

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN MÔN HỌC: CHUYÊN ĐỀ ASP.NET
HỌC KỲ I, NĂM HỌC 2024-2025
XÂY DỰNG WEBSITE BÁN GIÀY

Giảng viên hướng dẫn:

TS. Nguyễn Nhứt Lam

Sinh viên thực hiện:

Họ tên: Võ Quang Thịnh

MSSV: 170123282

Lớp: DT23TTK10

Vĩnh Long, tháng 10 năm 2025

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN MÔN HỌC: CHUYÊN ĐỀ ASP.NET
HỌC KỲ I, NĂM HỌC 2024-2025
XÂY DỰNG WEBSITE BÁN GIÀY

Giảng viên hướng dẫn:

TS. Nguyễn Nhứt Lam

Sinh viên thực hiện:

Họ tên: Võ Quang Thịnh

MSSV: 170123282

Lớp: DT23TTK10

Vĩnh Long, tháng 10 năm 2025

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

(Ký tên và ghi rõ họ tên)

[illegible]

(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Trước hết, em xin bày tỏ lòng biết ơn đến toàn thể quý Thầy/Cô Trường Đại học Trà Vinh, đặc biệt là Khoa Công nghệ Thông tin đã tận tình giảng dạy, truyền đạt cho em những kiến thức nền tảng cũng như chuyên ngành trong suốt thời gian học tập tại trường. Những bài giảng, những định hướng quý báu đã trở thành hành trang quan trọng để em có thể vận dụng vào thực tế và hoàn thành đồ án này.

Em xin gửi lời cảm ơn sâu sắc nhất đến Thầy TS. Nguyễn Nhứt Lam – người trực tiếp hướng dẫn, đã luôn tận tâm chỉ bảo, định hướng, đưa ra những góp ý xác đáng và hỗ trợ em khắc phục những khó khăn trong quá trình nghiên cứu và phát triển website. Sự kiên nhẫn, tận tình và trách nhiệm của Thầy chính là động lực to lớn giúp em kiên trì hoàn thành đồ án.

Bên cạnh đó, em cũng xin cảm ơn các Thầy/Cô trong Hội đồng phản biện đã dành thời gian đọc, đánh giá và đưa ra những nhận xét quý báu để em có cơ hội nhìn nhận lại công việc của mình, đồng thời rút ra nhiều bài học kinh nghiệm cho con đường nghề nghiệp sau này.

Em cũng vô cùng biết ơn sự động viên, ủng hộ hết lòng của gia đình – chỗ dựa vững chắc để em an tâm học tập và nghiên cứu. Em xin cảm ơn bạn bè, đồng nghiệp đã luôn đồng hành, chia sẻ, đóng góp ý kiến và giúp đỡ em trong quá trình tìm kiếm tài liệu, thử nghiệm và hoàn thiện sản phẩm.

Mặc dù đã nỗ lực hết sức, nhưng do kiến thức và kinh nghiệm thực tiễn còn hạn chế, đồ án chắc chắn không tránh khỏi những thiếu sót. Em kính mong nhận được sự thông cảm và những góp ý chân thành từ quý Thầy/Cô để có thể hoàn thiện hơn trong tương lai.

Một lần nữa, em xin chân thành cảm ơn tất cả những sự giúp đỡ và đồng hành đã tạo nên động lực để em hoàn thành đồ án này.

Xin trân trọng cảm ơn!

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN	1
1.1. Lý do chọn đề tài	1
1.2. Mục tiêu của đồ án	1
1.3. Phạm vi nghiên cứu và triển khai	2
1.4. Tóm tắt công nghệ sử dụng	3
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT	4
2.1. Khái quát về ASP.NET.....	4
2.2. Các mô hình lập trình	5
2.2.1. Mô hình Web Forms.....	5
2.2.2. Mô hình MVC (Model – View – Controller).....	6
2.2.3. Mô hình Razor Pages.....	7
2.3. Công nghệ liên quan.....	8
2.3.1. Cơ sở dữ liệu – SQL Server	8
2.3.2. ORM – Entity Framewor.....	8
CHƯƠNG 3: HIỆN THỰC HOÁ NGHIÊN CỨU	9
3.1. Phân tích yêu cầu.....	9
3.1.1. Yêu cầu chức năng	9
3.1.2. Yêu cầu phi chức năng	10
3.2. Thiết kế cơ sở dữ liệu.....	11
3.3. Thiết kế chức năng	20
CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU	21
4.1. Cấu trúc dự án.....	21
4.2. Mô tả chi tiết.....	23

MỤC LỤC

4.2.1. Đăng nhập.....	23
4.2.2. Đăng ký	24
4.2.3. Hiển thị danh sách sản phẩm và tìm kiếm sản phẩm	25
4.2.4. Thêm sản phẩm vào giỏ hàng.....	26
4.2.5. Cập nhật số lượng trong giỏ hàng	27
4.2.6. Xóa sản phẩm trong giỏ hàng.....	28
4.3. Giao diện chính.....	29
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	34
5.1. Tóm tắt các chức năng đã hoàn thành	34
5.2. Demo kết quả đạt được.....	34
5.3. Đánh giá, so sánh mục tiêu ban đầu	39
5.4. Hạn chế và hướng phát triển.....	40
5.5. Kết luận.....	40
DANH MỤC TÀI LIỆU THAM KHẢO.....	41

DANH MỤC HÌNH ẢNH

Hình 1. Mô hình Web Forms.....	6
Hình 2. Mô hình MVC (Model - View - Controller)	7
Hình 3. Mô hình Razor Pages.....	7
Hình 4. Mô hình cơ sở dữ liệu.....	11
Hình 5. Usecase diagram tổng quát.....	20
Hình 6. Cấu trúc thư mục dự án	23
Hình 7. Xử lý chức năng đăng nhập.....	24
Hình 8. Xử lý chức năng đăng ký.....	25
Hình 9. Xử lý chức năng hiển thị danh sách sản phẩm và tìm kiếm sản phẩm.....	26
Hình 10. Xử lý chức năng thêm sản phẩm vào giỏ hàng	27
Hình 11. Xử lý chức năng cập nhật số lượng trong giỏ hàng.....	28
Hình 12. Xử lý chức năng xóa sản phẩm trong giỏ hàng	29
Hình 13. Màn hình Trang chủ	29
Hình 14. Màn hình Trang chủ - Sản phẩm mới.....	30
Hình 15. Màn hình Danh mục sản phẩm.....	30
Hình 16. Màn hình đăng nhập	31
Hình 17. Màn hình Đăng ký	31
Hình 18. Màn hình Giỏ hàng.....	32
Hình 19. Màn hình điền thông tin giao hàng.....	32
Hình 20. Màn hình Blog tin tức.....	33
Hình 21. Màn hình thông tin liên hệ.....	33
Hình 22. Màn hình Trang chủ	35
Hình 23. Màn hình Trang chủ - Sản phẩm mới.....	35
Hình 24. Màn hình Danh mục sản phẩm.....	36

DANH MỤC HÌNH ẢNH

Hình 25. Màn hình đăng nhập	36
Hình 26. Màn hình Đăng ký	37
Hình 27. Màn hình Giỏ hàng	37
Hình 28. Màn hình điền thông tin giao hàng	38
Hình 29. Màn hình Blog tin tức	38
Hình 30. Màn hình thông tin liên hệ	39

DANH MỤC BẢNG BIỂU

Bảng 1. Bảng Orders	12
Bảng 2. Bảng OrderDetails.....	12
Bảng 3. Bảng Products	13
Bảng 4. Bảng Categories	14
Bảng 5. Bảng Brands	14
Bảng 6. BảngAspNetUsers.....	15
Bảng 7. BảngAspNetRoles.....	16
Bảng 8. BảngAspNetUserRoles	16
Bảng 9. BảngAspNetUserClaims	17
Bảng 10. BảngAspNetRoleClaims	17
Bảng 11. BảngAspNetUserLogins	18
Bảng 12. BảngAspNetUserTokens.....	18
Bảng 13. Bảng __EFMigrationsHistory	19

TÓM TẮT ĐỒ ÁN

Đề tài “**Xây dựng Website bán giày**” bằng ASP.NET được thực hiện với mục tiêu phát triển một hệ thống thương mại điện tử hỗ trợ hoạt động mua bán giày trực tuyến, vừa đáp ứng nhu cầu thực tiễn vừa giúp sinh viên vận dụng kiến thức đã học vào thực hành. Trong quá trình nghiên cứu, em đã tiến hành phân tích yêu cầu hệ thống và xác định các chức năng cốt lõi như quản lý sản phẩm, giỏ hàng, đặt hàng, quản lý khách hàng và quản trị hệ thống. ASP.NET được lựa chọn làm nền tảng phát triển nhờ khả năng tích hợp tốt với SQL Server, tính bảo mật cao và hỗ trợ đầy đủ cho xây dựng ứng dụng web hiện đại. Hệ thống được triển khai theo mô hình ba lớp nhằm đảm bảo khả năng mở rộng và dễ dàng bảo trì, đồng thời cơ sở dữ liệu được thiết kế với các ràng buộc hợp lý để duy trì tính toàn vẹn dữ liệu.

Kết quả của đồ án là một website hoàn chỉnh cho phép người dùng đăng ký, đăng nhập, tìm kiếm sản phẩm, thêm vào giỏ hàng, đặt hàng và theo dõi tình trạng đơn hàng, trong khi quản trị viên có thể quản lý danh mục, sản phẩm, khách hàng và đơn hàng. Giao diện được xây dựng trực quan, thân thiện với người dùng và hỗ trợ các tính năng tìm kiếm, phân trang, lọc sản phẩm để nâng cao trải nghiệm. Thông qua quá trình thực hiện, em không chỉ củng cố kiến thức về ASP.NET, SQL Server và mô hình phát triển ứng dụng web mà còn rèn luyện kỹ năng phân tích, thiết kế hệ thống cũng như tổ chức và quản lý một dự án phần mềm thực tế.

CHƯƠNG 1: TỔNG QUAN

1.1. Lý do chọn đề tài

Trong những năm gần đây, cùng với sự phát triển mạnh mẽ của công nghệ thông tin, Internet đã trở thành một phần không thể thiếu trong đời sống con người. Các hoạt động kinh doanh truyền thống dần được chuyển đổi sang môi trường trực tuyến nhằm mở rộng thị trường, giảm chi phí vận hành và tiếp cận khách hàng nhanh chóng hơn. Trong bối cảnh đó, thương mại điện tử đã và đang phát triển vượt bậc, trở thành xu hướng tất yếu của thời đại số.

Ngành thời trang và giày dép là một trong những lĩnh vực có nhu cầu tiêu thụ cao và thường xuyên được người tiêu dùng quan tâm. Tuy nhiên, việc mua sắm giày dép theo hình thức truyền thống vẫn còn tồn tại nhiều hạn chế như tốn thời gian, không thuận tiện trong việc so sánh sản phẩm, hoặc phụ thuộc vào vị trí địa lý. Chính vì vậy, việc xây dựng một website bán giày trực tuyến giúp người dùng có thể dễ dàng tra cứu, lựa chọn, đặt mua sản phẩm mọi lúc mọi nơi là một giải pháp thiết thực.

Bên cạnh đó, từ góc độ học tập, việc lựa chọn đề tài này còn giúp người thực hiện có cơ hội vận dụng kiến thức đã học về lập trình web, cơ sở dữ liệu, phân tích và thiết kế hệ thống thông tin vào một sản phẩm cụ thể. Qua đó, người thực hiện có thể củng cố kỹ năng kỹ thuật, hiểu rõ quy trình phát triển phần mềm theo mô hình thực tế, và tạo nền tảng cho công việc chuyên môn sau khi tốt nghiệp.

Với những lý do nêu trên, đề tài “Xây dựng Website Bán Giày” sử dụng ASP.NET Framework và SQL Server được lựa chọn với mục tiêu vừa mang tính thực tiễn, vừa mang tính học thuật, đáp ứng được nhu cầu của người dùng và mục tiêu đào tạo của ngành công nghệ thông tin.

1.2. Mục tiêu của đề án

Mục tiêu tổng quát của đề án là xây dựng một hệ thống website bán giày trực tuyến có khả năng hỗ trợ người dùng tra cứu, mua sắm và đặt hàng trực tiếp thông qua Internet. Hệ thống được thiết kế hướng tới sự đơn giản, dễ sử dụng, ổn định và bảo mật, phù hợp với quy mô của các cửa hàng vừa và nhỏ.

Cụ thể, đề án hướng đến các mục tiêu sau:

- Xây dựng giao diện thân thiện và hiện đại, giúp người dùng dễ dàng tương tác và thực hiện các thao tác như xem sản phẩm, tìm kiếm, thêm vào giỏ hàng và đặt hàng nhanh chóng.
- Phát triển hệ thống quản trị (Admin) cho phép người quản lý có thể thêm, sửa, xóa sản phẩm; theo dõi đơn hàng; quản lý thông tin khách hàng và tình trạng giao dịch.
- Thiết kế cơ sở dữ liệu chặt chẽ và hợp lý, đảm bảo tính toàn vẹn và an toàn cho dữ liệu.
- Ứng dụng công nghệ ASP.NET Framework kết hợp với SQL Server, qua đó thực hành các kỹ năng lập trình hướng đối tượng, xử lý sự kiện và tương tác giữa tầng giao diện – tầng xử lý – tầng dữ liệu.
- Kiểm thử và triển khai ứng dụng trên môi trường thực tế (localhost hoặc mạng nội bộ), đánh giá hiệu năng và tính ổn định của hệ thống.

1.3. Phạm vi nghiên cứu và triển khai

Đề tài được thực hiện trong phạm vi một website bán hàng trực tuyến chuyên về giày dép, có chức năng cơ bản của một hệ thống thương mại điện tử, nhưng chưa mở rộng đến các tính năng nâng cao như thanh toán online qua cổng ngân hàng hoặc vận hành trên môi trường điện toán đám mây.

Về phạm vi người dùng, hệ thống được chia thành hai nhóm chính:

- Người dùng thông thường (khách hàng): có thể truy cập website, xem thông tin sản phẩm, tìm kiếm, xem chi tiết, thêm vào giỏ hàng và đặt hàng. Khách hàng cũng có thể đăng ký tài khoản để theo dõi lịch sử mua hàng.
- Người quản trị (admin): có quyền đăng nhập vào trang quản trị để thực hiện các thao tác quản lý như thêm mới, chỉnh sửa hoặc xóa sản phẩm, xem danh sách đơn hàng, xác nhận và xử lý các đơn hàng của khách hàng.

Về phạm vi chức năng, hệ thống bao gồm các phần chính như:

- Quản lý danh mục và thông tin chi tiết của sản phẩm (mã, tên, loại giày, giá, số lượng, hình ảnh).
- Tìm kiếm và lọc sản phẩm theo tên, loại hoặc giá.

- Chức năng giỏ hàng và đặt hàng trực tuyến.
- Quản lý tài khoản người dùng và phân quyền.
- Trang quản trị giúp quản lý toàn bộ hoạt động bán hàng.

Về phạm vi kỹ thuật, hệ thống được xây dựng dựa trên ASP.NET Framework (C#), sử dụng SQL Server để quản lý cơ sở dữ liệu. Giao diện được thiết kế bằng HTML, CSS, Bootstrap, JavaScript và JQuery, đảm bảo tính tương thích và trải nghiệm tốt trên nhiều thiết bị. Quá trình lập trình và chạy thử được thực hiện trên môi trường Visual Studio, triển khai và kiểm thử cục bộ qua IIS Express hoặc localhost.

1.4. Tóm tắt công nghệ sử dụng

Đề tài sử dụng nhiều công nghệ khác nhau, trong đó ASP.NET Framework đóng vai trò là nền tảng chính để phát triển website. ASP.NET là một framework mạnh mẽ của Microsoft, hỗ trợ xây dựng các ứng dụng web động, có khả năng tương tác cao, bảo mật tốt và dễ dàng mở rộng. Với ngôn ngữ lập trình C#, hệ thống có thể xử lý logic nghiệp vụ, tương tác cơ sở dữ liệu, và thực hiện các thao tác phía server một cách hiệu quả.

Cơ sở dữ liệu của hệ thống được xây dựng trên Microsoft SQL Server, một hệ quản trị cơ sở dữ liệu quan hệ phổ biến, giúp lưu trữ và truy vấn dữ liệu nhanh chóng, an toàn. SQL Server cho phép người phát triển dễ dàng thiết kế các bảng dữ liệu, ràng buộc khóa ngoại, và thực hiện các truy vấn phức tạp phục vụ cho việc xử lý nghiệp vụ trong website.

Phần giao diện người dùng được xây dựng bằng HTML và CSS, kết hợp Bootstrap để tối ưu hiển thị trên nhiều loại màn hình khác nhau, giúp website có giao diện trực quan, hiện đại và thân thiện. Ngoài ra, JavaScript và JQuery được sử dụng để tăng tính tương tác, xử lý các hiệu ứng động và thao tác trên trình duyệt mà không cần tải lại toàn bộ trang.

Công cụ phát triển chính được sử dụng là Visual Studio, một môi trường lập trình tích hợp (IDE) mạnh mẽ, hỗ trợ gỡ lỗi, quản lý dự án và triển khai ứng dụng một cách dễ dàng.

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1. Khái quát về ASP.NET

ASP.NET là một nền tảng phát triển ứng dụng web mạnh mẽ được Microsoft giới thiệu lần đầu vào năm 2002, hoạt động trên nền tảng .NET Framework. ASP.NET cho phép lập trình viên xây dựng các ứng dụng web, dịch vụ web, và website động bằng nhiều ngôn ngữ lập trình khác nhau như C#, VB.NET, hoặc F#.

ASP.NET không chỉ hỗ trợ mô hình phát triển hướng đối tượng (OOP) mà còn cung cấp một môi trường thực thi an toàn, hiệu năng cao, cùng khả năng tích hợp sâu với các công nghệ Microsoft khác như SQL Server, Azure và Visual Studio. Nhờ đó, ASP.NET trở thành một trong những công nghệ hàng đầu được sử dụng trong phát triển các hệ thống web doanh nghiệp hiện nay.

- Ưu điểm:

- Hiệu năng cao: ASP.NET biên dịch mã thành dạng Microsoft Intermediate Language (MSIL), giúp cải thiện tốc độ xử lý so với các nền tảng dựa trên ngôn ngữ kịch bản (script-based).
- Bảo mật mạnh mẽ: Hỗ trợ cơ chế xác thực người dùng (authentication), phân quyền truy cập (authorization) và mã hóa dữ liệu (encryption) tích hợp sẵn.
- Tích hợp dễ dàng với C# và .NET Framework: Cho phép sử dụng các thư viện, API và công cụ trong hệ sinh thái .NET, giúp tăng năng suất lập trình.
- Quản lý trạng thái tốt: ASP.NET hỗ trợ cơ chế State Management linh hoạt như Session, ViewState, Cookies, giúp duy trì dữ liệu người dùng trong quá trình tương tác.
- Công cụ hỗ trợ phát triển mạnh: Visual Studio cung cấp môi trường lập trình trực quan, gỡ lỗi dễ dàng và hỗ trợ kéo – thả trong thiết kế giao diện web.
- Khả năng mở rộng và bảo trì cao: Hỗ trợ mô hình MVC (Model-View-Controller), tách biệt logic nghiệp vụ, giao diện và dữ liệu, giúp hệ thống dễ dàng mở rộng và bảo trì.

- Khuyết điểm:

- Chi phí triển khai cao: ASP.NET thường yêu cầu môi trường máy chủ Windows

Server và IIS (Internet Information Services), dẫn đến chi phí vận hành lớn hơn so với các công nghệ mã nguồn mở.

- Tính phụ thuộc vào hệ sinh thái Microsoft: Khó khăn trong việc triển khai trên các nền tảng không thuộc Microsoft.
- Độ phức tạp cao đối với người mới bắt đầu: Việc nắm vững ASP.NET đòi hỏi hiểu biết sâu về C#, .NET Framework và cấu trúc ứng dụng web.

Kiến trúc của ASP.NET được thiết kế theo mô hình 3 lớp (Three-Tier Architecture) gồm:

- Presentation Layer (Tầng giao diện): Là phần hiển thị trực tiếp với người dùng, bao gồm các trang web (.aspx, .cshtml) chứa mã HTML, CSS, JavaScript và các điều khiển (controls). Tầng này chịu trách nhiệm nhận dữ liệu từ người dùng và hiển thị kết quả từ tầng xử lý.
- Business Logic Layer (Tầng nghiệp vụ): Chứa các lớp (classes) xử lý logic nghiệp vụ của ứng dụng. Đây là nơi xử lý các yêu cầu từ giao diện người dùng, kiểm tra dữ liệu, áp dụng quy tắc kinh doanh, và giao tiếp với tầng dữ liệu.
- Data Access Layer (Tầng truy cập dữ liệu): Chịu trách nhiệm kết nối và tương tác với cơ sở dữ liệu (SQL Server). Các thao tác như thêm, xóa, sửa, truy vấn dữ liệu đều được thực hiện tại tầng này, đảm bảo tính độc lập giữa dữ liệu và logic nghiệp vụ.

Mô hình kiến trúc nhiều tầng của ASP.NET giúp tăng khả năng tái sử dụng mã nguồn, dễ bảo trì, nâng cấp và đảm bảo tính mở rộng khi hệ thống phát triển.

2.2. Các mô hình lập trình

ASP.NET hỗ trợ nhiều mô hình phát triển ứng dụng web khác nhau, tùy thuộc vào quy mô và nhu cầu của dự án. Ba mô hình phổ biến nhất gồm Web Forms, MVC, và Razor Pages.

2.2.1. Mô hình Web Forms

ASP.NET Web Forms là mô hình lập trình ra đời đầu tiên trong ASP.NET, dựa trên khái niệm event-driven (hướng sự kiện), tương tự như lập trình ứng dụng desktop. Web Forms cho phép lập trình viên kéo thả các thành phần (controls) như textbox, button, gridview lên giao diện và xử lý sự kiện trong code-behind (file .aspx.cs).

- Ưu điểm của Web Forms:

- Dễ học, dễ sử dụng cho người mới bắt đầu.
- Hỗ trợ kéo thả trực quan trong Visual Studio.
- Giảm bớt mã HTML nhờ các điều khiển máy chủ.

- Nhược điểm của Web Forms:

- Tách biệt giữa giao diện và logic chưa rõ ràng.
- Hiệu năng thấp hơn trong các ứng dụng lớn.
- Khó kiểm soát mã HTML sinh ra tự động.



Hình 1. Mô hình Web Forms

2.2.2. Mô hình MVC (Model – View – Controller)

ASP.NET MVC ra đời nhằm khắc phục các hạn chế của Web Forms. Mô hình này chia ứng dụng thành ba phần riêng biệt:

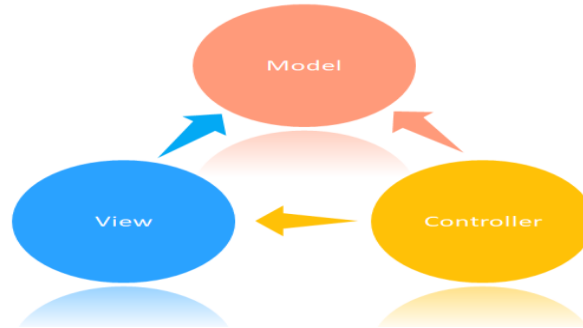
- Model: Quản lý dữ liệu và các quy tắc nghiệp vụ.
- View: Hiển thị giao diện người dùng.
- Controller: Tiếp nhận yêu cầu (request), xử lý và trả về kết quả phù hợp.

- Ưu điểm của MVC:

- Phân tách rõ ràng giữa các tầng, dễ mở rộng và bảo trì.
- Hỗ trợ tốt cho các nguyên tắc phát triển hiện đại như TDD (Test-Driven Development).
- Cho phép kiểm soát hoàn toàn HTML, CSS, JavaScript.
- Dễ dàng tích hợp với các framework front-end như Angular, React, Vue.

- Nhược điểm của MVC:

- Cấu trúc phức tạp hơn, cần nhiều thời gian học.
- Đòi hỏi lập trình viên hiểu sâu về mô hình kiến trúc phần mềm.



Hình 2. Mô hình MVC (Model – View – Controller)

2.2.3. Mô hình Razor Pages

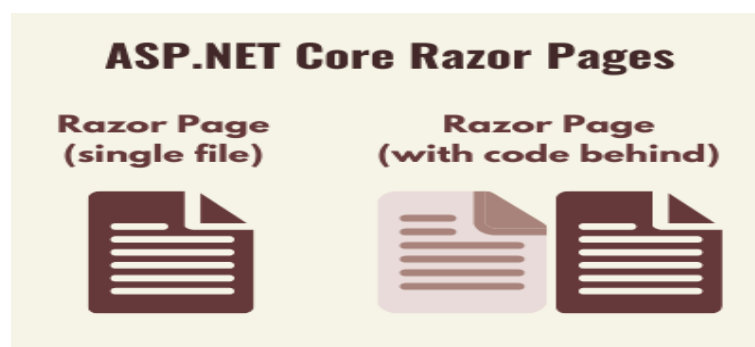
Razor Pages được giới thiệu trong phiên bản ASP.NET Core nhằm đơn giản hóa mô hình MVC. Mỗi trang Razor hoạt động như một view và controller nhò gọn trong cùng một file, giúp lập trình nhanh và dễ quản lý.

- Ưu điểm của Razor Pages:

- Cấu trúc gọn nhẹ, dễ phát triển với các ứng dụng quy mô vừa và nhỏ.
- Tối ưu cho các dự án không cần nhiều controller phức tạp.
- Tốc độ phản hồi cao, mã ngắn gọn, dễ bảo trì.

- Nhược điểm của Razor Pages:

- Chưa phù hợp cho các dự án lớn có nghiệp vụ phức tạp.
- Khó tách biệt hoàn toàn giữa giao diện và logic nếu không tuân thủ chặt chẽ cấu trúc.



Hình 3. Mô hình Razor Pages

2.3. Công nghệ liên quan

2.3.1. Cơ sở dữ liệu – SQL Server

Microsoft SQL Server là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) do Microsoft phát triển, được sử dụng phổ biến trong các ứng dụng ASP.NET. SQL Server hỗ trợ ngôn ngữ truy vấn T-SQL (Transact-SQL), cho phép thực hiện các thao tác như thêm, xóa, sửa, truy vấn dữ liệu một cách hiệu quả.

Một số đặc điểm nổi bật của SQL Server:

- Hỗ trợ giao dịch (transaction) đảm bảo tính toàn vẹn dữ liệu.
- Có khả năng mở rộng và tối ưu hóa truy vấn tốt.
- Cung cấp cơ chế bảo mật mạnh, hỗ trợ xác thực người dùng (authentication) và phân quyền (authorization).
- Tích hợp trực tiếp với Visual Studio, giúp lập trình viên dễ dàng quản lý cơ sở dữ liệu ngay trong môi trường phát triển.

Trong đồ án “Website Bán Giày”, SQL Server được sử dụng để lưu trữ toàn bộ dữ liệu sản phẩm, khách hàng, đơn hàng và thông tin quản trị.

2.3.2. ORM – Entity Framework

Entity Framework (EF) là một ORM (Object Relational Mapping) mạnh mẽ do Microsoft phát triển, cho phép lập trình viên thao tác với cơ sở dữ liệu bằng đối tượng (object) thay vì câu lệnh SQL thuần túy. EF tự động ánh xạ (mapping) giữa các lớp trong ứng dụng và bảng dữ liệu trong cơ sở dữ liệu, giúp giảm thiểu lỗi và tăng hiệu quả phát triển.

Lợi ích của việc sử dụng Entity Framework:

- Giảm bớt việc viết mã SQL thủ công.
- Hỗ trợ LINQ (Language Integrated Query) để truy vấn dữ liệu dưới dạng đối tượng.
- Dễ dàng di chuyển dữ liệu giữa các môi trường nhờ cơ chế Migration.
- Giúp mã nguồn dễ đọc, dễ bảo trì và tuân thủ nguyên tắc separation of concerns.

Trong dự án này, Entity Framework đóng vai trò là cầu nối giữa tầng nghiệp vụ và cơ sở dữ liệu, đảm bảo dữ liệu được truy xuất nhanh, chính xác và an toàn.

CHƯƠNG 3: HIỆN THỰC HOÁ NGHIÊN CỨU

3.1. Phân tích yêu cầu

3.1.1. Yêu cầu chức năng

- Chức năng dành cho Khách hàng:

- Đăng ký tài khoản: Người dùng có thể tạo tài khoản mới bằng cách cung cấp thông tin cá nhân cơ bản (họ tên, email, số điện thoại, mật khẩu).
- Đăng nhập/Đăng xuất: Cho phép người dùng đăng nhập vào hệ thống để sử dụng các chức năng nâng cao và đăng xuất khi kết thúc phiên làm việc.
- Tìm kiếm và duyệt sản phẩm: Người dùng có thể tìm kiếm sản phẩm theo tên, danh mục hoặc từ khóa, đồng thời duyệt danh sách sản phẩm với phân trang và lọc theo giá, loại sản phẩm.
- Xem chi tiết sản phẩm: Người dùng xem được thông tin chi tiết của sản phẩm như tên, hình ảnh, giá bán, mô tả, tình trạng tồn kho.
- Giỏ hàng: Người dùng có thể thêm sản phẩm vào giỏ hàng, cập nhật số lượng hoặc xóa sản phẩm ra khỏi giỏ.
- Đặt hàng: Cho phép khách hàng xác nhận đơn hàng từ giỏ hàng, nhập thông tin giao hàng và gửi đơn đặt hàng đến hệ thống.
- Theo dõi đơn hàng: Khách hàng có thể xem trạng thái đơn hàng (chờ xác nhận, đang giao, hoàn tất, hủy).
- Quản lý tài khoản cá nhân: Người dùng có thể cập nhật thông tin cá nhân, thay đổi mật khẩu.

- Chức năng dành cho Quản trị viên:

- Quản lý danh mục sản phẩm: Cho phép thêm, sửa, xóa các danh mục sản phẩm.
- Quản lý sản phẩm: Thêm mới sản phẩm, chỉnh sửa thông tin sản phẩm, cập nhật tồn kho, xóa sản phẩm khi cần thiết.
- Quản lý khách hàng: Theo dõi danh sách khách hàng, xem thông tin chi tiết, khóa hoặc mở tài khoản khi cần.

- Quản lý đơn hàng: Xem danh sách đơn hàng, cập nhật trạng thái đơn hàng (xác nhận, giao hàng, hoàn tất, hủy).
- Báo cáo – thống kê: Thống kê số lượng đơn hàng, sản phẩm bán chạy, doanh thu cơ bản để phục vụ công tác quản trị.

3.1.2. Yêu cầu phi chức năng

Yêu cầu phi chức năng mô tả chất lượng hoạt động của hệ thống, bao gồm hiệu năng, bảo mật, tính ổn định và khả năng mở rộng. Đây là những yếu tố không trực tiếp thể hiện qua tính năng, nhưng có ảnh hưởng lớn đến trải nghiệm và hiệu quả sử dụng hệ thống.

- Hiệu năng:

- Hệ thống phải đáp ứng nhanh chóng với thời gian phản hồi trung bình dưới 3 giây cho các thao tác cơ bản (xem sản phẩm, thêm giỏ hàng).
- Dữ liệu truy vấn và hiển thị phải được tối ưu hóa bằng cách sử dụng caching, phân trang và chỉ truy vấn các trường cần thiết.

- Bảo mật:

- Thông tin tài khoản người dùng và mật khẩu được mã hóa khi lưu trữ trong cơ sở dữ liệu.
- Hệ thống phân quyền rõ ràng giữa người dùng và quản trị viên.
- Kiểm tra và ngăn chặn các lỗi bảo mật phổ biến như SQL Injection, XSS (Cross-Site Scripting) hoặc CSRF (Cross-Site Request Forgery).

- Khả năng mở rộng (Scalability):

- Hệ thống được thiết kế với kiến trúc nhiều lớp, dễ dàng mở rộng chức năng trong tương lai như tích hợp thanh toán online hoặc hệ thống vận chuyển.

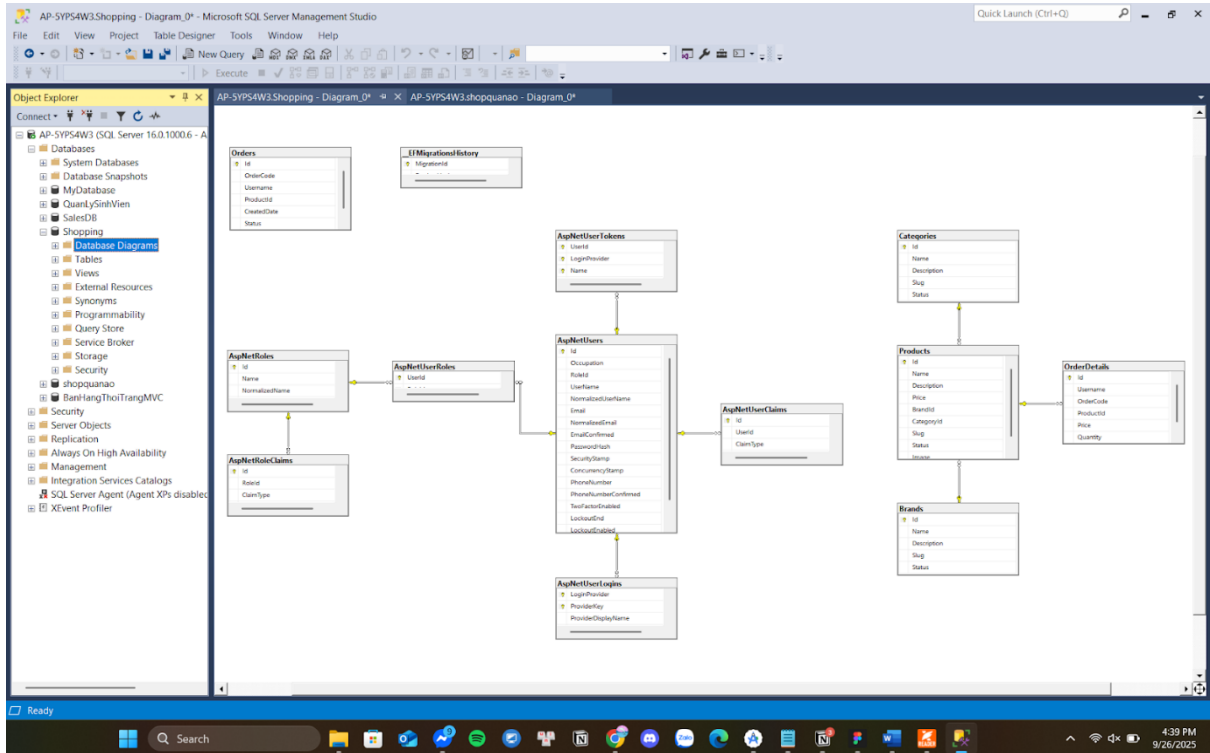
- Giao diện và trải nghiệm người dùng:

- Giao diện trực quan, dễ sử dụng, hỗ trợ tiếng Việt đầy đủ.
- Website hiển thị tốt trên các thiết bị khác nhau (desktop, tablet, smartphone) nhờ thiết kế Responsive với Bootstrap.

- Tính tương thích:

- Website hoạt động ổn định trên các trình duyệt phổ biến như Google Chrome, Microsoft Edge, Firefox, Safari.
- Tương thích tốt với các hệ điều hành Windows, macOS và Android.

3.2. Thiết kế cơ sở dữ liệu



Hình 4. Mô hình cơ sở dữ liệu

Bảng Orders

Tên trường	Kiểu dữ liệu	Ghi chú
Id	INT	PK (Khóa chính)
OrderCode	NVARCHAR(MAX)	Mã đơn hàng duy nhất
Username	NVARCHAR(MAX)	FK → AspNetUsers(Username)
ProductID	BIGINT	Mã sản phẩm

Tên trường	Kiểu dữ liệu	Ghi chú
CreatedDate	DATETIME	Ngày tạo đơn
Status	INT	Trạng thái đơn hàng

Bảng 1. Bảng Orders

Bảng OrderDetails

Tên trường	Kiểu dữ liệu	Ghi chú
Id	INT	PK
Username	NVARCHAR(MAX)	FK → AspNetUsers(UserName)
OrderCode	NVARCHAR(MAX)	FK → Orders(OrderCode)
ProductId	BIGINT	FK → Products(Id)
Price	DECIMAL(18,2)	Giá tại thời điểm mua
Quantity	INT	Số lượng

Bảng 2. Bảng OrderDetails

Bảng Products

Tên trường	Kiểu dữ liệu	Ghi chú
Id	BIGINT	PK

Tên trường	Kiểu dữ liệu	Ghi chú
Name	NVARCHAR(MAX)	Tên sản phẩm
Description	NVARCHAR(MAX)	Mô tả
Price	DECIMAL(18,2)	Giá
BrandId	INT	FK → Brands(Id)
CategoryId	INT	FK → Categories(Id)
Slug	NVARCHAR(MAX)	Chuỗi định danh URL
Status	BIT	Trạng thái
Image	NVARCHAR(MAX)	Đường dẫn ảnh

Bảng 3. Bảng Products

Bảng Categories

Tên trường	Kiểu dữ liệu	Ghi chú
Id	INT	PK
Name	NVARCHAR(MAX)	Tên loại
Description	NVARCHAR(MAX)	Mô tả
Slug	NVARCHAR(MAX)	Chuỗi định danh

Tên trường	Kiểu dữ liệu	Ghi chú
Status	INT	Trạng thái

*Bảng 4. Bảng Categories***Bảng Brands**

Tên trường	Kiểu dữ liệu	Ghi chú
Id	INT	PK
Name	NVARCHAR(MAX)	Tên thương hiệu
Description	NVARCHAR(MAX)	Mô tả
Slug	NVARCHAR(MAX)	Chuỗi định danh
Status	INT	Trạng thái

*Bảng 5. Bảng Brands***Bảng AspNetUsers**

Tên trường	Kiểu dữ liệu	Ghi chú
Id	NVARCHAR(450)	PK
Occupation	NVARCHAR(MAX)	Nghề nghiệp
RoleId	NVARCHAR(MAX)	FK → AspNetRoles(Id)

Tên trường	Kiểu dữ liệu	Ghi chú
UserName	NVARCHAR(256)	Tên đăng nhập
NormalizedUserName	NVARCHAR(256)	Dùng để tìm kiếm nhanh
Email	NVARCHAR(256)	Email
NormalizedEmail	NVARCHAR(256)	Chuẩn hóa Email
EmailConfirmed	BIT	Xác thực email
PasswordHash	NVARCHAR(MAX)	Mật khẩu mã hóa
SecurityStamp	NVARCHAR(MAX)	Token bảo mật
ConcurrencyStamp	NVARCHAR(MAX)	Token kiểm tra xung đột
PhoneNumber	NVARCHAR(MAX)	SĐT
PhoneNumberConfirmed	BIT	Xác thực SĐT
TwoFactorEnabled	BIT	Xác thực 2 lớp
LockoutEnd	DATETIMEOFFSET	Hạn khóa tài khoản
LockoutEnabled	BIT	Cho phép khóa
AccessFailedCount	INT	Số lần đăng nhập sai

Bảng 6. Bảng *AspNetUsers*

BảngAspNetRoles

Tên trường	Kiểu dữ liệu	Ghi chú
Id	NVARCHAR(450)	PK
Name	NVARCHAR(256)	Tên vai trò
NormalizedName	NVARCHAR(256)	Tên chuẩn hóa

*Bảng 7. BảngAspNetRoles***BảngAspNetUserRoles**

Tên trường	Kiểu dữ liệu	Ghi chú
UserId	NVARCHAR(450)	PK, FK → AspNetUsers(Id)
RoleId	NVARCHAR(450)	PK, FK → AspNetRoles(Id)

*Bảng 8. BảngAspNetUserRoles***BảngAspNetUserClaims**

Tên trường	Kiểu dữ liệu	Ghi chú
Id	INT	PK

Tên trường	Kiểu dữ liệu	Ghi chú
UserId	NVARCHAR(450)	FK → AspNetUsers(Id)
ClaimType	NVARCHAR(MAX)	Kiểu claim

*Bảng 9. Bảng AspNetUserClaims***Bảng AspNetRoleClaims**

Tên trường	Kiểu dữ liệu	Ghi chú
Id	INT	PK
RoleId	NVARCHAR(450)	FK → AspNetRoles(Id)
ClaimType	NVARCHAR(MAX)	Quyền gán cho Role
ClaimValue	NVARCHAR(MAX)	Giá trị cho Role

*Bảng 10. Bảng AspNetRoleClaims***Bảng AspNetUserLogins**

Tên trường	Kiểu dữ liệu	Ghi chú
LoginProvider	NVARCHAR(450)	PK

Tên trường	Kiểu dữ liệu	Ghi chú
ProviderKey	NVARCHAR(450)	PK
ProviderDisplayName	NVARCHAR(MAX)	Tên provider
UserId	NVARCHAR(450)	FK → AspNetUsers(Id)

Bảng 11. Bảng AspNetUserLogins

Bảng AspNetUserTokens

Tên trường	Kiểu dữ liệu	Ghi chú
UserId	NVARCHAR(450)	PK, FK → AspNetUsers(Id)
LoginProvider	NVARCHAR(450)	PK
Name	NVARCHAR(450)	PK
Value	NVARCHAR(MAX)	Token value

Bảng 12. Bảng AspNetUserTokens

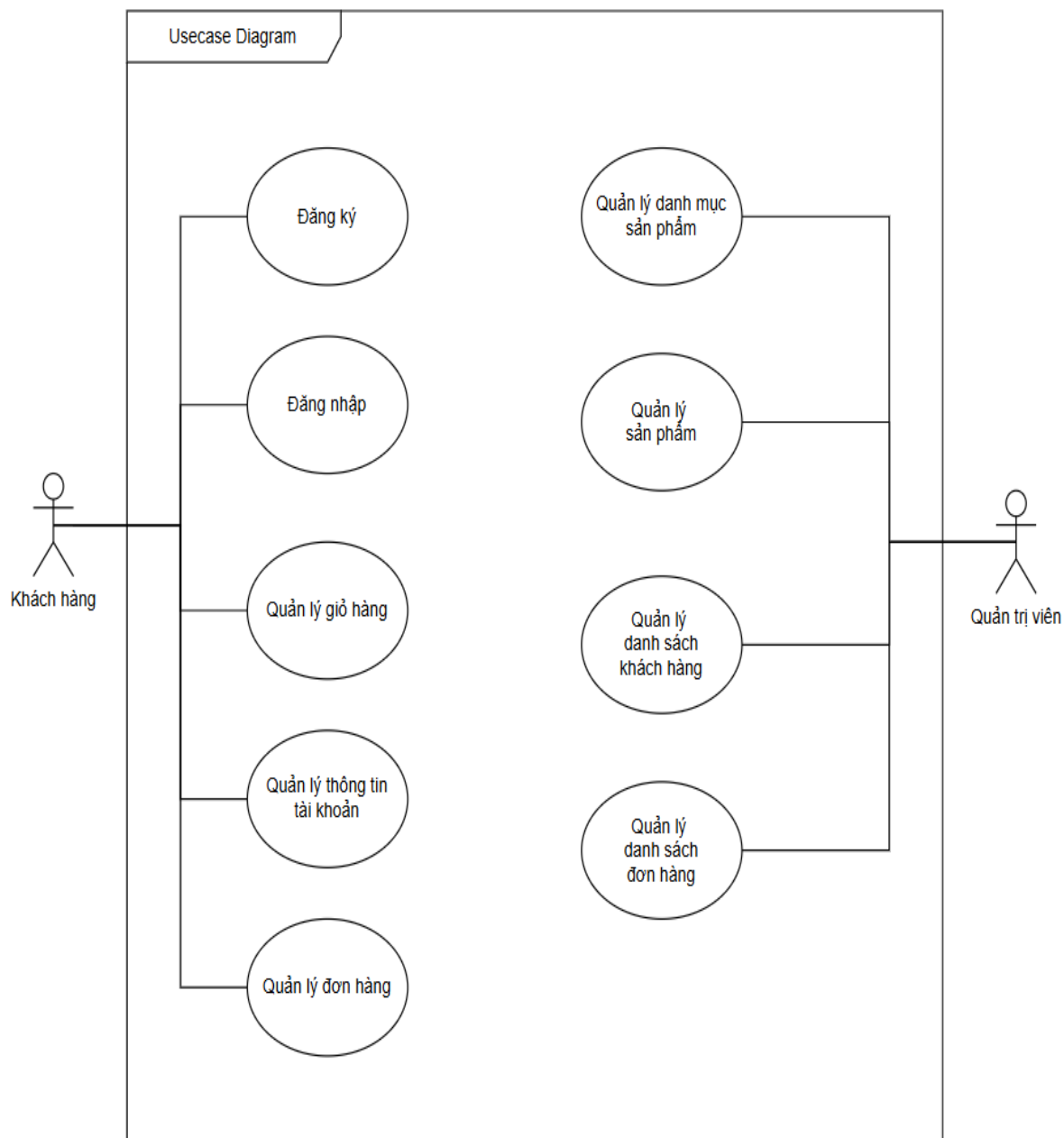
Bảng __EFMigrationsHistory

Tên trường	Kiểu dữ liệu	Ghi chú
MigrationId	NVARCHAR(150)	PK

Tên trường	Kiểu dữ liệu	Ghi chú
ProductVersion	NVARCHAR(32)	Phiên bản EF Core

Bảng 13. Bảng __EFMigrationsHistory

3.3. Thiết kế chức năng



Hình 5. Usecase diagram tổng quát

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

4.1. Cấu trúc dự án

Website bán giày là một trang web thương mại điện tử được phát triển theo mô hình MVC - chia tách rõ ràng phần hiển thị (View), xử lý nghiệp vụ (Controller) và dữ liệu (Model).

Các thành phần được tổ chức trong Solution Explorer giúp dễ quản lý, mở rộng và bảo trì hệ thống.

+ Controllers:

- Thư mục này chứa các lớp điều khiển (controller), chịu trách nhiệm xử lý logic nghiệp vụ và điều hướng luồng dữ liệu giữa Model và View.

- Một số controller chính gồm:

- HomeController.cs – quản lý trang chủ, chức năng đăng ký nhận tin (Subscribe), và các trang giới thiệu (About, Contact).
- ProductsController.cs – xử lý hiển thị danh sách, chi tiết sản phẩm.
- ShoppingCartController.cs – xử lý toàn bộ nghiệp vụ giỏ hàng, thanh toán và đặt hàng.
- AccountController.cs, ManageController.cs – quản lý tài khoản người dùng (đăng ký, đăng nhập, đổi mật khẩu, v.v.).
- NewsController.cs, MenuController.cs, ContactController.cs – quản lý tin tức, danh mục và liên hệ.

+ Models:

- Chứa các lớp đại diện cho dữ liệu và đối tượng nghiệp vụ của ứng dụng, ví dụ:

- ApplicationDbContext – lớp ngữ cảnh làm việc với Entity Framework, quản lý kết nối đến cơ sở dữ liệu.
- EF (Entity Framework Models) – chứa các model như Product, Order, OrderDetail, ShoppingCart, Subscribe,...

- Các lớp hỗ trợ như Payments, Common, Utils,... phục vụ cho các chức năng mở rộng như gửi email, thanh toán VNPay.

+ Views:

- Thư mục này chứa các tệp giao diện người dùng (HTML + Razor).

- Mỗi Controller có một thư mục con tương ứng, ví dụ:

- Views/Home/ chứa các View như Index.cshtml, About.cshtml, Contact.cshtml, Partial_Subscribe.cshtml.

- Views/ShoppingCart/ chứa Index.cshtml, CheckOut.cshtml, Partial_Item_Cart.cshtml,...

- Views/Products/ chứa giao diện hiển thị sản phẩm, chi tiết sản phẩm.

+ Content:

- Chứa các tệp tĩnh như CSS, hình ảnh, template email, v.v.

- Ví dụ: thư mục Content/templates lưu các file mẫu email HTML được dùng trong quá trình gửi mail xác nhận đơn hàng.

+ Scripts:

- Chứa các file JavaScript/jQuery phục vụ tương tác phía client — ví dụ thêm sản phẩm vào giỏ, cập nhật số lượng, xử lý Ajax.

+ Common:

- Thư mục tiện ích, chứa các lớp dùng chung trong toàn dự án, chẳng hạn như:

- Hàm SendMail() để gửi email.

- Hàm FormatNumber(), GetIpAddress() dùng trong nhiều Controller.

+ App_Start:

- Chứa các file cấu hình hệ thống khi ứng dụng khởi động:

- RouteConfig.cs – cấu hình định tuyến URL đến Controller tương ứng.

- FilterConfig.cs, BundleConfig.cs, Startup.Auth.cs – thiết lập các bộ lọc, gói tài nguyên, và cấu hình xác thực.

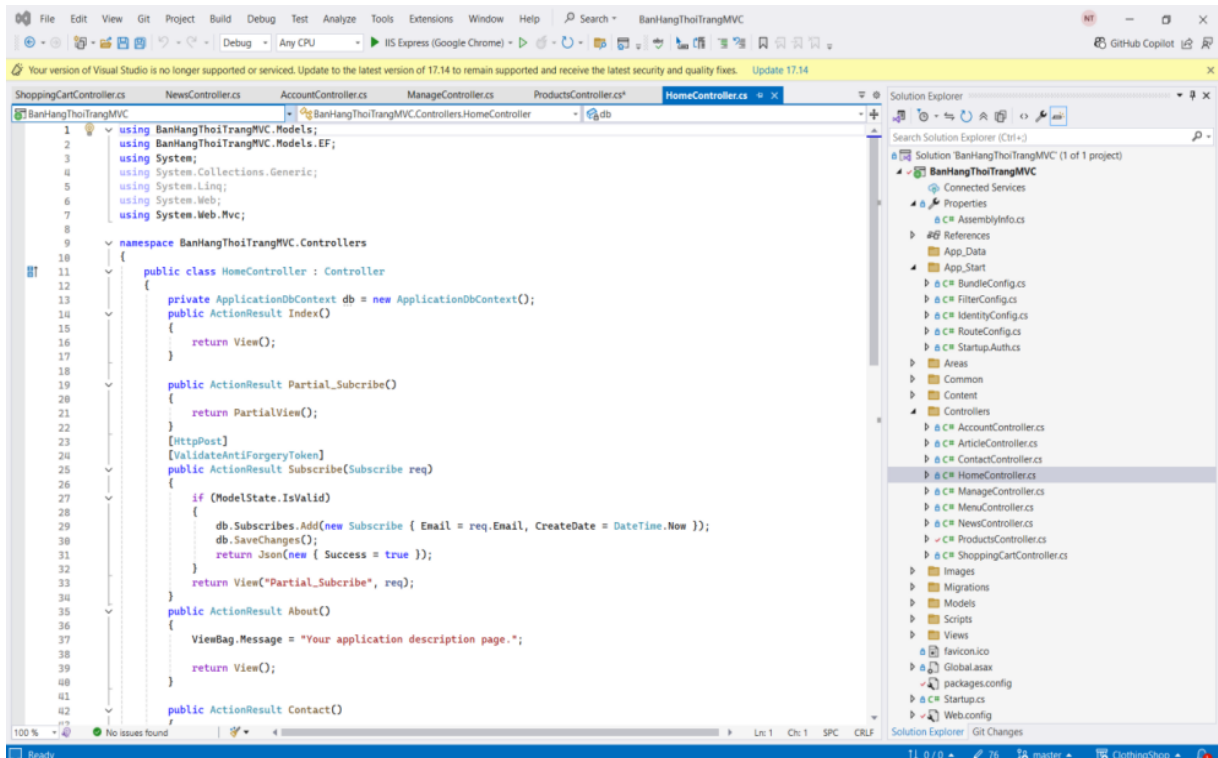
+ Global.asax

- Là tệp khởi động của ứng dụng, định nghĩa các sự kiện toàn cục như

Application_Start, Session_Start,...

+ Web.config

- Tập cấu hình chính của ứng dụng web, chứa thông tin như:
- Chuỗi kết nối đến cơ sở dữ liệu.
- Cấu hình xác thực, bảo mật.
- Cấu hình VNPay, Email Admin,...



Hình 6. Cấu trúc thư mục dự án

4.2. Mô tả chi tiết

4.2.1. Đăng nhập

- Bước 1: Người dùng nhập thông tin vào form đăng nhập.
- Bước 2: Hệ thống SignInManager.PasswordSignInAsync() để xác thực người dùng:
- Nếu đúng: đăng nhập thành công → chuyển về trang chủ.
- Nếu sai: hiển thị lỗi "Invalid login attempt."

```
//  
// GET: /Account/Login  
[AllowAnonymous]  
public ActionResult Login(string returnUrl)  
{  
    ViewBag.ReturnUrl = returnUrl;  
    return View();  
}  
  
//  
// POST: /Account/Login  
[HttpPost]  
[AllowAnonymous]  
[ValidateAntiForgeryToken]  
public async Task<ActionResult> Login(LoginViewModel model, string returnUrl)  
{  
    if (!ModelState.IsValid)  
    {  
        return View(model);  
    }  
  
    // This doesn't count login failures towards account lockout  
    // To enable password failures to trigger account lockout, change to shouldLockout: true  
    var result = await SignInManager.PasswordSignInAsync(model.UserName, model.Password, model.RememberMe, shouldLockout);  
    switch (result)  
    {  
        case SignInStatus.Success:  
            return RedirectToLocal(returnUrl);  
        case SignInStatus.LockedOut:  
            return View("Lockout");  
        case SignInStatus.RequiresVerification:  
            return RedirectToAction("SendCode", new { ReturnUrl = returnUrl, RememberMe = model.RememberMe });  
        case SignInStatus.Failure:  
        default:  
            ModelState.AddModelError("", "Invalid login attempt.");  
            return View(model);  
    }  
}
```

Hình 7. Xử lý chức năng đăng nhập

4.2.2. Đăng ký

- Bước 1: Người dùng nhập thông tin vào form đăng ký.
- Bước 2: Hệ thống kiểm tra tính hợp lệ của dữ liệu nhập.
- Bước 3: Tạo đối tượng ApplicationUser chứa thông tin người dùng
- Bước 4: Dùng UserManager.CreateAsync(user, password) để lưu vào database (bảngAspNetUsers). Gán quyền mặc định "Customer" cho người mới.
- Bước 5: Sau khi đăng ký xong → chuyển hướng người dùng về trang đăng nhập.

```
//
// GET: /Account/Register
[AllowAnonymous]
public ActionResult Register()
{
    return View();
}

//
// POST: /Account/Register
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Register(RegisterViewModel model)
{
    if (ModelState.IsValid)
    {
        var user = new ApplicationUser { UserName = model.Email, Fullname = model.FullName, Phone = model.Phone, Email = model.Email };
        var result = await UserManager.CreateAsync(user, model.Password);
        if (result.Succeeded)
        {
            if (user != null)
            {
                UserManager.AddToRole(user.Id, "Customer");
            }
            /*await SignInManager.SignInAsync(user, isPersistent:false, rememberBrowser:false);*/

            // For more information on how to enable account confirmation and password reset please visit https://go.microsoft.com/fwlink/?LinkID=320755
            // Send an email with this link
            // string code = await UserManager.GenerateEmailConfirmationTokenAsync(user.Id);
            // var callbackUrl = Url.Action("ConfirmEmail", "Account", new { userId = user.Id, code = code }, protocol: Request.Url.Scheme);
            // await UserManager.SendEmailAsync(user.Id, "Confirm your account", "Please confirm your account by clicking the following link: " + callbackUrl);

            return RedirectToAction("Login", "Account");
        }
        AddErrors(result);
    }

    // If we got this far, something failed, redisplay form
    return View(model);
}
```

Hình 8. Xử lý chức năng đăng ký

4.2.3. Hiện thị danh sách sản phẩm và tìm kiếm sản phẩm

- Bước 1: Khởi tạo đối tượng ApplicationDbContext db để truy xuất dữ liệu từ bảng Products

- Bước 2: Nhận tham số tìm kiếm từ người dùng. Phương thức Index nhận 2 tham số:

+ Searchtext: chuỗi tìm kiếm mà người dùng nhập vào (có thể rỗng).

+ id: tham số tùy chọn

- Bước 3: Hệ thống truy vấn bảng Products, sắp xếp theo ID giảm dần để hiển thị các sản phẩm mới nhất trước. Sau đó lọc ra những sản phẩm đang hoạt động (IsActive = true).

- Bước 4: Nếu người dùng nhập nội dung tìm kiếm, hệ thống tiếp tục lọc danh sách sản phẩm theo Tên sản phẩm (Title) chứa chuỗi tìm kiếm, hoặc bí danh (Alias) chứa chuỗi tìm kiếm.

- Bước 5: Sau khi lọc xong, hệ thống gửi danh sách sản phẩm (items) đến View tương ứng. Tại đây, giao diện người dùng sẽ hiển thị danh sách sản phẩm phù hợp với điều kiện tìm kiếm.

```
namespace BanHangThoiTrangMVC.Controllers
{
    public class ProductsController : Controller
    {
        private ApplicationDbContext db = new ApplicationDbContext();
        // GET: Products
        public ActionResult Index(string Searchtext, int? id)
        {
            /*var items = db.Products.Where(x => x.IsActive).Take(12).ToList();*/
            IEnumerable<Product> items = db.Products.OrderByDescending(x => x.Id);
            items = items.Where(x => x.IsActive).Take(12).ToList();

            if (!string.IsNullOrEmpty(Searchtext))
            {
                items = items.Where(x => x.Alias.Contains(Searchtext) || x.Title.Contains(Searchtext));
            }

            return View(items);
        }
    }
}
```

Hình 9. Xử lý chức năng hiển thị danh sách sản phẩm và tìm kiếm sản phẩm

4.2.4. Thêm sản phẩm vào giỏ hàng

- Bước 1: Nhận dữ liệu từ client

+ Phương thức AddToCart(int id, int quantity) nhận hai tham số:

- id: mã sản phẩm cần thêm vào giỏ.
- quantity: số lượng sản phẩm mà người dùng muốn thêm.

- Bước 2: Kiểm tra và lấy thông tin sản phẩm

+ Hệ thống truy vấn bảng Products trong ApplicationDbContext để lấy thông tin sản phẩm có Id = id.

+ Nếu sản phẩm tồn tại, tiến hành tạo đối tượng ShoppingCartItem chứa các thông tin cần thiết: ProductId, ProductName, CategoryName, Alias, Price, Quantity, ProductImg.

- Bước 3: Xác định giá bán

+ Nếu sản phẩm có PriceSale > 0, giá bán là PriceSale, ngược lại lấy Price gốc.

+ Tính TotalPrice = Quantity × Price.

- Bước 4: Thêm vào giỏ hàng

+ Nếu Session chưa có giỏ hàng (Session["Cart"] == null), khởi tạo mới đối

tượng ShoppingCart.

+ Gọi phương thức AddToCart(item, quantity) để thêm sản phẩm vào danh sách cart.Items.

+ Cập nhật lại Session["Cart"].

- Bước 5: Trả kết quả về client

+ Trả về Json thông báo thành công, kèm số lượng sản phẩm hiện có trong giỏ hàng.

```
[HttpPost]
[Authorize]
public ActionResult AddToCart(int id, int quantity)
{
    var code = new { Success = false, msg = "", code = -1, Count = 0 };
    var db = new ApplicationDbContext();
    var checkProduct = db.Products.FirstOrDefault(x => x.Id == id);
    ShoppingCart cart = (ShoppingCart)Session["Cart"];
    if (cart == null)
    {
        cart = new ShoppingCart();
    }
    ShoppingCartItem item = new ShoppingCartItem
    {
        ProductId = checkProduct.Id,
        ProductName = checkProduct.Title,
        CategoryName = checkProduct.ProductCategory.Title,
        Alias = checkProduct.Alias,
        Quantity = quantity
    };
    if (checkProduct.ProductImages.FirstOrDefault(x => x.IsDefault) != null)
    {
        item.ProductImg = checkProduct.ProductImages.FirstOrDefault(x => x.IsDefault).Image;
    }
    item.Price = checkProduct.Price;
    if (checkProduct.PriceSale > 0)
    {
        item.Price = (decimal)checkProduct.PriceSale;
    }
    item.TotalPrice = item.Quantity * item.Price;
    cart.AddToCart(item, quantity);
    Session["Cart"] = cart;
    code = new { Success = true, msg = "Thêm Sản Phẩm Vào Giỏ Hàng Thành Công!", code = 1, Count = cart.Items.Count };
    return Json(code);
}
```

Hình 10. Xử lý chức năng thêm sản phẩm vào giỏ hàng

4.2.5. Cập nhật số lượng trong giỏ hàng

- Bước 1: Nhận dữ liệu cập nhật

+ Phương thức Update(int id, int quantity) nhận:

- id: mã sản phẩm cần cập nhật.
- quantity: số lượng mới.

- Bước 2: Lấy giỏ hàng từ Session

- + Hệ thống truy xuất ShoppingCart từ Session["Cart"].
- Bước 3: Cập nhật số lượng
 - + Nếu giỏ hàng tồn tại, gọi phương thức cart.UpdateQuantity(id, quantity) để thay đổi số lượng của sản phẩm tương ứng.
- Bước 4: Trả phản hồi
 - + Trả về Json thông báo kết quả ("Cập nhật thành công" nếu hợp lệ).

```
[HttpPost]
public ActionResult Update(int id, int quantity)
{
    ShoppingCart cart = (ShoppingCart)Session["Cart"];
    if (cart != null)
    {
        cart.UpdateQuantity(id, quantity);
        return Json(new { Success = true, msg = "Cập Nhật Thành Công" });
    }
    return Json(new { Success = false });
}
```

Hình 11. Xử lý chức năng cập nhật số lượng trong giỏ hàng

4.2.6. Xóa sản phẩm trong giỏ hàng

- Bước 1: Nhận mã sản phẩm
 - + Phương thức Delete(int id) nhận tham số id là mã sản phẩm cần xóa.
- Bước 2: Lấy giỏ hàng từ Session
 - + Hệ thống kiểm tra Session["Cart"].
 - + Nếu giỏ hàng tồn tại, tìm sản phẩm có ProductId = id trong cart.Items.
- Bước 3: Xóa sản phẩm
 - + Nếu sản phẩm tồn tại, gọi cart.Remove(id) để xóa khỏi danh sách.
 - + Cập nhật lại Session.
- Bước 4: Trả phản hồi
 - + Trả về Json thông báo thành công và số lượng sản phẩm còn lại trong giỏ.

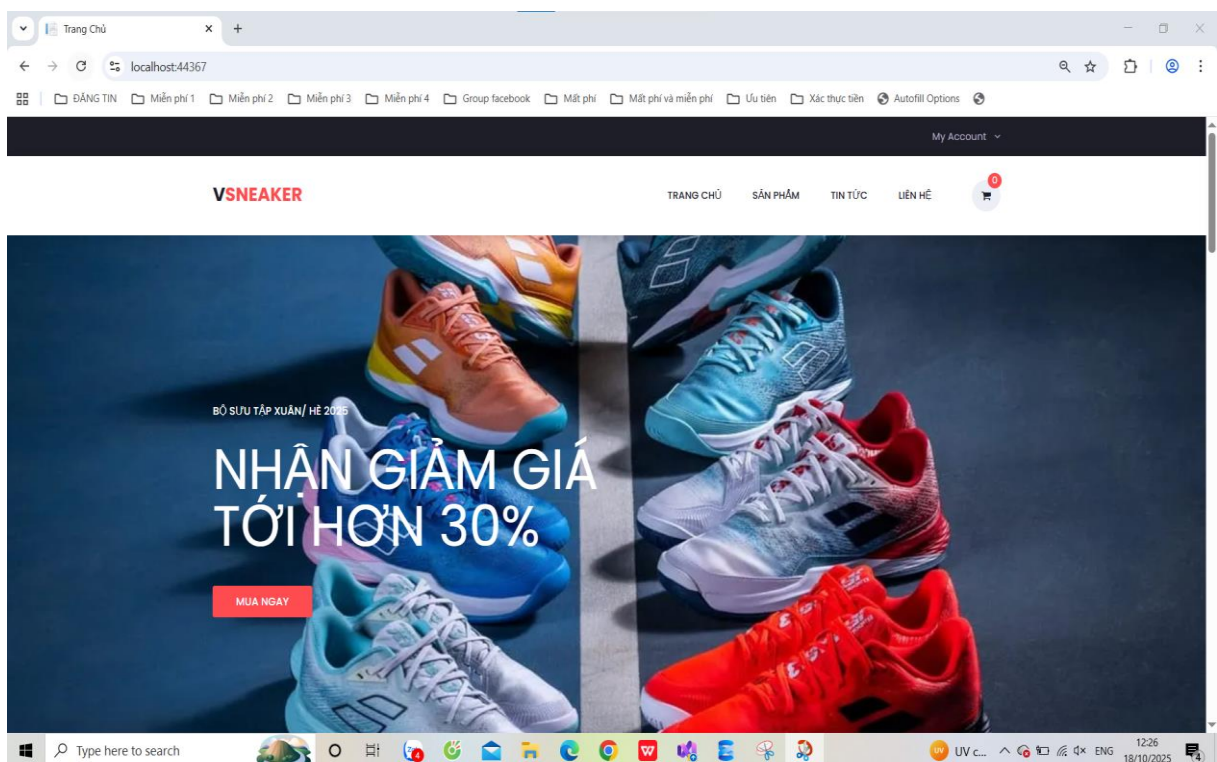
```

[HttpPost]
public ActionResult Delete(int id)
{
    var code = new { Success = false, msg = "", code = -1, Count = 0 };
    ShoppingCart cart = (ShoppingCart)Session["Cart"];
    if (cart != null)
    {
        var checkProduct = cart.Items.FirstOrDefault(x => x.ProductId == id);
        if (checkProduct != null)
        {
            cart.Remove(id);
            code = new { Success = true, msg = "", code = 1, Count = cart.Items.Count };
        }
    }
    return Json(code);
}

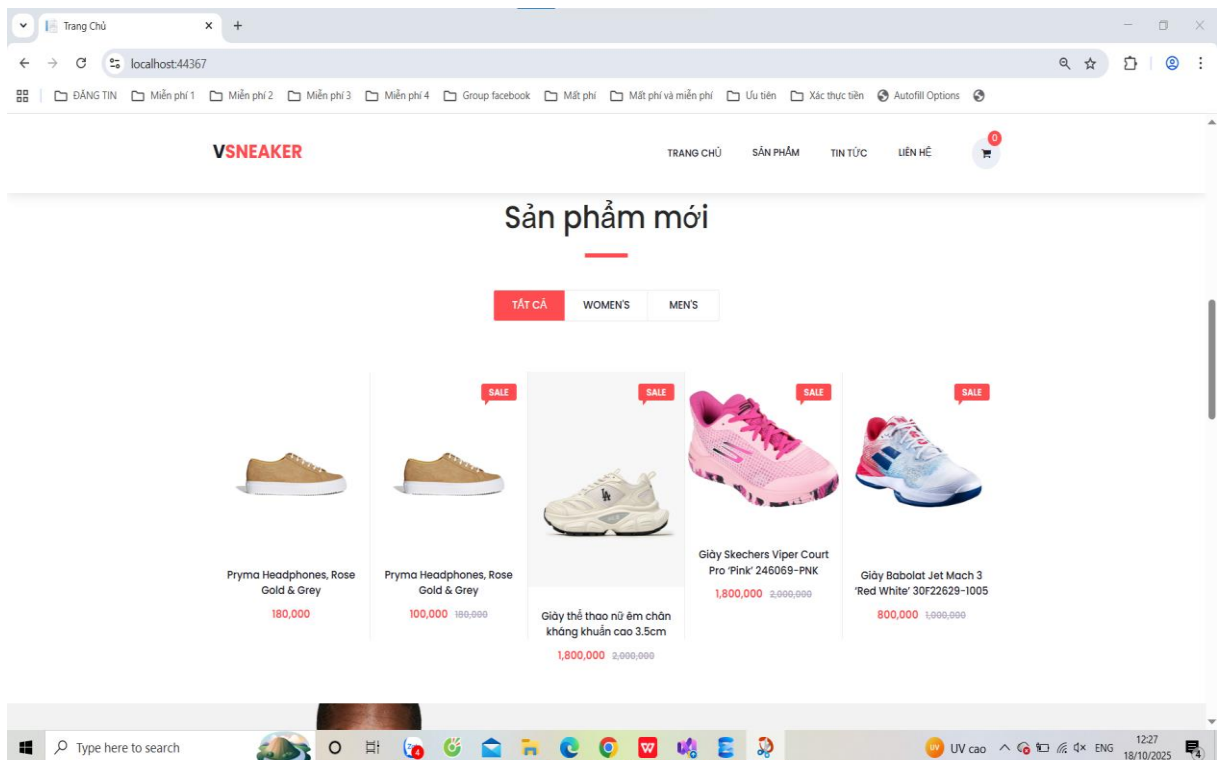
```

Hình 12. Xử lý chức năng xóa sản phẩm trong giỏ hàng

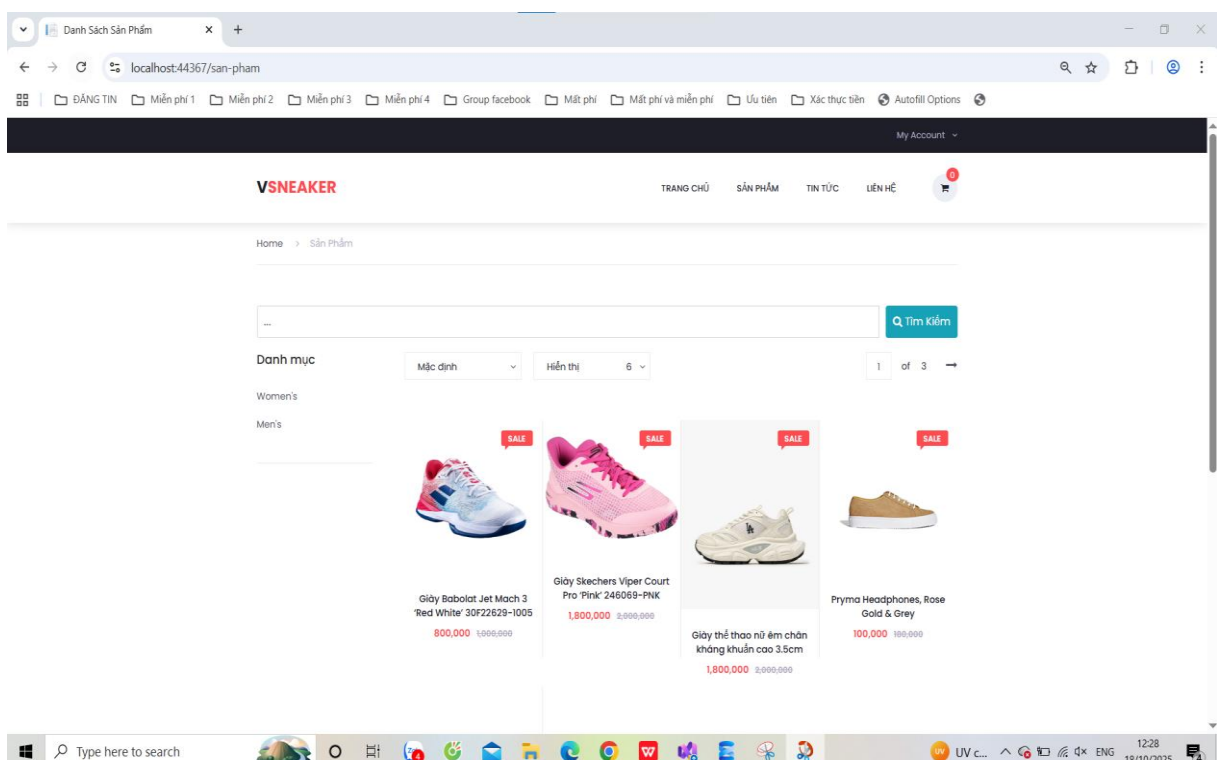
4.3. Giao diện chính



Hình 13. Màn hình Trang chủ



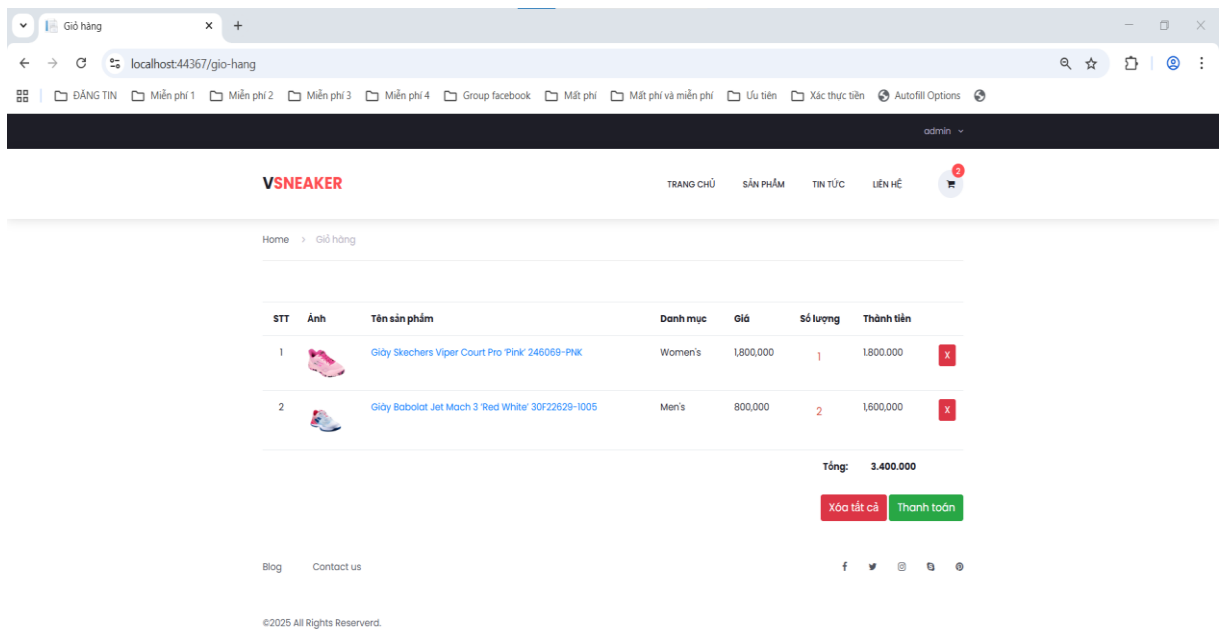
Hình 14. Màn hình Trang chủ - Sản phẩm mới



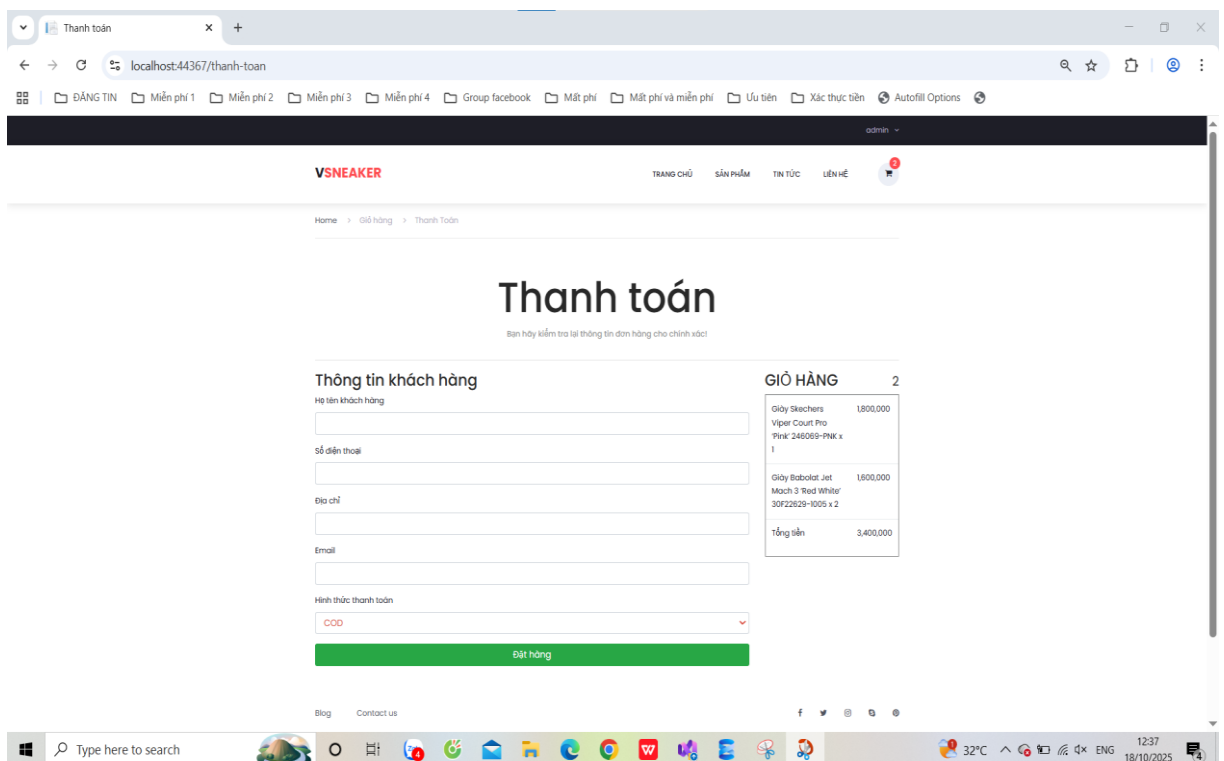
Hình 15. Màn hình Danh mục sản phẩm

Hình 16. Màn hình đăng nhập

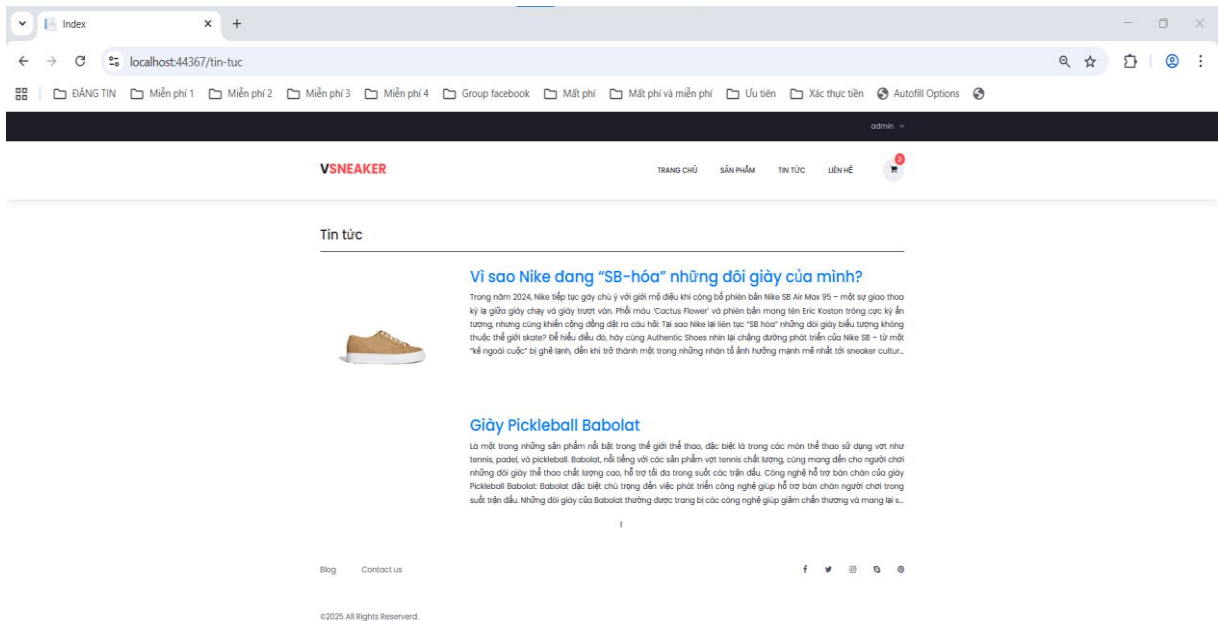
Hình 17. Màn hình Đăng ký



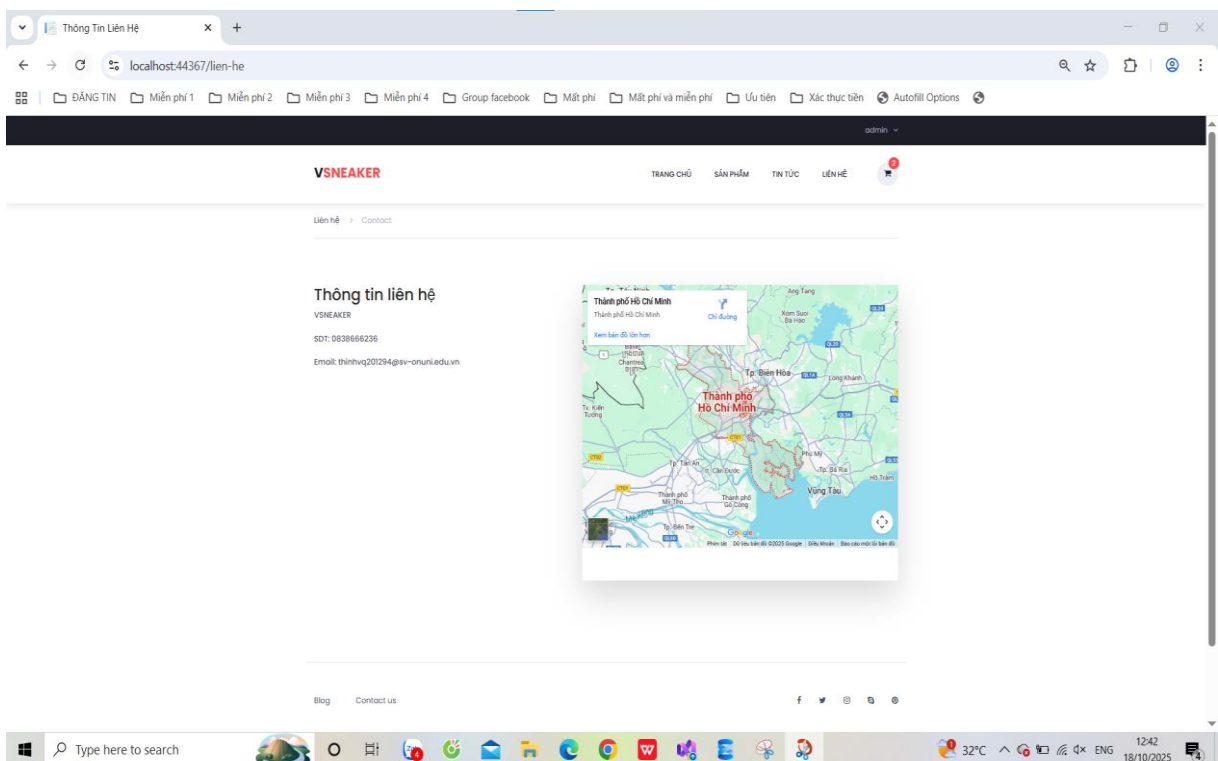
Hình 18. Màn hình Giỏ hàng



Hình 19. Màn hình điền thông tin giao hàng



Hình 20. Màn hình Blog tin tức



Hình 21. Màn hình thông tin liên hệ

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Tóm tắt các chức năng đã hoàn thành

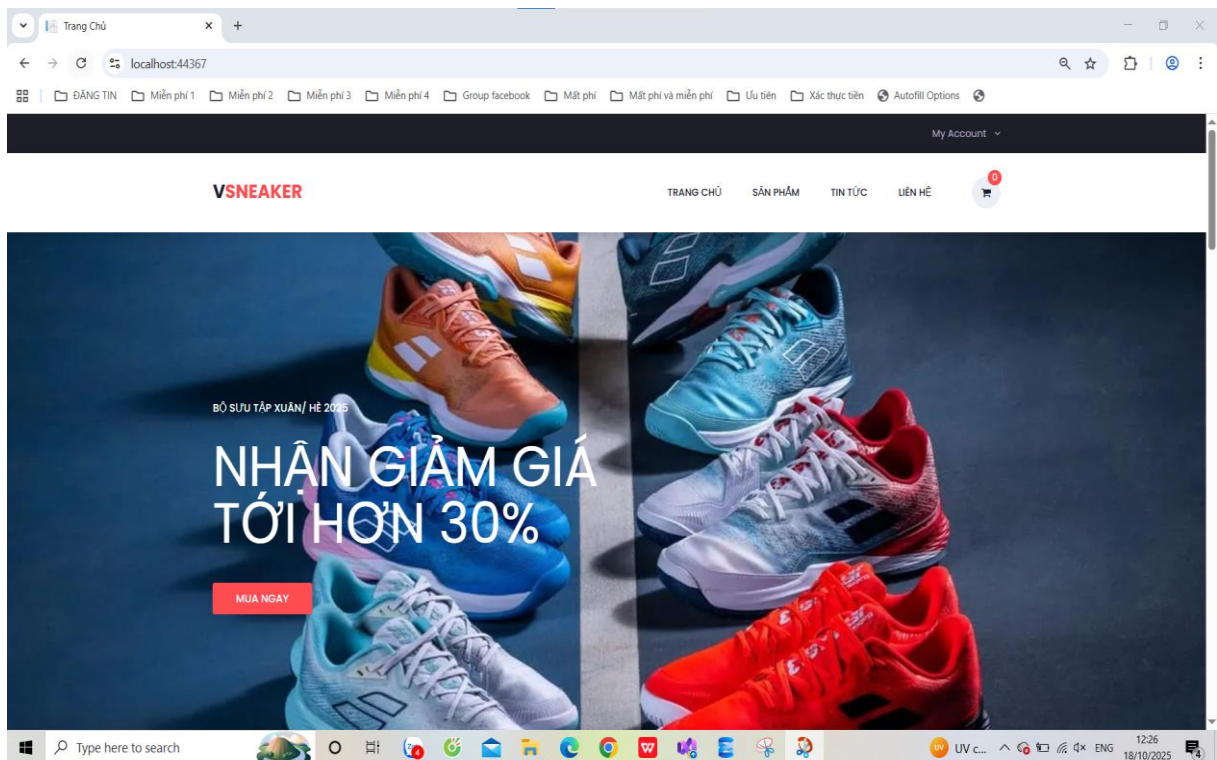
Sau quá trình nghiên cứu, phân tích, thiết kế và triển khai, hệ thống Website Bán Giày trực tuyến đã được xây dựng và hoàn thiện với các chức năng cơ bản đáp ứng được mục tiêu đề ra ban đầu.

Cụ thể, các chức năng chính đã được thực hiện gồm:

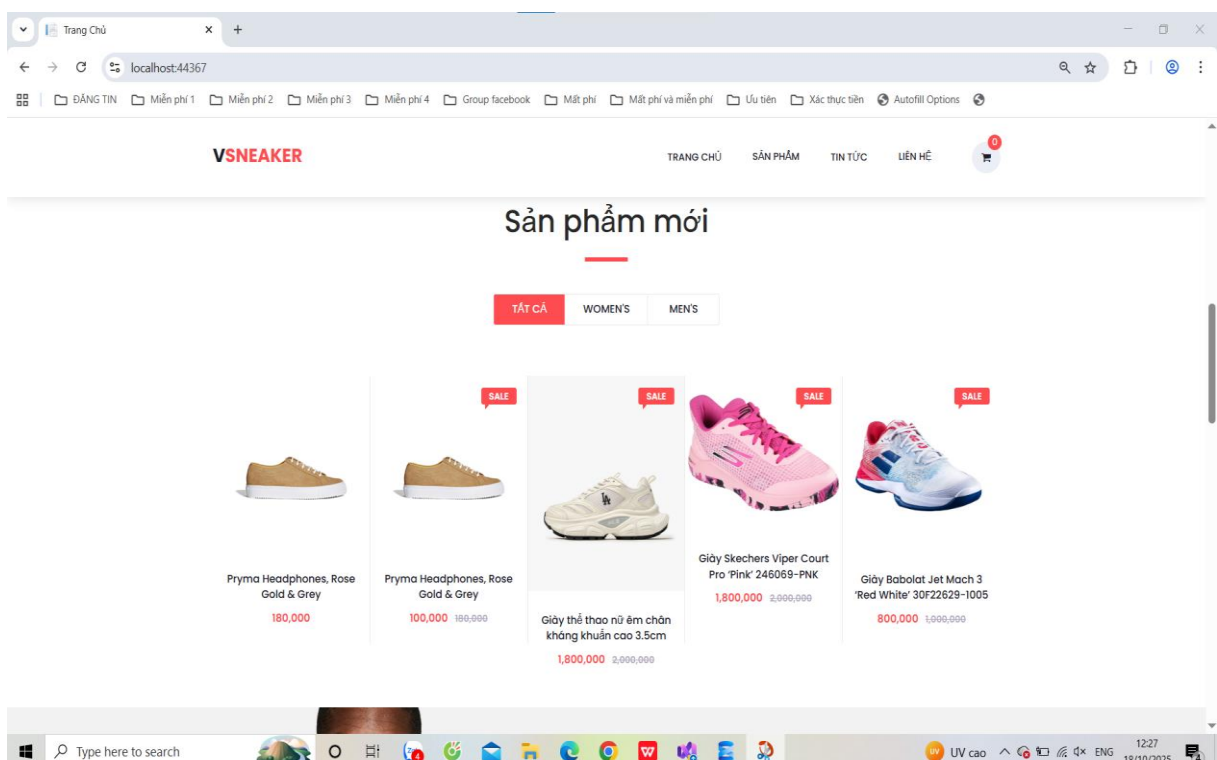
- Chức năng đăng ký tài khoản: Người dùng có thể tạo tài khoản cá nhân bằng cách cung cấp thông tin cần thiết như họ tên, email, số điện thoại và mật khẩu. Dữ liệu được lưu trữ an toàn trong cơ sở dữ liệu SQL Server.
- Chức năng đăng nhập/đăng xuất: Hệ thống hỗ trợ người dùng đăng nhập vào tài khoản để sử dụng các chức năng như mua hàng, xem lịch sử đơn hàng. Cơ chế xác thực đảm bảo chỉ người dùng hợp lệ mới được truy cập vào các chức năng riêng tư.
- Chức năng tìm kiếm sản phẩm: Website cho phép người dùng tra cứu sản phẩm theo từ khóa, loại giày hoặc mức giá. Kết quả được hiển thị nhanh chóng và chính xác, giúp người dùng dễ dàng tìm được sản phẩm mong muốn.
- Chức năng giỏ hàng: Người dùng có thể thêm sản phẩm vào giỏ, thay đổi số lượng, hoặc xóa sản phẩm không mong muốn. Dữ liệu giỏ hàng được lưu tạm thời và đồng bộ khi người dùng tiến hành đặt hàng.
- Chức năng đặt hàng: Sau khi xác nhận thông tin giỏ hàng, người dùng có thể nhập địa chỉ giao hàng, chọn phương thức thanh toán (hiện tại là thanh toán khi nhận hàng – COD) và xác nhận đơn hàng. Hệ thống ghi nhận đơn, cập nhật cơ sở dữ liệu, đồng thời cho phép người dùng theo dõi trạng thái đơn hàng trong tài khoản.

Bên cạnh đó, hệ thống quản trị cũng được phát triển với các chức năng cơ bản như quản lý sản phẩm, đơn hàng, và thông tin người dùng, giúp đảm bảo hoạt động của website diễn ra ổn định, thống nhất.

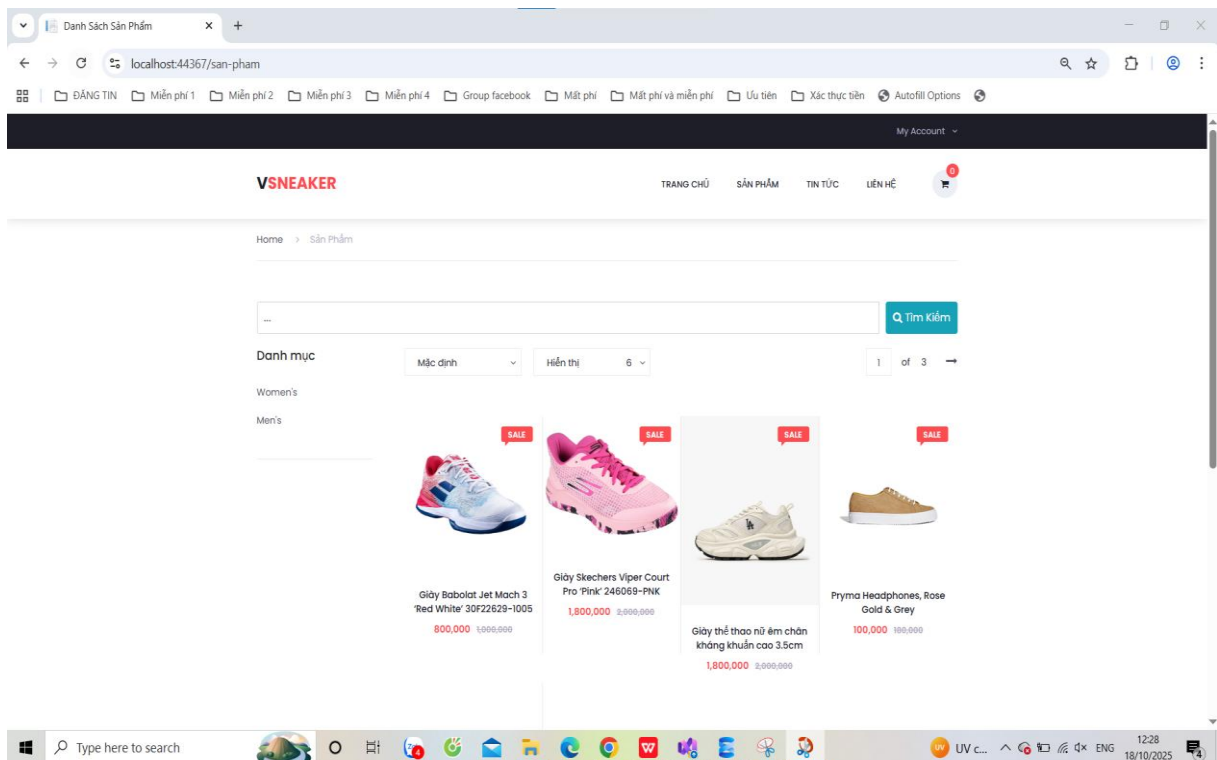
5.2. Demo kết quả đạt được



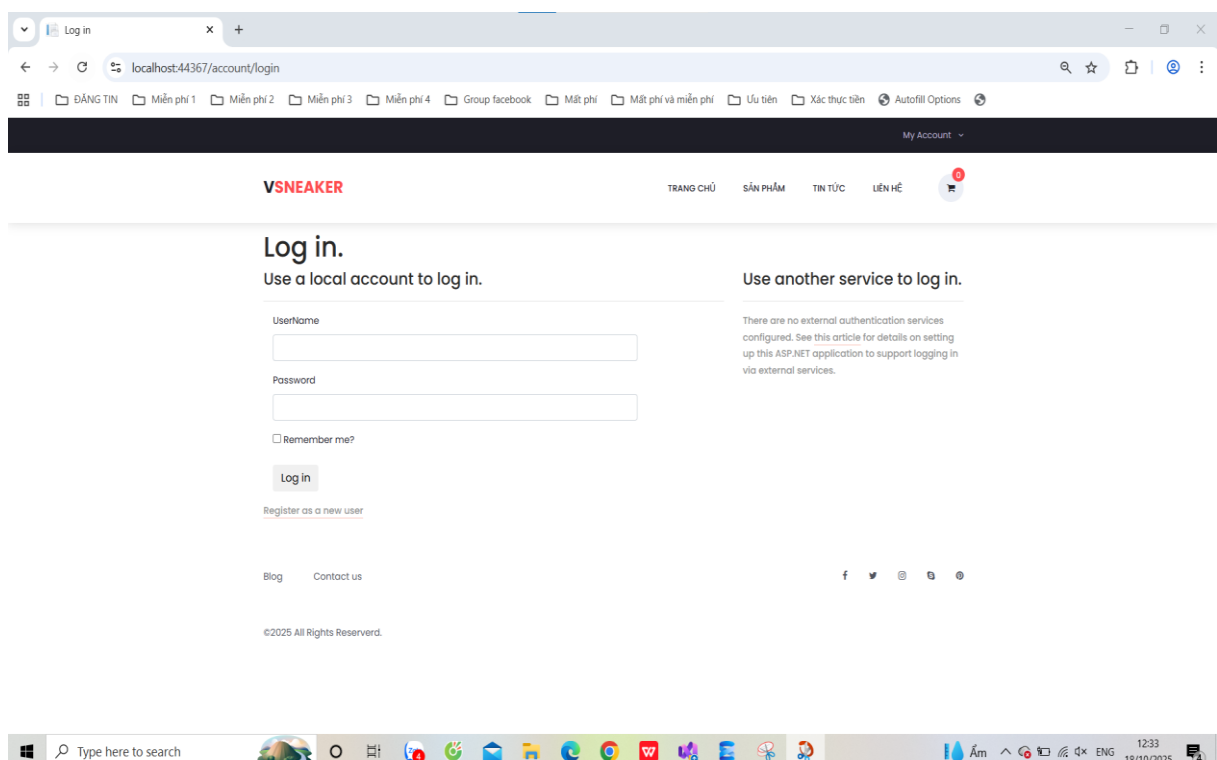
Hình 22. Màn hình Trang chủ



Hình 23. Màn hình Trang chủ - Sản phẩm mới



Hình 24. Màn hình Danh mục sản phẩm



Hình 25. Màn hình đăng nhập

Register.

Create a new account.

Email

FullName

Phone

Password

Confirm password

Register

Blog Contact us

f t i s e

Hình 26. Màn hình Đăng ký

Home > Giỏ hàng

STT	Ảnh	Tên sản phẩm	Danh mục	Giá	Số lượng	Thành tiền
1		Giày Skechers Viper Court Pro 'Pink' 246069-PNK	Women's	1,800,000	1	1,800,000
2		Giày Babolat Jet Mach 3 'Red White' 30F22629-1005	Men's	800,000	2	1,600,000

Tổng: 3.400.000

Xóa tất cả Thanh toán

Blog Contact us

f t i s e

©2025 All Rights Reserved.

Hình 27. Màn hình Giỏ hàng

Thanh toán

Bạn hãy kiểm tra lại thông tin đơn hàng cho chính xác!

Thông tin khách hàng

Họ tên khách hàng

Số điện thoại

Địa chỉ

Email

Hình thức thanh toán

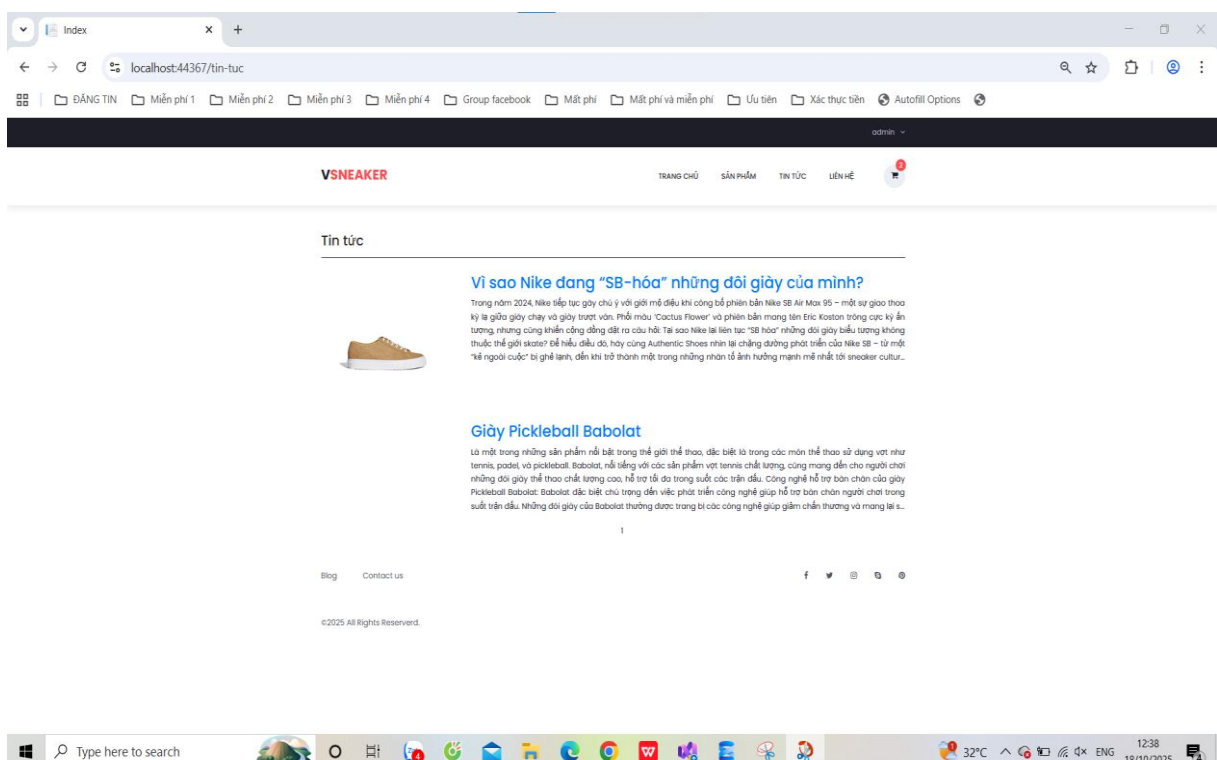
COD

Đặt hàng

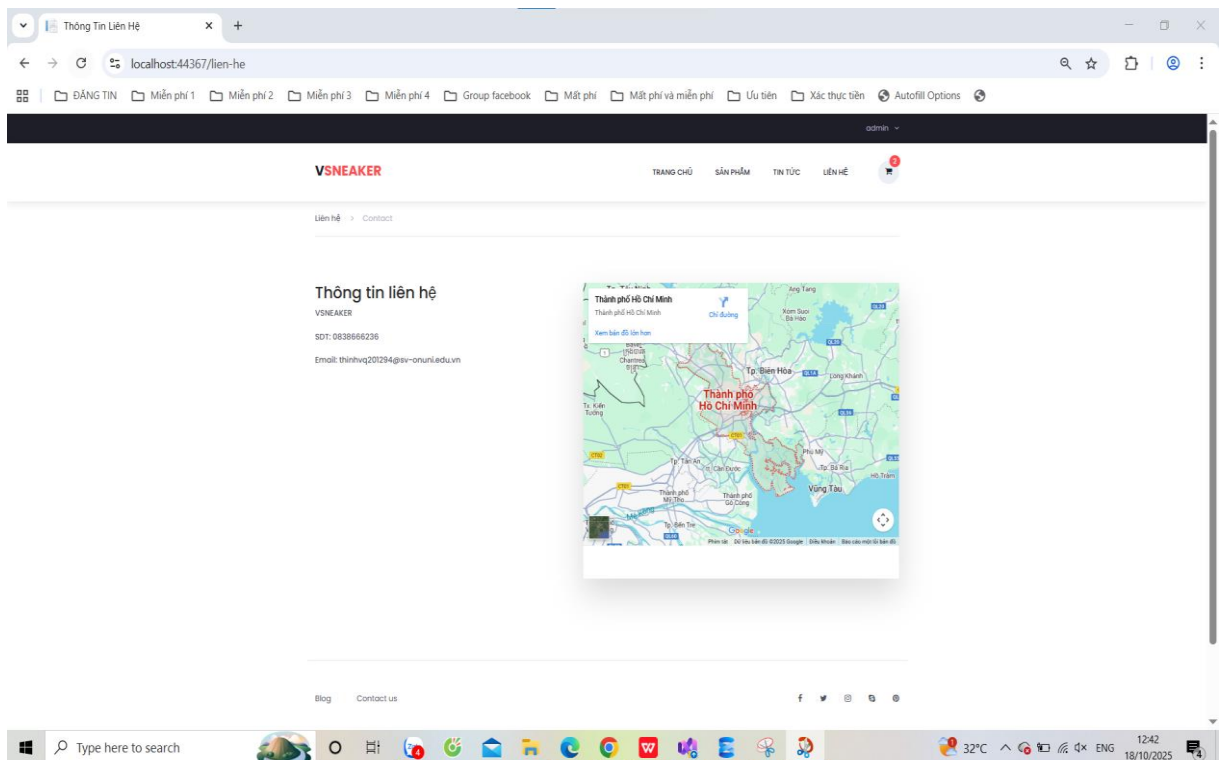
GIỎ HÀNG 2

Giày Skechers Viper Court Pro Pink 240069-PRK x 1	1,800,000
Giày Babolat Jet Moch 3 Red White 30F22029-1005 x 2	1,600,000
Tổng tiền	3,400,000

Hình 28. Màn hình điền thông tin giao hàng



Hình 29. Màn hình Blog tin tức



Hình 30. Màn hình thông tin liên hệ

5.3. Đánh giá, so sánh mục tiêu ban đầu

Khi so sánh kết quả đạt được với mục tiêu ban đầu được đề ra ở chương 1, có thể nhận thấy rằng hầu hết các yêu cầu chính đã được đáp ứng đầy đủ.

Về mặt chức năng, các tính năng quan trọng như đăng ký, đăng nhập, hiển thị và tìm kiếm sản phẩm, giỏ hàng, đặt hàng, quản lý dữ liệu đã được triển khai thành công và hoạt động ổn định. Hệ thống cơ bản đáp ứng được quy trình bán hàng trực tuyến tiêu chuẩn.

Về mặt kỹ thuật, website được phát triển trên nền tảng ASP.NET Framework kết hợp với SQL Server, đúng theo công nghệ dự kiến. Kiến trúc phân tầng (3-layer) được áp dụng hợp lý, giúp mã nguồn rõ ràng, dễ mở rộng.

Về mặt giao diện và trải nghiệm người dùng, giao diện website được thiết kế bằng HTML, CSS và Bootstrap, đảm bảo thân thiện, dễ sử dụng và tương thích trên nhiều thiết bị.

Về tính ổn định, website hoạt động ổn định trong quá trình kiểm thử với dữ liệu thực nghiệm. Các thao tác như tìm kiếm, thêm giỏ hàng, đặt hàng được xử lý nhanh và chính xác.

Tuy nhiên, vẫn còn một số tính năng mở rộng và yếu tố nâng cao chưa được thực hiện đầy đủ do giới hạn về thời gian và phạm vi của đề án, như tích hợp thanh toán trực tuyến, hệ thống quản lý kho nâng cao hoặc tính năng đánh giá sản phẩm.

5.4. Hạn chế và hướng phát triển

Mặc dù hệ thống đã cơ bản hoàn thiện, song vẫn còn nhiều tiềm năng để nâng cấp và mở rộng nhằm đáp ứng tốt hơn nhu cầu thực tế. Trong tương lai, website có thể được tích hợp thêm nhiều tính năng quan trọng như: cổng thanh toán trực tuyến (VNPay, MoMo, PayPal...) để tạo sự thuận tiện cho khách hàng; hệ thống gợi ý sản phẩm dựa trên trí tuệ nhân tạo nhằm cá nhân hóa trải nghiệm mua sắm; tích hợp đánh giá, bình luận của khách hàng để tăng tính tương tác và tin cậy cho sản phẩm.

Bên cạnh đó, giao diện người dùng cần được tối ưu hóa hơn nữa theo hướng thiết kế đáp ứng (responsive design) để phù hợp với nhiều thiết bị, đặc biệt là điện thoại di động và máy tính bảng. Việc nâng cao các cơ chế bảo mật như mã hóa dữ liệu, chống tấn công SQL Injection, XSS và CSRF cũng cần được chú trọng nhằm bảo vệ thông tin cá nhân và giao dịch của người dùng. Ngoài ra, hệ thống có thể mở rộng thêm tính năng quản lý khuyến mãi, mã giảm giá, tích điểm thành viên để tăng trải nghiệm khách hàng và hỗ trợ hoạt động marketing.

Trong định hướng lâu dài, việc triển khai hệ thống trên nền tảng điện toán đám mây (Azure, AWS) sẽ giúp cải thiện khả năng mở rộng, tăng tính sẵn sàng và tối ưu chi phí vận hành. Đồng thời, phát triển thêm ứng dụng di động (Android, iOS) đồng bộ với website cũng sẽ là một hướng phát triển tiềm năng, giúp mở rộng kênh tiếp cận khách hàng và đáp ứng xu hướng thương mại điện tử hiện đại.

5.5. Kết luận

Nhìn chung, đề án đã đạt được 80–90% so với kế hoạch đặt ra ban đầu, không chỉ là một website bán giày trực tuyến có tính ứng dụng thực tiễn mà còn là một minh chứng cho khả năng kết hợp giữa lý thuyết đã học và thực tiễn triển khai, đồng thời thể hiện được khả năng áp dụng tổng hợp kiến thức lập trình web, thiết kế cơ sở dữ liệu và triển khai website thực tế, góp phần giúp các cửa hàng, doanh nghiệp vừa và nhỏ mở ra nhiều hướng phát triển tiềm năng trong tương lai.

DANH MỤC TÀI LIỆU THAM KHẢO

[1]. A. Freeman, *Pro ASP.NET Core 6: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages*, 10th ed., Apress, New York, 2022.

[2]. Microsoft Corporation, “ASP.NET Documentation,” *Microsoft Learn*, 2024. [Trực tuyến]. Có tại: <https://learn.microsoft.com/en-us/aspnet/>

[Truy cập: 06/10/2025].

[3]. Microsoft Corporation, “ADO.NET Overview,” *Microsoft Learn*, 2024. [Trực tuyến]. Có tại: <https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/>

[Truy cập: 06/10/2025].