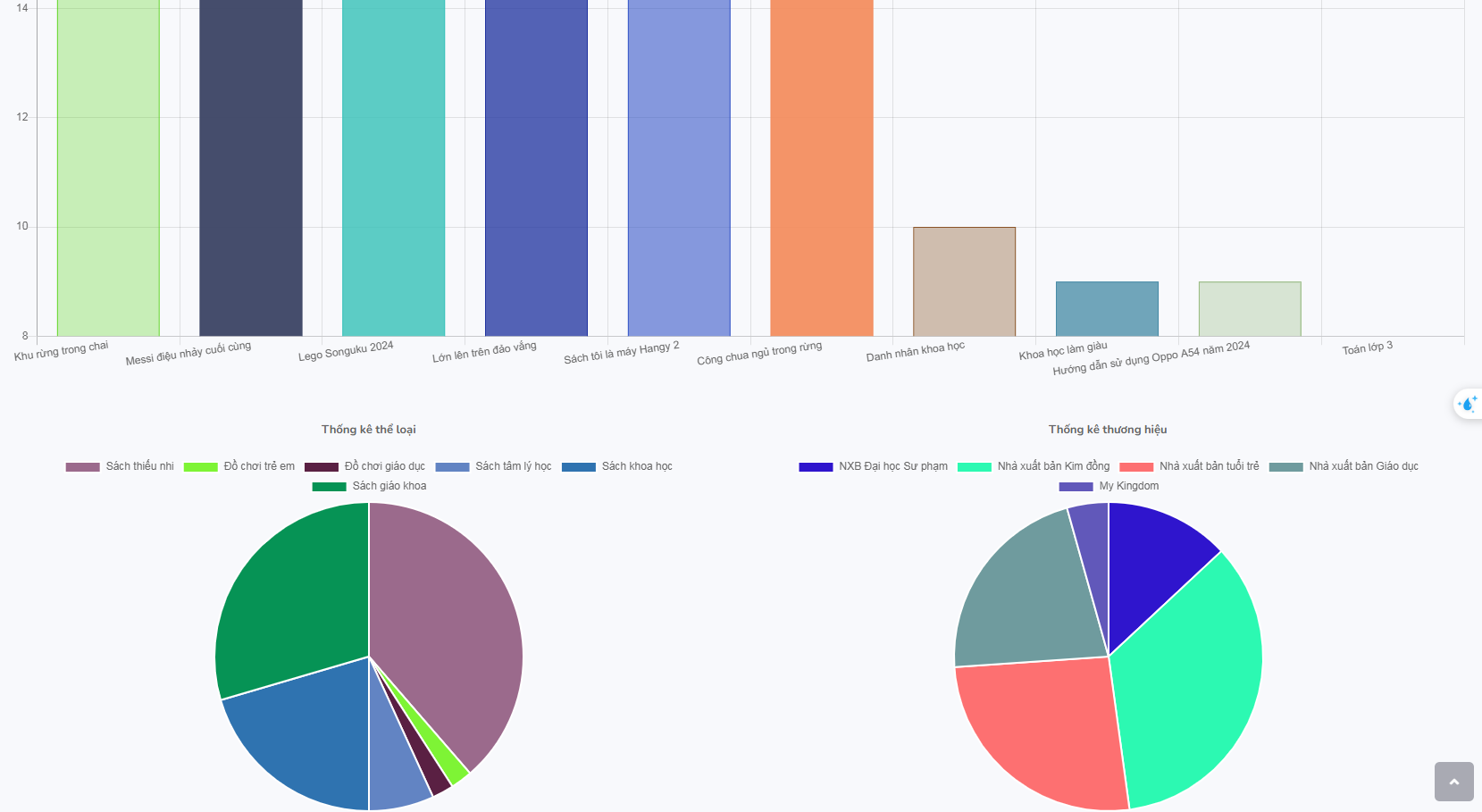
### **4.3.13. Quản lý nhân viên**



# Hình 4.15 : Giao diện “Quản lý nhân viên”

### **4.3.14. Thống kê**





# Hình 4.16 : Giao diện “Thống kê”

# **CHƯƠNG 5 : TỔNG KẾT VÀ ĐÁNH GIÁ**

## **5.1. KẾT QUẢ ĐẠT ĐƯỢC**

Qua quá trình thực hiện đề tài “Xây dựng website bán sách bằng ASP.NET Core” chúng em đã đạt được những kết quả sau:

* Hoàn thiện được các chức năng: Trong quá trình thực hiện đồ án em đã xây dựng được các chức năng như xem hàng, xem hàng qua tác giả, thương hiệu, thể loại, xem chi tiết sản phẩm, tích hợp được thanh toán online bằng vnpay, paypal, giúp khách hàng quản lý được đơn hàng, giúp nhân viên và quản trị viên quản lý được sản phẩm, thể loại, khách hàng và xem được báo cáo thống kê.
* Giỏ hàng và thanh toán: Hệ thống giỏ hàng hoạt động hiệu quả, cho phép người dùng thêm, xóa, và cập nhật số lượng sách trong giỏ hàng. Chức năng thanh toán được tích hợp với hai phương thức thanh toán trực tuyến phổ biến là vnpay và paypal, đảm bảo giao dịch an toàn và thuận tiện.
* Báo cáo và thống kê: Hệ thống báo cáo và thống kê cung cấp cho quản trị viên cái nhìn tổng quan về doanh thu, số lượng sản phẩm bán ra, các sản phẩm được bán nhiều nhất. Điều này hỗ trợ quản trị viên trong việc ra quyết định kinh doanh và chiến lược marketing.
* Giao diện thân thiện và tương tác: Giao diện website được thiết kế thân thiện với người dùng, dễ dàng tương tác và điều hướng. Đặc biệt, giao diện này được tối ưu hóa cho cả thiết bị di động và máy tính để bàn, đảm bảo trải nghiệm người dùng tốt trên mọi nền tảng.
* Bảo mật: Website được xây dựng với các biện pháp bảo mật tiên tiến nhằm bảo vệ thông tin người dùng và dữ liệu giao dịch.
* Áp dụng được kiến trúc Onion: Website được xây dựng theo kiến trúc Onion giúp tách biệt các thành phần ứng dụng, đảm bảo tính độc lập và dễ dàng kiểm thử. Nó tăng cường khả năng bảo trì và mở rộng ứng dụng, giảm thiểu sự phụ thuộc giữa các lớp.

## **5.2. HƯỚNG PHÁT TRIỂN**

Đề xuất phát triển: Để hệ thống “Website bán sách bằng ASP.NET Core ” có thể có nhiều chức năng hơn phù hợp với người tiêu dùng và mang đến những trải nghiệm mới mẻ cùng với việc cập nhật liên tục trong tương lai với mong muốn hệ thống có thể được sử dụng chính thức và đưa những sản phẩm thông tin hữu ích đến người dùng, em xin đề xuất một số hướng phát triển tiếp theo:

* Tối ưu hóa hiệu suất: Tiếp tục tối ưu hóa hiệu suất hệ thống để đảm bảo đáp ứng nhu cầu người dùng, đặc biệt là trong các quá trình xử lý lớn.
* Mở rộng tính năng: Phát triển thêm các tính năng mới như chat trực tiếp với nhân viên hỗ trợ khách hàng, tạo được tính năng tích điểm khách hàng qua các lần mua hàng, theo dõi đơn hàng và thông báo tự động để tăng được khả năng trải nghiệm cho khách hàng.
* Tích hợp AI: Phát triển thêm tính năng tư vấn sản phẩm bằng chat bot AI để khách hàng có thêm những sự lựa chọn phù hợp khi mua sản phẩm.

# **TÀI LIỆU THAM KHẢO**

1. Josephine Bush, “*Learn SQL Database Programming*” , ACB Bookstore, 2020.
2. Microsoft, <https://learn.microsoft.com/vi-vn/dotnet/welcome>.
3. Tài liệu ASP.NET Core, <https://xuanthulab.net/>
4. Tài liệu boostrap, <https://getbootstrap.com/docs/>
5. Tài liệu Jquery, <https://jquery.com/>
6. Tài liệu tích hợp thanh toán VnPay: <https://sandbox.vnpayment.vn/apis/>
7. Tài liệu tích hợp thanh toán PayPal: <https://developer.paypal.com/>