

TRƯỜNG ĐẠI HỌC YERSIN ĐÀ LẠT
KHOA CÔNG NGHỆ THÔNG TIN



YERSIN UNIVERSITY

BÁO CÁO MÔN HỌC
LẬP TRÌNH WEB 1

**Phát triển ứng dụng web thương mại điện tử
hỗ trợ kinh doanh thực phẩm và thức ăn
nhanh sử dụng ASP.NET Core MVC**

GVHD : Nguyễn Đức Tấn

SVTH : Nguyễn Chí Hải

Mã số SV : 2301010067

Khóa học : 2024 - 2025

Đà Lạt, tháng 6 - 2025

PHẦN NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm:

Ngày ... tháng ... năm...

Ký và ghi rõ họ tên

LỜI NÓI ĐẦU

Trước hết chúng em xin gửi lời cảm ơn đến thầy Nguyễn Đức Tấn, người đã hướng dẫn chúng em rất nhiều trong suốt quá trình tìm hiểu nghiên cứu và hoàn thành đồ án này từ lý thuyết đến ứng dụng. Sự hướng dẫn của thầy đã giúp chúng em có thêm được những hiểu biết và cách thức tạo nên một website bán hàng trực tuyến cho cửa hàng thực phẩm và đồ ăn nhanh. Do lĩnh vực của đề tài này còn tương đối rộng và kiến thức của nhóm chúng em còn nhiều hạn chế nên đồ án không thể tránh khỏi sai sót. Kính mong sự chỉ bảo thêm của các thầy và những sự góp ý của các bạn sinh viên trong lớp để nhóm có thể hoàn thiện đề tài nghiên cứu của mình hơn nữa.

Một lần nữa, nhóm chúng em xin chân thành cảm ơn các thầy trong khoa Công nghệ thông tin – Trường Đại học Yersin, những người đã nhiệt tình giảng dạy và truyền đạt những kiến thức cần thiết trong suốt thời gian chúng em học tập tại trường, đặc biệt là thầy Nguyễn Đức Tấn đã tận tình giúp đỡ và chỉ bảo giúp chúng em hoàn thành đồ án này.

MỤC LỤC

Contents

LỜI NÓI ĐẦU	3
MỤC LỤC	4
DANH MỤC HÌNH.....	6
DANH MỤC TỪ NGỮ VIẾT TẮT.....	7
MỞ ĐẦU	8
Chương 1: Cơ sở lý thuyết.....	9
1.1 Khái niệm MVC.....	9
1.2 Đặc điểm của MVC	9
1.3 Nguyên lý hoạt động của MVC	10
1.4 Công nghệ ASP.NET Core và MVC trong ASP.NET Core.....	11
Chương 2: Xây dựng ứng dụng thực tế	14
2.1 Bài toán thực tế	14
2.2 Ý nghĩa thực tế.....	15
2.3 Yêu cầu chức năng.....	16
2.4 Yêu cầu phi chức năng.....	17
2.5 Sơ đồ Use-Case.....	18
2.5.1 Use-case người dùng.....	19
2.5.2 Use-case Quản lý	28
2.6 Thiết kế cơ sở dữ liệu.....	40
2.7 Thiết kế thành phần MVC.....	50
2.7.1 Model	50
2.7.2 View	52
2.7.3 Controller	53
2.8 Triển khai và cài đặt.....	54
2.8.1 Mô tả môi trường	54
2.8.2 Thư viện framework/ thư viện	57
2.8.3 Các bước triển khai	58
Chương 3: Kết quả chương trình	63

3.1 Demo Chức năng chính	63
3.1.1 Trang chủ (Sau khi đăng nhập của quản lý)	63
3.1.2 Trang chủ (Sau khi đăng nhập của người dùng)	63
3.1.3 Cửa hàng	64
3.1.4 Đánh giá	64
3.1.5 Chi tiết sản phẩm	65
3.1.6 Thanh toán	65
3.1.7 Đơn hàng.....	66
3.1.8 Quản trị của quản lý	66
3.2 Đánh giá, nhận xét	67
Kết luận	68
1.Kết quả đạt được	68
2.Khuyết điểm.....	68
3.Hướng phát triển	68
Tài liệu tham khảo	69

DANH MỤC HÌNH

Ảnh 1.Sơ đồ Use-case chung	18
Ảnh 2.Sơ đồ Use-Case người dùng.....	19
Ảnh 3. Sơ đồ Use-case quản lý	28
Ảnh 4.Sơ đồ cơ sở dữ liệu.....	40
Ảnh 5.Cấu trúc dự án	50
Ảnh 6.Model Product.....	51
Ảnh 7. View của trang Giới thiệu	53
Ảnh 8. Controller của Trang chủ	54
Ảnh 9. Ảnh minh họa cho các ngôn ngữ FE	55
Ảnh 10. Ảnh minh họa cho ASP.Net Core	55
Ảnh 11. Ảnh minh họa cho MSSV	56
Ảnh 12. Ảnh minh họa cho Git.....	57
Ảnh 13. Ảnh thư viện và frameworks.....	58
Ảnh 14. Ảnh cấu trúc của View	59
Ảnh 15. Ảnh cấu trúc của Model	60
Ảnh 16. Ảnh cấu trúc của các Controller và Repository	61
Ảnh 17. Cấu trúc của ASP.NET Core Identity	62
Ảnh 18. Ảnh giao diện trang chủ quản lý	63
Ảnh 19. Ảnh giao diện trang chủ người dùng.....	63
Ảnh 20. Ảnh giao diện cửa hàng	64
Ảnh 21. Ảnh giao diện trang đánh giá	64
Ảnh 22. Ảnh giao diện trang chi tiết sản phẩm.....	65
Ảnh 23. Ảnh giao diện trang thanh toán	65
Ảnh 24. Ảnh giao diện trang đơn hàng	66
Ảnh 25. Ảnh giao diện trang quản trị của quản lý	66

DANH MỤC TỪ NGỮ VIẾT TẮT

STT	Tên viết tắt	Tên đầy đủ	Dịch ra tiếng Việt (Nếu là tiếng nước ngoài)
1	MVC	Model-View-Controller	
2	IoT	Internet of Things	Internet vạn vật
3	HTTP	Hypertext Transfer Protocol	Giao thức truyền tải siêu văn bản
4	LP	Luồng phụ	
5	CSDL	Cơ sở dữ liệu	
6	PK	Primary key	Khóa chính
7	FK	Foreign key	Khóa ngoại
9	GUID	Globally Unique Identifier	Mã định danh duy nhất toàn cục

MỞ ĐẦU

Trong thời đại công nghệ số bùng nổ, xu hướng tiêu dùng trực tuyến ngày càng trở nên phổ biến, đặc biệt là trong lĩnh vực thực phẩm và đồ ăn nhanh – nơi mà sự tiện lợi và tốc độ phục vụ được đặt lên hàng đầu. Người tiêu dùng hiện nay có xu hướng đặt món ăn và thực phẩm ngay tại nhà chỉ với vài thao tác đơn giản trên thiết bị di động hoặc máy tính, thay vì phải đến cửa hàng trực tiếp.

Đối với các doanh nghiệp kinh doanh thực phẩm và đồ ăn nhanh, việc xây dựng một hệ thống bán hàng trực tuyến không chỉ giúp nâng cao trải nghiệm khách hàng mà còn là giải pháp hiệu quả để mở rộng thị trường, tăng doanh thu và quản lý hoạt động kinh doanh một cách chuyên nghiệp. Một website thương mại điện tử được thiết kế bài bản sẽ cho phép khách hàng lựa chọn sản phẩm, đặt hàng nhanh chóng, đồng thời giúp doanh nghiệp dễ dàng quản lý đơn hàng, sản phẩm, khách hàng và xử lý thanh toán.

Từ thực tiễn trên, em thực hiện đề tài “Xây dựng website thương mại điện tử kinh doanh sản phẩm thực phẩm và đồ ăn nhanh” với mục tiêu phát triển một nền tảng website thân thiện, dễ sử dụng dành cho cả người dùng và quản trị viên. Website không chỉ đáp ứng nhu cầu đặt hàng tiện lợi cho khách hàng mà còn hỗ trợ quản lý sản phẩm, đơn hàng, người dùng, thống kê doanh thu, và các chức năng cần thiết khác cho phía quản trị.

Thông qua dự án này, em không chỉ vận dụng kiến thức đã học về lập trình web, cơ sở dữ liệu và thiết kế hệ thống mà còn rèn luyện thêm kỹ năng lên kế hoạch, phân tích yêu cầu và giải quyết các bài toán thực tế. Em kỳ vọng hệ thống này sẽ là giải pháp hiệu quả góp phần nâng cao hiệu suất kinh doanh trong lĩnh vực thực phẩm và đồ ăn nhanh trong bối cảnh chuyển đổi số hiện nay.

Chương 1: Cơ sở lý thuyết

1.1 Khái niệm MVC

MVC là viết tắt của Model-View-Controller, một mẫu kiến trúc phần mềm được sử dụng rộng rãi để phát triển các ứng dụng có giao diện người dùng. Mục tiêu chính của MVC là tách biệt các thành phần của ứng dụng thành ba khối chức năng chính, mỗi khối đảm nhiệm một vai trò cụ thể, giúp quản lý độ phức tạp của ứng dụng và tăng cường khả năng bảo trì. Sự ra đời của MVC được coi là một bước tiến quan trọng trong việc tổ chức mã nguồn, đặc biệt là trong các ứng dụng web và ứng dụng desktop phức tạp, nơi mà việc quản lý tương tác giữa dữ liệu, logic nghiệp vụ và giao diện người dùng trở thành một thách thức lớn.

Trước khi MVC xuất hiện, các ứng dụng thường có cấu trúc monolithic, nơi mọi chức năng đều được đặt chung trong một khối mã, dẫn đến khó khăn trong việc phát triển, kiểm thử và mở rộng. MVC đã giải quyết vấn đề này bằng cách đưa ra một phương pháp tiếp cận có tổ chức hơn, cho phép các nhà phát triển làm việc độc lập trên từng phần của ứng dụng mà không làm ảnh hưởng đến các phần khác. Điều này không chỉ nâng cao hiệu suất làm việc của nhóm mà còn giảm thiểu đáng kể số lượng lỗi phát sinh trong quá trình phát triển và bảo trì. Mẫu MVC đã trở thành một nền tảng cơ bản cho nhiều framework phát triển web phổ biến hiện nay như Ruby on Rails, Django, Laravel, ASP.NET MVC, và Spring MVC, minh chứng cho tính hiệu quả và sự linh hoạt của nó trong việc xây dựng các ứng dụng hiện đại.

1.2 Đặc điểm của MVC

Mẫu kiến trúc MVC sở hữu nhiều đặc điểm nổi bật giúp nó trở thành một lựa chọn ưu việt trong phát triển phần mềm.

Đầu tiên và quan trọng nhất là tính tách biệt rõ ràng các mối quan tâm. Mỗi thành phần (Model, View, Controller) đảm nhiệm một vai trò riêng biệt và độc lập, giảm thiểu sự phụ thuộc lẫn nhau. Điều này có nghĩa là các nhà phát triển có thể thay đổi giao diện người dùng (View) mà không cần phải thay đổi logic nghiệp vụ (Model) hoặc ngược lại. Sự tách biệt này giúp tăng cường tính module hóa của ứng dụng, làm cho mã nguồn dễ hiểu, dễ quản lý và dễ bảo trì hơn rất nhiều.

Thứ hai, MVC hỗ trợ mạnh mẽ khả năng tái sử dụng mã nguồn. Ví dụ, cùng một Model có thể được sử dụng với nhiều View khác nhau, cho phép hiển thị dữ liệu theo nhiều cách khác nhau tùy thuộc vào yêu cầu của ứng dụng hoặc loại thiết bị truy cập. Điều này đặc

biệt hữu ích trong các ứng dụng đa nền tảng hoặc khi cần cung cấp nhiều giao diện người dùng cho cùng một tập dữ liệu. Thứ ba, MVC tăng cường khả năng kiểm thử. Khi các thành phần được tách biệt, việc viết các bài kiểm thử đơn vị cho từng thành phần trở nên dễ dàng hơn. Ví dụ, bạn có thể kiểm thử logic nghiệp vụ của Model mà không cần phải khởi tạo View hoặc Controller, giúp quá trình kiểm thử nhanh chóng và hiệu quả hơn. Đặc điểm thứ tư là phát triển song song. Nhờ sự tách biệt rõ ràng, nhiều lập trình viên có thể làm việc trên các phần khác nhau của ứng dụng đồng thời mà không bị chồng chéo hay xung đột quá nhiều. Một người có thể tập trung vào việc thiết kế giao diện (View), trong khi người khác xây dựng logic nghiệp vụ (Model), và một người khác nữa xử lý luồng điều khiển (Controller). Điều này giúp tăng tốc độ phát triển và giảm thời gian đưa sản phẩm ra thị trường. Cuối cùng, MVC cải thiện khả năng mở rộng của ứng dụng. Khi ứng dụng phát triển và cần thêm các tính năng mới, việc thêm các Model, View hoặc Controller mới mà không làm ảnh hưởng đến các phần hiện có sẽ dễ dàng hơn nhiều. Điều này giúp ứng dụng có thể phát triển linh hoạt theo thời gian và đáp ứng được các yêu cầu thay đổi của người dùng hoặc thị trường.

1.3 Nguyên lý hoạt động của MVC

Nguyên lý hoạt động của MVC xoay quanh sự tương tác phối hợp giữa ba thành phần chính: Model, View và Controller, tạo thành một chu trình xử lý dữ liệu và hiển thị thông tin một cách có tổ chức. Khi một người dùng tương tác với ứng dụng, ví dụ như nhấp vào một nút hoặc gửi một biểu mẫu, Controller là thành phần đầu tiên tiếp nhận yêu cầu. Controller đóng vai trò như một bộ điều phối, nó phân tích yêu cầu từ người dùng và xác định hành động cần thực hiện. Dựa trên yêu cầu đó, Controller sẽ tương tác với Model. Model là trái tim của ứng dụng, chứa toàn bộ logic nghiệp vụ, dữ liệu và các quy tắc quản lý dữ liệu. Nó không chỉ đơn thuần là một cơ sở dữ liệu mà còn bao gồm các đối tượng đại diện cho dữ liệu, các phương thức xử lý, lưu trữ, truy vấn, và xác thực dữ liệu. Khi Controller yêu cầu, Model sẽ xử lý các tác vụ liên quan đến dữ liệu, ví dụ như lấy dữ liệu từ cơ sở dữ liệu, cập nhật thông tin, hoặc thực hiện các phép tính phức tạp.

Sau khi Model hoàn thành việc xử lý và cung cấp dữ liệu đã được cập nhật hoặc kết quả của một thao tác, Controller sẽ tiếp tục tương tác với View. View là thành phần chịu trách nhiệm hiển thị dữ liệu cho người dùng. Nó không chứa bất kỳ logic nghiệp vụ nào, mà chỉ tập trung vào việc trình bày thông tin một cách trực quan và dễ hiểu. Controller sẽ truyền dữ liệu từ Model đến View, và View sẽ sử dụng dữ liệu đó để tạo ra giao diện người dùng phù hợp. Quá trình này có thể bao gồm việc định dạng dữ liệu, sắp xếp thông tin, hoặc tạo ra các phần tử đồ họa.

Quan trọng là View không tự mình lấy dữ liệu mà nó nhận dữ liệu từ Controller. Điều này giúp đảm bảo sự tách biệt rõ ràng giữa logic nghiệp vụ và giao diện người dùng. Sau khi View hoàn tất việc tạo giao diện, nó sẽ trả về kết quả hiển thị cho người dùng. Chu trình này lặp lại mỗi khi có một tương tác mới từ phía người dùng. Có thể hình dung một cách đơn giản là: người dùng đưa yêu cầu (qua View), Controller nhận yêu cầu và chỉ đạo Model xử lý, Model xử lý xong và trả dữ liệu về cho Controller, Controller đưa dữ liệu đó cho View để hiển thị lại cho người dùng. Sự phối hợp nhịp nhàng này giữa ba thành phần đảm bảo rằng mỗi phần tử có một trách nhiệm rõ ràng, giảm thiểu sự phụ thuộc lẫn nhau và tạo ra một kiến trúc ứng dụng linh hoạt, dễ mở rộng và bảo trì.

1.4 Công nghệ ASP.NET Core và MVC trong ASP.NET Core

ASP.NET Core là một nền tảng phát triển web mã nguồn mở và đa nền tảng, được Microsoft phát triển để xây dựng các ứng dụng web hiện đại, dịch vụ đám mây, và các ứng dụng IoT. Khác với các phiên bản ASP.NET trước đây chỉ chạy trên Windows, ASP.NET Core đã được thiết kế lại hoàn toàn để có thể hoạt động trên nhiều hệ điều hành khác nhau như Windows, Linux và macOS, mang lại sự linh hoạt chưa từng có cho các nhà phát triển. Một trong những đặc điểm nổi bật của ASP.NET Core là tính mô-đun hóa cao, cho phép các nhà phát triển chỉ bao gồm những thành phần cần thiết cho ứng dụng của mình, giúp giảm kích thước ứng dụng và tăng hiệu suất. Nền tảng này cũng được xây dựng dựa trên khái niệm middleware (Phần mềm trung gian) và dependency injection (Chèn phụ thuộc), giúp cấu trúc ứng dụng trở nên linh hoạt, dễ kiểm thử và mở rộng. Hiệu suất của ASP.NET Core cũng được cải thiện đáng kể so với các phiên bản trước, xếp hạng cao trong các bài kiểm tra benchmark về hiệu năng cho các ứng dụng web. Đặc biệt, ASP.NET Core được thiết kế để hỗ trợ phát triển các ứng dụng dựa trên microservices, cho phép xây dựng các hệ thống lớn bằng cách chia nhỏ thành các dịch vụ độc lập, dễ quản lý và triển khai. Cộng đồng ASP.NET Core cũng rất lớn mạnh, với nhiều tài liệu, thư viện và công cụ hỗ trợ, giúp các nhà phát triển dễ dàng học hỏi và triển khai các dự án phức tạp. Sự ra đời của ASP.NET Core đánh dấu một bước tiến lớn trong chiến lược phát triển phần mềm của Microsoft, hướng tới một nền tảng mở và đa nền tảng, đáp ứng nhu cầu ngày càng cao của thị trường công nghệ thông tin toàn cầu.

Trong khuôn khổ của ASP.NET Core, mẫu kiến trúc MVC đóng một vai trò trung tâm và là một trong những cách tiếp cận phổ biến nhất để xây dựng các ứng dụng web. ASP.NET Core MVC kế thừa và phát triển những ưu điểm của mẫu MVC truyền thống, đồng thời tích hợp chặt chẽ với các tính năng mới của ASP.NET Core để mang lại một trải nghiệm phát triển mạnh mẽ và hiệu quả. Về cơ bản, MVC trong ASP.NET Core vẫn tuân thủ nguyên tắc tách biệt các mối quan tâm: Model đại diện cho dữ liệu và logic nghiệp vụ,

View chịu trách nhiệm hiển thị giao diện người dùng, và Controller xử lý các yêu cầu và điều phối tương tác giữa Model và View.

Tuy nhiên, ASP.NET Core MVC còn mang đến nhiều cải tiến đáng kể. Chẳng hạn, nó hỗ trợ Routing mạnh mẽ, cho phép định nghĩa các địa chỉ thân thiện và linh hoạt, ánh xạ tới các hành động (actions) trong Controller. Ví dụ, một địa chỉ như `/Products/Details/5` có thể được ánh xạ tới phương thức `Details` trong `ProductsController` với tham số id là 5. Ngoài ra, Action Filters là một tính năng mạnh mẽ khác, cho phép thực thi logic trước hoặc sau khi một hành động được thực thi, hữu ích cho các tác vụ như xác thực, ủy quyền, hoặc ghi log. Model Binding trong ASP.NET Core MVC tự động ánh xạ dữ liệu từ yêu cầu HTTP (ví dụ: dữ liệu từ biểu mẫu, chuỗi truy vấn) vào các đối tượng Model trong phương thức của Controller, giúp giảm thiểu đáng kể mã lặp lại. Validation cũng được tích hợp sâu, cho phép định nghĩa các quy tắc kiểm tra tính hợp lệ của dữ liệu trực tiếp trên Model, và Framework sẽ tự động kiểm tra và báo lỗi nếu dữ liệu không hợp lệ trước khi xử lý.

Để minh họa nguyên lý hoạt động của MVC trong ASP.NET Core một cách trực quan và liên mạch, hãy hình dung một kịch bản đơn giản: quá trình đăng ký tài khoản mới trên một website.

Minh họa về quy trình Đăng ký tài khoản mới:

Khi một người dùng truy cập vào một trang web được xây dựng bằng ASP.NET Core MVC và mong muốn tạo một tài khoản mới, mọi thứ bắt đầu từ View. View là trang mà người dùng nhìn thấy, nơi hiển thị form đăng ký với các trường như tên đăng nhập, mật khẩu, email. View này được thiết kế để trực quan và dễ sử dụng, hoàn toàn không chứa bất kỳ logic xử lý dữ liệu nào, nó chỉ có nhiệm vụ hiển thị thông tin và nhận các tương tác từ người dùng. Khi người dùng điền đầy đủ các thông tin cần thiết vào form và nhấn nút "Đăng ký", một yêu cầu sẽ được gửi đi, mang theo dữ liệu người dùng vừa nhập.

Yêu cầu này ngay lập tức được chuyển tới Controller, đóng vai trò như một người quản lý trung tâm hoặc một nhân viên lễ tân của hệ thống. Controller nhận yêu cầu từ người dùng và hiểu rằng họ muốn tạo một tài khoản mới. Nó không trực tiếp lưu dữ liệu hay kiểm tra tính hợp lệ, mà nhiệm vụ của nó là điều phối. Controller sẽ thu thập dữ liệu người dùng đã nhập thông qua một cơ chế gọi là Model Binding, tự động ánh xạ thông tin từ form vào một đối tượng dữ liệu cụ thể, ví dụ như đối tượng `User` hoặc `RegisterViewModel`. Sau khi dữ liệu được thu thập, Controller sẽ kiểm tra sơ bộ tính hợp lệ của nó dựa trên các quy tắc đã được định nghĩa sẵn trong Model (như độ dài tối thiểu của mật khẩu, định dạng email chính xác).

Tiếp theo, Controller sẽ giao phó công việc kiểm tra và lưu trữ dữ liệu này cho Model. Model chính là trái tim của ứng dụng, chuyên trách về mọi thứ liên quan đến dữ liệu và logic nghiệp vụ. Model sẽ thực hiện các thao tác chuyên sâu hơn: kiểm tra xem tên đăng nhập đã tồn tại chưa, mã hóa mật khẩu, và cuối cùng là lưu thông tin tài khoản mới vào cơ sở dữ liệu. Model hoàn toàn độc lập với giao diện người dùng; nó chỉ tập trung vào việc quản lý dữ liệu và đảm bảo các quy tắc kinh doanh được tuân thủ. Trong trường hợp dữ liệu không hợp lệ (ví dụ, tên đăng nhập đã được sử dụng), Model sẽ báo cáo lại cho Controller về lỗi cụ thể. Nếu mọi thứ thành công, Model sẽ xác nhận việc tạo tài khoản.

Khi Model hoàn thành nhiệm vụ của mình, dù là thành công hay thất bại, nó sẽ gửi kết quả trở lại cho Controller. Dựa vào kết quả này, Controller sẽ đưa ra quyết định tiếp theo. Nếu tài khoản được tạo thành công, Controller có thể hướng dẫn người dùng đến một trang chúc mừng hoặc tự động đăng nhập họ vào hệ thống. Để làm điều này, Controller sẽ "chỉ đạo" View hiển thị trang tương ứng. Ngược lại, nếu có lỗi xảy ra (ví dụ, mật khẩu quá yếu), Controller sẽ "nói" với View rằng "hãy hiển thị lại form đăng ký cùng với các thông báo lỗi cụ thể để người dùng biết và sửa chữa".

Cuối cùng, View sẽ nhận dữ liệu và chỉ dẫn từ Controller, sau đó tự động cập nhật giao diện người dùng. Nếu đăng ký thành công, người dùng sẽ thấy thông báo chào mừng hoặc trang chủ của tài khoản. Nếu có lỗi, họ sẽ thấy lại form đăng ký, nhưng lần này các trường nhập liệu có thể được điền sẵn thông tin người dùng vừa nhập, kèm theo các dòng chữ đỏ cảnh báo lỗi bên dưới mỗi trường cần sửa. Toàn bộ quy trình này diễn ra một cách liên tục và mượt mà, cho thấy sự phối hợp nhịp nhàng giữa ba thành phần Model, View và Controller, giúp xây dựng một ứng dụng web mạnh mẽ, có tổ chức và dễ dàng mở rộng.

Chương 2: Xây dựng ứng dụng thực tế

2.1 Bài toán thực tế

Trong bối cảnh toàn cầu hóa và sự phát triển vượt bậc của công nghệ thông tin, đặc biệt là xu hướng số hóa trong mọi lĩnh vực của đời sống, thói quen tiêu dùng của xã hội đã có những chuyển biến sâu sắc. Người tiêu dùng ngày nay không chỉ tìm kiếm sản phẩm chất lượng mà còn đề cao sự tiện lợi, nhanh chóng và trải nghiệm mua sắm cá nhân hóa. Trong ngành thực phẩm và đồ ăn nhanh, điều này càng trở nên rõ nét hơn bao giờ hết, khi mà quỹ thời gian hạn hẹp khiến nhiều người ưu tiên các giải pháp mua sắm trực tuyến. Từ đó, bài toán trọng tâm được đặt ra là làm thế nào để thiết lập một nền tảng thương mại điện tử chuyên biệt, không chỉ đơn thuần là một kênh bán hàng mà còn phải là một giải pháp toàn diện để đáp ứng những nhu cầu đa dạng và phức tạp của thị trường. Hiện tại, dù có sự xuất hiện của một số ứng dụng giao đồ ăn và các siêu thị trực tuyến, vẫn tồn tại một khoảng trống lớn về một hệ thống tích hợp, có khả năng phục vụ một cách liền mạch cho cả nhu cầu về thực phẩm tươi sống, thực phẩm đóng gói và đồ ăn nhanh, đồng thời đảm bảo chất lượng, an toàn vệ sinh thực phẩm và sự minh bạch về nguồn gốc sản phẩm.

Thách thức đầu tiên và quan trọng nhất nằm ở việc xây dựng một kiến trúc kỹ thuật vững chắc và linh hoạt, có khả năng chịu tải cao để xử lý một lượng lớn các giao dịch đồng thời, từ hàng ngàn đơn hàng thực phẩm tươi mỗi ngày đến hàng chục ngàn lượt đặt đồ ăn nhanh vào giờ cao điểm. Hệ thống này cần phải có tính sẵn sàng cao, đảm bảo hoạt động liên tục mà không gặp sự cố, đồng thời phải bảo mật tuyệt đối thông tin cá nhân và dữ liệu thanh toán của khách hàng. Bên cạnh đó, việc thiết kế một giao diện người dùng trực quan và trải nghiệm người dùng mượt mà là yếu tố then chốt để thu hút và giữ chân khách hàng. Nền tảng phải cung cấp các tính năng tìm kiếm thông minh, bộ lọc chi tiết theo loại sản phẩm, giá cả, thương hiệu, hoặc thậm chí là các thuộc tính chuyên biệt như thực phẩm hữu cơ, không gluten. Tính năng gợi ý sản phẩm cá nhân hóa, dựa trên lịch sử mua hàng và sở thích của người dùng, cũng sẽ nâng cao đáng kể trải nghiệm mua sắm. Thách thức khác là việc tích hợp hiệu quả với các hệ thống quản lý đơn hàng. Cuối cùng, bài toán còn bao gồm việc xây dựng một hệ thống phản hồi và đánh giá hiệu quả, cho phép người dùng chia sẻ trải nghiệm của họ, từ đó giúp doanh nghiệp liên tục cải thiện chất lượng sản phẩm và dịch vụ. Việc giải quyết thành công bài toán này không chỉ mang lại lợi ích kinh tế mà còn góp phần vào việc nâng cao chất lượng cuộc sống và sự tiện nghi cho xã hội.

2.2 Ý nghĩa thực tế

Việc triển khai một website thương mại điện tử chuyên biệt để kinh doanh sản phẩm thực phẩm và đồ ăn nhanh mang lại ý nghĩa thực tiễn vô cùng to lớn, tạo ra những giá trị bền vững cho cả người tiêu dùng, cộng đồng doanh nghiệp và thậm chí là toàn bộ nền kinh tế số.

Đối với người tiêu dùng, nền tảng này đóng vai trò như một giải pháp đột phá, giúp họ tiếp cận nguồn thực phẩm và đồ ăn tiện lợi hơn bao giờ hết. Trong bối cảnh cuộc sống hiện đại ngày càng bận rộn, áp lực công việc và gia đình khiến quỹ thời gian eo hẹp, khả năng mua sắm thực phẩm tươi sống hay đặt món ăn nhanh chỉ với vài cú nhấp chuột trên điện thoại hay máy tính là một lợi ích không thể phủ nhận. Người dân có thể dễ dàng duyệt qua hàng ngàn sản phẩm, so sánh giá cả, đọc đánh giá từ những người mua trước, và lựa chọn những món đồ phù hợp nhất với nhu cầu và sở thích cá nhân, mà không cần phải di chuyển đến siêu thị hay nhà hàng truyền thống. Điều này không chỉ tiết kiệm đáng kể thời gian và công sức mà còn giảm thiểu chi phí đi lại. Hơn nữa, việc nền tảng cung cấp thông tin minh bạch về nguồn gốc sản phẩm, thành phần dinh dưỡng và các chứng nhận an toàn vệ sinh thực phẩm sẽ góp phần nâng cao ý thức về sức khỏe và giúp người tiêu dùng đưa ra những quyết định mua sắm thông thái hơn, từ đó cải thiện chất lượng bữa ăn hàng ngày. Khả năng thanh toán đa dạng (qua thẻ, ví điện tử) và dịch vụ giao hàng tận nơi nhanh chóng, chính xác sẽ hoàn thiện trải nghiệm mua sắm tiện nghi và đáng tin cậy.

Về phía doanh nghiệp, website thương mại điện tử mở ra một kênh phân phối hoàn toàn mới, phá vỡ mọi rào cản địa lý và thời gian, giúp họ tiếp cận một lượng lớn khách hàng tiềm năng mà không cần đầu tư quá nhiều vào mặt bằng vật lý. Các nhà cung cấp thực phẩm, nông trại nhỏ, hay các cửa hàng ăn uống có thể dễ dàng đưa sản phẩm của mình lên trực tuyến, mở rộng thị trường và tăng doanh thu đáng kể. Nền tảng này cho phép doanh nghiệp tối ưu hóa quy trình quản lý hoạt động, từ việc kiểm soát tồn kho một cách chính xác, xử lý đơn hàng tự động cho đến phân tích dữ liệu về hành vi mua sắm của khách hàng. Thông qua các công cụ phân tích tích hợp, doanh nghiệp có thể hiểu rõ hơn về sở thích, xu hướng tiêu dùng, từ đó đưa ra các chiến lược marketing nhắm mục tiêu hiệu quả, phát triển sản phẩm mới và tối ưu hóa chuỗi cung ứng. Đặc biệt, đối với các doanh nghiệp vừa và nhỏ hay các hộ kinh doanh cá thể, một website thương mại điện tử chuyên nghiệp là cơ hội vàng để xây dựng thương hiệu trực tuyến, tạo dựng uy tín và cạnh tranh công bằng hơn với các chuỗi lớn, giảm thiểu chi phí vận hành truyền thống. Đây không chỉ là một công cụ bán hàng mà còn là một cầu nối quan trọng giữa nhà cung cấp và người tiêu dùng, thúc đẩy sự chuyển đổi số mạnh mẽ trong ngành thực phẩm và dịch vụ ăn uống, góp phần vào sự phát triển chung của nền kinh tế số và nâng cao tiện ích xã hội.

2.3 Yêu cầu chức năng

Đối với người dùng thông thường, tức là khách hàng, các chức năng cốt lõi xoay quanh trải nghiệm mua sắm tiện lợi và cá nhân hóa. Đầu tiên, người dùng cần có khả năng quản lý tài khoản một cách linh hoạt, bao gồm việc đăng ký tài khoản mới một cách dễ dàng thông qua email hoặc số điện thoại, đăng nhập nhanh chóng, và đăng xuất khi không sử dụng. Sau khi đăng nhập, họ phải có quyền cập nhật thông tin cá nhân như tên, địa chỉ giao hàng chi tiết, số điện thoại liên hệ, cũng như quản lý mật khẩu để đảm bảo an toàn. Một tính năng không thể thiếu là khả năng xem lại toàn bộ lịch sử đơn hàng đã đặt, giúp họ dễ dàng theo dõi chi tiêu và các món đồ đã mua. Thứ hai, để tìm kiếm sản phẩm mong muốn, hệ thống phải cung cấp chức năng tìm kiếm và duyệt sản phẩm mạnh mẽ. Điều này bao gồm khả năng tìm kiếm chính xác theo từ khóa (tên sản phẩm, mô tả), áp dụng các bộ lọc đa dạng như danh mục (rau củ, thịt tươi, các loại hạt, trái cây, các loại gia vị). Sau khi tìm thấy, người dùng cần có khả năng xem chi tiết sản phẩm để nắm rõ thông tin đầy đủ nhất, bao gồm mô tả chi tiết, thành phần, giá cả, nhiều hình ảnh chất lượng cao từ các góc độ khác nhau, và quan trọng nhất là các đánh giá, bình luận từ những người mua trước, cung cấp cái nhìn khách quan về chất lượng sản phẩm và dịch vụ.

Khi đã chọn được sản phẩm, chức năng quản lý giỏ hàng trở nên thiết yếu. Người dùng phải dễ dàng thêm các mặt hàng vào giỏ, điều chỉnh số lượng sản phẩm, hoặc loại bỏ bất kỳ mặt hàng nào không còn cần thiết. Sau khi hoàn tất việc lựa chọn, quá trình đặt hàng phải diễn ra một cách trực quan và ít bước nhất có thể, bắt đầu bằng việc xác nhận địa chỉ giao hàng, xem lại tổng quan đơn hàng một lần cuối trước khi xác nhận. Ngay sau khi đặt hàng, hệ thống cần có chức năng theo dõi đơn hàng, cho phép người dùng kiểm tra trạng thái hiện tại (đang xử lý, đã xác nhận, đã hủy) và nhận thông báo cập nhật qua email hoặc tin nhắn. Cuối cùng, để xây dựng cộng đồng và nâng cao chất lượng, người dùng cần có khả năng đánh giá và bình luận về sản phẩm đã mua hoặc trải nghiệm dịch vụ giao hàng, và một kênh hỗ trợ và liên hệ rõ ràng để giải đáp thắc mắc, xử lý sự cố.

Đối với người quản trị hệ thống (Admin), các chức năng tập trung vào việc vận hành và duy trì toàn bộ website một cách hiệu quả. Chức năng quản lý danh mục sản phẩm cho phép Admin thêm mới, chỉnh sửa hoặc xóa bỏ các danh mục để tổ chức sản phẩm một cách logic. Kế đến là quản lý sản phẩm chi tiết, bao gồm khả năng thêm sản phẩm mới với đầy đủ thông tin từ tên, mô tả, giá, hình ảnh, cũng như sửa đổi thông tin hiện có hoặc xóa vĩnh viễn. Chức năng quản lý đơn hàng là trung tâm của mọi hoạt động, cho phép Admin xem tổng quan tất cả các đơn hàng, cập nhật trạng thái đơn hàng theo từng giai đoạn (ví dụ: từ "đang chờ xác nhận" đến "đã xác nhận"), tìm kiếm đơn hàng theo nhiều tiêu chí. Đồng thời,

quản lý người dùng là cần thiết để Admin có thể xem danh sách khách hàng, tìm kiếm, kích hoạt hoặc khóa tài khoản khi cần. Cuối cùng, các báo cáo và thống kê về doanh thu, số lượng đơn hàng, sản phẩm bán chạy nhất là không thể thiếu để Admin đưa ra quyết định kinh doanh chiến lược.

2.4 Yêu cầu phi chức năng

Yêu cầu phi chức năng là những tiêu chí quan trọng đánh giá chất lượng của hệ thống, chúng không mô tả những gì hệ thống làm mà mô tả cách hệ thống thực hiện các chức năng đó. Đây là những yếu tố quyết định đến trải nghiệm tổng thể của người dùng và sự ổn định, an toàn của ứng dụng.

Trước hết, hiệu năng là một yếu tố then chốt. Website phải đảm bảo thời gian tải trang nhanh chóng, lý tưởng là không quá 3-5 giây ngay cả khi người dùng truy cập từ các thiết bị có kết nối mạng chậm hoặc trong điều kiện có hàng ngàn người dùng truy cập cùng lúc. Thời gian xử lý các thao tác quan trọng như thêm sản phẩm vào giỏ hàng hoặc xác nhận đơn hàng cần phải được tối ưu hóa, diễn ra dưới 2 giây để tránh gây khó chịu cho người dùng.

Đặc tính quan trọng tiếp theo của một hệ thống hiệu quả là khả năng mở rộng. Một website thương mại điện tử chuyên về thực phẩm và đồ ăn nhanh cần được thiết kế để dễ dàng thích nghi và phát triển cùng với quy mô kinh doanh. Điều này có nghĩa là, hệ thống phải có khả năng xử lý lượng người dùng và giao dịch tăng đột biến trong các chiến dịch khuyến mãi lớn hay vào mùa cao điểm mà không bị giảm sút hiệu năng. Kiến trúc cần đủ linh hoạt để cho phép tích hợp thêm các chức năng mới, mở rộng danh mục sản phẩm hay liên kết với các đối tác bên ngoài như các cổng thanh toán hay dịch vụ giao hàng mới một cách liền mạch, không làm ảnh hưởng lớn đến cấu trúc nền tảng hiện có. Sự linh hoạt này đảm bảo website có thể liên tục cải tiến và phát triển theo nhu cầu thị trường, tránh việc phải xây dựng lại toàn bộ hệ thống khi quy mô hoạt động tăng lên đáng kể.

Bên cạnh khả năng mở rộng, khả năng sử dụng là yếu tố then chốt ảnh hưởng trực tiếp đến trải nghiệm của người tiêu dùng. Giao diện người dùng phải được thiết kế một cách trực quan, dễ hiểu và dễ điều hướng, giúp người dùng dễ dàng tìm kiếm sản phẩm, thêm vào giỏ hàng và hoàn tất các tác vụ mong muốn mà không cần nhiều hướng dẫn hay mất thời gian làm quen. Quy trình đặt hàng cần được tối giản hóa, chỉ với vài bước cần thiết và cung cấp phản hồi rõ ràng ở mỗi giai đoạn, tạo cảm giác thuận tiện và nhanh chóng cho người mua. Website cũng cần phải tương thích tốt trên nhiều nền tảng và thiết bị khác nhau, bao gồm các trình duyệt web phổ biến và đặc biệt là trên các thiết bị di động (điện thoại thông minh, máy tính bảng) thông qua thiết kế đáp ứng hoặc ứng dụng di động riêng biệt, đảm bảo trải nghiệm liền mạch dù người dùng truy cập từ bất cứ đâu. Cuối cùng, khả năng

bảo trì của mã nguồn là yếu tố quyết định đến chi phí và hiệu quả trong dài hạn. Mã nguồn phải được viết rõ ràng, có cấu trúc logic, tuân thủ các nguyên tắc thiết kế phần mềm tốt, dễ dàng cho các nhà phát triển khác hiểu, sửa đổi và nâng cấp hệ thống trong tương lai mà không gây ra lỗi phát sinh, từ đó đảm bảo tuổi thọ và sự phát triển bền vững của ứng dụng.

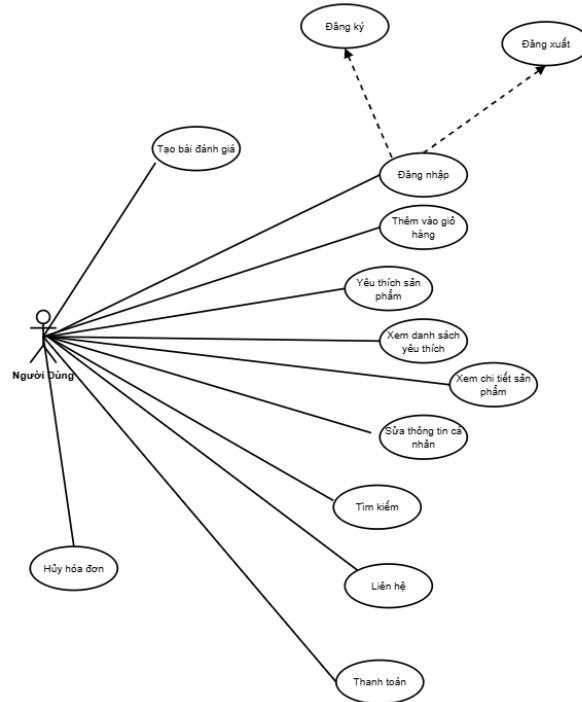
2.5 Sơ đồ Use-Case



Ảnh 1. Sơ đồ Use-case chung

Sơ đồ Use-Case tổng thể các chức năng của website thương mại điện tử, thể hiện rõ vai trò của hai tác nhân chính: Người Dùng và Quản Lý. Người Dùng là trung tâm của các tương tác mua sắm, từ việc Đăng ký, Đăng nhập, Đăng xuất, đến Thêm vào giỏ hàng, Thanh toán và Hủy hóa đơn, cùng với các tác vụ phụ trợ như Tạo bài đánh giá, Yêu thích sản phẩm, Xem chi tiết sản phẩm, và Liên hệ. Trong khi đó, tác nhân Quản Lý có quyền kiểm soát toàn bộ hệ thống, thực hiện các tác vụ quản trị từ Quản lý sản phẩm, Quản lý bài viết, Quản lý vai trò, Quản lý thư viện ảnh, cho đến Xác nhận hóa đơn và xem Thống kê & Báo cáo. Mỗi quan hệ giữa Đăng nhập với Đăng ký và Đăng xuất cũng được thể hiện rõ, tạo nên một cái nhìn tổng quan về phạm vi và luồng hoạt động chính của ứng dụng.

2.5.1 Use-case người dùng



Ảnh 2. Sơ đồ Use-Case người dùng

-Đặc tả use-case người dùng

1) Đăng kí

Mô tả	Cho phép người dùng mới tạo một tài khoản trên hệ thống để có thể thực hiện các giao dịch và quản lý thông tin cá nhân.
Tác nhân chính	Người Dùng
Luồng chính	<ol style="list-style-type: none"> 1. Người Dùng truy cập trang Đăng ký. 2. Hệ thống hiển thị form Đăng ký. 3. Người Dùng nhập thông tin cần thiết (tên đăng nhập, email, mật khẩu, xác nhận mật khẩu, v.v.). 4. Người Dùng nhấn nút "Đăng ký". 5. Hệ thống kiểm tra tính hợp lệ của dữ liệu. 6. Hệ thống lưu thông tin tài khoản mới vào cơ sở dữ liệu. 7. Hệ thống thông báo Đăng ký thành 8. Hệ thống tự động đăng nhập người dùng hoặc chuyển hướng đến trang Đăng nhập.

Luồng ngoại lệ	<p>LP1: Thông tin không hợp lệ:</p> <ol style="list-style-type: none"> Hệ thống phát hiện dữ liệu thiếu hoặc sai định dạng. Hệ thống hiển thị thông báo lỗi tương ứng bên cạnh trường nhập liệu. Người Dùng sửa lỗi và nhấn “Đăng ký”. <p>LP2: Tên đăng nhập/Email đã tồn tại:</p> <ol style="list-style-type: none"> Hệ thống kiểm tra và phát hiện tên đăng nhập hoặc email đã có trong hệ thống. Hệ thống hiển thị thông báo lỗi rõ ràng. Người Dùng nhập thông tin khác và nhấn “Đăng ký”.
----------------	--

2) Đăng nhập

Mô tả	Cho phép người dùng đã có tài khoản truy cập vào các chức năng yêu cầu xác thực của hệ thống.
Tác nhân chính	Người Dùng
Luồng chính	<ol style="list-style-type: none"> Người Dùng truy cập trang Đăng nhập. Hệ thống hiển thị form Đăng nhập. Người Dùng nhập tên đăng nhập/email và mật khẩu. Người Dùng nhấn nút "Đăng nhập". Hệ thống xác thực thông tin đăng nhập. Nếu thông tin đúng, Hệ thống cấp quyền truy cập và chuyển hướng Người Dùng đến trang chính
Luồng ngoại lệ	<p>LP1: Sai thông tin đăng nhập:</p> <ol style="list-style-type: none"> Hệ thống thông báo "Tên đăng nhập hoặc mật khẩu không đúng". Người Dùng có thể thử lại hoặc chọn "Quên mật khẩu". <p>LP2: Tài khoản bị khóa:</p> <ol style="list-style-type: none"> Hệ thống thông báo tài khoản đã bị khóa (do quá nhiều lần thử sai hoặc bị Admin khóa). Người Dùng được hướng dẫn liên hệ hỗ trợ hoặc chờ thời gian mở khóa. <p>LP3: Chưa xác thực Email/Số điện thoại (nếu có yêu cầu):</p>

	a. Hệ thống thông báo yêu cầu xác thực email/số điện thoại. b. Hướng dẫn người dùng thực hiện xác thực.
--	--

3) Đăng xuất

Mô tả	Cho phép người dùng kết thúc phiên làm việc an toàn trên hệ thống.
Tác nhân chính	Người Dùng
Luồng chính	1. Người Dùng đang ở trạng thái đã đăng nhập. 2. Người Dùng nhấn vào nút "Đăng xuất". 3. Hệ thống xóa phiên làm việc của người dùng. 4. Hệ thống chuyển hướng Người Dùng về trang Đăng nhập hoặc Trang Chủ (ở trạng thái chưa đăng nhập).
Luồng ngoại lệ	Không có luồng phụ đáng kể, đây là một hành động đơn giản.

4) Tìm kiếm

Mô tả	Cho phép người dùng tìm kiếm sản phẩm hoặc thông tin cụ thể trên website.
Tác nhân chính	Người Dùng
Luồng chính	1. Người Dùng nhập từ khóa vào ô tìm kiếm. 2. Người Dùng nhấn Enter hoặc biểu tượng tìm kiếm. 3. Hệ thống thực hiện tìm kiếm trong cơ sở dữ liệu sản phẩm, blog, hoặc danh mục. 4. Hệ thống hiển thị trang kết quả tìm kiếm với danh sách các sản phẩm/bài viết/thông tin liên quan.
Luồng ngoại lệ	LP1: Không tìm thấy kết quả: a. Hệ thống thông báo "Không tìm thấy kết quả phù hợp với từ khóa của bạn." b. Gợi ý các từ khóa liên quan hoặc sản phẩm phổ biến.

5) Xem chi tiết sản phẩm

Mô tả	Cho phép người dùng xem thông tin đầy đủ về một sản phẩm cụ thể.
Tác nhân chính	Người Dùng
Luồng chính	<ol style="list-style-type: none"> 1. Người Dùng nhấp vào tên hoặc hình ảnh của một sản phẩm từ trang danh sách hoặc kết quả tìm kiếm. 2. Hệ thống tải và hiển thị trang chi tiết sản phẩm, bao gồm: tên, giá, mô tả, hình ảnh lớn, thông tin nhà cung cấp, đánh giá trung bình, và các thông tin liên quan khác.
Luồng ngoại lệ	<p>LP1: Sản phẩm không tồn tại:</p> <ol style="list-style-type: none"> a. Hệ thống thông báo "Sản phẩm không tìm thấy" hoặc "Sản phẩm không còn kinh doanh". b. Chuyển hướng người dùng về trang danh sách sản phẩm hoặc trang chủ.

6) Thêm vào giỏ hàng

Mô tả	Cho phép người dùng thêm một sản phẩm vào giỏ hàng để chuẩn bị mua sắm.
Tác nhân chính	Người Dùng
Luồng chính	<ol style="list-style-type: none"> 1. Người Dùng đang xem trang chi tiết sản phẩm hoặc danh sách sản phẩm. 2. Người Dùng chọn số lượng sản phẩm (nếu có tùy chọn). 3. Người Dùng nhấn nút "Thêm vào giỏ hàng". 4. Hệ thống kiểm tra số lượng tồn kho và thêm sản phẩm vào giỏ hàng của người dùng. 5. Hệ thống hiển thị thông báo "Đã thêm vào giỏ hàng" hoặc cập nhật biểu tượng giỏ hàng.
Luồng ngoại lệ	<p>LP1: Sản phẩm hết hàng/không đủ số lượng:</p> <ol style="list-style-type: none"> a. Hệ thống thông báo "Sản phẩm đã hết hàng" hoặc "Số lượng yêu cầu vượt quá tồn kho". b. Không thêm sản phẩm vào giỏ.

7) Yêu thích sản phẩm

Mô tả	Cho phép người dùng đánh dấu các sản phẩm quan tâm để dễ dàng xem lại sau này.
Tác nhân chính	Người Dùng
Luồng chính	<ol style="list-style-type: none"> 1. Người Dùng đang xem trang chi tiết sản phẩm hoặc danh sách sản phẩm. 2. Người Dùng nhấn vào biểu tượng "Yêu thích" 3. Hệ thống lưu sản phẩm này vào danh sách yêu thích của người dùng. 4. Hệ thống thay đổi biểu tượng để thể hiện sản phẩm đã được yêu thích.
Luồng ngoại lệ	LP1: Sản phẩm đã có trong danh sách yêu thích: a. Hệ thống thông báo hoặc không làm gì, giữ nguyên trạng thái.

8) Xem danh sách yêu thích

Mô tả	Cho phép người dùng xem lại các sản phẩm mà họ đã đánh dấu yêu thích.
Tác nhân chính	Người Dùng
Luồng chính	<ol style="list-style-type: none"> 1. Người Dùng truy cập trang "Danh sách yêu thích" từ menu hoặc profile cá nhân. 2. Hệ thống tải và hiển thị danh sách các sản phẩm đã được người dùng yêu thích. 3. Người Dùng có thể nhấp vào từng sản phẩm để xem chi tiết hoặc xóa khỏi danh sách.
Luồng ngoại lệ	LP1: Danh sách yêu thích trống: a. Hệ thống hiển thị thông báo "Chưa có sản phẩm nào trong danh sách yêu thích của bạn." b. Gợi ý khách hàng duyệt sản phẩm.

9) Sửa thông tin cá nhân

Mô tả	Cho phép người dùng cập nhật các thông tin cá nhân trong tài khoản của mình.
Tác nhân chính	Người Dùng

Luồng chính	<ol style="list-style-type: none"> 1. Người Dùng truy cập trang "Thông tin cá nhân" hoặc "Quản lý tài khoản". 2. Hệ thống hiển thị form với thông tin cá nhân hiện tại. 3. Người Dùng chỉnh sửa các trường thông tin mong muốn (tên, địa chỉ, số điện thoại...). 4. Người Dùng nhấn nút "Lưu thay đổi". 5. Hệ thống kiểm tra tính hợp lệ của dữ liệu. 6. Hệ thống cập nhật thông tin vào cơ sở dữ liệu. 7. Hệ thống thông báo "Cập nhật thành công".
Luồng ngoại lệ	<p>LP1: Dữ liệu không hợp lệ:</p> <ol style="list-style-type: none"> a. Hệ thống thông báo lỗi tương ứng. b. Người Dùng sửa lỗi và thử lại.

10) Tạo bài đánh giá

Mô tả	Cho phép người dùng để lại nhận xét và đánh giá sao cho sản phẩm đã mua.
Tác nhân chính	Người Dùng
Luồng chính	<ol style="list-style-type: none"> 1. Người Dùng truy cập trang chi tiết sản phẩm đã mua hoặc mục "Lịch sử đơn hàng" và chọn sản phẩm để đánh giá. 2. Hệ thống hiển thị form đánh giá (cho phép chọn số sao và nhập bình luận). 3. Người Dùng nhập điểm đánh giá và/hoặc bình luận. 4. Người Dùng nhấn nút "Gửi đánh giá". 5. Hệ thống kiểm tra tính hợp lệ của đánh giá (ví dụ: không để trống). 6. Hệ thống lưu đánh giá vào cơ sở dữ liệu và cập nhật điểm đánh giá trung bình của sản phẩm. 7. Hệ thống thông báo "Đánh giá của bạn đã được gửi".
Luồng ngoại lệ	<p>LP1: Đánh giá không hợp lệ:</p> <ol style="list-style-type: none"> a. Hệ thống thông báo lỗi nếu đánh giá trống hoặc không đúng định dạng. b. Người Dùng sửa và gửi lại. <p>LP2: Đã đánh giá sản phẩm này:</p>

	a. Hệ thống thông báo "Bạn đã đánh giá sản phẩm này" và cho phép chỉnh sửa đánh giá cũ thay vì tạo mới.
--	---

11) Liên hệ

Mô tả	Cung cấp kênh để người dùng gửi câu hỏi, góp ý, hoặc yêu cầu hỗ trợ đến quản trị viên.
Tác nhân chính	Người Dùng
Luồng chính	<ol style="list-style-type: none"> 1. Người Dùng truy cập trang "Liên hệ". 2. Hệ thống hiển thị form liên hệ (họ tên, email, số điện thoại, chủ đề, nội dung). 3. Người Dùng điền thông tin và nội dung tin nhắn. 4. Người Dùng nhấn nút "Gửi tin nhắn". 5. Hệ thống kiểm tra dữ liệu và gửi tin nhắn về hệ thống quản trị. 6. Hệ thống thông báo "Tin nhắn của bạn đã được gửi thành công".
Luồng ngoại lệ	LP1: Thông tin không hợp lệ: <ol style="list-style-type: none"> a. Hệ thống báo lỗi nếu thông tin liên hệ không đầy đủ hoặc sai định dạng. b. Người Dùng sửa lỗi và gửi lại.

12) Thanh toán

Mô tả	Cho phép người dùng hoàn tất quá trình mua sắm bằng cách thanh toán cho đơn hàng trong giỏ hàng.
Tác nhân chính	Người Dùng
Luồng chính	<ol style="list-style-type: none"> 1. Người Dùng đã có sản phẩm trong giỏ hàng và nhấn "Tiến hành thanh toán" từ trang giỏ hàng. 2. Hệ thống hiển thị trang Thanh toán (có thể theo bước). 3. Người Dùng xác nhận địa chỉ giao hàng và thời gian giao hàng. 4. Người Dùng chọn phương thức thanh toán (COD, chuyển khoản, ví điện tử, thẻ tín dụng).

	<p>5. Nếu chọn thanh toán trực tuyến, Người Dùng cung cấp thông tin thanh toán (ví dụ: chi tiết thẻ).</p> <p>6. Người Dùng xem lại tóm tắt đơn hàng và xác nhận.</p> <p>7. Hệ thống gửi yêu cầu thanh toán đến Hệ thống Thanh toán (bên ngoài).</p> <p>8. Hệ thống Thanh toán xử lý giao dịch.</p> <p>9. Hệ thống nhận kết quả thanh toán.</p> <p>10. Nếu thành công, Hệ thống tạo hóa đơn, lưu thông tin vào CSDL và gửi xác nhận đơn hàng/hóa đơn cho Người Dùng.</p> <p>11. Hệ thống chuyển hướng Người Dùng đến trang "Đặt hàng thành công".</p>
Luồng ngoại lệ	<p>LP1: Thanh toán không thành công:</p> <p>a. Hệ thống Thanh toán báo lỗi (ví dụ: thẻ không đủ tiền, lỗi kỹ thuật).</p> <p>b. Hệ thống thông báo "Thanh toán thất bại" và yêu cầu Người Dùng thử lại hoặc chọn phương thức khác.</p> <p>LP2: Thông tin giao hàng/Thanh toán không hợp lệ:</p> <p>a. Hệ thống báo lỗi và yêu cầu Người Dùng sửa thông tin.</p> <p>LP3: Hệ thống Thanh toán không khả dụng:</p> <p>a. Hệ thống thông báo lỗi kỹ thuật và khuyến nghị thử lại sau hoặc chọn phương thức khác.</p>

13) Hủy hóa đơn

Mô tả	Cho phép người dùng hủy một đơn hàng đã đặt trước khi nó được xử lý hoặc giao hàng.
Tác nhân chính	Người Dùng
Luồng chính	<p>1. Người Dùng truy cập trang "Lịch sử đơn hàng".</p> <p>2. Người Dùng chọn đơn hàng muốn hủy (chỉ áp dụng cho đơn hàng ở trạng thái cho phép hủy).</p> <p>3. Người Dùng nhấn nút "Hủy đơn hàng".</p> <p>4. Hệ thống yêu cầu xác nhận hủy.</p>

	<p>5. Người Dùng xác nhận hủy.</p> <p>6. Hệ thống kiểm tra trạng thái đơn hàng.</p> <p>7. Nếu đơn hàng đủ điều kiện hủy, Hệ thống cập nhật trạng thái đơn hàng thành "Đã hủy" trong CSDL.</p> <p>8. Hệ thống thông báo "Đơn hàng đã được hủy thành công".</p>
Luồng ngoại lệ	<p>LP1: Đơn hàng không đủ điều kiện hủy:</p> <p>a. Hệ thống thông báo "Đơn hàng này không thể hủy do đã được xử lý/đang giao."</p> <p>b. Hướng dẫn Người Dùng liên hệ bộ phận hỗ trợ nếu vẫn muốn hủy.</p>

2.5.2 Use-case Quản lý



Ảnh 3. Sơ đồ Use-case quản lý

-Đặc tả use-case quản lý

1) Đăng kí

Mô tả	Cho phép người dùng mới tạo một tài khoản trên hệ thống để có thể thực hiện các giao dịch và quản lý thông tin cá nhân.
Tác nhân chính	Người Dùng
Luồng chính	1. Người Dùng truy cập trang Đăng ký.

	<p>2. Hệ thống hiển thị form Đăng ký.</p> <p>3. Người Dùng nhập thông tin cần thiết (tên đăng nhập, email, mật khẩu, xác nhận mật khẩu, v.v.).</p> <p>4. Người Dùng nhấn nút "Đăng ký".</p> <p>5. Hệ thống kiểm tra tính hợp lệ của dữ liệu.</p> <p>6. Hệ thống lưu thông tin tài khoản mới vào cơ sở dữ liệu.</p> <p>7. Hệ thống thông báo Đăng ký thành</p> <p>8. Hệ thống tự động đăng nhập người dùng hoặc chuyển hướng đến trang Đăng nhập.</p>
Luồng ngoại lệ	<p>LP1: Thông tin không hợp lệ: Hệ thống phát hiện dữ liệu thiếu hoặc sai định dạng. Hệ thống hiển thị thông báo lỗi tương ứng bên cạnh trường nhập liệu. Người Dùng sửa lỗi và nhấn "Đăng ký".</p> <p>LP2: Tên đăng nhập/Email đã tồn tại:</p> <ol style="list-style-type: none"> Hệ thống kiểm tra và phát hiện tên đăng nhập hoặc email đã có trong hệ thống. Hệ thống hiển thị thông báo lỗi rõ ràng. Người Dùng nhập thông tin khác và nhấn "Đăng ký".

2) Đăng nhập

Mô tả	Cho phép người dùng đã có tài khoản truy cập vào các chức năng yêu cầu xác thực của hệ thống.
Tác nhân chính	Người Dùng
Luồng chính	<p>1. Người Dùng truy cập trang Đăng nhập.</p> <p>2. Hệ thống hiển thị form Đăng nhập.</p> <p>3. Người Dùng nhập tên đăng nhập/email và mật khẩu.</p> <p>4. Người Dùng nhấn nút "Đăng nhập".</p> <p>5. Hệ thống xác thực thông tin đăng nhập.</p> <p>6. Nếu thông tin đúng, Hệ thống cấp quyền truy cập và chuyển hướng Người Dùng đến trang chính</p>
Luồng ngoại lệ	<p>LP1: Sai thông tin đăng nhập: Hệ thống thông báo "Tên đăng nhập hoặc mật khẩu không đúng".</p>

	<p>b. Người Dùng có thể thử lại hoặc chọn "Quên mật khẩu".</p> <p>LP2: Tài khoản bị khóa:</p> <p>a. Hệ thống thông báo tài khoản đã bị khóa (do quá nhiều lần thử sai hoặc bị Admin khóa).</p> <p>b. Người Dùng được hướng dẫn liên hệ hỗ trợ hoặc chờ thời gian mở khóa.</p> <p>LP3: Chưa xác thực Email/Số điện thoại (nếu có yêu cầu):</p> <p>a. Hệ thống thông báo yêu cầu xác thực email/số điện thoại.</p> <p>b. Hướng dẫn người dùng thực hiện xác thực.</p>
--	---

3) Đăng xuất

Mô tả	Cho phép người dùng kết thúc phiên làm việc an toàn trên hệ thống.
Tác nhân chính	Người Dùng
Luồng chính	<ol style="list-style-type: none"> 1. Người Dùng đang ở trạng thái đã đăng nhập. 2. Người Dùng nhấn vào nút "Đăng xuất". 3. Hệ thống xóa phiên làm việc của người dùng. 4. Hệ thống chuyển hướng Người Dùng về trang Đăng nhập hoặc Trang Chủ (ở trạng thái chưa đăng nhập).
Luồng ngoại lệ	Không có luồng phụ đáng kể, đây là một hành động đơn giản.

4) Tìm kiếm

Mô tả	Cho phép người dùng tìm kiếm sản phẩm hoặc thông tin cụ thể trên website.
Tác nhân chính	Người Dùng
Luồng chính	<ol style="list-style-type: none"> 1. Người Dùng nhập từ khóa vào ô tìm kiếm.

	<p>2. Người Dùng nhấn Enter hoặc biểu tượng tìm kiếm.</p> <p>3. Hệ thống thực hiện tìm kiếm trong cơ sở dữ liệu sản phẩm, blog, hoặc danh mục.</p> <p>4. Hệ thống hiển thị trang kết quả tìm kiếm với danh sách các sản phẩm/bài viết/thông tin liên quan.</p>
Luồng ngoại lệ	<p>LP1: Không tìm thấy kết quả:</p> <p>a. Hệ thống thông báo "Không tìm thấy kết quả phù hợp với từ khóa của bạn."</p> <p>b. Gợi ý các từ khóa liên quan hoặc sản phẩm phổ biến.</p>

5) Xem chi tiết sản phẩm

Mô tả	Cho phép người dùng xem thông tin đầy đủ về một sản phẩm cụ thể.
Tác nhân chính	Người Dùng
Luồng chính	<p>1. Người Dùng nhấp vào tên hoặc hình ảnh của một sản phẩm từ trang danh sách hoặc kết quả tìm kiếm.</p> <p>2. Hệ thống tải và hiển thị trang chi tiết sản phẩm, bao gồm: tên, giá, mô tả, hình ảnh lớn, thông tin nhà cung cấp, đánh giá trung bình, và các thông tin liên quan khác.</p>
Luồng ngoại lệ	<p>LP1: Sản phẩm không tồn tại:</p> <p>a. Hệ thống thông báo "Sản phẩm không tìm thấy" hoặc "Sản phẩm không còn kinh doanh".</p> <p>b. Chuyển hướng người dùng về trang danh sách sản phẩm hoặc trang chủ.</p>

6) Thêm vào giỏ hàng

Mô tả	Cho phép người dùng thêm một sản phẩm vào giỏ hàng để chuẩn bị mua sắm.
Tác nhân chính	Người Dùng
Luồng chính	<p>1. Người Dùng đang xem trang chi tiết sản phẩm hoặc danh sách sản phẩm.</p> <p>2. Người Dùng chọn số lượng sản phẩm</p> <p>3. Người Dùng nhấn nút "Thêm vào giỏ hàng".</p>

	<p>4. Hệ thống kiểm tra số lượng tồn kho và thêm sản phẩm vào giỏ hàng của người dùng.</p> <p>5. Hệ thống hiển thị thông báo "Đã thêm vào giỏ hàng" hoặc cập nhật biểu tượng giỏ hàng.</p>
Luồng ngoại lệ	<p>LP1: Sản phẩm hết hàng/không đủ số lượng:</p> <p>a. Hệ thống thông báo "Sản phẩm đã hết hàng" hoặc "Số lượng yêu cầu vượt quá tồn kho".</p> <p>b. Không thêm sản phẩm vào giỏ.</p>

7) Yêu thích sản phẩm

Mô tả	Cho phép người dùng đánh dấu các sản phẩm quan tâm để dễ dàng xem lại sau này.
Tác nhân chính	Người Dùng
Luồng chính	<p>1. Người Dùng đang xem trang chi tiết sản phẩm hoặc danh sách sản phẩm.</p> <p>2. Người Dùng nhấn vào biểu tượng "Yêu thích"</p> <p>3. Hệ thống lưu sản phẩm này vào danh sách yêu thích của người dùng.</p> <p>4. Hệ thống thay đổi biểu tượng để thể hiện sản phẩm đã được yêu thích.</p>
Luồng ngoại lệ	<p>LP1: Sản phẩm đã có trong danh sách yêu thích:</p> <p>a. Hệ thống thông báo hoặc không làm gì, giữ nguyên trạng thái.</p>

8) Xem danh sách yêu thích

Mô tả	Cho phép người dùng xem lại các sản phẩm mà họ đã đánh dấu yêu thích.
Tác nhân chính	Người Dùng
Luồng chính	<p>1. Người Dùng truy cập trang "Danh sách yêu thích" từ menu hoặc profile cá nhân.</p> <p>2. Hệ thống tải và hiển thị danh sách các sản phẩm đã được người dùng yêu thích.</p> <p>3. Người Dùng có thể nhấp vào từng sản phẩm.</p>

	phẩm để xem chi tiết hoặc xóa khỏi danh sách.
Luồng ngoại lệ	LP1: Danh sách yêu thích trống: a. Hệ thống hiển thị thông báo "Chưa có sản phẩm nào trong danh sách yêu thích của bạn." b. Gợi ý khách hàng duyệt sản phẩm.

9) Sửa thông tin cá nhân

Mô tả	Cho phép người dùng cập nhật các thông tin cá nhân trong tài khoản của mình.
Tác nhân chính	Người Dùng
Luồng chính	1. Người Dùng truy cập trang "Thông tin cá nhân" hoặc "Quản lý tài khoản". 2. Hệ thống hiển thị form với thông tin cá nhân hiện tại. 3. Người Dùng chỉnh sửa các trường thông tin mong muốn (tên, địa chỉ, số điện thoại...) 4. Người Dùng nhấn nút "Lưu thay đổi". 5. Hệ thống kiểm tra tính hợp lệ của dữ liệu. 6. Hệ thống cập nhật thông tin vào cơ sở dữ liệu. 7. Hệ thống thông báo "Cập nhật thành công".
Luồng ngoại lệ	LP1: Dữ liệu không hợp lệ: a. Hệ thống thông báo lỗi tương ứng. b. Người Dùng sửa lỗi và thử lại.

10) Liên hệ

Mô tả	Cung cấp kênh để người dùng gửi câu hỏi, góp ý, hoặc yêu cầu hỗ trợ đến quản trị viên.
Tác nhân chính	Người Dùng
Luồng chính	1. Người Dùng truy cập trang "Liên hệ". 2. Hệ thống hiển thị form liên hệ (họ tên, email, số điện thoại, chủ đề, nội dung).

	<p>3. Người Dùng điền thông tin và nội dung tin nhắn.</p> <p>4. Người Dùng nhấn nút "Gửi tin nhắn".</p> <p>5. Hệ thống kiểm tra dữ liệu và gửi tin nhắn về hệ thống quản trị.</p> <p>6. Hệ thống thông báo "Tin nhắn của bạn đã được gửi thành công".</p>
Luồng ngoại lệ	<p>LP1: Thông tin không hợp lệ:</p> <p>a. Hệ thống báo lỗi nếu thông tin liên hệ không đầy đủ hoặc sai định dạng.</p> <p>b. Người Dùng sửa lỗi và gửi lại.</p>

11) Thanh toán

Mô tả	Cho phép người dùng hoàn tất quá trình mua sắm bằng cách thanh toán cho đơn hàng trong giỏ hàng.
Tác nhân chính	Người Dùng
Luồng chính	<p>1. Người Dùng đã có sản phẩm trong giỏ hàng và nhấn "Tiến hành thanh toán" từ trang giỏ hàng.</p> <p>2. Hệ thống hiển thị trang Thanh toán (có thể theo bước).</p> <p>3. Người Dùng xác nhận địa chỉ giao hàng và thời gian giao hàng.</p> <p>4. Người Dùng chọn phương thức thanh toán (COD, chuyển khoản, ví điện tử, thẻ tín dụng).</p> <p>5. Nếu chọn thanh toán trực tuyến, Người Dùng cung cấp thông tin thanh toán (ví dụ: chi tiết thẻ).</p> <p>6. Người Dùng xem lại tóm tắt đơn hàng và xác nhận.</p> <p>7. Hệ thống gửi yêu cầu thanh toán đến Hệ thống Thanh toán (bên ngoài).</p> <p>8. Hệ thống Thanh toán xử lý giao dịch.</p> <p>9. Hệ thống nhận kết quả thanh toán.</p>

	<p>10. Nếu thành công, Hệ thống tạo hóa đơn, lưu thông tin vào CSDL và gửi xác nhận đơn hàng/hóa đơn cho Người Dùng.</p> <p>11. Hệ thống chuyển hướng Người Dùng đến trang "Đặt hàng thành công".</p>
Luồng ngoại lệ	<p>LP1: Thanh toán không thành công:</p> <p>a. Hệ thống Thanh toán báo lỗi (ví dụ: thẻ không đủ tiền, lỗi kỹ thuật).</p> <p>b. Hệ thống thông báo "Thanh toán thất bại" và yêu cầu Người Dùng thử lại hoặc chọn phương thức khác.</p> <p>LP2: Thông tin giao hàng/Thanh toán không hợp lệ:</p> <p>a. Hệ thống báo lỗi và yêu cầu Người Dùng sửa thông tin.</p> <p>LP3: Hệ thống Thanh toán không khả dụng:</p> <p>a. Hệ thống thông báo lỗi kỹ thuật và khuyến nghị thử lại sau hoặc chọn phương thức khác.</p>

12) Quản lý sản phẩm

Mô tả	Cho phép Quản lý thực hiện các thao tác thêm, sửa, xóa và xem thông tin sản phẩm trên hệ thống.
Tác nhân chính	Quản lý
Luồng chính	<p>1. Quản lý đăng nhập vào trang quản trị.</p> <p>2. Quản lý truy cập mục "Quản lý sản phẩm".</p> <p>3. Hệ thống hiển thị danh sách các sản phẩm hiện có.</p> <p>4. Quản lý có thể chọn:</p> <p>a. Thêm sản phẩm mới: Nhập thông tin (tên, giá, mô tả, hình ảnh, danh mục, số lượng tồn kho), nhấn "Lưu".</p> <p>b. Sửa sản phẩm: Chọn một sản phẩm, chỉnh sửa thông tin, nhấn "Cập nhật".</p> <p>c. Xóa sản phẩm: Chọn một hoặc nhiều sản phẩm, xác nhận xóa.</p> <p>5. Hệ thống xử lý yêu cầu và cập nhật dữ liệu.</p>

	6. Hệ thống thông báo kết quả thao tác.
Luồng ngoại lệ	<p>LP1: Thiếu/Sai thông tin khi thêm/sửa:</p> <p>a. Hệ thống báo lỗi các trường thông tin không hợp lệ hoặc bị thiếu.</p> <p>b. Quản lý sửa lỗi và thử lại.</p> <p>LP2: Lỗi hệ thống khi thao tác:</p> <p>a. Hệ thống hiển thị thông báo lỗi kỹ thuật.</p> <p>b. Quản lý kiểm tra lại hoặc liên hệ bộ phận kỹ thuật.</p> <p>LP3: Sản phẩm không tồn tại (khi sửa/xóa):</p> <p>a. Hệ thống thông báo sản phẩm không tìm thấy.</p>

13) Quản lý bài viết

Mô tả	Cho phép Quản lý thêm, sửa, xóa các bài viết (blog, tin tức) trên website.
Tác nhân chính	Quản lý
Luồng chính	<p>1. Quản lý đăng nhập vào trang quản trị.</p> <p>2. Quản lý truy cập mục "Quản lý bài viết".</p> <p>3. Hệ thống hiển thị danh sách các bài viết.</p> <p>4. Quản lý có thể chọn:</p> <p>a. Thêm bài viết mới: Nhập tiêu đề, nội dung, hình ảnh đại diện, nhấn "Đăng".</p> <p>b. Sửa bài viết: Chọn một bài viết, chỉnh sửa nội dung, nhấn "Cập nhật".</p> <p>c. Xóa bài viết: Chọn một hoặc nhiều bài viết, xác nhận xóa.</p> <p>5. Hệ thống xử lý yêu cầu và cập nhật dữ liệu.</p> <p>6. Hệ thống thông báo kết quả thao tác.</p>
Luồng ngoại lệ	<p>LP1: Thiếu/Sai thông tin khi thêm/sửa:</p> <p>a. Hệ thống báo lỗi các trường thông tin không hợp lệ hoặc bị thiếu.</p> <p>b. Quản lý sửa lỗi và thử lại.</p> <p>LP2: Lỗi hệ thống khi thao tác:</p> <p>a. Hệ thống hiển thị thông báo lỗi kỹ thuật.</p> <p>b. Quản lý kiểm tra lại hoặc liên hệ bộ phận kỹ thuật.</p>

14) Quản lý vai trò

Mô tả	Cho phép Quản lý quản lý các vai trò (roles) trong hệ thống và gán/hủy gán vai trò cho người dùng.
Tác nhân chính	Quản lý
Luồng chính	<ol style="list-style-type: none"> 1. Quản lý đăng nhập vào trang quản trị. 2. Quản lý truy cập mục "Quản lý vai trò". 3. Hệ thống hiển thị danh sách các vai trò hiện có. 4. Quản lý có thể: <ol style="list-style-type: none"> a. Thêm vai trò mới: Nhập tên vai trò. b. Sửa tên vai trò: Chọn vai trò cần sửa. c. Xóa vai trò: Chọn vai trò cần xóa (kiểm tra không có người dùng nào đang giữ vai trò này). d. Gán/Hủy gán vai trò cho người dùng: Chọn người dùng, chọn vai trò cần gán/hủy. 5. Hệ thống xử lý yêu cầu và cập nhật dữ liệu. 6. Hệ thống thông báo kết quả thao tác.
Luồng ngoại lệ	<p>LP1: Tên vai trò đã tồn tại (khi thêm):</p> <ol style="list-style-type: none"> a. Hệ thống báo lỗi tên vai trò bị trùng. b. Quản lý nhập tên khác. <p>LP2: Vai trò đang được sử dụng (khi xóa):</p> <ol style="list-style-type: none"> a. Hệ thống thông báo không thể xóa vai trò đang được gán cho người dùng. b. Quản lý phải hủy gán vai trò đó khỏi tất cả người dùng trước khi xóa.

15) Quản lý thư viện ảnh

Mô tả	Cho phép Quản lý tải lên, sắp xếp và xóa các hình ảnh trong thư viện, phục vụ cho sản phẩm, bài viết, v.v.
-------	--

Tác nhân chính	Quản lý
Luồng chính	<ol style="list-style-type: none"> 1. Quản lý đăng nhập vào trang quản trị. 2. Quản lý truy cập mục "Quản lý thư viện ảnh". 3. Hệ thống hiển thị danh sách các hình ảnh đã tải lên. 4. Quản lý có thể: <ol style="list-style-type: none"> a. Tải lên hình ảnh mới: Chọn file hình ảnh từ máy tính. b. Xóa hình ảnh: Chọn hình ảnh và xác nhận xóa. c. Sắp xếp/Phân loại hình ảnh: Di chuyển hình ảnh vào các thư mục hoặc gắn tag. 5. Hệ thống xử lý yêu cầu (tải lên/xóa/sắp xếp). 6. Hệ thống thông báo kết quả thao tác.
Luồng ngoại lệ	<p>LP1: Lỗi khi tải lên file:</p> <ol style="list-style-type: none"> a. Hệ thống báo lỗi file không hợp lệ (định dạng, kích thước) hoặc lỗi đường truyền. b. Quản lý kiểm tra file và thử lại. <p>LP2: Hình ảnh đang được sử dụng:</p> <ol style="list-style-type: none"> a. Hệ thống cảnh báo hoặc ngăn chặn việc xóa hình ảnh đang được liên kết với sản phẩm/bài viết. b. Quản lý phải hủy liên kết trước khi xóa.

16) Xác nhận hóa đơn

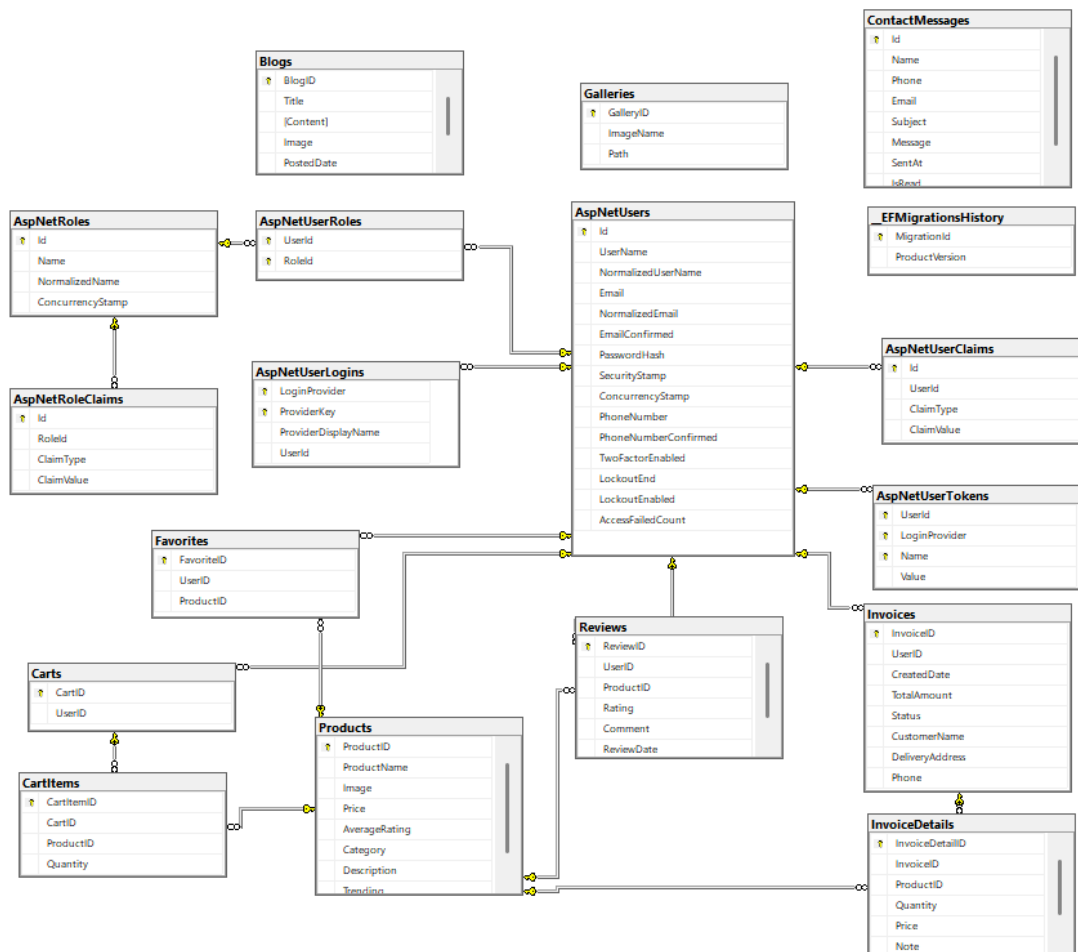
Mô tả	Cho phép Quản lý xem và cập nhật trạng thái của các đơn hàng/hóa đơn từ khách hàng.
Tác nhân chính	Quản lý
Luồng chính	<ol style="list-style-type: none"> 1. Quản lý đăng nhập vào trang quản trị. 2. Quản lý truy cập mục "Quản lý đơn hàng" hoặc "Hóa đơn". 3. Hệ thống hiển thị danh sách các hóa đơn, thường sắp xếp theo trạng thái hoặc thời gian. 4. Quản lý xem chi tiết một hóa đơn (thông tin khách hàng, sản phẩm, tổng tiền).

	<p>5. Quản lý cập nhật trạng thái hóa đơn (ví dụ: từ "Chờ xác nhận" sang "Đang xử lý", "Đang giao", "Đã giao", "Đã hủy").</p> <p>6. Hệ thống lưu trạng thái mới và có thể gửi thông báo cho khách hàng.</p> <p>7. Hệ thống thông báo cập nhật thành công.</p>
Luồng ngoại lệ	<p>LP1: Hóa đơn không tồn tại:</p> <p>a. Hệ thống thông báo không tìm thấy hóa đơn.</p> <p>LP2: Lỗi cập nhật trạng thái:</p> <p>a. Hệ thống báo lỗi nếu có vấn đề kỹ thuật hoặc trạng thái chuyển đổi không hợp lệ.</p> <p>b. Quản lý thử lại hoặc kiểm tra hệ thống.</p>

17) Thống kê & Báo cáo

Mô tả	Cung cấp cho Quản lý các báo cáo về doanh thu, số lượng đơn hàng, sản phẩm bán chạy để phân tích kinh doanh.
Tác nhân chính	Quản lý
Luồng chính	<p>1. Quản lý đăng nhập vào trang quản trị.</p> <p>2. Quản lý truy cập mục "Thống kê & Báo cáo".</p> <p>3. Hệ thống hiển thị các loại báo cáo có sẵn (ví dụ: báo cáo doanh thu theo ngày/tháng/năm, sản phẩm bán chạy nhất, số lượng khách hàng mới).</p> <p>4. Quản lý chọn loại báo cáo và thiết lập các bộ lọc (thời gian, danh mục).</p> <p>5. Hệ thống truy vấn dữ liệu và hiển thị báo cáo dưới dạng biểu đồ hoặc bảng.</p> <p>6. Quản lý có thể xuất báo cáo ra file (ví dụ: Excel, PDF).</p>
Luồng ngoại lệ	<p>LP1: Dữ liệu báo cáo không có:</p> <p>a. Hệ thống thông báo không có dữ liệu cho khoảng thời gian hoặc bộ lọc đã chọn.</p> <p>LP2: Lỗi hệ thống khi tạo báo cáo:</p> <p>a. Hệ thống báo lỗi kỹ thuật.</p> <p>b. Quản lý thử lại hoặc báo cáo kỹ thuật.</p>

2.6 Thiết kế cơ sở dữ liệu



Ảnh 4. Sơ đồ cơ sở dữ liệu

-Bảng AspNetUsers

Tên Cột	Kiểu Dữ liệu	Mô tả
Id	NVARCHAR(450)	Khóa chính (PK) duy nhất cho mỗi người dùng. Thường là một chuỗi GUID.

UserName	NVARCHAR(256)	Tên đăng nhập của người dùng. Phải là duy nhất.
NormalizedUserName	NVARCHAR(256)	Phiên bản chuẩn hóa của tên đăng nhập, dùng để tìm kiếm không phân biệt chữ hoa/chữ thường.
Email	NVARCHAR(256)	Địa chỉ email của người dùng.
NormalizedEmail	NVARCHAR(256)	Phiên bản chuẩn hóa của email.
EmailConfirmed	BIT	Cờ (flag) cho biết email của người dùng đã được xác nhận hay chưa (True/False).
PasswordHash	NVARCHAR(MAX)	Chuỗi băm (hash) của mật khẩu người dùng. Mật khẩu gốc không bao giờ được lưu trữ.
SecurityStamp	NVARCHAR(MAX)	Một chuỗi ngẫu nhiên thay đổi mỗi khi người dùng thay đổi mật khẩu hoặc thông tin nhạy cảm, dùng để vô hiệu hóa token cũ.
ConcurrencyStamp	NVARCHAR(MAX)	Một chuỗi được sử dụng cho mục đích kiểm soát đồng thời khi cập nhật bản ghi.
PhoneNumber	NVARCHAR(MAX)	Số điện thoại liên hệ của người dùng.
PhoneNumberConfirmed	BIT	Cờ cho biết số điện thoại đã được xác nhận hay chưa.
TwoFactorEnabled	BIT	Cờ cho biết tính năng xác thực hai yếu tố có được bật cho tài khoản này hay không.
LockoutEnd	DATETIMEOFFSET	Thời điểm kết thúc khóa tài khoản. Nếu tài khoản bị khóa do quá nhiều lần đăng nhập sai, cột này sẽ chứa thời gian tài khoản được mở khóa lại.

LockoutEnabled	BIT	Cờ cho biết có áp dụng khóa tài khoản khi đăng nhập sai hay không.
AccessFailedCount	INT	Số lần đăng nhập sai liên tiếp.

-BảngAspNetRoles

Tên Cột	Kiểu Dữ liệu	Mô tả
Id	NVARCHAR(450)	Khóa chính (PK) duy nhất cho mỗi vai trò.
Name	NVARCHAR(256)	Tên của vai trò (ví dụ: "Admin", "Customer", "Shipper"). Phải là duy nhất.
NormalizedName	NVARCHAR(256)	Phiên bản chuẩn hóa của tên vai trò.
ConcurrencyStamp	NVARCHAR(MAX)	Chuỗi dùng cho kiểm soát đồng thời khi cập nhật vai trò.

-BảngAspNetUserRoles

Tên Cột	Kiểu Dữ liệu	Mô tả
UserId	NVARCHAR(450)	Khóa ngoại (FK) tham chiếu đến Id của bảng AspNetUsers.
RoleId	NVARCHAR(450)	Khóa ngoại (FK) tham chiếu đến Id của bảng AspNetRoles.
(Khóa chính)	(N/A)	Kết hợp của UserId và RoleId tạo thành khóa chính (PK), đảm bảo mỗi cặp người dùng-vai trò là duy nhất.

-BảngAspNetUserClaims

Tên Cột	Kiểu Dữ liệu	Mô tả
Id	INT	Khóa chính (PK) duy nhất cho mỗi yêu cầu.
UserId	NVARCHAR(450)	Khóa ngoại (FK) tham chiếu đến Id của bảngAspNetUsers.
ClaimType	NVARCHAR(MAX)	Loại yêu cầu (ví dụ: "Address", "DateOfBirth", "Permission.EditProduct").
ClaimValue	NVARCHAR(MAX)	Giá trị của yêu cầu (ví dụ: "123 Main St", "1990-01-01", "true").

-Bảng AspNetUserLogins

Tên Cột	Kiểu Dữ liệu	Mô tả
LoginProvider	NVARCHAR(450)	Tên của nhà cung cấp đăng nhập bên ngoài (ví dụ: "Google", "Facebook").
ProviderKey	NVARCHAR(450)	Khóa duy nhất từ nhà cung cấp bên ngoài dùng để xác định người dùng đó trên hệ thống của họ.
ProviderDisplayName	NVARCHAR(MAX)	Tên hiển thị của nhà cung cấp đăng nhập.
UserId	NVARCHAR(450)	Khóa ngoại (FK) tham chiếu đến Id của bảngAspNetUsers.
(Khóa chính)	(N/A)	Kết hợp của LoginProvider và ProviderKey tạo thành khóa chính (PK).

-Bảng AspNetUserTokens

Tên Cột	Kiểu Dữ liệu	Mô tả
---------	--------------	-------

UserId	NVARCHAR(450)	Khóa ngoại (FK) tham chiếu đến Id của bảng AspNetUsers.
LoginProvider	NVARCHAR(450)	Tên của nhà cung cấp token (ví dụ: "Identity", "Google").
Name	NVARCHAR(450)	Tên của token (ví dụ: "AccessToken", "RefreshToken").
Value	NVARCHAR(MAX)	Giá trị của token.
(Khóa chính)	(N/A)	Kết hợp của UserId, LoginProvider, và Name tạo thành khóa chính (PK).

-Bảng Products

Tên Cột	Kiểu Dữ liệu	Mô tả
ProductID	INT	Khóa chính (PK) duy nhất cho mỗi sản phẩm.
ProductName	NVARCHAR(MAX)	Tên đầy đủ của sản phẩm.
Image	NVARCHAR(MAX)	Đường dẫn hoặc tên file của hình ảnh đại diện chính của sản phẩm.
Price	DECIMAL	Giá bán của sản phẩm.
AverageRating	FLOAT	Điểm đánh giá trung bình của sản phẩm (ví dụ: 4.5/5 sao).
Category	NVARCHAR(MAX)	Tên danh mục của sản phẩm
Description	NVARCHAR(MAX)	Mô tả chi tiết về sản phẩm.

-Bảng Carts

Tên Cột	Kiểu Dữ liệu	Mô tả
CartID	INT	Khóa chính (PK) duy nhất cho mỗi giỏ hàng.
UserID	NVARCHAR(450)	Khóa ngoại (FK) tham chiếu đến Id của bảngAspNetUsers, liên kết giỏ hàng với người dùng cụ thể.

-Bảng CartItems

Tên Cột	Kiểu Dữ liệu	Mô tả
CartItemID	INT	Khóa chính (PK) duy nhất cho mỗi mặt hàng trong giỏ hàng.
CartID	INT	Khóa ngoại (FK) tham chiếu đến CartID của bảng Carts, xác định giỏ hàng mà mặt hàng này thuộc về.
ProductID	INT	Khóa ngoại (FK) tham chiếu đến ProductID của bảng Products, xác định sản phẩm có trong mặt hàng này.
Quantity	INT	Số lượng của sản phẩm trong giỏ hàng.

-Bảng Invoices

Tên Cột	Kiểu Dữ liệu	Mô tả
InvoiceID	INT	Khóa chính (PK) duy nhất cho mỗi hóa đơn.
UserID	NVARCHAR(450)	Khóa ngoại (FK) tham chiếu đến Id của bảng AspNetUsers, xác định người dùng đã tạo hóa đơn này.
CreateDate	DATETIME	Thời điểm hóa đơn được tạo.
TotalAmount	DECIMAL	Tổng số tiền của hóa đơn (bao gồm cả thuế, phí vận chuyển, sau giảm giá).
Status	NVARCHAR(MAX)	Trạng thái hiện tại của hóa đơn (ví dụ: "Chờ xác nhận", "Đang xử lý", "Đang giao", "Đã giao", "Đã hủy").
CustomerName	NVARCHAR(MAX)	Tên khách hàng (có thể khác với UserName nếu là khách vắng lai hoặc đặt cho người khác).
DeliveryAddress	NVARCHAR(MAX)	Địa chỉ chi tiết để giao hàng.
Phone	NVARCHAR(MAX)	Số điện thoại liên hệ cho việc giao hàng.

-Bảng InvoiceDetails

Tên Cột	Kiểu Dữ liệu	Mô tả
InvoiceDetailID	INT	Khóa chính (PK) duy nhất cho mỗi chi tiết hóa đơn.
InvoiceID	INT	Khóa ngoại (FK) tham chiếu đến InvoiceID của bảng Invoices, xác định hóa đơn mà chi tiết này thuộc về.
ProductID	INT	Khóa ngoại (FK) tham chiếu đến ProductID của bảng Products, xác định sản phẩm trong chi tiết hóa đơn.

Quantity	INT	Số lượng của sản phẩm đó trong hóa đơn.
Price	DECIMAL	Giá của sản phẩm tại thời điểm đặt hàng. Lưu ý: Giá này có thể khác với giá hiện tại trong bảng Products nếu giá đã thay đổi sau khi đặt hàng.
Note	NVARCHAR(MAX)	Ghi chú thêm cho sản phẩm cụ thể này trong hóa đơn (ví dụ: "ít đường", "không hành").

-Bảng Favorites

Tên Cột	Kiểu Dữ liệu	Mô tả
FavoriteID	INT	Khóa chính (PK) duy nhất cho mỗi mục yêu thích.
UserID	NVARCHAR(450)	Khóa ngoại (FK) tham chiếu đến Id của bảngAspNetUsers, xác định người dùng.
ProductID	INT	Khóa ngoại (FK) tham chiếu đến ProductID của bảng Products, xác định sản phẩm yêu thích.

-Bảng Reviews

ReviewID	INT	Khóa chính (PK) duy nhất cho mỗi đánh giá.
UserID	NVARCHAR(450)	Khóa ngoại (FK) tham chiếu đến Id của bảngAspNetUsers, xác định người dùng đã đánh giá.
ProductID	INT	Khóa ngoại (FK) tham chiếu đến ProductID của bảng Products, xác định sản phẩm được đánh giá.
Rating	INT	Điểm đánh giá (thường từ 1 đến 5 sao).
Comment	NVARCHAR(MAX)	Nội dung bình luận của người dùng.
ReviewDate	DATETIME	Thời điểm đánh giá được gửi.

-Bảng Blogs

Tên Cột	Kiểu Dữ liệu	Mô tả
BlogID	INT	Khóa chính (PK) duy nhất cho mỗi bài blog.
Title	NVARCHAR(MAX)	Tiêu đề của bài blog.
Content	NVARCHAR(MAX)	Nội dung chi tiết của bài blog.
Image	NVARCHAR(MAX)	Đường dẫn hoặc tên file của hình ảnh đại diện cho bài blog.
PostedDate	DATETIME	Thời điểm bài blog được đăng tải.

-Bảng Galleries

Tên Cột	Kiểu Dữ liệu	Mô tả
GalleryID	INT	Khóa chính (PK) duy nhất cho mỗi mục thư viện ảnh.
ImageName	NVARCHAR(MAX)	Tên của tệp hình ảnh.
Path	NVARCHAR(MAX)	Đường dẫn đến tệp hình ảnh trên server hoặc CDN.

-Bảng ContactMessages

Tên Cột	Kiểu Dữ liệu	Mô tả
Id	INT	Khóa chính (PK) duy nhất cho mỗi tin nhắn liên hệ.
Name	NVARCHAR(MAX)	Tên của người gửi tin nhắn.
Phone	NVARCHAR(MAX)	Số điện thoại của người gửi.
Email	NVARCHAR(MAX)	Địa chỉ email của người gửi.
Subject	NVARCHAR(MAX)	Chủ đề của tin nhắn.
Message	NVARCHAR(MAX)	Nội dung chi tiết của tin nhắn.

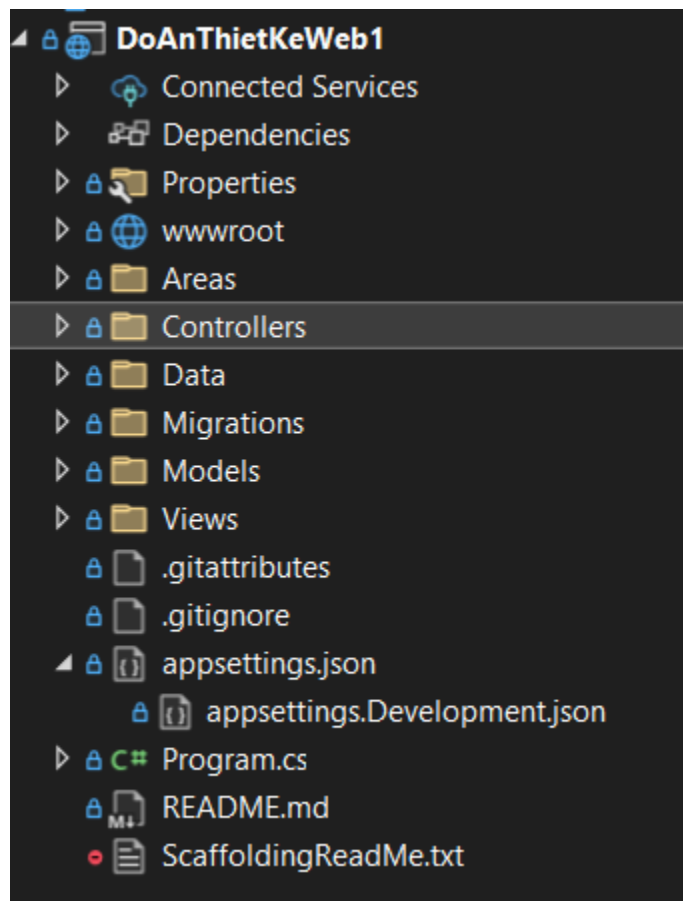
SendAt	DATETIME	Thời điểm tin nhắn được gửi.
--------	----------	------------------------------

-Bảng EFMigrationsHistory

Tên Cột	Kiểu Dữ liệu	Mô tả
MigrationId	NVARCHAR(150)	Mã định danh duy nhất cho mỗi migration (ví dụ: "20230101000000_InitialCreate").
ProductVersion	NVARCHAR(32)	Phiên bản của Entity Framework Core đã tạo ra migration này.

2.7 Thiết kế thành phần MVC

Trong kiến trúc MVC, ba thành phần cốt lõi của MVC được tổ chức rõ ràng để đảm bảo luồng hoạt động mượt mà. Controllers là nơi tiếp nhận mọi yêu cầu từ người dùng, đóng vai trò như "người điều phối" chính. Chúng chịu trách nhiệm giao tiếp với Models, nơi chứa dữ liệu và logic nghiệp vụ, để lấy thông tin cần thiết hoặc thực hiện các thao tác xử lý. Sau khi dữ liệu được Model cung cấp, Controller sẽ lựa chọn Views phù hợp. Các View này, tập trung vào việc trình bày giao diện, sẽ nhận dữ liệu từ Controller và hiển thị chúng cho người dùng một cách trực quan, hoàn thành chu trình xử lý yêu cầu và phản hồi trong kiến trúc MVC.



Ảnh 5. Cấu trúc dự án

2.7.1 Model

Trong kiến trúc MVC, Model có nhiệm vụ đại diện cho dữ liệu sản phẩm và các mối quan hệ nghiệp vụ liên quan. Model được thiết kế để tương tác với cơ sở dữ liệu và cung cấp dữ liệu cho các thành phần Controller và View.

Ví dụ cho Model:

```
37 references
✓ public partial class Product
{
    27 references
    public string ProductId { get; set; } = null!;

    23 references
    public string? ProductName { get; set; }

    16 references
    public string? Image { get; set; }

    15 references
    public int Price { get; set; }

    10 references
    public decimal? AverageRating { get; set; }

    13 references
    public string? Category { get; set; }

    8 references
    public string? Description { get; set; }

    8 references
    public bool Trending { get; set; }

    1 reference
    public virtual ICollection<CartItem> CartItems { get; set; } = new List<CartItem>();

    1 reference
    public virtual ICollection<Favorite> Favorites { get; set; } = new List<Favorite>();

    1 reference
    public virtual ICollection<InvoiceDetail> InvoiceDetails { get; set; } = new List<InvoiceDetail>();

    1 reference
    public virtual ICollection<Review> Reviews { get; set; } = new List<Review>();
}
```

Ảnh 6. Model Product

Thuộc tính cơ bản của Model trên:

ProductId: Định danh duy nhất cho mỗi sản phẩm (khóa chính)

ProductName: Tên sản phẩm hiển thị cho người dùng

Image: Đường dẫn đến hình ảnh đại diện sản phẩm

Price: Giá bán sản phẩm (kiểu số nguyên)

AverageRating: Điểm đánh giá trung bình (kiểu thập phân)

Category: Danh mục phân loại sản phẩm

Description: Thông tin mô tả chi tiết sản phẩm

Trending: Cờ đánh dấu sản phẩm đang hot (true/false)

Quan hệ với các Model khác:

CartItems: Thiết lập quan hệ một-nhiều với giỏ hàng, cho phép theo dõi các sản phẩm được thêm vào giỏ

Favorites: Quản lý danh sách người dùng đã yêu thích sản phẩm

InvoiceDetails: Liên kết với chi tiết hóa đơn để phục vụ chức năng thanh toán

Reviews: Lưu trữ các đánh giá của khách hàng về sản phẩm

Vai trò trong ứng dụng:

Model Product đảm nhiệm ba chức năng chính trong hệ thống. Thứ nhất, nó đóng vai trò trung tâm trong việc quản lý thông tin sản phẩm, bao gồm cả dữ liệu cơ bản và các thuộc tính mở rộng. Thứ hai, model này thiết lập các mối quan hệ nghiệp vụ quan trọng với các thành phần khác như giỏ hàng, đánh giá và hóa đơn. Cuối cùng, nó cung cấp nền tảng dữ liệu cho các thao tác hiển thị danh sách sản phẩm, tìm kiếm, lọc và sắp xếp sản phẩm theo nhiều tiêu chí khác nhau.

2.7.2 View

Trong kiến trúc MVC, View đóng vai trò quan trọng trong việc hiển thị dữ liệu từ Model dưới dạng giao diện người dùng trực quan thông qua các công nghệ web như HTML, CSS và JavaScript. View không chứa bất kỳ logic nghiệp vụ phức tạp nào mà chỉ tập trung vào nhiệm vụ trình bày dữ liệu một cách rõ ràng và thân thiện với người dùng.

View thực hiện các tương tác cơ bản với người dùng như nhận dữ liệu đầu vào từ form, hiển thị các thông báo trạng thái, và điều hướng giữa các trang khác nhau.

Ví dụ cho View:

```
@model IEnumerable<DoAnThietKeWeb1.Models.Gallery>
{
    ViewData["Title"] = "Giới thiệu - Goroco Shop";
}

<div class="heading">
    <h1>Về Chúng Tôi</h1>
    <p> <a asp-action="HomeIndex" asp-controller="Home">Trang Chủ >></a> Giới Thiệu </p>
</div>

<section class="about">
    <div class="image">
        
    </div>
    <div class="content">
        <span>Chào Mừng Đến Với Cửa Hàng Chúng Tôi</span>
        <h3>Thực Phẩm Tươi Sạch Và Hữu Cơ</h3>
        <p>Chúng tôi cung cấp các sản phẩm sạch, an toàn và thân thiện với môi trường. Với tiêu chí "Chất lượng là ưu tiên hàng đầu", cửa hàng luôn đảm bảo nguồn hàng tươi ngon mỗi ngày cho khách hàng.</p>
        <p>Hãy đồng hành cùng chúng tôi để chăm sóc sức khỏe gia đình bạn bằng những sản phẩm hữu cơ chất lượng, giá cả hợp lý và phục vụ tận tâm.</p>
        <a href="#" class="btn">Xem Thêm</a>
    </div>
</section>

<section class="gallery">
    <h1 class="title"> Bộ Sưu Tập <span>Hình Ảnh</span> <a href="#">Xem Tất Cả >></a> </h1>
    <div class="box-container">
        @foreach (var image in Model)
        {
            <div class="box">
                
                <div class="icons">
                    <a href="#" class="fas fa-eye"></a>
                    <a href="#" class="fas fa-heart"></a>
                    <a href="#" class="fas fa-share-alt"></a>
                </div>
            </div>
        }
    </div>
</section>
```

Ảnh 7. View của trang Giới thiệu

Giao diện trên được thiết kế để giới thiệu cửa hàng Goroco. Trang được chia thành hai phần chính:

Phần "Về Chúng Tôi": Đây là khu vực giới thiệu cửa hàng, sử dụng bố cục hai cột hiệu quả. Một bên là hình ảnh minh họa lớn. Bên còn lại là nội dung chữ, nêu bật thông điệp "Thực Phẩm Tươi Sạch Và Hữu Cơ" cùng với cam kết về chất lượng, sự an toàn và lợi ích sức khỏe của sản phẩm. Nút "Xem Thêm" khuyến khích người dùng tìm hiểu sâu hơn, tạo sự tò mò.

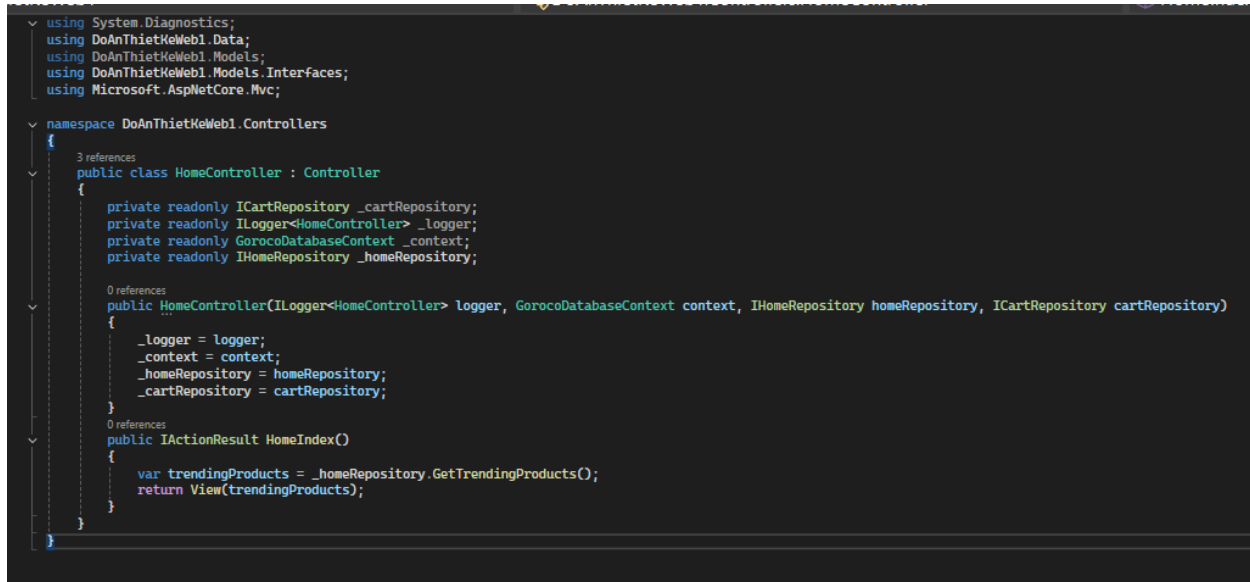
Phần "Bộ Sưu Tập Hình Ảnh": Khu vực này tập trung vào việc trưng bày hình ảnh sản phẩm hoặc hoạt động của cửa hàng. Các ảnh được trình bày dưới dạng lưới gọn gàng, dễ nhìn. Đặc biệt, việc tích hợp các biểu tượng tương tác (như xem chi tiết, yêu thích, chia sẻ) khi di chuột qua mỗi ảnh.

2.7.3 Controller

Trong mô hình kiến trúc MVC, Controller đóng vai trò trung tâm và cực kỳ quan trọng, có thể ví như một "người điều phối" hoặc "bộ não" của ứng dụng web. Nhiệm vụ chính của Controller là tiếp nhận và xử lý các yêu cầu từ người dùng, những yêu cầu này thường

được gửi đến thông qua giao diện người dùng (View) hoặc trực tiếp từ trình duyệt. Khi một yêu cầu đến, Controller sẽ phân tích nó để xác định hành động cần thực hiện.

Ví dụ cho Controller:



```
using System.Diagnostics;
using DoAnThietKeWeb1.Data;
using DoAnThietKeWeb1.Models;
using DoAnThietKeWeb1.Models.Interfaces;
using Microsoft.AspNetCore.Mvc;

namespace DoAnThietKeWeb1.Controllers
{
    public class HomeController : Controller
    {
        private readonly ICartRepository _cartRepository;
        private readonly ILogger<HomeController> _logger;
        private readonly GorocoDatabaseContext _context;
        private readonly IHomeRepository _homeRepository;

        public HomeController(ILogger<HomeController> logger, GorocoDatabaseContext context, IHomeRepository homeRepository, ICartRepository cartRepository)
        {
            _logger = logger;
            _context = context;
            _homeRepository = homeRepository;
            _cartRepository = cartRepository;
        }

        public IActionResult HomeIndex()
        {
            var trendingProducts = _homeRepository.GetTrendingProducts();
            return View(trendingProducts);
        }
    }
}
```

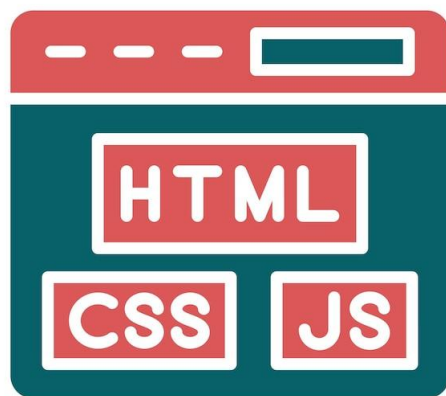
Ảnh 8. Controller của Trang chủ

Chức năng chính của Controller này là khi người dùng truy cập trang chủ, phương thức này sẽ được kích hoạt. Nhiệm vụ của nó là yêu cầu nhà cung cấp dữ liệu trang chủ để lấy về danh sách các sản phẩm đang được ưa chuộng hoặc "hot" nhất. Sau khi nhận được danh sách sản phẩm này, Controller sẽ gửi toàn bộ dữ liệu đó tới giao diện (View) để hiển thị cho người dùng.

2.8 Triển khai và cài đặt

2.8.1 Mô tả môi trường

Trong quá trình thiết kế và triển khai website, em đã xây dựng một môi trường làm việc toàn diện, nơi mỗi thành phần đều đóng góp vào sự vận hành mượt mà và hiệu quả của ứng dụng theo kiến trúc MVC.



Ảnh 9. Ảnh minh họa cho các ngôn ngữ FE

Bắt đầu với giao diện người dùng (frontend), em sử dụng bộ ba công nghệ quen thuộc và mạnh mẽ: HTML để tạo nên cấu trúc xương sống của mọi trang web, định hình nội dung và cách bố cục các phần tử; CSS được dùng để tạo phong cách, màu sắc, phông chữ và đảm bảo tính thẩm mỹ, nhất quán cho toàn bộ giao diện, từ bố cục responsive cho các thiết bị khác nhau đến từng chi tiết nhỏ nhất như nút bấm hay form nhập liệu, mang lại trải nghiệm thị giác hấp dẫn cho người dùng. Song song đó, JavaScript đóng vai trò quan trọng trong việc tạo ra các tương tác động, nâng cao trải nghiệm người dùng bằng cách xử lý các sự kiện phía client, thực hiện các hiệu ứng hình ảnh, kiểm tra dữ liệu đầu vào trước khi gửi lên máy chủ, hoặc tải nội dung động mà không cần tải lại toàn bộ trang, giúp website trở nên sống động và phản hồi nhanh chóng hơn. Sự kết hợp chặt chẽ giữa HTML, CSS và JavaScript chính là nền tảng vững chắc để xây dựng một View đẹp mắt và linh hoạt trong mô hình MVC.



Ảnh 10. Ảnh minh họa cho ASP.Net Core

Tiếp theo, để xử lý các yêu cầu phức tạp từ người dùng và quản lý dữ liệu hiệu quả ở phần backend, em tin tưởng vào sức mạnh của C# chạy trên nền tảng ASP.NET Core. C# được sử dụng để xây dựng các Controller – bộ não của ứng dụng MVC, nơi tiếp nhận mọi yêu cầu từ người dùng, điều phối việc xử lý logic nghiệp vụ và tương tác với dữ liệu. Đồng thời, C# cũng là ngôn ngữ để phát triển Model, đại diện cho các thực thể dữ liệu và các quy tắc nghiệp vụ cốt lõi của website. ASP.NET Core cung cấp một framework mạnh mẽ và linh hoạt, cho phép em xây dựng các API hiệu quả, quản lý phiên làm việc, xác thực và ủy quyền người dùng một cách an toàn, đồng thời tối ưu hóa hiệu năng và khả năng mở rộng của ứng dụng. Môi trường ASP.NET Core không chỉ hỗ trợ phát triển nhanh chóng mà còn giúp ứng dụng có thể triển khai trên nhiều hệ điều hành khác nhau, từ Windows đến Linux, mang lại sự linh hoạt tối đa cho việc vận hành.



Ảnh 11. Ảnh minh họa cho hệ quản trị cơ sở dữ liệu SQL Server

Tất cả dữ liệu quan trọng của website, từ thông tin sản phẩm, đơn hàng đến hồ sơ người dùng, đều được lưu trữ và quản lý trong hệ quản trị cơ sở dữ liệu SQL Server. Việc lựa chọn SQL Server đảm bảo tính toàn vẹn, bảo mật và hiệu năng cao cho dữ liệu, vốn là xương sống của mọi trang thương mại điện tử. C# trong ASP.NET Core sẽ tương tác với SQL Server thông qua các công nghệ như Entity Framework Core, giúp em dễ dàng định nghĩa các Model và thực hiện các thao tác CRUD (Tạo, Đọc, Cập nhật, Xóa) một cách hiệu quả mà không cần viết quá nhiều mã SQL thô. Mỗi liên kết giữa Controller và Model, với Model được ánh xạ trực tiếp tới các bảng trong SQL Server, tạo nên một luồng dữ liệu minh bạch và nhất quán.



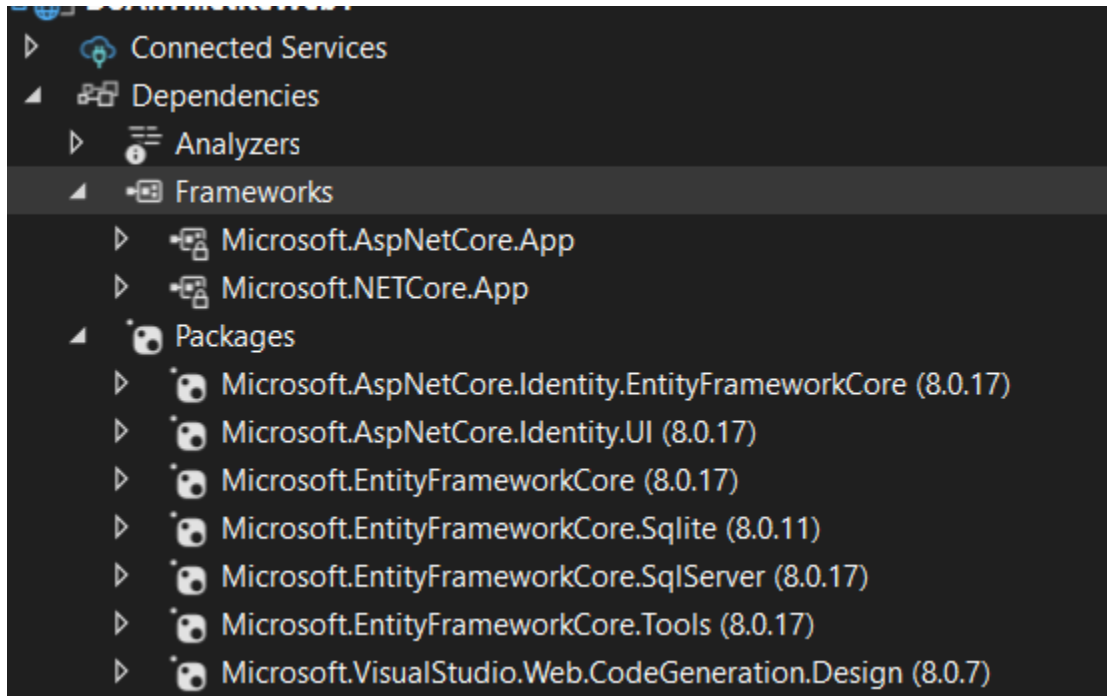
Ảnh 12. Ảnh minh họa cho Git

Cuối cùng, để đảm bảo quá trình phát triển được kiểm soát chặt chẽ và cộng tác hiệu quả, em sử dụng Git kết hợp với GitHub làm hệ thống quản lý phiên bản. Git cho phép em theo dõi mọi thay đổi trong mã nguồn, dễ dàng quay lại các phiên bản trước nếu cần, và đặc biệt là quản lý các nhánh phát triển để làm việc độc lập trên các tính năng mới mà không ảnh hưởng đến phiên bản chính của website. GitHub không chỉ cung cấp một nơi lưu trữ mã nguồn an toàn trên đám mây mà còn tạo ra một nền tảng cộng tác mạnh mẽ, nơi các thành viên trong nhóm có thể xem xét mã, báo cáo lỗi và đóng góp vào dự án một cách có tổ chức, đảm bảo chất lượng và tiến độ của quá trình phát triển website.

2.8.2 Thư viện framework/ thư viện

Trong quá trình xây dựng website, em đã dựa trên nền tảng vững chắc của các framework và thư viện mạnh mẽ. Cụ thể, em sử dụng ASP.NET Core làm khung sườn chính, cung cấp toàn bộ kiến trúc MVC và các dịch vụ cốt lõi để ứng dụng có thể hoạt động hiệu quả. Đi kèm với đó là .NET Core Runtime, môi trường cần thiết để chạy mã lệnh. Để quản lý và tương tác với cơ sở dữ liệu, em tích hợp Entity Framework Core, giúp em dễ dàng làm việc với dữ liệu thông qua các đối tượng C# mà không cần viết trực tiếp SQL. Bên cạnh đó, hệ thống xác thực và quản lý người dùng được xây dựng nhanh chóng nhờ các gói ASP.NET Core Identity, bao gồm cả giao diện người dùng có sẵn. Ngoài ra, các thư viện công cụ như Entity Framework Core Tools và Code Generation Design đã hỗ trợ

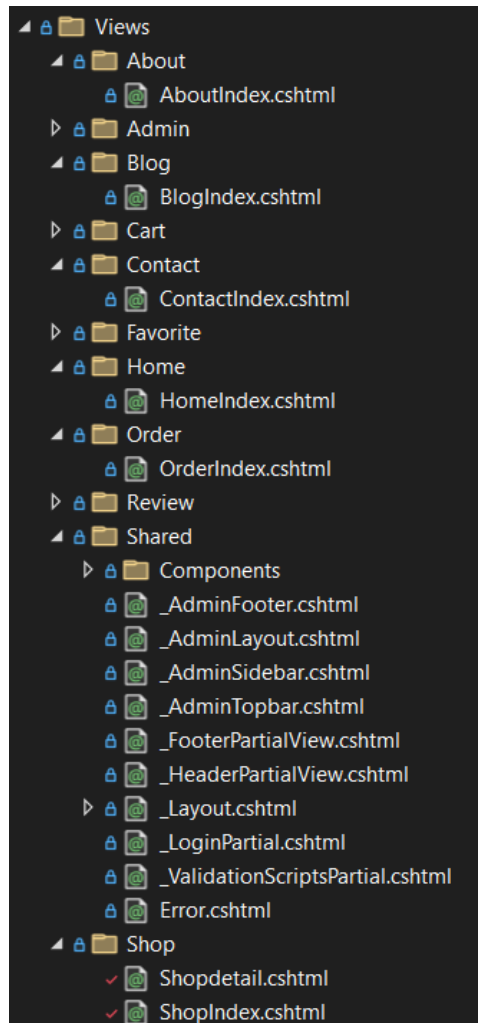
em rất nhiều trong việc tự động hóa các tác vụ phát triển, giúp quá trình triển khai dự án MVC của em trở nên nhanh chóng và trôi chảy hơn.



Ảnh 13. Ảnh thư viện và frameworks

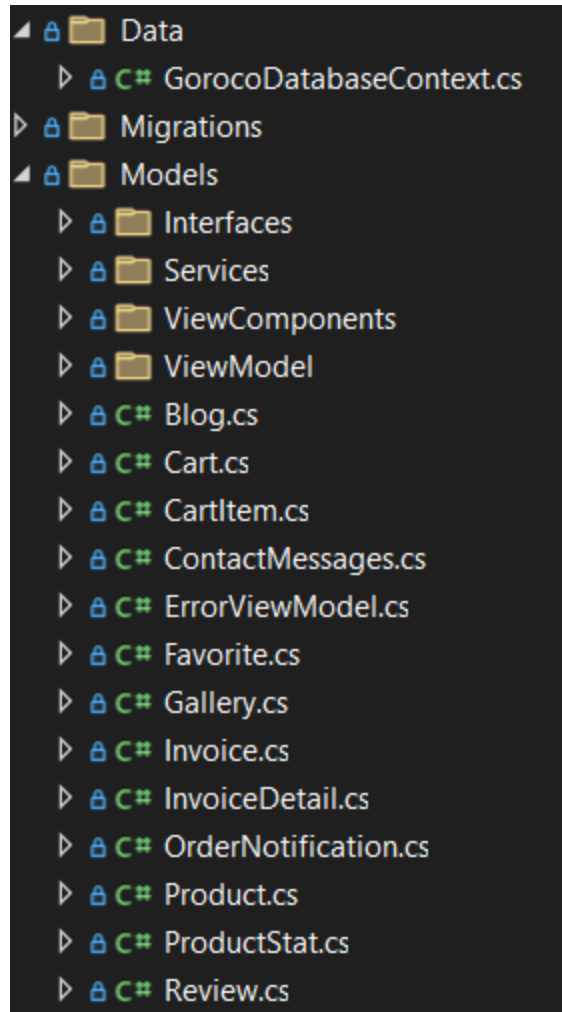
2.8.3 Các bước triển khai

Để xây dựng website này, em đã thực hiện một quy trình triển khai có hệ thống và toàn diện, bắt đầu từ việc thiết lập cấu trúc cơ bản vững chắc cho đến hoàn thiện các tính năng phức tạp và tối ưu hóa giao diện người dùng. Quy trình này bao gồm nhiều giai đoạn chính, mỗi giai đoạn đều được thực hiện một cách tỉ mỉ để đảm bảo chất lượng và hiệu quả của sản phẩm cuối cùng.



Ảnh 14. Ảnh cấu trúc của View

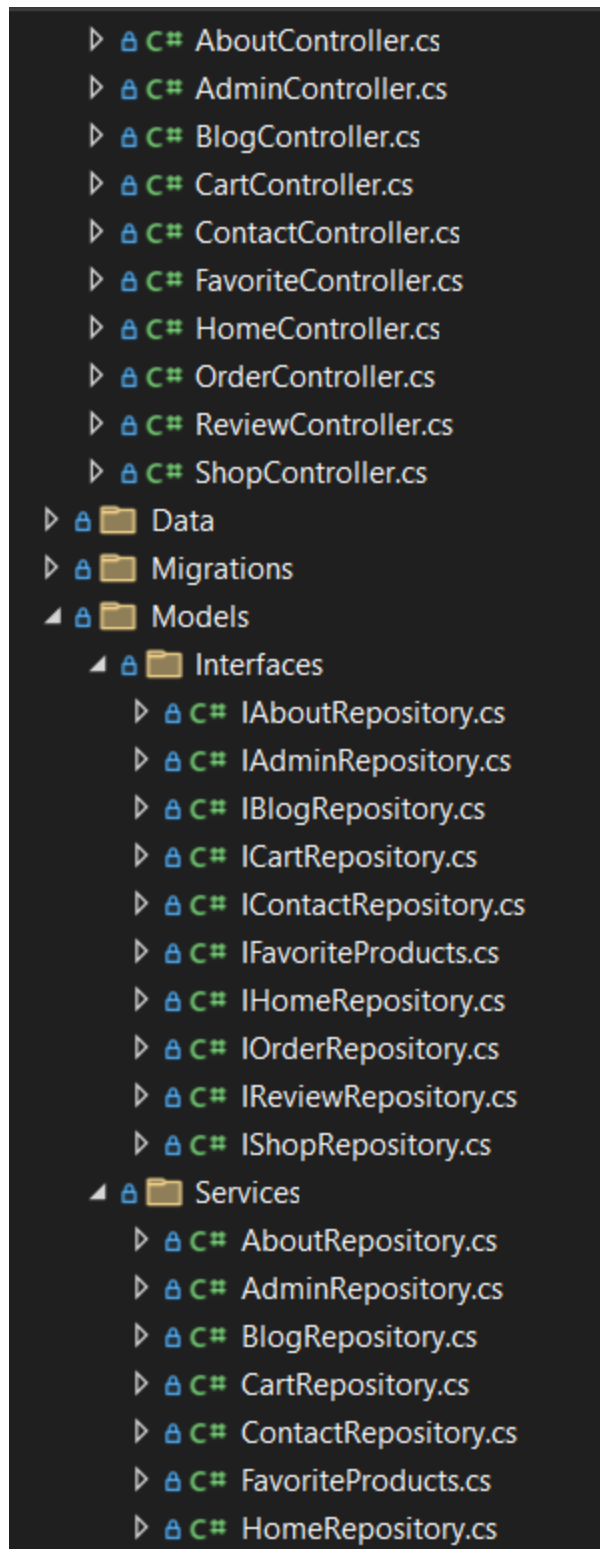
Đầu tiên, em tập trung vào việc định hình kiến trúc MVC (Model-View-Controller), một mô hình thiết kế phần mềm mạnh mẽ và quen thuộc, giúp tách biệt các thành phần của ứng dụng thành ba phần chính nhằm dễ dàng quản lý, phát triển và bảo trì. Cụ thể, trong giai đoạn phát triển View, em đã tạo ra các giao diện người dùng cho từng trang web. Mỗi View chịu trách nhiệm hiển thị thông tin và tương tác với người dùng, được xây dựng bằng cách sử dụng các công nghệ frontend phù hợp như HTML, CSS và JavaScript, đảm bảo giao diện thân thiện và trải nghiệm người dùng tối ưu. Cấu trúc của các View được tổ chức một cách logic, dễ dàng điều hướng và mở rộng, với các thư mục con tương ứng với từng chức năng hoặc module của website như Home, Admin, Shop, Blog, Cart, Order, hay Review. Đồng thời, các View dùng chung như header, footer, sidebar hay layout chính được đặt trong thư mục Shared và Components, giúp tái sử dụng code hiệu quả và đảm bảo tính nhất quán của giao diện trên toàn bộ website.



Ảnh 15. Ảnh cấu trúc của Model

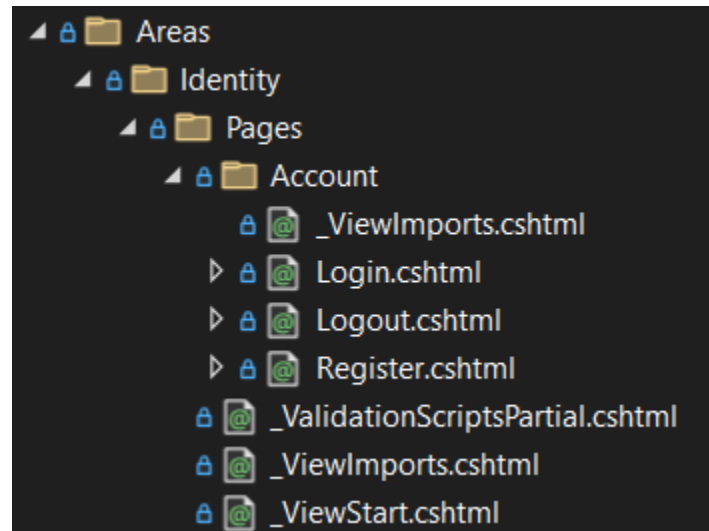
Song song đó, em xây dựng các Model để biểu diễn dữ liệu và logic nghiệp vụ. Trong thư mục Model này, em đặc biệt tổ chức thêm hai thư mục con là Services và Interfaces. Thư mục Interfaces chứa các định nghĩa giao diện cho từng repository, đảm bảo tính nhất quán và khả năng mở rộng trong việc truy cập dữ liệu của từng trang. Sau khi có các interface, em tiến hành tạo lớp DbContext để thiết lập cầu nối với cơ sở dữ liệu. Bước này là cực kỳ quan trọng vì nó định nghĩa ngữ cảnh làm việc với dữ liệu.

Tiếp theo, em tập trung vào phần dữ liệu: em lần lượt tạo từng class (lớp) trong Model để ánh xạ tới các bảng cụ thể trong cơ sở dữ liệu. Sau khi định nghĩa xong cấu trúc dữ liệu này, em sử dụng tính năng Add-Migration của Entity Framework Core để tạo ra các mã script cần thiết cho việc thay đổi cấu trúc cơ sở dữ liệu, và sau đó thực hiện Update-Database để áp dụng những thay đổi đó lên cơ sở dữ liệu thực tế.



Ảnh 16. Ảnh cấu trúc của các Controller và Repository

Khi cấu trúc dữ liệu đã sẵn sàng, em bắt đầu đi sâu vào phần logic. Em lần lượt viết các Controller cho từng trang, chịu trách nhiệm xử lý các yêu cầu từ người dùng. Đồng thời, em cũng phát triển các lớp Repository tương ứng với từng trang, chúng sẽ hiện thực hóa các interface đã định nghĩa trước đó, chứa toàn bộ logic tương tác với cơ sở dữ liệu. Sau khi Controller và Repository được kết nối, em quay lại chỉnh sửa các View, điều chỉnh chúng để có thể nhận và hiển thị dữ liệu đã được lấy từ cơ sở dữ liệu thông qua Controller và Repository, đảm bảo thông tin luôn được cập nhật chính xác.



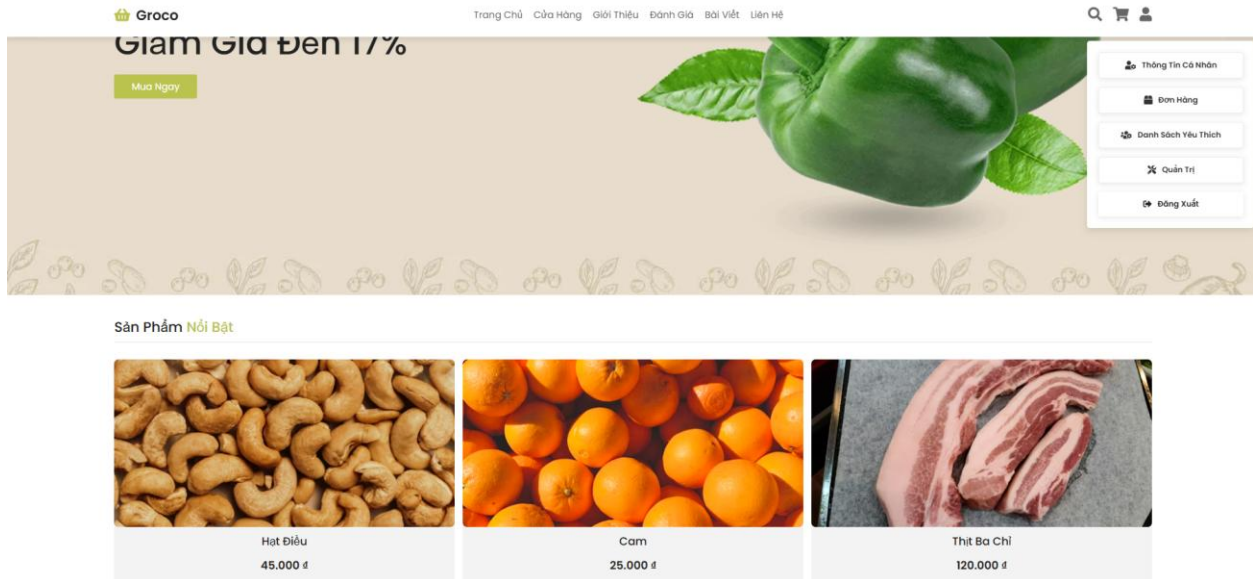
Ảnh 17. Cấu trúc của ASP.NET Core Identity

Cuối cùng, để hoàn thiện các tính năng quan trọng của một website, em đã tích hợp thư viện ASP.NET Core Identity vào dự án. Thư viện này cho phép em dễ dàng triển khai các chức năng xác thực người dùng như đăng nhập, đăng xuất và đăng ký tài khoản một cách bảo mật và hiệu quả. Sau khi các chức năng nền tảng và bảo mật đã ổn định, em dành thời gian để thiết kế và hoàn thiện giao diện người dùng, bao gồm cả giao diện riêng biệt cho quản trị viên (admin) và người dùng thông thường (user), đảm bảo mỗi đối tượng đều có trải nghiệm phù hợp và tối ưu nhất khi sử dụng website.

Chương 3: Kết quả chương trình

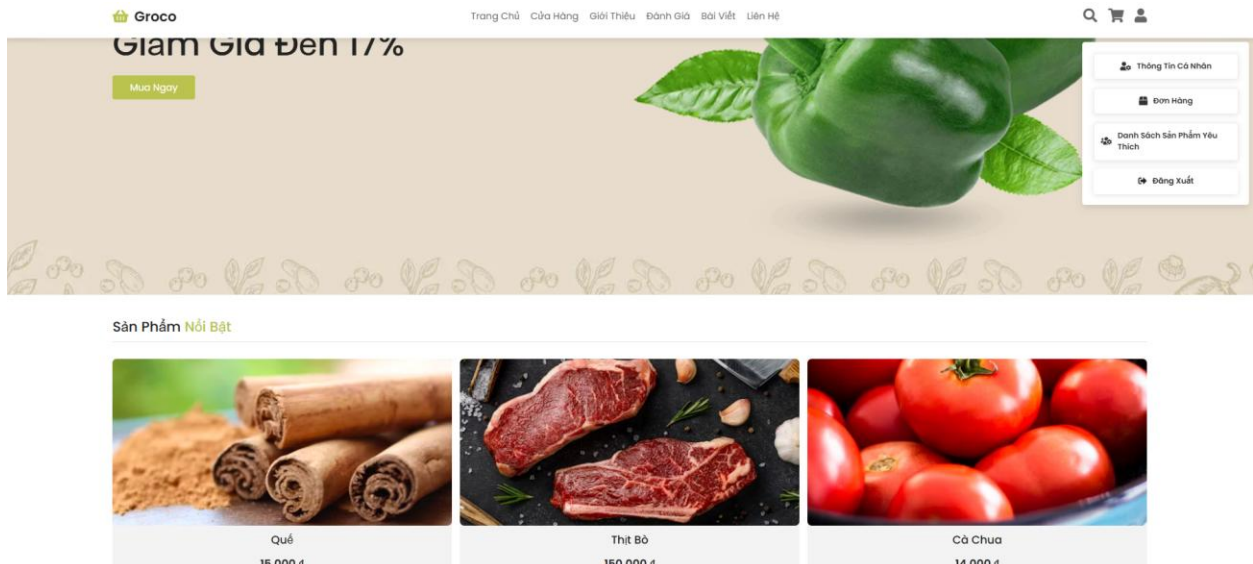
3.1 Demo Chức năng chính

3.1.1 Trang chủ (Sau khi đăng nhập của quản lý)



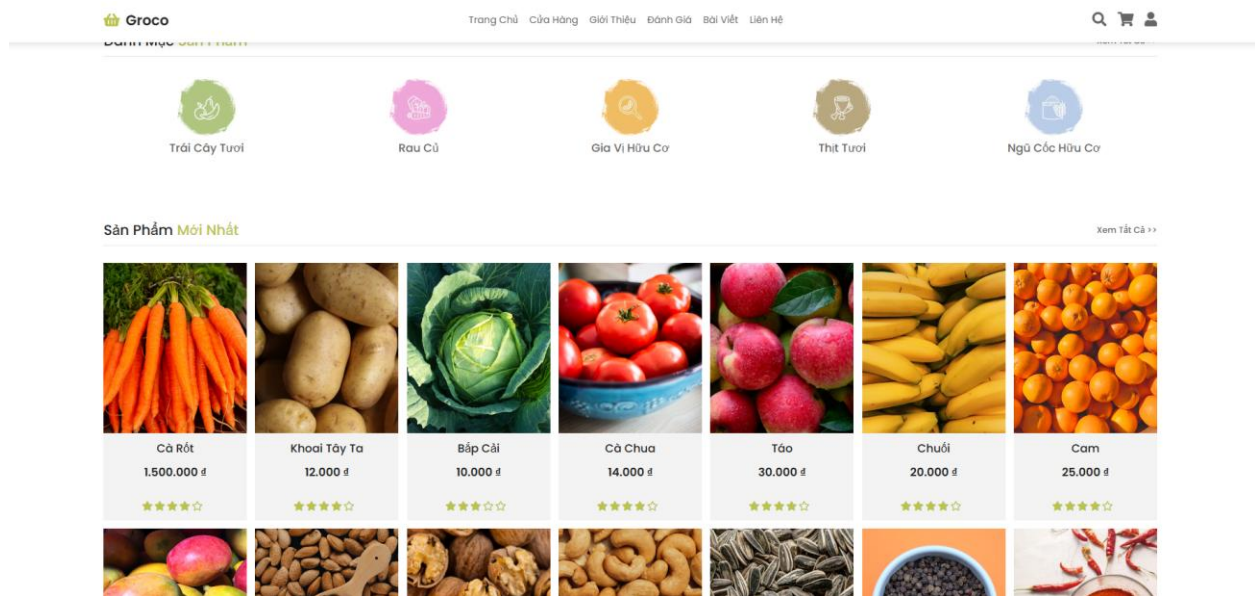
Ảnh 18. Ảnh giao diện trang chủ quản lý

3.1.2 Trang chủ (Sau khi đăng nhập của người dùng)



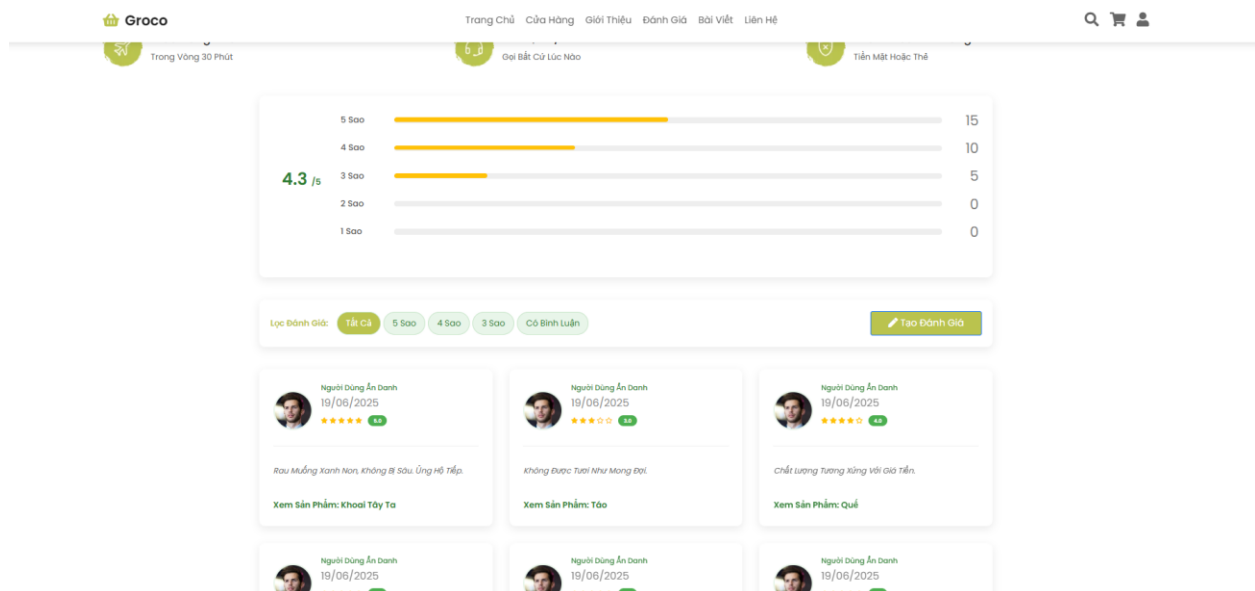
Ảnh 19. Ảnh giao diện trang chủ người dùng

3.1.3 Cửa hàng



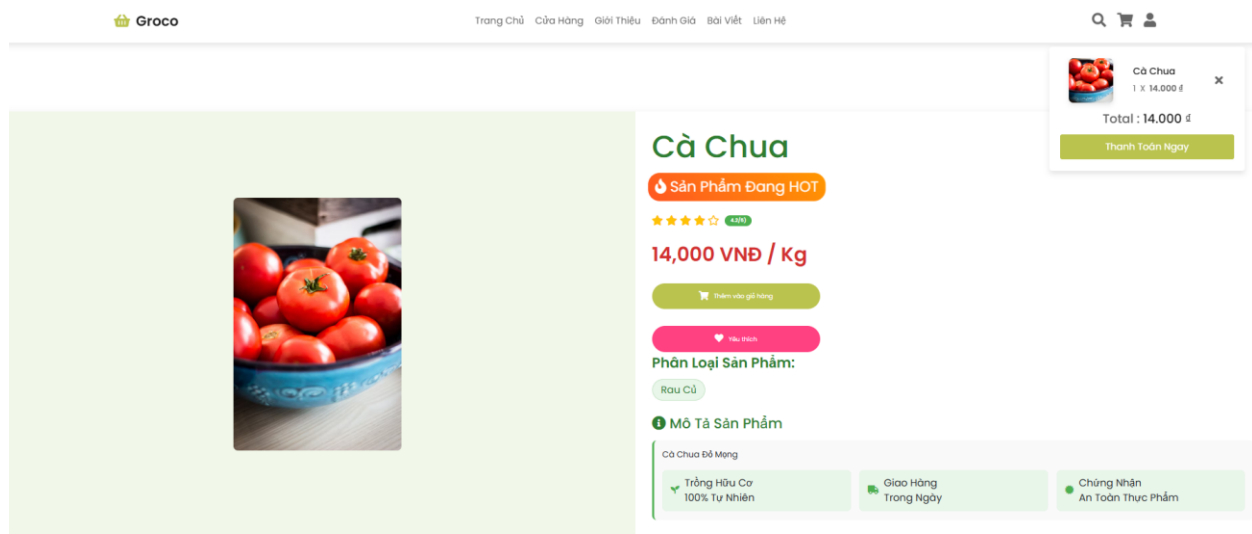
Ảnh 20. Ảnh giao diện cửa hàng

3.1.4 Đánh giá



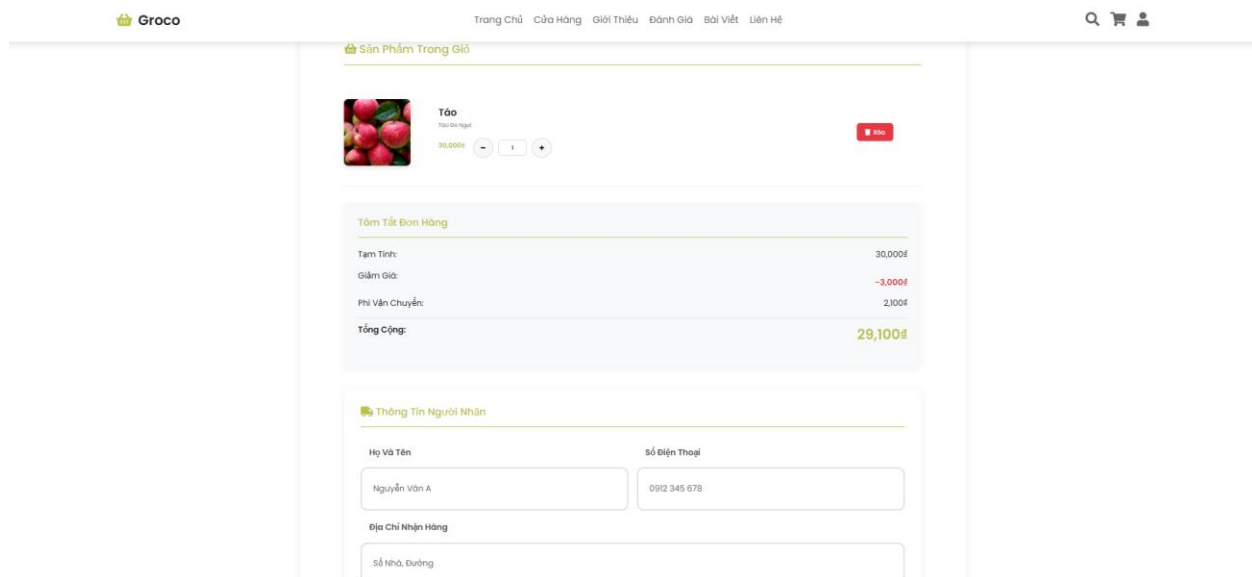
Ảnh 21. Ảnh giao diện trang đánh giá

3.1.5 Chi tiết sản phẩm



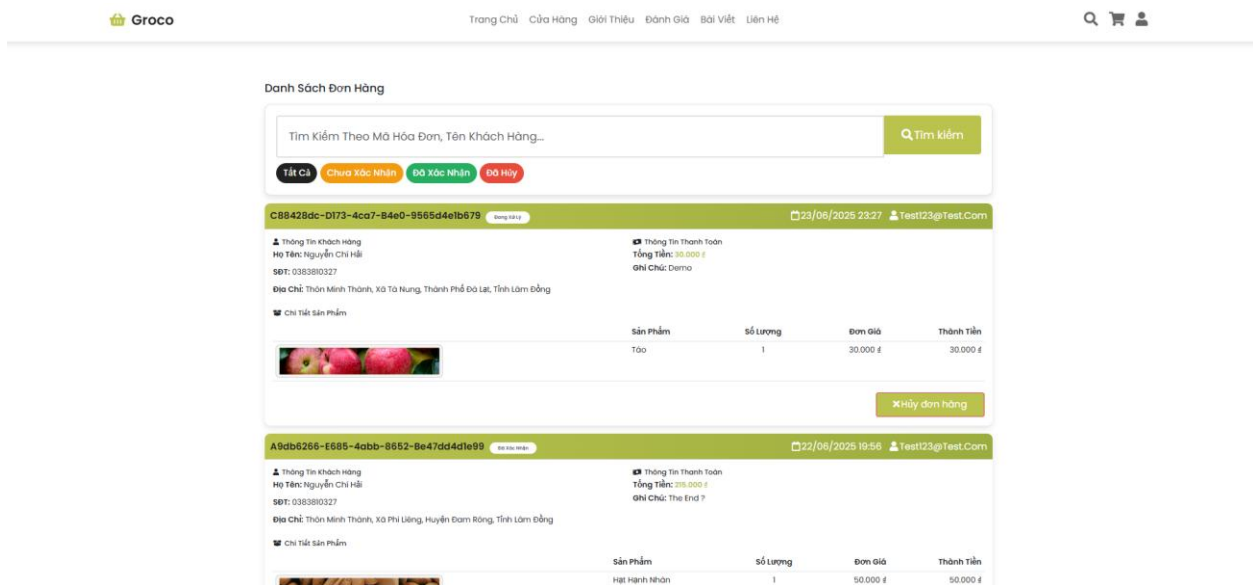
Ảnh 22. Ảnh giao diện trang chi tiết sản phẩm

3.1.6 Thanh toán



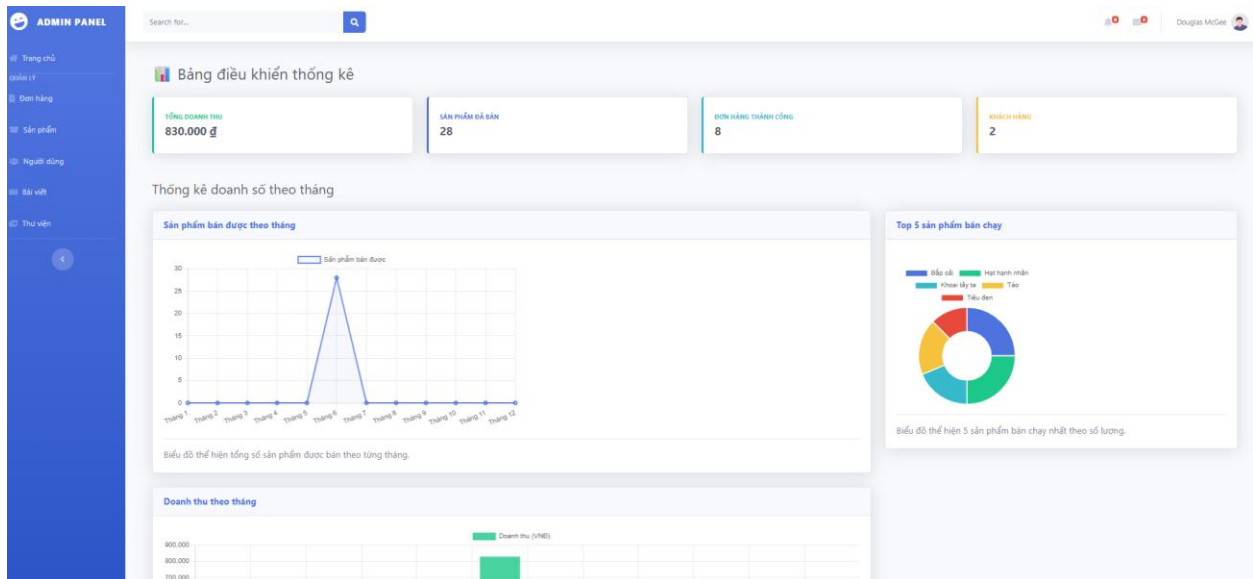
Ảnh 23. Ảnh giao diện trang thanh toán

3.1.7 Đơn hàng



Ảnh 24. Ảnh giao diện trang đơn hàng

3.1.8 Quản trị của quản lý



Ảnh 25. Ảnh giao diện trang quản trị của quản lý

3.2 Đánh giá, nhận xét

Quá trình xây dựng website này là một trải nghiệm học hỏi sâu sắc, giúp em không chỉ áp dụng lý thuyết mà còn giải quyết các vấn đề thực tế trong phát triển web. Em đã vận dụng thành công mô hình kiến trúc MVC để tạo ra một cấu trúc dự án rõ ràng, từ việc phân chia Controller, Model đến View, đảm bảo tính dễ đọc và khả năng mở rộng. Đồng thời, việc làm chủ các thao tác truy vấn và quản lý dữ liệu với Entity Framework Core cùng khả năng tích hợp các thư viện xác thực người dùng đã giúp website có đầy đủ chức năng cốt lõi và tương tác mượt mà. Tuy nhiên, em cũng nhận thấy cần có sự tối ưu hơn nữa để đạt được khả năng đồng bộ hóa dữ liệu và giao diện tức thời giữa các phiên làm việc và người dùng, một yếu tố quan trọng để nâng cao trải nghiệm người dùng trong tương lai.

Kết luận

1. Kết quả đạt được

Website đã được triển khai thành công dựa trên mô hình kiến trúc MVC, cho thấy sự nắm vững về cách tổ chức và phân chia các thành phần Controller, Model và View một cách hợp lý và chuẩn mực. Em đã xây dựng được hệ thống truy vấn và thao tác dữ liệu hiệu quả với cơ sở dữ liệu thông qua Entity Framework Core, đảm bảo các chức năng thêm, sửa, xóa và hiển thị dữ liệu hoạt động trơn tru. Đặc biệt, việc tích hợp các thư viện hỗ trợ xác thực người dùng đã giúp hoàn thiện các tính năng quan trọng như đăng ký, đăng nhập và đăng xuất, tạo nên một nền tảng vững chắc cho hệ thống quản lý người dùng của website.

2. Khuyết điểm

Mặc dù đã đạt được nhiều thành quả, website vẫn còn một số hạn chế đáng kể. Đầu tiên, là vấn đề liên quan đến khả năng đồng bộ hóa: các thay đổi về dữ liệu hoặc trạng thái giao diện có thể chưa được cập nhật tức thời và liên mạch trên nhiều phiên làm việc hoặc giữa các người dùng khác nhau. Thứ hai, do hạn chế về nguồn lực và thời gian, một số tính năng quan trọng và hữu ích cho trải nghiệm người dùng vẫn còn bị thiếu sót. Điển hình như khả năng yêu thích sản phẩm, tạo bài đánh giá và nhận xét sản phẩm, hoặc chức năng chỉnh sửa thông tin cá nhân của tài khoản người dùng vẫn chưa được triển khai hoàn chỉnh. Những thiếu sót này làm giảm đi một phần tính tương tác và cá nhân hóa của website..

3. Hướng phát triển

Trong tương lai, em dự định tập trung cải thiện khả năng đồng bộ hóa của website bằng cách nghiên cứu và áp dụng các công nghệ thời gian thực như WebSocket hoặc SignalR để đảm bảo cập nhật dữ liệu và giao diện ngay lập tức. Bên cạnh đó, em sẽ ưu tiên bổ sung các tính năng còn thiếu sót đã nêu trên, bao gồm phát triển chức năng yêu thích sản phẩm, cho phép người dùng đăng bài đánh giá, hoàn thiện phần quản lý thông tin cá nhân và nhiều tính năng còn thiếu sót khác. Đồng thời, em sẽ tiếp tục tối ưu hóa hiệu suất truy vấn, tăng cường bảo mật cho các giao dịch nhạy cảm, và mở rộng thêm các tính năng nâng cao khác nhằm mang lại một trải nghiệm thương mại điện tử toàn diện và chuyên nghiệp hơn cho người dùng.

Tài liệu tham khảo

1. Thiết kế trang web thương mại điện tử với MVC, Nguyễn Chí Hải, <https://github.com/Nguyen-Chi-Hai/DoAnThietKeWeb1> .
2. Tài liệu hướng dẫn ASP.NET Core MVC, Microsoft Learn, <https://docs.microsoft.com/en-us/aspnet/core/mvc/> .
3. Tài liệu học tập về AJAX (Asynchronous JavaScript and XML), Mozilla Developer Network, [Making network requests with JavaScript - Learn web development | MDN](#).