

ỦY BAN NHÂN DÂN TP HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



Họ và tên sinh viên : Võ Hải Hà

BÁO CÁO
THỰC TẬP TỐT NGHIỆP

Công ty thực tập : Viện Y dược học Dân tộc
Chuyên gia hướng dẫn : Chế Hồng Trung
Giảng viên hướng dẫn : Nguyễn Trung Tín

TP. Hồ Chí Minh, tháng năm 24

ỦY BAN NHÂN DÂN TP HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



Họ và tên sinh viên : Võ Hải Hà

BÁO CÁO
THỰC TẬP TỐT NGHIỆP

Công ty thực tập : Viện Y dược học Dân tộc

Chuyên gia hướng dẫn : Chế Hồng Trung

Giảng viên hướng dẫn : Nguyễn Trung Tín

TP. Hồ Chí Minh, tháng 10 năm 2024

MỤC LỤC

MỤC LỤC.....	2
DANH MỤC HÌNH ẢNH.....	4
NHẬN XÉT CỦA CHUYÊN GIA DOANH NGHIỆP	5
NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN	6
LỜI MỞ ĐẦU	1
CHƯƠNG 1. GIỚI THIỆU	2
1.1. Giới thiệu về Viện Y dược học Dân tộc.....	2
1.2. Giới thiệu về phòng Công nghệ thông tin	5
1.3. Sơ đồ tổ chức Viện Y dược học Dân tộc	6
1.4. Nhiệm vụ thực tập	6
1.5. Kết luận chương 1.....	6
CHƯƠNG 2. TỔNG QUAN VỀ ĐỀ TÀI THỰC TẬP	8
2.1. Xác định đề tài nghiên cứu	8
2.2. Đối tượng nghiên cứu	8
2.3. Phạm vi nghiên cứu và ứng dụng.....	10
CHƯƠNG 3: TỔNG QUAN VỀ PHẦN MỀM.....	11
3.1. Yêu cầu về phần mềm	11
3.2. Đối tượng người dùng	11
CHƯƠNG 4: XÂY DỰNG PHẦN MỀM	12
4.1. Xây dựng giao diện phần mềm	12
4.2. Xây dựng model dữ liệu	53
4.3. Xây dựng chức năng phần mềm.....	57
4.4. Cấu trúc dự án	81
CHƯƠNG 5: KẾT QUẢ THỰC TẬP	82
5.1. Phát triển kỹ năng chuyên môn.....	82
5.2. Hiểu biết về ngành y tế.....	82
5.3. Kinh nghiệm làm việc nhóm	82
5.4. Tư duy phát triển bản thân.....	83

BẢNG GHI NHẬN KẾT QUẢ THỰC TẬP HÀNG TUẦN	84
BẢNG ĐÁNH GIÁ QUÁ TRÌNH THỰC TẬP TỐT NGHIỆP	86
PHIẾU ĐÁNH GIÁ KẾT QUẢ THỰC TẬP TỐT NGHIỆP	87
CHƯƠNG 6. KẾT LUẬN VÀ KIẾN NGHỊ.....	88
6.1. Kết luận	88
6.2. Kiến nghị	88
TÀI LIỆU THAM KHẢO	89

DANH MỤC HÌNH ẢNH

Hình 1: Logo Viện Y dược học Dân tộc	5
Hình 2: Sơ đồ Viện Y dược học Dân tộc	6
Hình 3: Giao diện trang chủ	12
Hình 4: Giao diện trang chủ admin	12
Hình 5: Giao diện Login.....	13
Hình 6: Giao diện đăng kí	14
Hình 7: Giao diện kích hoạt tài khoản	15
Hình 8: Giao diện danh mục sản phẩm	16
Hình 9: Giao diện chi tiết sản phẩm.....	17
Hình 10: Giao diện giỏ hàng trống.....	19
Hình 11: Giao diện giỏ hàng khi có hàng	20
Hình 12: Giao diện danh sách đơn hàng	22
Hình 13: Giao diện chi tiết đơn hàng	24
Hình 14: Giao diện danh sách tài khoản	26
Hình 15: Giao diện chỉnh sửa tài khoản.....	28
Hình 16: Giao diện chi tiết tài khoản	30
Hình 17: Giao diện tạo tài khoản mới	32
Hình 18: Giao diện danh sách sản phẩm.....	34
Hình 19: Giao diện tạo sản phẩm mới.....	36
Hình 20: Giao diện chỉnh sửa sản phẩm	38
Hình 21: Giao diện chi tiết sản phẩm.....	41
Hình 22: Giao diện xác nhận xoá sản phẩm.....	43
Hình 23: Giao diện danh sách danh mục	46
Hình 24: Giao diện chỉnh sửa danh mục	48
Hình 25: Giao diện chi tiết danh mục	49
Hình 26: Giao diện khởi tạo danh mục mới.....	51
Hình 27: Giao diện xác nhận xoá danh mục	52

[illegible]

[illegible]

LỜI MỞ ĐẦU

Công nghệ thông tin đang trở thành sức mạnh thúc đẩy phát triển đáng kể trong mọi lĩnh vực trên toàn cầu, và Việt Nam không thể ngồi ngoài cuộc. Đất nước chúng ta đã bắt đầu đầu tư đầy quyết tâm vào ngành công nghệ thông tin, và sự hiện diện của nó đã trở thành hạt nhân không thể thiếu cho các cơ quan, doanh nghiệp.

Trong thế giới công nghiệp ngày nay, bất kể lĩnh vực hoạt động của cơ quan, doanh nghiệp là gì, công nghệ thông tin đóng vai trò cốt lõi trong việc cải thiện hiệu quả hoạt động và tối ưu hóa quy trình công việc.

Mặc dù công nghệ thông tin là một lĩnh vực đang còn trẻ, nhưng tiềm năng của nó tại Việt Nam là rất lớn. Sử dụng công nghệ thông tin hiệu quả sẽ giúp việc quản lý trở nên dễ dàng hơn, đồng thời tiết kiệm được nhiều công sức và chi phí lao động quý báu. Với tầm nhìn và chiến lược phù hợp, chúng ta sẽ tiếp tục xây dựng môi trường thuận lợi để công nghệ thông tin tỏa sáng và thúc đẩy đất nước phát triển vượt bậc. Việc tận dụng và phát triển nguồn lực này là một bước đi đúng hướng, hướng tới một tương lai thịnh vượng và hiện đại cho Việt Nam.

Với thời gian học tập ngắn, bản thân lại vừa học, vừa hoàn thành công việc của cơ quan, nên báo cáo khó tránh khỏi những thiếu sót, rất mong sự góp ý của quý giảng viên/chuyên gia để tiểu luận được hoàn thiện hơn.

Xin trân trọng cảm ơn

CHƯƠNG 1. GIỚI THIỆU

1.1. GIỚI THIỆU VỀ VIỆN Y DƯỢC HỌC DÂN TỘC

Sau giải phóng, cách mạng Việt Nam chuyển sang giai đoạn mới - giai đoạn cả nước đi lên chủ nghĩa xã hội, khắc phục từng bước những hậu quả nặng nề của chiến tranh; bước đầu phân bố lại lực lượng lao động xã hội; tăng cường xây dựng cơ sở vật chất - kỹ thuật của nền kinh tế quốc dân. Những đoàn cán bộ y tế dưới sự chỉ đạo của Bộ Y tế Xã hội và Thương binh từ Trung ương Cục miền Nam - đã được phân công tiếp quản các cơ sở hạ tầng tại thành phố Hồ Chí Minh để thực hiện công tác y tế thời kỳ mới.

Bộ Y tế Xã hội và Thương binh định hướng xây dựng cơ sở Đông y thứ 2 tại miền Nam, điều động đội ngũ cán bộ được đào tạo về y học cổ truyền vào tiếp quản. Đó là lý do ra đời thành lập Viện Y dược Học Dân Tộc (gọi tắt là Viện) tại Quyết định số 43/YTXHTB-TC vào ngày 24 tháng 12 năm 1975 do BS. Dương Quỳnh Hoa, Bộ trưởng Bộ Y tế Xã hội và Thương binh ký ban hành với địa chỉ chính hiện nay tại số 273 - 275 đường Nguyễn Văn Trỗi, phường 10, quận Phú Nhuận, TP. Hồ Chí Minh.

Qua gần 49 năm phát triển, tính đến 2024, Viện Y dược Học Dân Tộc hiện là bệnh viện chuyên khoa hạng I, trực thuộc Sở Y tế TP. Hồ Chí Minh, đảm nhiệm chức năng chính là điều trị tất cả các loại bệnh trong lĩnh vực y học cổ truyền kết hợp y học hiện đại cho người dân các tỉnh thành thuộc khu vực phía Nam. Bên cạnh đó, được Bộ Y tế (BYT) phân công là tuyến trung ương về chuyên môn kỹ thuật trong lĩnh vực y học cổ truyền kết hợp y học hiện đại, Viện Y dược Học Dân Tộc đã thực hiện công tác chỉ đạo tuyến, triển khai đề án vệ tinh cho các bệnh viện y học cổ truyền và khoa y học cổ truyền của các bệnh viện đa khoa ở TP. Hồ Chí Minh và 23 tỉnh, thành miền Nam. Qua đó, Viện Y dược Học Dân Tộc đã hợp tác với các đơn vị chuyên khoa trong và ngoài nước xây dựng mạng lưới điều trị bệnh cho khu vực phía Nam, đồng thời phát triển mạng lưới phòng chống các bệnh lý mãn tính, không lây nhiễm đạt hiệu quả, chất lượng phục vụ nhân dân TP. Hồ Chí Minh nói riêng, cả nước nói chung.

Ngoài chức năng, nhiệm vụ chính là khám, tầm soát, chẩn đoán, chăm sóc và điều trị chuyên sâu trong lĩnh vực y học cổ truyền kết hợp y học hiện đại, với đội ngũ nhân lực có trình độ chuyên môn cao, Viện Y dược Học Dân Tộc còn là trung tâm đào

tạo lâm sàng cho các trường đào tạo y khoa trên địa bàn thành phố, cả nước và là trung tâm nghiên cứu lâm sàng với nhiều đề tài hợp tác trong và ngoài nước.

Với nhiều thành tích trong suốt gần 49 năm qua, nhiều tập thể và cá nhân của Viện Y dược Học Dân Tộc TP. Hồ Chí Minh đã được Đảng, Nhà nước trao tặng nhiều phần thưởng, danh hiệu cao quý, trong đó nổi bật là Huân chương lao động hạng I (2021) dành cho tập thể lãnh đạo và cán bộ, viên chức, người lao động của Viện Y dược Học Dân Tộc.

Lịch sử hình thành và phát triển Viện Y dược Học Dân Tộc có thể được chia thành 2 giai đoạn chính:

- Giai đoạn 1975 – 1985: Viện Y dược Học Dân Tộc thuộc Bộ Y tế quản lý.
- Giai đoạn 1985 – nay: Viện Y dược Học Dân Tộc do Ủy ban Nhân dân, Sở Y tế TP. Hồ Chí Minh quản lý.

Trải qua cùng thăng trầm của lịch sử và Thành phố, Viện Y dược Học Dân Tộc đã từng bước xây dựng, phát triển gắn liền với các cột mốc lịch sử quan trọng.

Ngày 24/12/1975, theo Quyết định số 43/YTXHTB-TC ngày 24/12/1975 do bác sĩ Dương Quỳnh Hoa – Bộ trưởng Bộ Y tế Xã hội và Thương binh ký, Viện Y dược Học Dân Tộc được thành lập trực thuộc Bộ Y tế Xã hội và Thương binh.

Ngày 05/01/1976, Ban lãnh đạo Viện họp và phân công nhiệm vụ như sau:

- GS, Lương y Lê Minh Xuân giữ chức vụ Viện trưởng
- 04 Phó Viện trưởng gồm: Đồng chí Nguyễn Văn Quyền, BS. Bùi Chí Hiếu, Đồng chí Lê Thanh Liêm và Đồng chí Võ Thành Thi.

Ngày 18/01/1977 Ông Hoàng Đình Cầu - Thứ trưởng Bộ Y tế ký Quyết định số 85/TC- QĐ công nhận và xác định tên các đơn vị sự nghiệp tại B2 cũ thuộc về ngành Y tế Trung ương trong đó có Viện Y dược Học Dân Tộc.

Căn cứ Quyết định số 161/BYT-QĐ ngày 02/3/1985 của Bộ trưởng Bộ Y tế ký chuyển giao Viện Y Dược Học Dân Tộc thành phố Hồ Chí Minh trực thuộc Bộ Y tế về cho Ủy ban Nhân dân thành phố Hồ Chí Minh quản lý toàn diện và Quyết định số 4026/QĐ-BYT ngày 20/10/2010 của Bộ trưởng Bộ Y tế về việc ban hành quy định phân công công tác chỉ đạo tuyến trong lĩnh vực khám, chữa bệnh, Viện Y Dược Học Dân Tộc thành phố Hồ Chí Minh thực hiện các nhiệm vụ khác do Bộ trưởng Bộ Y tế giao, cụ thể như sau :

- Nghiên cứu khoa học và tuyên truyền phổ cập những kiến thức Y học dân tộc trong cán bộ và nhân dân.

- Thừa kế và phát huy, chỉ đạo toàn diện công tác y dược học dân tộc trên địa bàn thành phố Hồ Chí Minh, đồng thời hỗ trợ giúp Cục quản lý Y Dược cổ truyền – Bộ Y tế trong việc tổ chức chỉ đạo thừa kế phát huy y, dược học dân tộc ở 19 tỉnh, thành miền Nam và 5 tỉnh Tây Nguyên.

- Khám và chữa bệnh bằng các phương pháp y học dân tộc cho cán bộ và nhân dân thành phố Hồ Chí Minh và các tỉnh trong cả nước.

- Tham gia đào tạo, bồi dưỡng cán bộ Đại học và Trung học y dược học dân tộc, đồng thời là cơ sở hướng dẫn thực hành cho Đại học Y Dược thành phố Hồ Chí Minh, Trường Đại học Y khoa Phạm Ngọc Thạch.

- Chỉ đạo tuyên truyền về Y dược học dân tộc trên địa bàn thành phố Hồ Chí Minh, đồng thời hỗ trợ giúp Cục quản lý Y Dược cổ truyền chỉ đạo, xây dựng phong trào Y dược dân tộc ở 19 tỉnh miền Nam (bao gồm thành phố Hồ Chí Minh) và 05 tỉnh Tây Nguyên.

Tổ chức bộ máy tại Viện gồm: Ban Giám đốc: 04 người. Viện có 10 phòng chức năng, 16 khoa chuyên môn và 1 trung tâm, tham gia khám chữa bệnh, đào tạo chuyên môn, sản xuất thuốc và chỉ đạo tuyên.

Liên tục từ năm 2015 đến nay, Viện Y dược Học Dân Tộc luôn là 01 trong 05 bệnh viện tại TP. Hồ Chí Minh đứng đầu về điểm quản lý chất lượng bệnh viện và là bệnh viện dẫn đầu điểm quản lý chất lượng bệnh viện trong các bệnh viện Y học cổ truyền toàn quốc.

Viện Y Dược Học Dân Tộc thực hiện cơ chế tự chủ một phần theo quy định và chịu sự quản lý, điều hành trực tiếp, toàn diện của Giám đốc Sở Y tế, Viện có tư cách pháp nhân, có trụ sở, có con dấu riêng và được mở tài khoản tại Kho bạc Nhà nước theo quy định của pháp luật, cụ thể:

- Tên đơn vị: Viện Y Dược Học Dân Tộc
- Trụ sở: 273-275 Nguyễn Văn Trỗi, P10, Q.Phú Nhuận, Thành phố Hồ Chí Minh
- Biểu tượng:



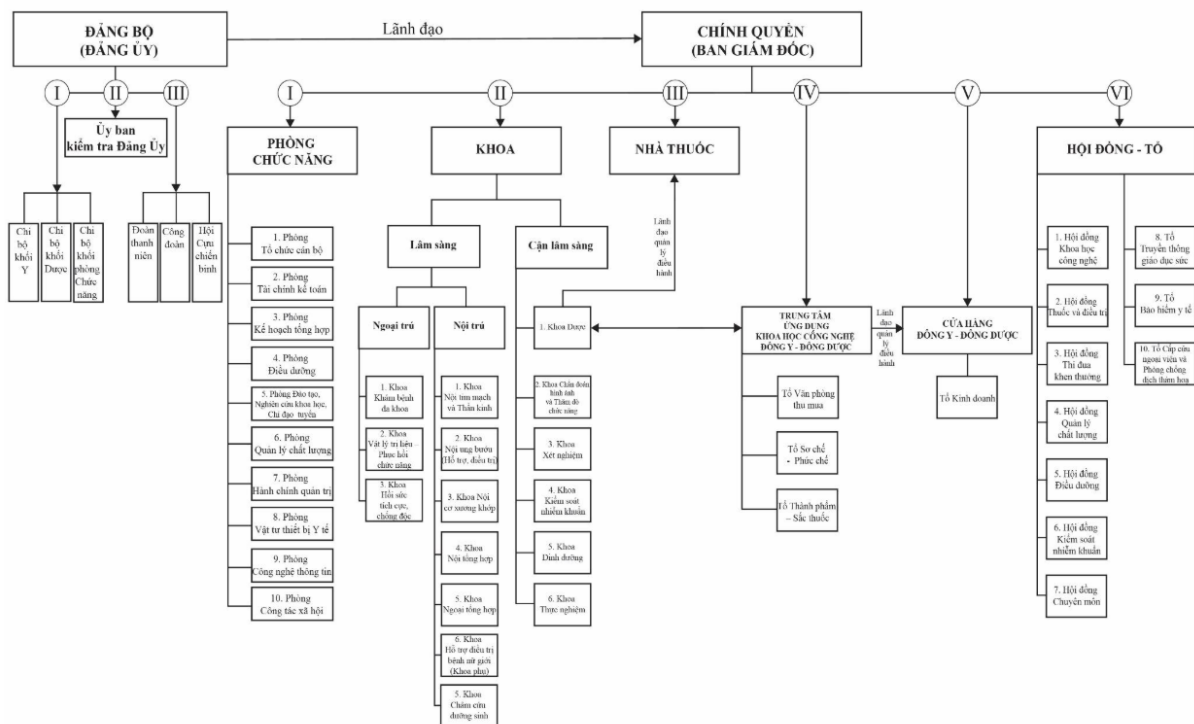
Hình 1: Logo Viện Y dược học Dân tộc

1.2. GIỚI THIỆU VỀ PHÒNG CÔNG NGHỆ THÔNG TIN

Là phòng nghiệp vụ chịu sự lãnh đạo trực tiếp của Phó Viện trưởng phụ trách chuyên môn và chịu trách nhiệm trước Ban giám đốc về xây dựng kế hoạch triển khai ứng dụng và phát triển CNTT trong hoạt động của Bệnh viện.

- Tổ chức triển khai các giải pháp đảm bảo an toàn và bảo mật cho hệ thống thông tin, cơ sở dữ liệu của bệnh viện.
- Tổ chức thiết kế, lắp đặt các thiết bị tin học cho các Khoa, Phòng, Trung tâm.
- Hướng dẫn, đào tạo nâng cao năng lực ứng dụng CNTT cho cán bộ, viên chức bệnh viện.
- Xây dựng hệ cơ sở dữ liệu của bệnh viện; Thực hiện công tác thống kê y tế theo quy định.
- Quản lý, duy trì hoạt động ổn định, có hiệu quả của hệ thống mạng, hệ thống thư điện tử, đảm bảo việc kết nối thông tin giữa Bệnh viện với Bộ Y tế và các đơn vị sự nghiệp có liên quan.
- Quản lý, phát triển trang thông tin điện tử (Website) của bệnh viện.
- Đảm nhiệm công tác sửa chữa, bảo dưỡng các thiết bị công nghệ thông tin.

1.3. SƠ ĐỒ TỔ CHỨC VIỆN Y DƯỢC HỌC DÂN TỘC



Hình 2: Sơ đồ Viện Y dược học Dân tộc

1.4. NHIỆM VỤ THỰC TẬP

Qua thời gian thực tập tại Viện em được học hỏi thêm nhiều kiến thức thực tế về hệ thống quản lý vận hành công nghệ thông tin của Viện, đồng thời được trải nghiệm tác phong làm việc chuyên nghiệp với các anh chị trong phòng Công nghệ thông tin.

- Gia nhập cơ quan với vai trò là thực tập sinh Helpdesk – Chuyên viên hỗ trợ Công nghệ thông tin, được phân công công tác tại phòng Công nghệ thông tin.
- Tìm hiểu văn hóa, lịch sử tại Viện, các quy định bảo mật thông tin, hệ thống máy chủ, hệ thống mạng, hệ thống máy tính và hệ thống camera của Viện.
- Tìm hiểu, áp dụng xây dựng dự án Website Thương mại Điện tử sử dụng công nghệ ASP.NET MVC NET 7.

1.5. KẾT LUẬN CHƯƠNG 1

Viện Y Dược Học Dân Tộc đã khẳng định được sự tín nhiệm từ người dân, thân nhân, bệnh nhân trong lĩnh vực Y – Dược – Học về Y học cổ truyền, bên cạnh việc cung cấp các dịch vụ trong lĩnh vực Y – Dược – Học thì Viện luôn tiên phong trong việc áp dụng và phát triển các công nghệ mới. Với việc luôn ưu tiên và đi đầu trong áp dụng và phát

triển các công nghệ mới đã tạo điều kiện thuận lợi cho các thực tập sinh như em có cơ hội tiếp cận tìm hiểu với các công nghệ mới áp dụng thực tế ra sao, tiếp cận với môi trường làm việc chuyên nghiệp và hiểu rõ hơn về quy trình làm việc.

CHƯƠNG 2. TỔNG QUAN VỀ ĐỀ TÀI THỰC TẬP

2.1. XÁC ĐỊNH ĐỀ TÀI NGHIÊN CỨU

2.1.1. Tên đề tài

Xây dựng website Thương mại Điện tử dùng để kinh doanh thuốc thực phẩm chức năng (không kê đơn) cho Viện Y dược học Dân tộc sử dụng ASP.NET MVC 7.

2.1.2. Mục tiêu tổng quát

Phát triển một trang website bán hàng cho phép người dùng truy cập và tìm mua các loại thuốc khác nhau (không kê đơn); cho phép người quản trị quản lý và xử lý các đơn hàng mua từ khách hàng.

2.1.3. Mục tiêu cụ thể

Cho phép người dùng đăng nhập, đăng kí, tìm kiếm sản phẩm theo từ khoá hoặc danh mục, mua sản phẩm và đánh giá sản phẩm đã mua.

Cho phép người quản trị đăng nhập vào hệ thống, quản lý các cơ sở dữ liệu bao gồm người dùng, sản phẩm, danh mục và xử lý đơn hàng.

2.2. ĐỐI TƯỢNG NGHIÊN CỨU

2.2.1. Ngôn ngữ lập trình C#:

C# (hay C-sharp) là một ngôn ngữ lập trình được phát triển bởi Microsoft vào năm 2000, với mục tiêu xây dựng một ngôn ngữ hiện đại, hướng đối tượng, và dựa trên nền tảng của C++ và Java. Trong các ứng dụng truyền thống trên Windows, mã nguồn được biên dịch trực tiếp thành mã thực thi; còn trong ứng dụng sử dụng .NET Framework, mã nguồn C# được biên dịch thành mã trung gian MSIL, sau đó được CLR biên dịch thành mã thực thi. Sự hỗ trợ của .NET Framework giúp việc phát triển ứng dụng Windows Forms, WPF, game, ứng dụng web và di động trở nên dễ dàng hơn. C# loại bỏ sự phức tạp của các ngôn ngữ tiền nhiệm và mang lại các tính năng hiện đại như xử lý ngoại lệ và thu gom bộ nhớ tự động. Là một ngôn ngữ hướng đối tượng thuần túy, C# đơn giản hóa lập trình với ít từ khóa nhưng vẫn mạnh mẽ và linh hoạt. Sự phát triển của .NET Core còn giúp C# có thể được sử dụng trên nhiều nền tảng hệ điều hành khác nhau như Windows, Linux, và MacOS.

2.2.2. ASP.NET Core

ASP.NET Core là một phiên bản cải tiến và mã nguồn mở của ASP.NET, do Microsoft phát triển và phát hành trên GitHub. Được thiết kế để tối ưu hóa hiệu suất và tiết kiệm tài nguyên, ASP.NET Core hỗ trợ đa nền tảng, bao gồm Windows, Mac và Linux. So với ASP.NET, ASP.NET Core mang lại nhiều lợi ích như khả năng xây dựng ứng dụng web UI và web API với mô hình MVC, tích hợp các framework phía client như AngularJS và Bootstrap, cũng như khả năng lưu trữ trên nhiều môi trường như IIS, HTTP.sys, Kestrel, Nginx, Apache và Docker. Đặc biệt, ASP.NET Core còn hỗ trợ xây dựng các dịch vụ HTTP với nhiều định dạng và tự động ánh xạ dữ liệu từ HTTP request tới các tham số của method action. Sự module hóa và tính mở của ASP.NET Core cùng với khả năng phát triển ứng dụng đa nền tảng và tối ưu cho môi trường đám mây đã làm cho nó trở nên hấp dẫn đối với các lập trình viên hiện đại.

2.2.3. SQL Server

SQL Server, hay còn được gọi là Microsoft SQL Server, là một hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft. Với việc cung cấp nhiều công cụ và tính năng như quản lý, lưu trữ, xử lý truy vấn dữ liệu, kiểm soát truy cập và xử lý giao dịch, SQL Server giúp tích hợp dữ liệu từ nhiều nguồn khác nhau. Nó còn hỗ trợ tạo báo cáo, phân tích và quản lý cơ sở dữ liệu trực quan thông qua giao diện người dùng hoặc các script lệnh SQL.

SQL Server được xây dựng dựa trên SQL, một ngôn ngữ lập trình tiêu chuẩn cho cơ sở dữ liệu quan hệ, và sử dụng Transact-SQL (T-SQL), phiên bản mở rộng của SQL do Microsoft phát triển, cung cấp các khả năng bổ sung như khai báo biến, thủ tục lưu trữ và xử lý ngoại lệ. Ban đầu chỉ hoạt động trên Windows, SQL Server từ năm 2016 đã mở rộng hỗ trợ sang Linux, với SQL Server 2017 chạy trên cả Windows và Linux.

Các công cụ đi kèm như SQL Server Management Studio giúp quản lý cơ sở dữ liệu dễ dàng trên cả môi trường 64-bit và 32-bit. T-SQL, ngôn ngữ thuộc sở hữu của Microsoft, cung cấp các tính năng nâng cao cho việc quản lý cơ sở dữ liệu. SQL Server đã không ngừng phát triển để đáp ứng nhu cầu đa dạng của người dùng, từ doanh nghiệp nhỏ đến các tổ chức lớn, nhờ vào sự linh hoạt và hiệu suất cao của mình.

2.2.4. Mô hình MVC

MVC (Model-View-Controller) là một mô hình thiết kế phần mềm hiệu quả, giúp phân tách ứng dụng thành ba phần chính: Model, View và Controller.

Model đại diện cho dữ liệu và logic nghiệp vụ của ứng dụng, chịu trách nhiệm quản lý dữ liệu và các quy tắc nghiệp vụ.

View là thành phần trực quan, chịu trách nhiệm hiển thị dữ liệu tới người dùng thông qua giao diện.

Controller là cầu nối giữa Model và View, xử lý các yêu cầu từ người dùng, gọi các phương thức trên Model và chọn View thích hợp để hiển thị kết quả.

Mô hình MVC giúp ứng dụng dễ dàng bảo trì, kiểm thử và mở rộng nhờ việc tách biệt rõ ràng giữa logic xử lý và giao diện. Điều này cũng cho phép các nhà phát triển làm việc song song trên các phần khác nhau của ứng dụng mà ít gây xung đột, đồng thời cải thiện khả năng tái sử dụng mã. MVC không chỉ được áp dụng rộng rãi trong việc phát triển web, mà còn trong nhiều ứng dụng phần mềm khác, nhờ vào tính linh hoạt và sự phân chia rõ ràng giữa các thành phần của nó.

2.2.5. Bootstraps

Bootstrap là một framework mạnh mẽ bao gồm các template HTML, CSS, và JavaScript, giúp phát triển website chuẩn responsive một cách nhanh chóng và dễ dàng. Với các thành tố cơ bản như typography, forms, buttons, tables, grids, navigation, và image carousels, Bootstrap giúp các lập trình viên tiết kiệm thời gian và công sức khi thiết kế giao diện web. Đây là một bộ sưu tập mã nguồn mở miễn phí, với các thuộc tính giao diện được quy định sẵn về kích thước, màu sắc, độ cao, độ rộng, giúp các designer dễ dàng sáng tạo ra những sản phẩm mới mẻ mà vẫn đảm bảo hiệu quả trong quá trình thiết kế. Bootstrap không chỉ hỗ trợ tối ưu hóa giao diện cho các thiết bị khác nhau mà còn giúp việc phát triển web trở nên nhanh chóng và hiệu quả hơn.

2.3. PHẠM VI NGHIÊN CỨU VÀ ỨNG DỤNG

Môi trường ứng dụng: Trình duyệt web.

Phạm vi người dùng: Các bệnh nhân đang khám tại Viện cũng như ngoài Viện có nhu cầu mua thuốc không kê đơn.

CHƯƠNG 3: TỔNG QUAN VỀ PHẦN MỀM

3.1. YÊU CẦU VỀ PHẦN MỀM

Quản lý sản phẩm: Người quản trị có thể thêm, sửa, xoá các sản phẩm thuốc men

Quản lý người dùng: Người quản trị có thể thêm, sửa các tài khoản người dùng

Quản lý danh mục: Người quản trị có thể thêm, sửa, xoá các danh mục sản phẩm thuốc men

Quản lý đơn hàng: Người quản trị có thể duyệt các đơn hàng của người mua

Đăng nhập / Đăng kí: Người dùng có thể tạo tài khoản khách hàng và đăng nhập vào hệ thống để mua hàng

Mua hàng: Người dùng có thể thêm sản phẩm vào giỏ hàng và tiến hành thanh toán đơn hàng

Đánh giá: Người dùng sau khi mua hàng có thể sử dụng chức năng đánh giá sản phẩm

3.2. ĐỐI TƯỢNG NGƯỜI DÙNG

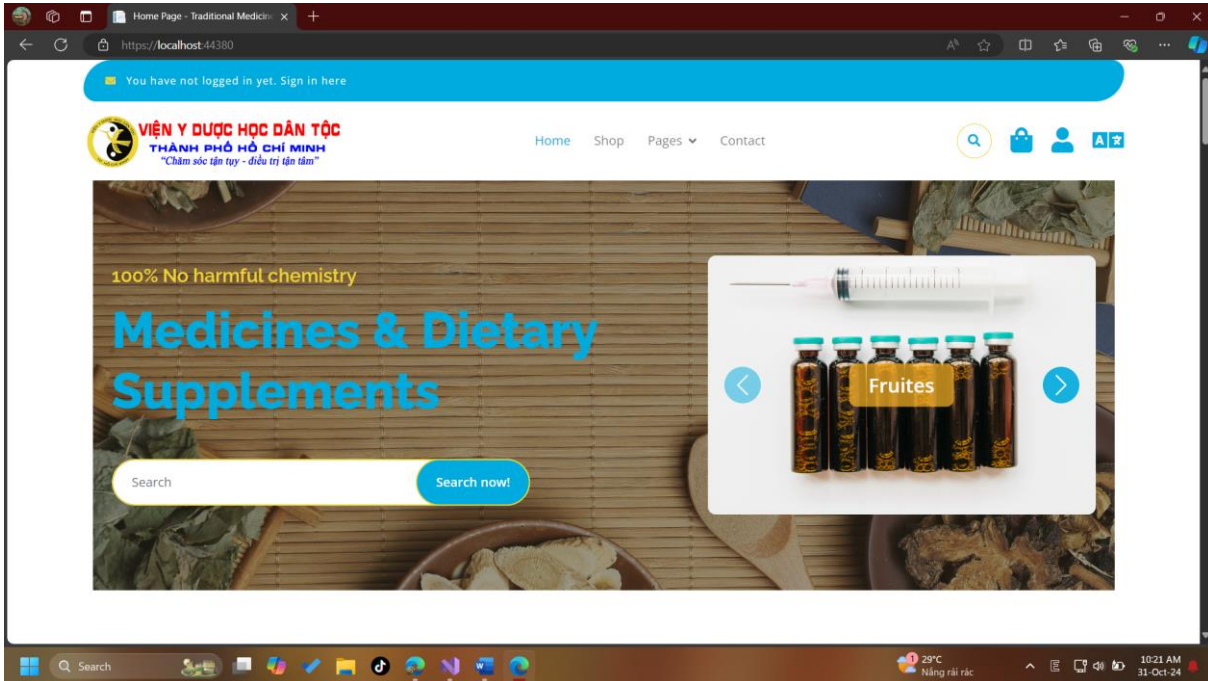
Đối với admin: cho phép quản lý các trường dữ liệu và duyệt đơn hàng người dùng

Đối với khách hàng: cho phép quản lý tài khoản cá nhân và mua hàng

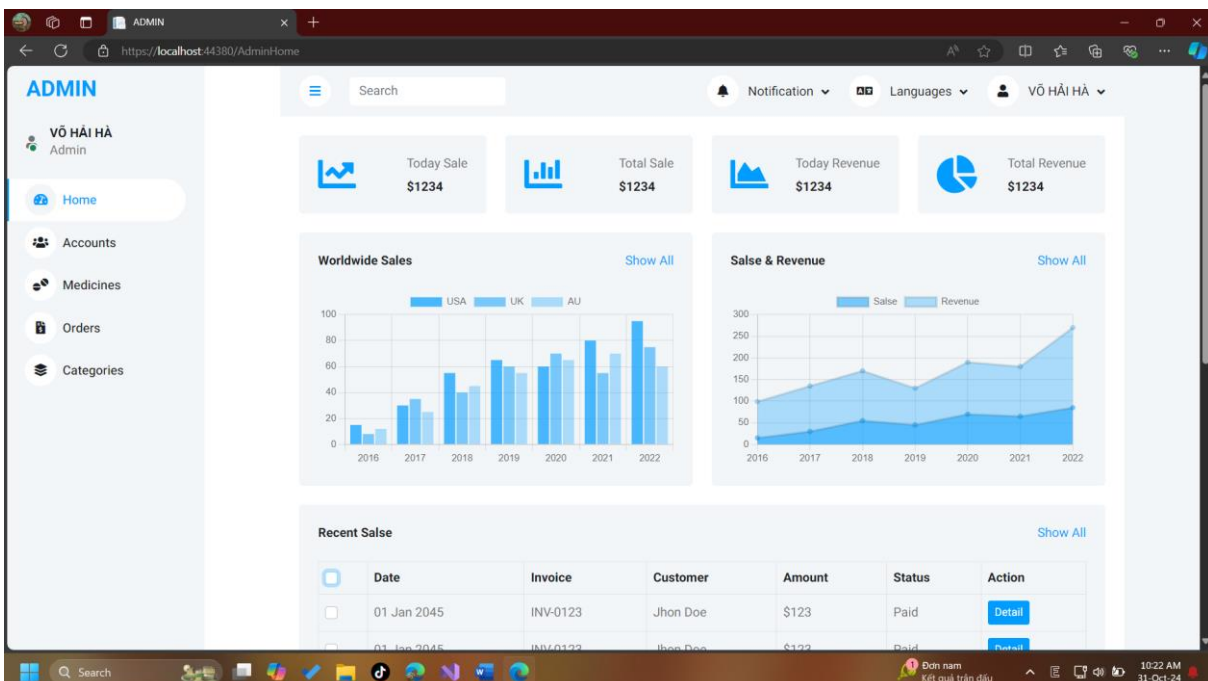
CHƯƠNG 4: XÂY DỰNG PHẦN MỀM

4.1. XÂY DỰNG GIAO DIỆN PHẦN MỀM

4.1.1. Trang chủ

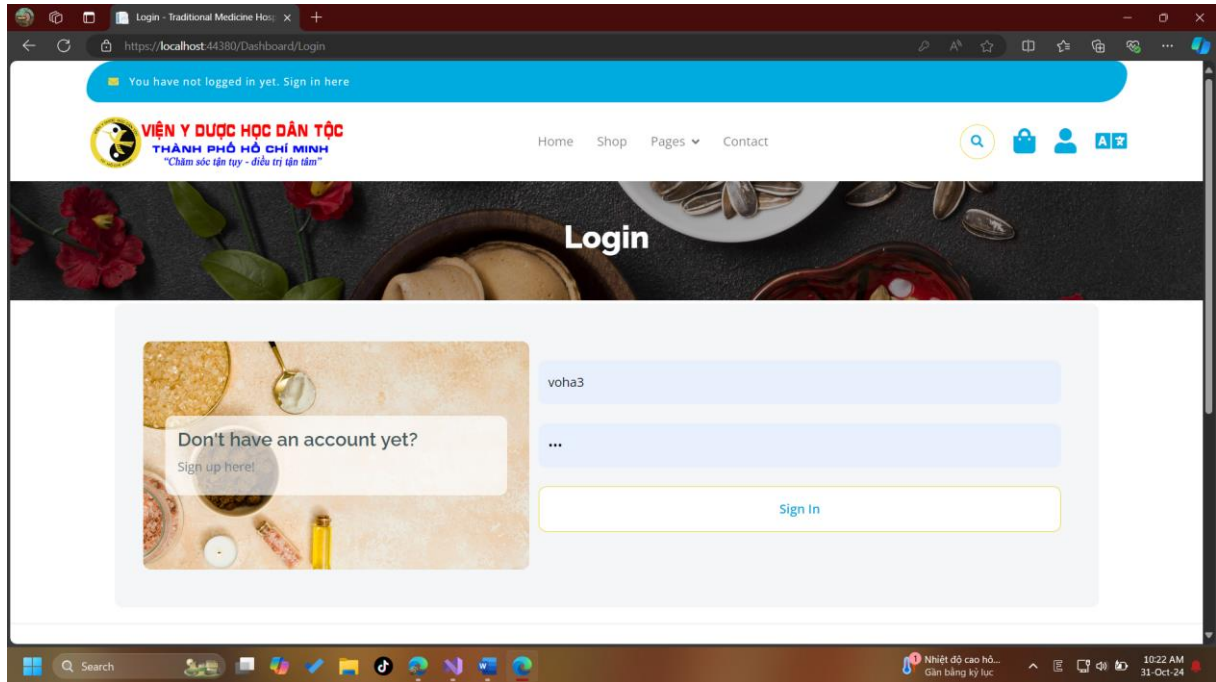


Hình 3: Giao diện trang chủ



Hình 4: Giao diện trang chủ admin

4.1.2. Đăng nhập vào hệ thống



Hình 5: Giao diện Login

@using ThucTap_ThuongMaiDienTu.Resources

@{

 ViewData["Title"] = "Login";

 Layout = "~/Views/Shared/_LayoutCustomer.cshtml";

}

 @if (ViewData.ModelState.Count > 0)

 {

 <div class="alert alert-danger">

 @Html.ValidationSummary(false, "", new { @class = "text-danger" })

 </div>

 }

 <form asp-action="Login" asp-controller="Dashboard">

 <input name="Username" type="text" class="w-100 form-control border-0 py-3 mb-4"

placeholder="@Resource.Username" required>

 <input name="Password" type="password" class="w-100 form-control border-0 py-3 mb-4"

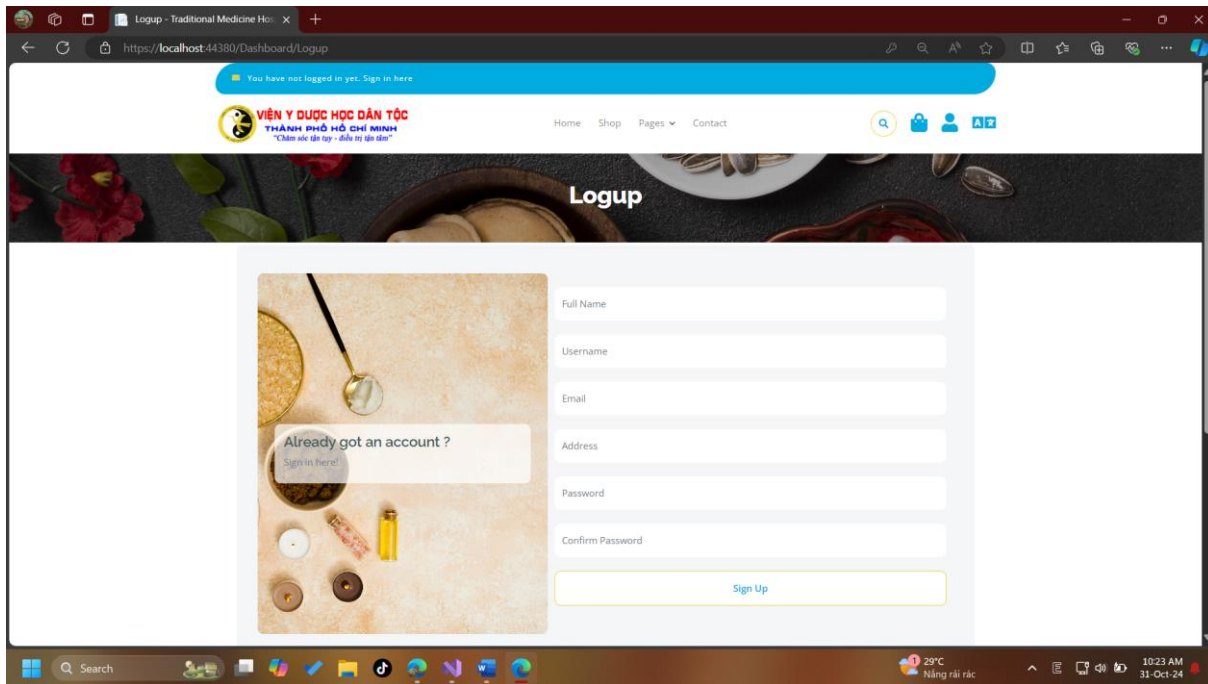
placeholder="@Resource.Password" required>

 <button class="w-100 btn form-control border-secondary py-3 bg-white text-primary"

type="submit">@Resource.Login</button>

 </form>

4.1.3. Đăng kí tài khoản mới



Hình 6: Giao diện đăng kí

@using ThucTap_ThuongMaiDienTu.Resources

@{

ViewData["Title"] = "Logup";

Layout = "~/Views/Shared/_LayoutCustomer.cshtml";

}

@if (ViewData.ModelState.Count > 0)

{

<div class="alert alert-danger">

@Html.ValidationSummary(false, "", new { @class = "text-danger" })

</div>

}

<form asp-action="Logup" asp-controller="Dashboard">

<input name="Name" type="text" class="w-100 form-control border-0 py-3 mb-4"

placeholder="@Resource.Name" required>

<input name="Username" type="text" class="w-100 form-control border-0 py-3 mb-4"

placeholder="@Resource.Username" required>

<input name="Email" type="email" class="w-100 form-control border-0 py-3 mb-4"

placeholder="Email" required>

<input name="Address" type="text" class="w-100 form-control border-0 py-3 mb-4"

placeholder="Address" required>

<input name="Password" type="password" class="w-100 form-control border-0 py-3 mb-4"

placeholder="@Resource.Password" required>

<input name="CFPassword" type="password" class="w-100 form-control border-0 py-3 mb-4"

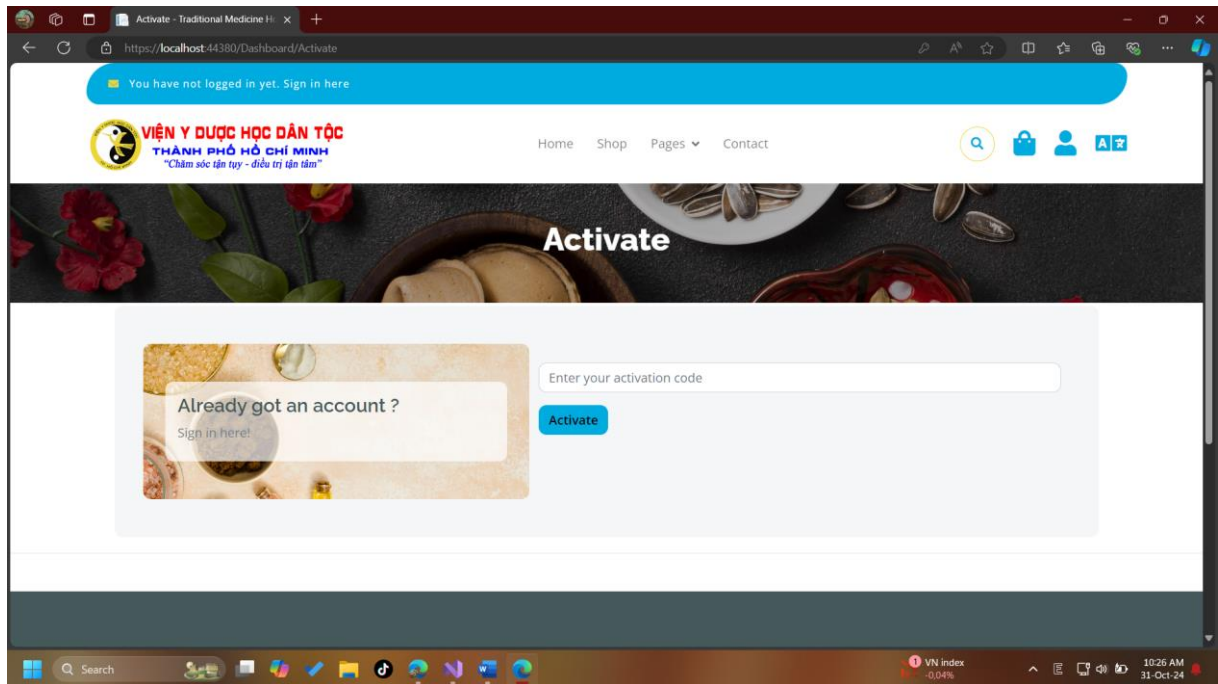
placeholder="@Resource.CFPassword" required>

```

        <button class="w-100 btn form-control border-secondary py-3 bg-white text-primary"
type="submit">@Resource.Logup</button>
    </form>

```

4.1.4. Xác thực tài khoản



Hình 7: Giao diện kích hoạt tài khoản

```

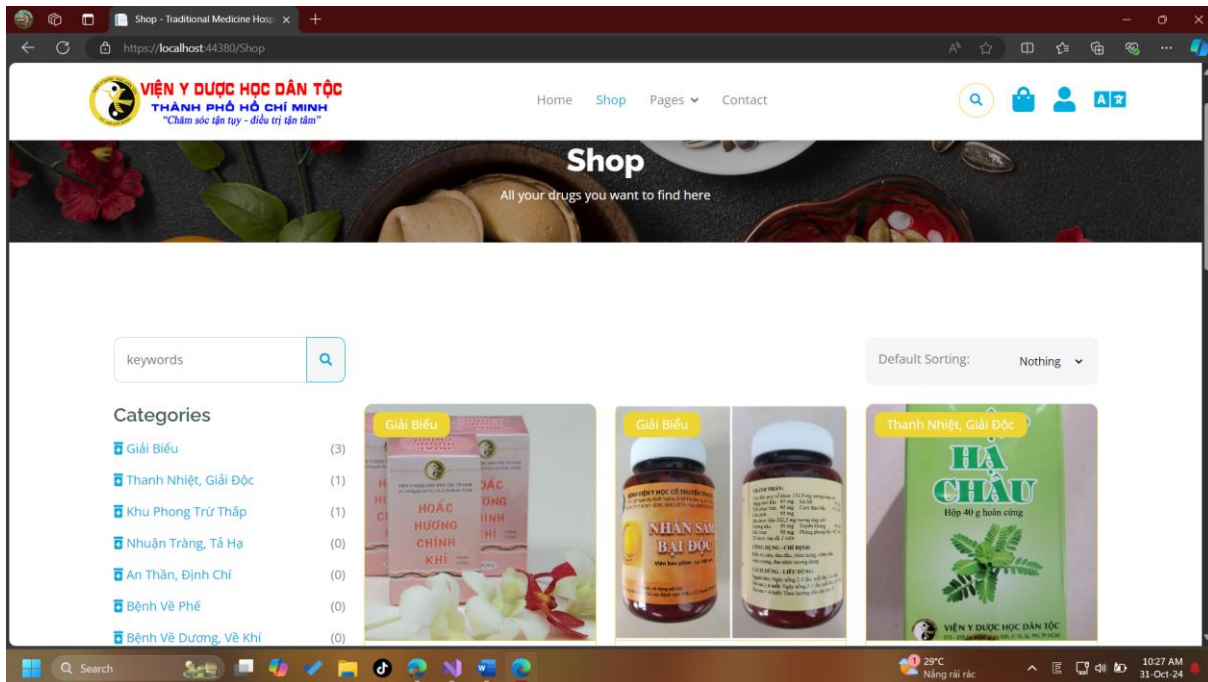
@using ThucTap_ThuongMaiDienTu.Resources
@{
    ViewData["Title"] = "Activate";
    Layout = "~/Views/Shared/_LayoutCustomer.cshtml";
}

@if (ViewData.ModelState.Count > 0)
{
    <div class="alert alert-danger">
        @Html.ValidationSummary(false, "", new { @class = "text-danger" })
    </div>
}

<form asp-action="Activate" asp-controller="Dashboard">
    <input name="Username" type="hidden" value="@TempData["Username"]" />
    <input name="Code" type="text" class="form-control" placeholder="Enter your activation code"
required />
    <button class="btn btn-primary mt-3" type="submit">Activate</button>
</form>

```


4.1.5. Danh mục sản phẩm



Hình 8: Giao diện danh mục sản phẩm

```
@model IEnumerable<ThucTap_ThuongMaiDienTu.ViewModels.MedicineVM>
@foreach (var item in Model)
{
    <div class="col-md-6 col-lg-6 col-xl-4" onclick="location.href='@Url.Action("ShopDetail", "Shop", new { id =
item.Id })';" style="cursor: pointer;">
        <div class="rounded position-relative fruite-item border border-secondary border-top-0 rounded-bottom">
            <div class="fruite-img border-bottom border-secondary">
                
            </div>
            <div class="text-white bg-secondary px-3 py-1 rounded position-absolute" style="top: 10px; left:
10px;">@item.Category</div>
            <div class="p-4">
                <h4>@item.Name</h4>
                <p>
                    @if (item.Description.Length > 100)
                    {
                        @item.Description.Substring(0, 100)
                    }
                    Else @item.Description
                </p>
                <div class="d-flex justify-content-between flex-lg-wrap">
                    <p class="text-dark fs-5 fw-bold mb-0">@$item.Price / @item.Pack</p>
                </div>
            </div>
        </div>
    </div>
}
```



```

<a href="#" class="btn border border-secondary rounded-pill px-3 text-primary"><i class="fa fa-shopping-bag me-2 text-primary"></i> Add to cart</a>

</div>

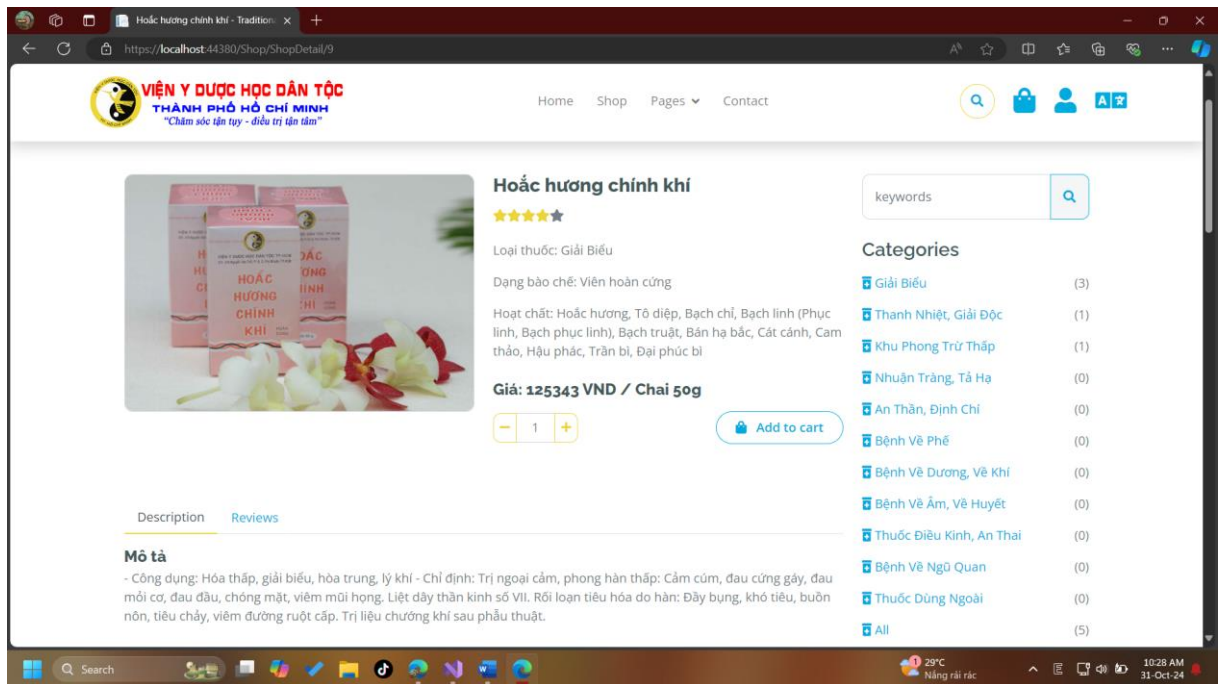
</div>

</div>

}

```

4.1.6. Chi tiết sản phẩm



Hình 9: Giao diện chi tiết sản phẩm

```
@model ThucTap_ThuongMaiDienTu.ViewModels.MedicineVM
```

```
@{
```

```
    ViewData["Title"] = Model.Name;
```

```
    Layout = "~/Views/Shared/_LayoutCustomer.cshtml";
```

```
}
```

```

<div class="container-fluid py-5 mt-5">
    <div class="container-fluid py-5 mt-5">
        <div class="container py-5">
            <div class="row g-4 mb-5">
                <div class="col-lg-8 col-xl-9">
                    <div class="row g-4">
                        <div class="col-lg-6">
                            <div class="border rounded">
                                <a href="#">
                                    
                                </a>
                            </div>
                        </div>

```

```

</div>
<div class="col-lg-6">
  <h4 class="fw-bold mb-3">@Model.Name</h4>
  <div class="d-flex mb-4">
    <i class="fa fa-star text-secondary"></i>
    <i class="fa fa-star text-secondary"></i>
    <i class="fa fa-star text-secondary"></i>
    <i class="fa fa-star text-secondary"></i>
    <i class="fa fa-star"></i>
  </div>
  <p class="mb-3">Loại thuốc: @Model.Category</p>
  <p class="mb-3">Dạng bào chế: @Model.Unit</p>
  <p class="mb-4">Hoạt chất: @Model.Ingredient</p>
  <h5 class="fw-bold mb-3">Giá: @Model.Price VND / @Model.Pack</h5>
  <form asp-action="AddToCart" asp-controller="Buy">
    <input type="hidden" name="productId" value="@Model.Id" />
    <input type="hidden" name="price" value="@Model.Price" />
    <div class="d-flex justify-content-between align-items-center mb-5">
      <div class="input-group quantity" style="flex-grow: 1; max-width: 150px;">
        <button type="button" class="btn btn-outline-secondary btn-sm d-flex align-items-center justify-content-center">
          <i class="fa fa-minus"></i>
        </button>
        <input type="text" class="form-control text-center border-1 shadow-none" name="quantity" value="1" min="1" style="font-size: 16px; max-width: 50px;">
        <button type="button" class="btn btn-plus btn-outline-secondary btn-sm d-flex align-items-center justify-content-center">
          <i class="fa fa-plus"></i>
        </button>
      </div>
      <button type="submit" class="btn btn-outline-primary rounded-pill px-4 py-2">
        <i class="fa fa-shopping-bag me-2"></i> Add to cart
      </button>
    </div>
  </form>
</div>
<div class="col-lg-12">
  <nav>
    <div class="nav nav-tabs mb-3">
      <button class="nav-link active border-white border-bottom-0" type="button" role="tab" id="nav-about-tab" data-bs-toggle="tab" data-bs-target="#nav-about" aria-controls="nav-about" aria-selected="true">
        Description

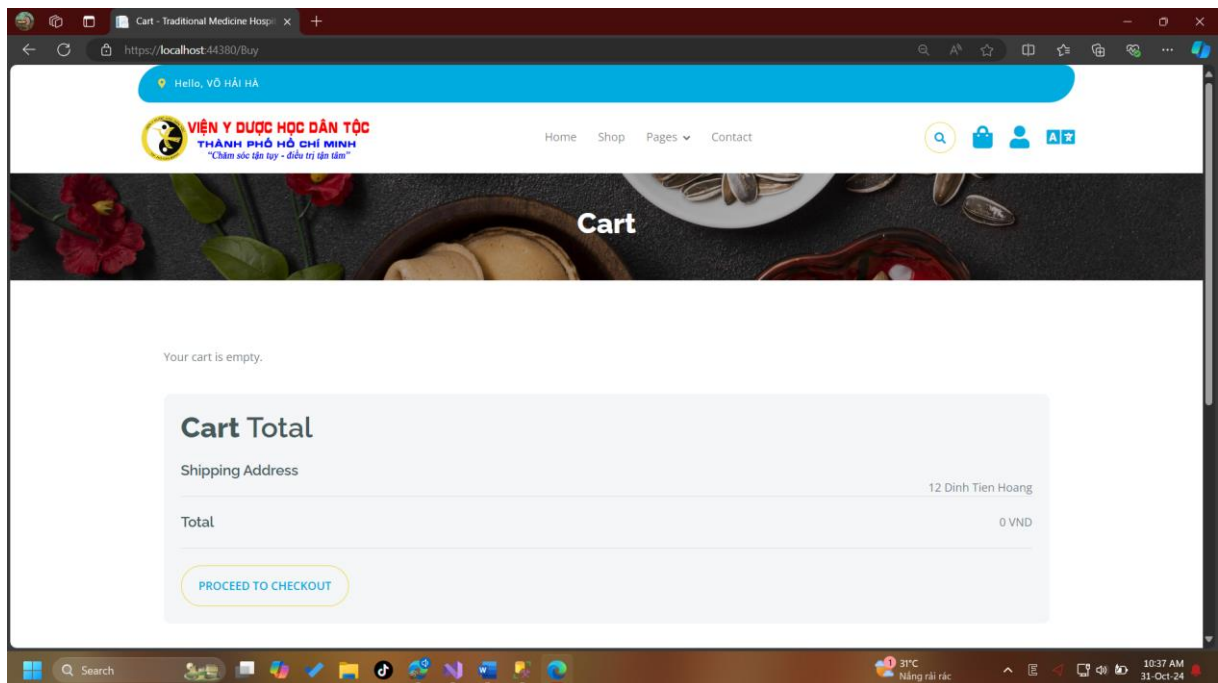
```

```

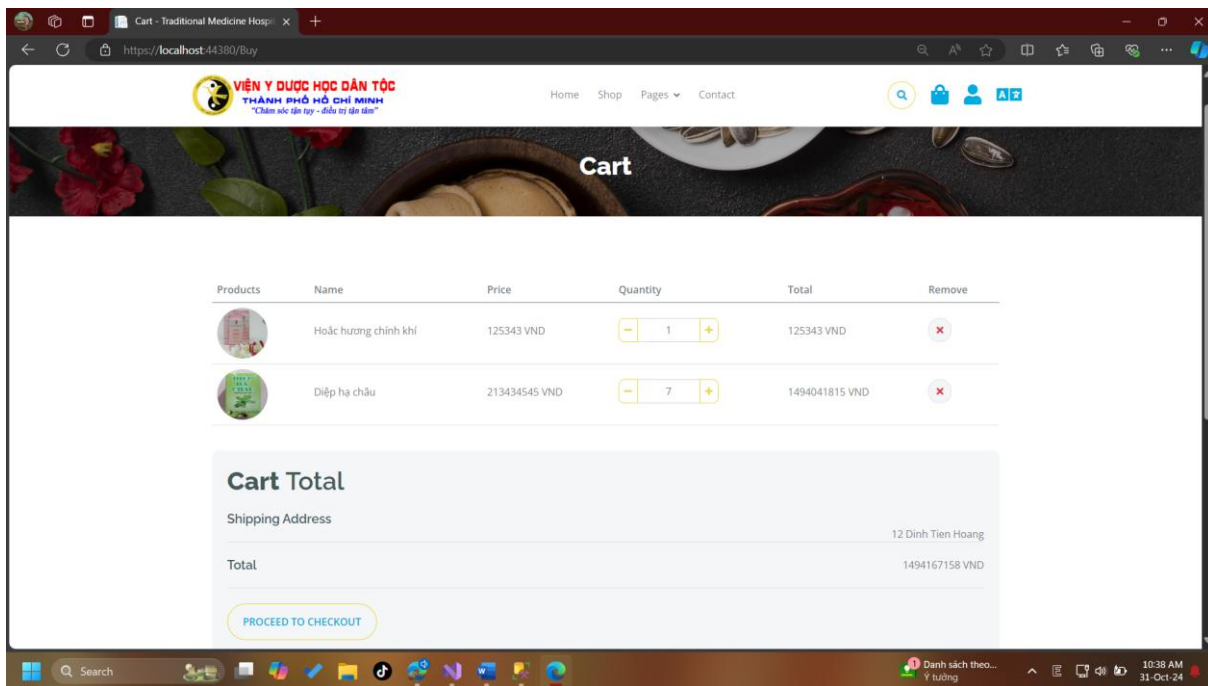
</button>
<button class="nav-link border-white border-bottom-0" type="button" role="tab"
  id="nav-mission-tab" data-bs-toggle="tab" data-bs-target="#nav-mission"
  aria-controls="nav-mission" aria-selected="false">
  Reviews
</button>
</div>
</nav>
<div class="tab-content mb-5">
  <div class="tab-pane active" id="nav-about" role="tabpanel" aria-labelledby="nav-about-tab">
    <h5 class="fw-bold mb-1">Mô tả</h5>
    <p class="mb-4" style="hyphens: auto;">
      @Html.Raw(Model.Description)
    </p>
    <h5 class="fw-bold mb-1">Chỉ định</h5>
    @Html.Raw(Model.Uses)
  </div>

```

4.1.7. Giỏ hàng của khách hàng



Hình 10: Giao diện giỏ hàng trống



Hình 11: Giao diện giỏ hàng khi có hàng

```
@model Cart
@{
    ViewData["Title"] = "Cart";
    Layout = "~/Views/Shared/_LayoutCustomer.cshtml";
}
<div class="container-fluid py-5">
    <div class="container py-5">
        <div class="row g-4 justify-content-end">
            <div class="col-12">
                @if (Model.CartDetails.Any())
                {
                    <div class="table-responsive">
                        <table class="table">
                            <thead>
                                <tr>
                                    <th scope="col">Products</th>
                                    <th scope="col">Name</th>
                                    <th scope="col">Price</th>
                                    <th scope="col">Quantity</th>
                                    <th scope="col">Total</th>
                                    <th scope="col">Remove</th>
                                </tr>
                            </thead>
                            <tbody>
                                @foreach (var item in Model.CartDetails)
```

```

{
    <tr>
        <th scope="row">
            <div class="d-flex align-items-center">
                
            </div>
        </th>
        <td>
            <p class="mb-0 mt-4">@item.Medicine.Name</p>
        </td>
        <td>
            <p class="mb-0 mt-4">@item.Medicine.Price VND</p>
        </td>
        <td>
            <div class="input-group quantity mt-3">
                <button type="button" class="btn btn-outline-secondary btn-sm d-flex align-items-center
justify-content-center">
                    <i class="fa fa-minus"></i>
                </button>
                <input type="text" class="form-control text-center border-1 shadow-none"
name="quantity" value="@item.Amount" min="1" style="font-size: 16px; max-width: 100px;">
                <button type="button" class="btn btn-plus btn-outline-secondary btn-sm d-flex align-
items-center justify-content-center">
                    <i class="fa fa-plus"></i>
                </button>
            </div>
        </td>
        <td>
            <p class="mb-0 mt-4">@item.Total VND</p>
        </td>
        <td>
            <button class="btn btn-md rounded-circle bg-light border mt-3">
                <i class="fa fa-times text-danger"></i>
            </button>
        </td>
    </tr>
}
</tbody>
</table>
</div>
</div>

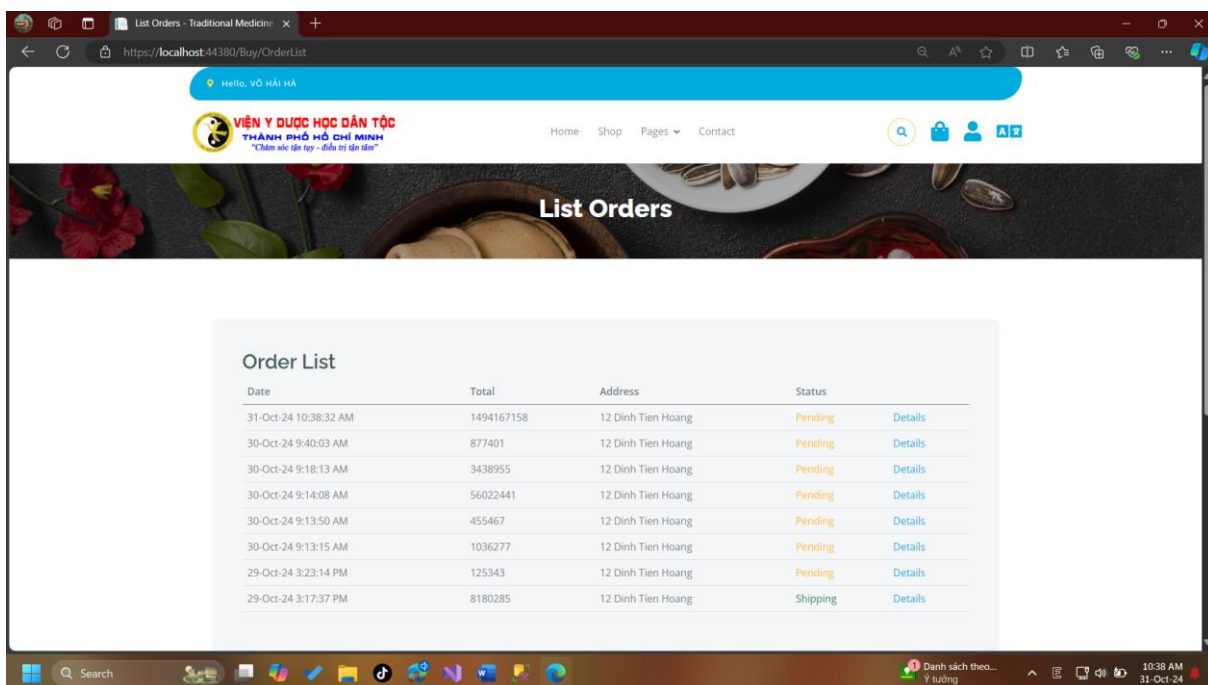
```

```

<form asp-action="Checkout" asp-controller="Buy">
    <input type="hidden" name="address" value="@Model.Account.Address" />
    <div class="col-12">
        <div class="bg-light rounded">
            <div class="p-4">
                <h1 class="display-6 mb-4">Cart <span class="fw-normal">Total</span></h1>
                <div class="d-flex justify-content-between">
                    <h5 class="mb-0 me-4">Shipping Address</h5>
                    <div>
                        <p class="mb-0 text-end">@Model.Account.Address</p>
                        <div class="border-top border-bottom py-4 mb-4 d-flex justify-content-between">
                            <h5 class="mb-0">Total</h5>
                            <p class="mb-0">@Model.Total VND</p>
                        </div>
                        <button class="btn border-secondary rounded-pill px-4 py-3 text-primary text-uppercase"
type="submit">
                            Proceed to Checkout
                        </button>
                    </div>
                </div>
            </div>
        </div>
    </div>
</form>

```

4.1.8. Quản lý đơn hàng của tài khoản



Hình 12: Giao diện danh sách đơn hàng

@model IEnumerable<ThucTap_ThuongMaiDienTu.Models.Order>

@using ThucTap_ThuongMaiDienTu.Resources

```

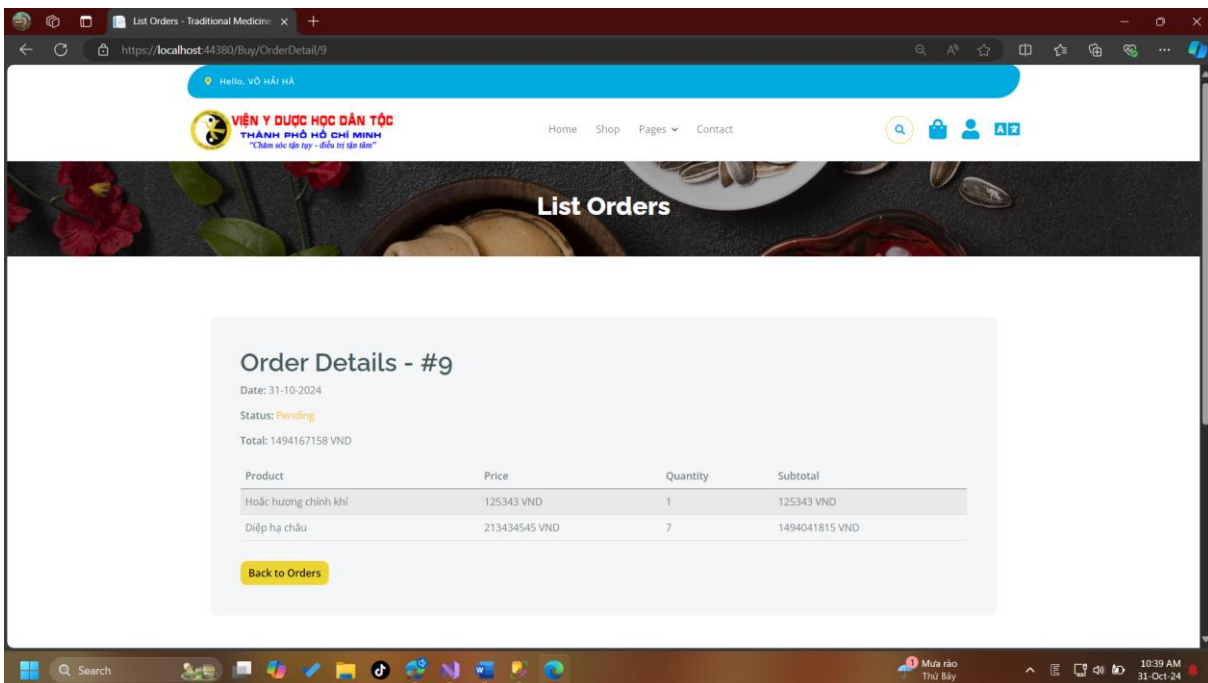
@{
    ViewData["Title"] = @Resource.OrderList;
    Layout = "~/Views/Shared/_LayoutCustomer.cshtml";
}
<div class="container-fluid contact py-5">
    <div class="container py-5">
        <div class="p-5 bg-light rounded">
            <div class="row g-4">
                <div class="col-12">
                    <h2>Order List</h2>
                    <table class="table">
                        <thead>
                            <tr>
                                <th>
                                    @Html.DisplayNameFor(model => model.Date)
                                </th>
                                <th>
                                    @Html.DisplayNameFor(model => model.Total)
                                </th>
                                <th>
                                    @Html.DisplayNameFor(model => model.Address)
                                </th>
                                <th>
                                    @Html.DisplayNameFor(model => model.Status)
                                </th>
                            <th></th>
                        </tr>
                    </thead>
                    <tbody>
                        @foreach (var item in Model)
                        {
                            <tr>
                                <td>
                                    @Html.DisplayFor(modelItem => item.Date)
                                </td>
                                <td>
                                    @Html.DisplayFor(modelItem => item.Total)
                                </td>
                                <td>
                                    @Html.DisplayFor(modelItem => item.Address)
                                </td>
                                <td>
                                    @if (item.Status)

```

```

        {
            <span class="text-success">Shipping</span>
        }
        else
        {
            <span class="text-warning">Pending</span>
        }
    </td>
    <td>
        <a asp-action="OrderDetail" asp-route-id="@item.Id">Details</a>
    </td>
</tr>
}
</tbody>
</table>

```



Hình 13: Giao diện chi tiết đơn hàng

```

@using ThucTap_ThuongMaiDienTu.Resources
@model ThucTap_ThuongMaiDienTu.Models.Order
@{
    ViewData["Title"] = @Resource.OrderList;
    Layout = "~/Views/Shared/_LayoutCustomer.cshtml";
}
<div class="container-fluid contact py-5">
    <div class="container py-5">
        <div class="p-5 bg-light rounded">
            <div class="row g-4">

```



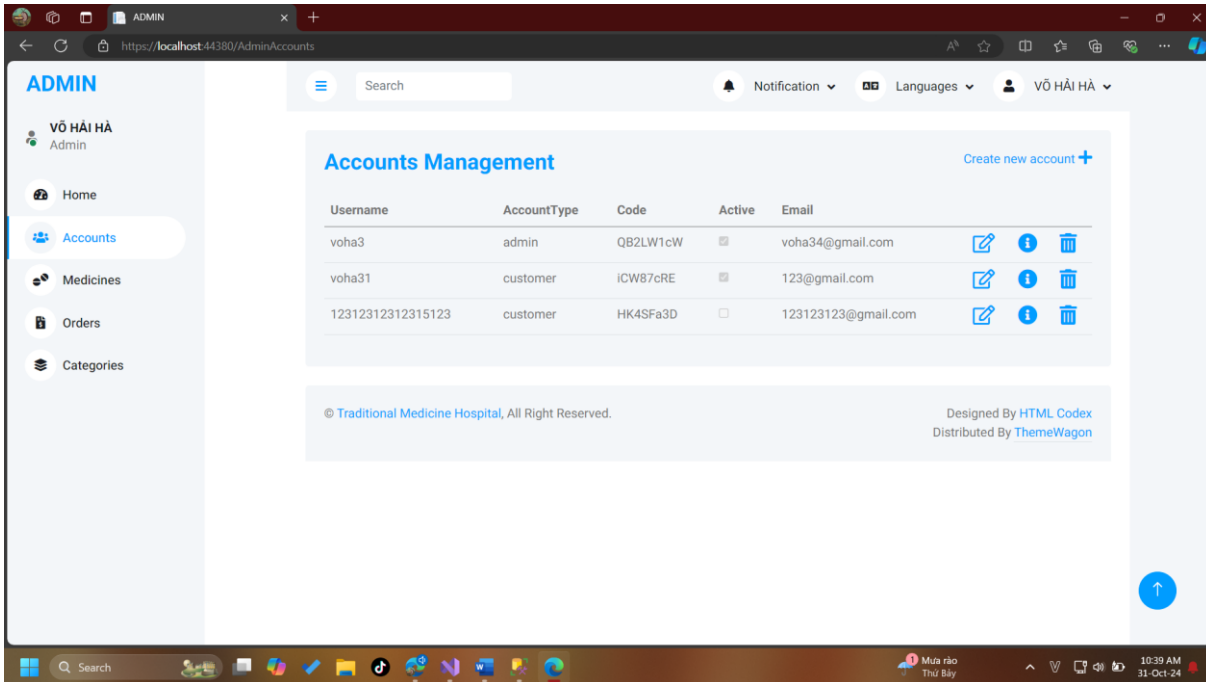
```

<div class="col-12">
  <h1>Order Details - #@Model.Id</h1>
  <div class="mb-4">
    <p><strong>Date:</strong> @Model.Date.ToString("dd/MM/yyyy")</p>
    <p>
      <strong>Status:</strong>
      <span class="@((Model.Status ? "text-success" : "text-warning"))">
        @(Model.Status ? "Completed" : "Pending")
      </span>
    </p>
    <p><strong>Total:</strong> @Model.Total VND</p>
  </div>
  @if (Model.OrderDetails.Any())
  {
    <div class="table-responsive">
      <table class="table table-striped">
        <thead>
          <tr>
            <th scope="col">Product</th>
            <th scope="col">Price</th>
            <th scope="col">Quantity</th>
            <th scope="col">Subtotal</th>
          </tr>
        </thead>
        <tbody>
          @foreach (var item in Model.OrderDetails)
          {
            <tr>
              <td>@item.Medicine.Name</td>
              <td>@item.MedicinePrice VND</td>
              <td>@item.Amount</td>
              <td>@item.SumPrice VND</td>
            </tr>
          }
        </tbody>
      </table>
    </div>
  }
  else
  {
    <p>No products in this order.</p>
  }
</div class="mt-3">

```

`<a asp-area="" asp-controller="Buy" asp-action="OrderList" class="btn btn-secondary">Back to Orders`

4.1.9. Quản lý tài khoản của admin



Hình 14: Giao diện danh sách tài khoản

@using ThucTap_ThuongMaiDienTu.Resources

@model IEnumerable<ThucTap_ThuongMaiDienTu.Models.Account>

@{

ViewData["Title"] = Resource.AccountsM;

ViewData["Active"] = "Accounts";

Layout = "~/Views/Shared/_LayoutAdmin.cshtml";

}

<!-- Table Start -->

<div class="container-fluid pt-4 px-4">

<div class="row g-4">

<div class="col-12">

<div class="bg-light rounded h-100 p-4">

<div class="d-flex justify-content-between">

<h3 class="mb-4 text-primary">@Resource.AccountsM</h3>

<a asp-action="Create">@Resource.AccountsN <i class="fa fa-plus" style="font-size: 20px;"></i>

</div>

<div class="table-responsive">

<table class="table">

<thead>

<tr>

<th>

@Html.DisplayNameFor(model => model.Username)

```

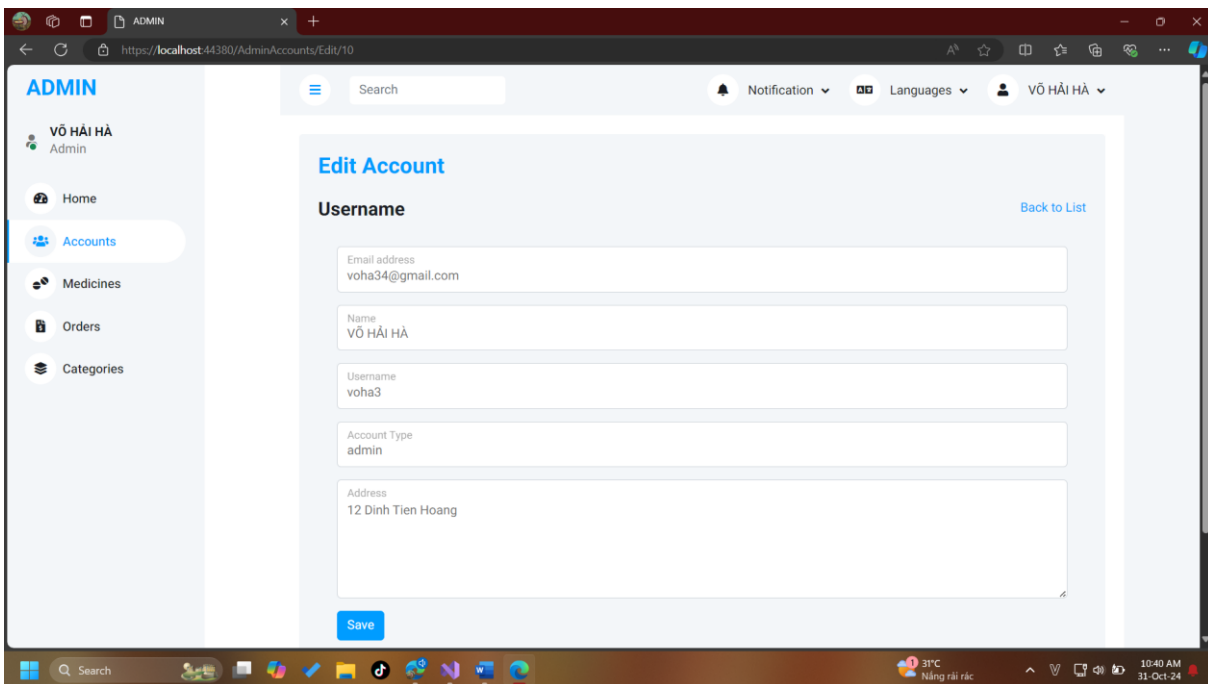
</th>
<th>
    @Html.DisplayNameFor(model => model.AccountType)
</th>
<th>
    @Html.DisplayNameFor(model => model.Code)
</th>
<th>
    @Html.DisplayNameFor(model => model.Active)
</th>
<th>
    @Html.DisplayNameFor(model => model.Email)
</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.Username)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.AccountType)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Code)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Active)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Email)
            </td>
            <td>
                <a asp-action="Edit" asp-route-id="@item.Id"><i class="far fa-edit" style="font-size:
25px"></i></a>
            </td>
            <td>

```

```

<a asp-action="Details" asp-route-id="@item.Id"><i class="fas fa-info-circle" style="font-size: 25px"></i></a>
</td>
<td>
<a asp-action="Delete" asp-route-id="@item.Id"><i class="fas fa-trash-alt" style="font-size: 25px"></i></a>
</td>
</tr>
}
</tbody>
</table>
<!-- Table End -->

```



Hình 15: Giao diện chỉnh sửa tài khoản

```
@model ThucTap_ThuongMaiDienTu.Models.Account
```

```
@using ThucTap_ThuongMaiDienTu.Resources
```

```
@{
```

```
ViewData["Title"] = Resource.AccountsM;
```

```
ViewData["Active"] = "Accounts";
```

```
Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
```

```
}
```

```
<!-- Table Start -->
```

```
<div class="container-fluid pt-4 px-4">
```

```
<div class="row g-4">
```

```
<div class="col-12">
```

```
<div class="bg-light rounded h-100 p-4">
```

```
<h3 class="mb-4 text-primary">Edit Account</h3>
```

```

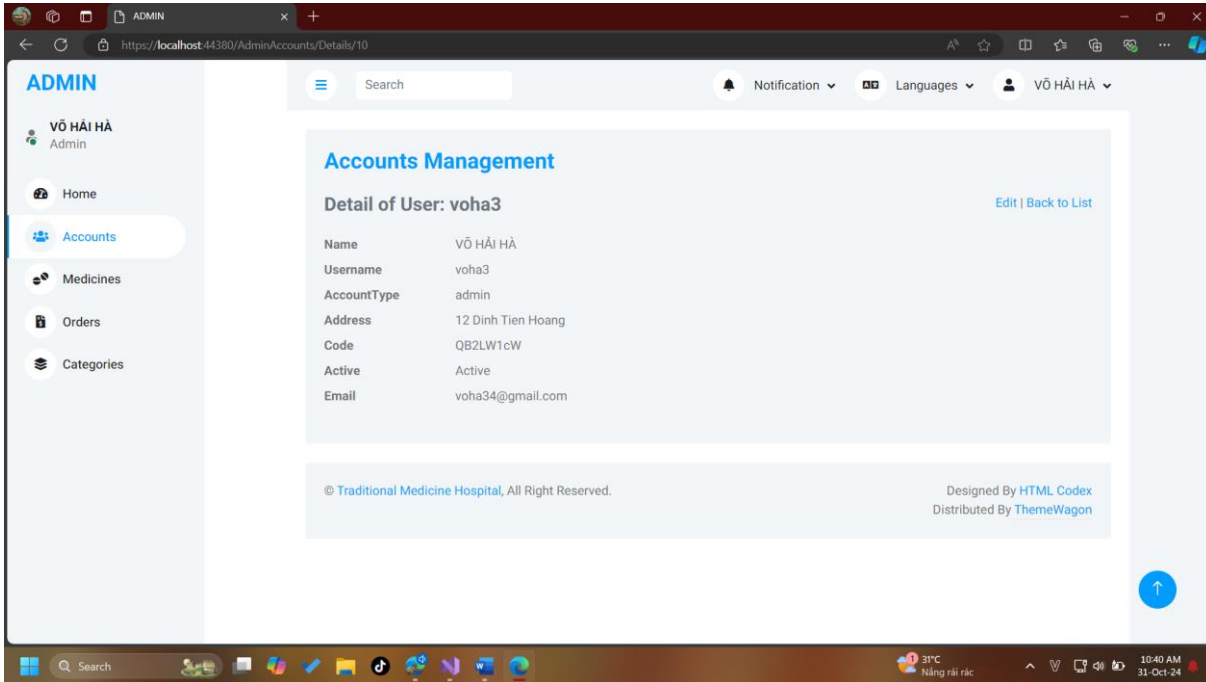
<div class="d-flex justify-content-between">
  <h4>@Html.DisplayNameFor(model => model.Username)</h4>
  <a asp-action="Index">Back to List</a>
</div>
<form asp-action="Edit">
  <div class="col-sm-12 col-xl-12">
    <div class="bg-light rounded h-100 p-4">
      <div asp-validation-summary="ModelOnly" class="text-danger"></div>
      <input type="hidden" asp-for="Id" />
      <input type="hidden" asp-for="Code" />
      <input type="hidden" asp-for="Password" />
      <input type="hidden" asp-for="Active" />
      <div class="form-floating mb-3">
        <input type="email" class="form-control" id="floatingInputEmail" asp-for="Email"
placeholder="name@example.com" />
        <label asp-for="Email">Email address</label>
        <span asp-validation-for="Email" class="text-danger"></span>
      </div>
      <div class="form-floating mb-3">
        <input type="text" class="form-control" id="floatingInputName" asp-for="Name"
placeholder="Full Name" />
        <label asp-for="Name">Name</label>
        <span asp-validation-for="Name" class="text-danger"></span>
      </div>
      <div class="form-floating mb-3">
        <input type="text" class="form-control" id="floatingInputUsername" asp-for="Username"
placeholder="Username" />
        <label asp-for="Username">Username</label>
        <span asp-validation-for="Username" class="text-danger"></span>
      </div>
      <div class="form-floating mb-3">
        <input type="text" class="form-control" id="floatingInputAccountType" asp-for="AccountType"
placeholder="Account Type" />
        <label asp-for="AccountType">Account Type</label>
        <span asp-validation-for="AccountType" class="text-danger"></span>
      </div>
      <div class="form-floating mb-3">
        <textarea class="form-control" placeholder="Address" asp-for="Address" id="floatingTextarea"
style="height: 150px;"></textarea>
        <label asp-for="Address">Address</label>
        <span asp-validation-for="Address" class="text-danger"></span>
      </div>
      <input type="submit" value="Save" class="btn btn-primary" />

```

```

    </div>
  </div>
  @section Scripts {
    @{
      await Html.RenderPartialAsync("_ValidationScriptsPartial");
    }
  }
</form>

```



Hình 16: Giao diện chi tiết tài khoản

```

@model ThucTap_ThuongMaiDienTu.Models.Account
@using ThucTap_ThuongMaiDienTu.Resources
@{
    ViewData["Title"] = Resource.AccountsM;
    ViewData["Active"] = "Accounts";
    Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
}
<!-- Table Start -->
<div class="container-fluid pt-4 px-4">
    <div class="row g-4">
        <div class="col-12">
            <div class="bg-light rounded h-100 p-4">
                <h3 class="mb-4 text-primary">
                    @Resource.AccountsM
                </h3>
                <div class="d-flex justify-content-between">
                    <h4 class="mb-4 text-secondary">

```

```

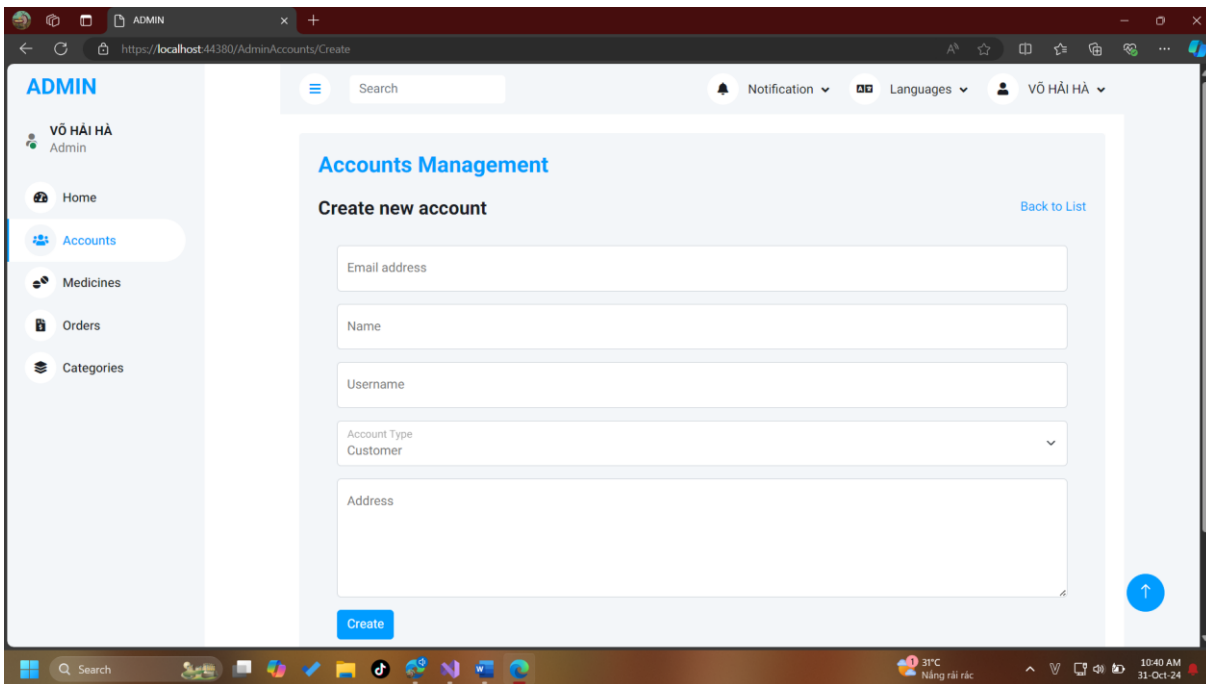
        @Resource.DetailsOfUser
        @Html.DisplayFor(model => model.Username)
    </h4>
    <div>
        <a asp-action="Edit" asp-route-id="@Model?.Id">Edit</a> |
        <a asp-action="Index">Back to List</a>
    </div>
</div>
<dl class="row">
    <dt class="col-sm-2">
        @Html.DisplayNameFor(model => model.Name)
    </dt>
    <dd class="col-sm-10">
        @Html.DisplayFor(model => model.Name)
    </dd>
    <dt class="col-sm-2">
        @Html.DisplayNameFor(model => model.Username)
    </dt>
    <dd class="col-sm-10">
        @Html.DisplayFor(model => model.Username)
    </dd>
    <dt class="col-sm-2">
        @Html.DisplayNameFor(model => model.AccountType)
    </dt>
    <dd class="col-sm-10">
        @Html.DisplayFor(model => model.AccountType)
    </dd>
    <dt class="col-sm-2">
        @Html.DisplayNameFor(model => model.Address)
    </dt>
    <dd class="col-sm-10">
        @Html.DisplayFor(model => model.Address)
    </dd>
    <dt class="col-sm-2">
        @Html.DisplayNameFor(model => model.Code)
    </dt>
    <dd class="col-sm-10">
        @Html.DisplayFor(model => model.Code)
    </dd>
    <dt class="col-sm-2">
        @Html.DisplayNameFor(model => model.Active)
    </dt>
    <dd class="col-sm-10">

```

```

@if (Model.Active)
{
    <span>Active</span>
}
else
{
    <span>Inactive</span>
}
</dd>
<dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Email)
</dt>
<dd class="col-sm-10">
    @Html.DisplayFor(model => model.Email)
</dd>
</dl>

```



Hình 17: Giao diện tạo tài khoản mới

```

@model ThucTap_ThuongMaiDienTu.Models.Account
@using ThucTap_ThuongMaiDienTu.Resources
@{
    ViewData["Title"] = Resource.AccountsM;
    ViewData["Active"] = "Accounts";
    Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
}
<!-- Table Start -->
<div class="container-fluid pt-4 px-4">

```



```

<div class="row g-4">
  <div class="col-12">
    <div class="bg-light rounded h-100 p-4">
      <h3 class="mb-4 text-primary">@Resource.AccountsM</h3>
      <div class="d-flex justify-content-between">
        <h4> @Resource.AccountsN</h4>
        <a asp-action="Index">Back to List</a>
      </div>
      <form asp-action="Create">
        <div class="col-sm-12 col-xl-12">
          <div class="bg-light rounded h-100 p-4">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-floating mb-3">
              <input type="email" class="form-control" id="floatingInputEmail" asp-for="Email"
placeholder="name@example.com" />
              <label asp-for="Email">Email address</label>
              <span asp-validation-for="Email" class="text-danger"></span>
            </div>
            <div class="form-floating mb-3">
              <input type="text" class="form-control" id="floatingInputName" asp-for="Name"
placeholder="Full Name" />
              <label asp-for="Name">Name</label>
              <span asp-validation-for="Name" class="text-danger"></span>
            </div>
            <div class="form-floating mb-3">
              <input type="text" class="form-control" id="floatingInputUsername" asp-for="Username"
placeholder="Username" />
              <label asp-for="Username">Username</label>
              <span asp-validation-for="Username" class="text-danger"></span>
            </div>
            <div class="form-floating mb-3">
              <select asp-for="AccountType" class="form-select" aria-label="Account type">
                <option value="customer" selected>Customer</option>
                <option value="admin">Admin</option>
              </select>
              <label asp-for="AccountType">Account Type</label>
              <span asp-validation-for="AccountType" class="text-danger"></span>
            </div>
            <div class="form-floating mb-3">
              <textarea class="form-control" placeholder="Address" asp-for="Address" id="floatingTextarea"
style="height: 150px;"></textarea>
              <label asp-for="Address">Address</label>
              <span asp-validation-for="Address" class="text-danger"></span>
            </div>
          </div>
        </div>
      </form>
    </div>
  </div>
</div>

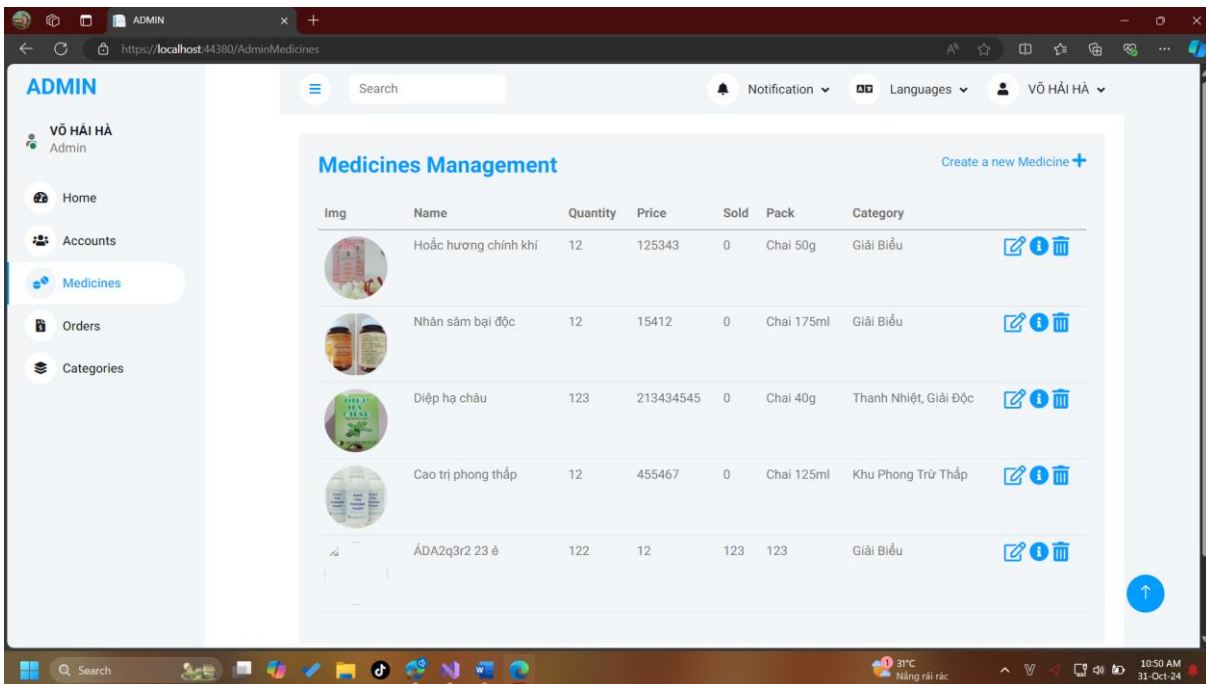
```

```

</div>
<input type="submit" value="Create" class="btn btn-primary" />
</div>
</div>
@section Scripts {
    @ {
        await Html.RenderPartialAsync("_ValidationScriptsPartial");
    }
}

```

4.1.10. Quản lý sản phẩm



Hình 18: Giao diện danh sách sản phẩm

```

@using ThucTap_ThuongMaiDienTu.Resources
@model IEnumerable<ThucTap_ThuongMaiDienTu.Models.Medicine>
@ {
    ViewData["Title"] = Resource.MedicinesM;
    ViewData["Active"] = "Medicines";
    Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
}
<!-- Table Start -->
<div class="container-fluid pt-4 px-4">
    <div class="row g-4">
        <div class="col-12">
            <div class="bg-light rounded h-100 p-4">
                <div class="d-flex justify-content-between">
                    <h3 class="mb-4 text-primary">@Resource.MedicinesM</h3>
                    <a asp-action="Create">@Resource.MedicinesN <i class="fa fa-plus" style="font-size: 20px;"></i></a>
                </div>
            </div>
        </div>
    </div>

```

```

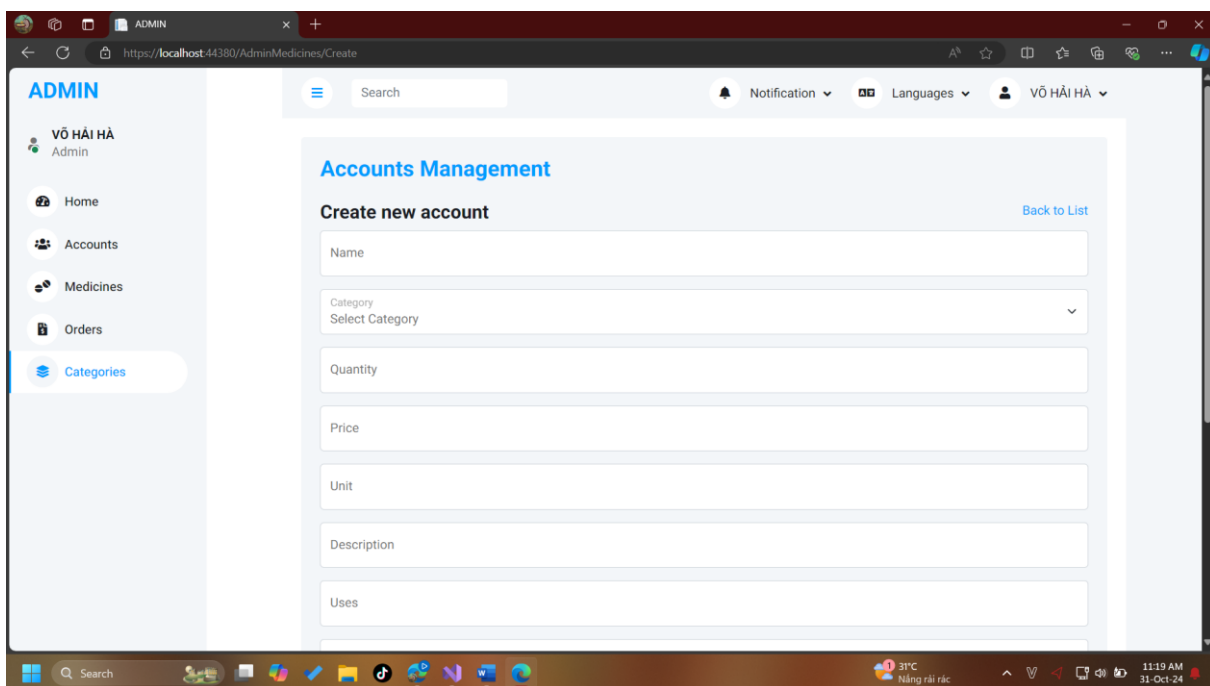
</div>
<div class="table-responsive">
  <table class="table">
    <thead>
      <tr>
        <th>
          @Html.DisplayNameFor(model => model.Image)
        </th>
        <th>
          @Html.DisplayNameFor(model => model.Name)
        </th>
        <th>
          @Html.DisplayNameFor(model => model.Quantity)
        </th>
        <th>
          @Html.DisplayNameFor(model => model.Price)
        </th>
        <th>
          @Html.DisplayNameFor(model => model.Sold)
        </th>
        <th>
          @Html.DisplayNameFor(model => model.Pack)
        </th>
        <th>
          @Html.DisplayNameFor(model => model.Category)
        </th>
      </tr>
    </thead>
    <tbody>
      @foreach (var item in Model)
      {
        <tr>
          <td>
            <div class="d-flex align-items-center">
              
            </div>
          </td>
          <td>
            @Html.DisplayFor(modelItem => item.Name)
          </td>
          <td>

```

```

        @Html.DisplayFor(modelItem => item.Quantity)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.Price)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.Sold)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.Pack)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.Category.Name)
    </td>
    <td>
        <a asp-action="Edit" asp-route-id="@item.Id"><i class="far fa-edit" style="font-size:
25px"></i></a>
        <a asp-action="Details" asp-route-id="@item.Id"><i class="fas fa-info-circle" style="font-
size: 25px"></i></a>
        <a asp-action="Delete" asp-route-id="@item.Id"><i class="fas fa-trash-alt" style="font-
size: 25px"></i></a>
    </td>
</tr>
}
</tbody>
</table>

```



Hình 19: Giao diện tạo sản phẩm mới

```

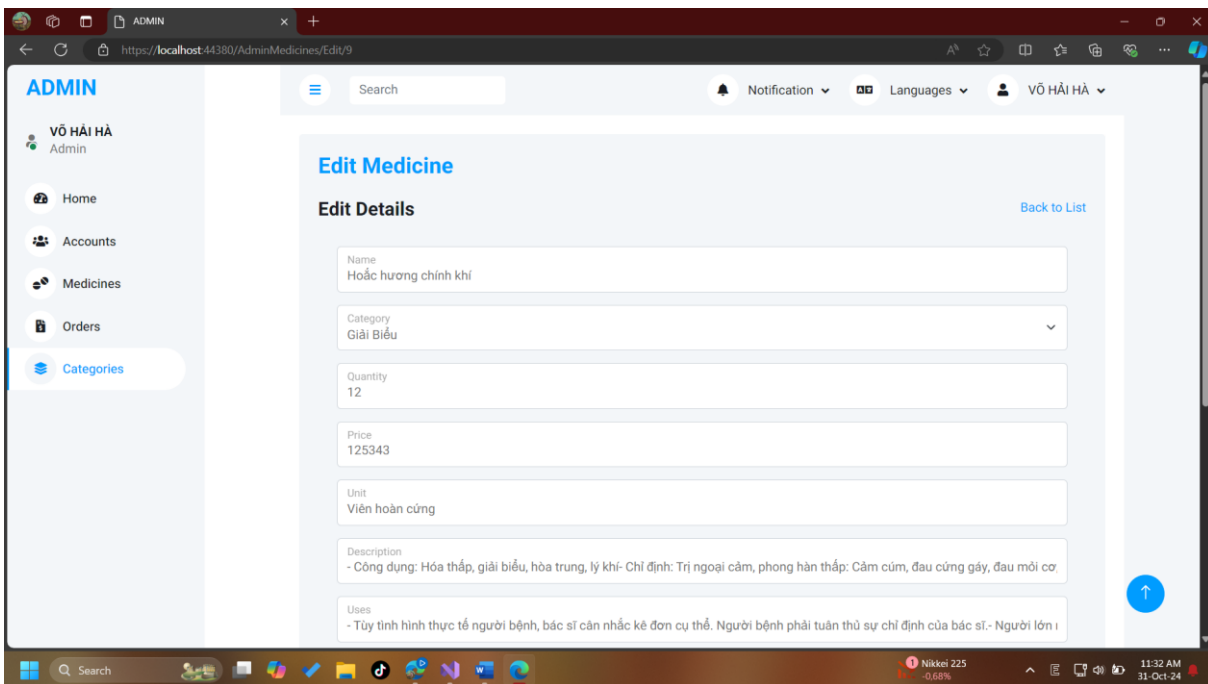
@model ThucTap_ThuongMaiDienTu.Models.Account
@using ThucTap_ThuongMaiDienTu.Resources
@{
    ViewData["Title"] = Resource.AccountsM;
    ViewData["Active"] = "Accounts";
    Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
}
<!-- Table Start -->
<div class="container-fluid pt-4 px-4">
    <div class="row g-4">
        <div class="col-12">
            <div class="bg-light rounded h-100 p-4">
                <h3 class="mb-4 text-primary">@Resource.AccountsM</h3>
                <div class="d-flex justify-content-between">
                    <h4> @Resource.AccountsN</h4>
                    <a asp-action="Index">Back to List</a>
                </div>
                <form asp-action="Create">
                    <div class="col-sm-12 col-xl-12">
                        <div class="bg-light rounded h-100 p-4">
                            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
                            <div class="form-floating mb-3">
                                <input type="email" class="form-control" id="floatingInputEmail" asp-for="Email"
placeholder="name@example.com" />
                                <label asp-for="Email">Email address</label>
                                <span asp-validation-for="Email" class="text-danger"></span>
                            </div>
                            <div class="form-floating mb-3">
                                <input type="text" class="form-control" id="floatingInputName" asp-for="Name"
placeholder="Full Name" />
                                <label asp-for="Name">Name</label>
                                <span asp-validation-for="Name" class="text-danger"></span>
                            </div>
                            <div class="form-floating mb-3">
                                <input type="text" class="form-control" id="floatingInputUsername" asp-for="Username"
placeholder="Username" />
                                <label asp-for="Username">Username</label>
                                <span asp-validation-for="Username" class="text-danger"></span>
                            </div>
                            <div class="form-floating mb-3">
                                <select asp-for="AccountType" class="form-select" aria-label="Account type">
                                    <option value="customer" selected>Customer</option>
                                    <option value="admin">Admin</option>
                                </select>
                            </div>

```

```

</select>
<label asp-for="AccountType">Account Type</label>
<span asp-validation-for="AccountType" class="text-danger"></span>
</div>
<div class="form-floating mb-3">
<textarea class="form-control" placeholder="Address" asp-for="Address" id="floatingTextarea"
style="height: 150px;"></textarea>
<label asp-for="Address">Address</label>
<span asp-validation-for="Address" class="text-danger"></span>
</div>
<input type="submit" value="Create" class="btn btn-primary" />
</div>
</div>
@section Scripts {
@{
await Html.RenderPartialAsync("_ValidationScriptsPartial");
}
}
</form>

```



Hình 20: Giao diện chỉnh sửa sản phẩm

```

@model ThucTap_ThuongMaiDienTu.ViewModels.MedicineCreateVM
@using ThucTap_ThuongMaiDienTu.Resources
@{
    ViewData["Title"] = Resource.CategoriesM;
    ViewData["Active"] = "Categories";
    Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
}

```

```

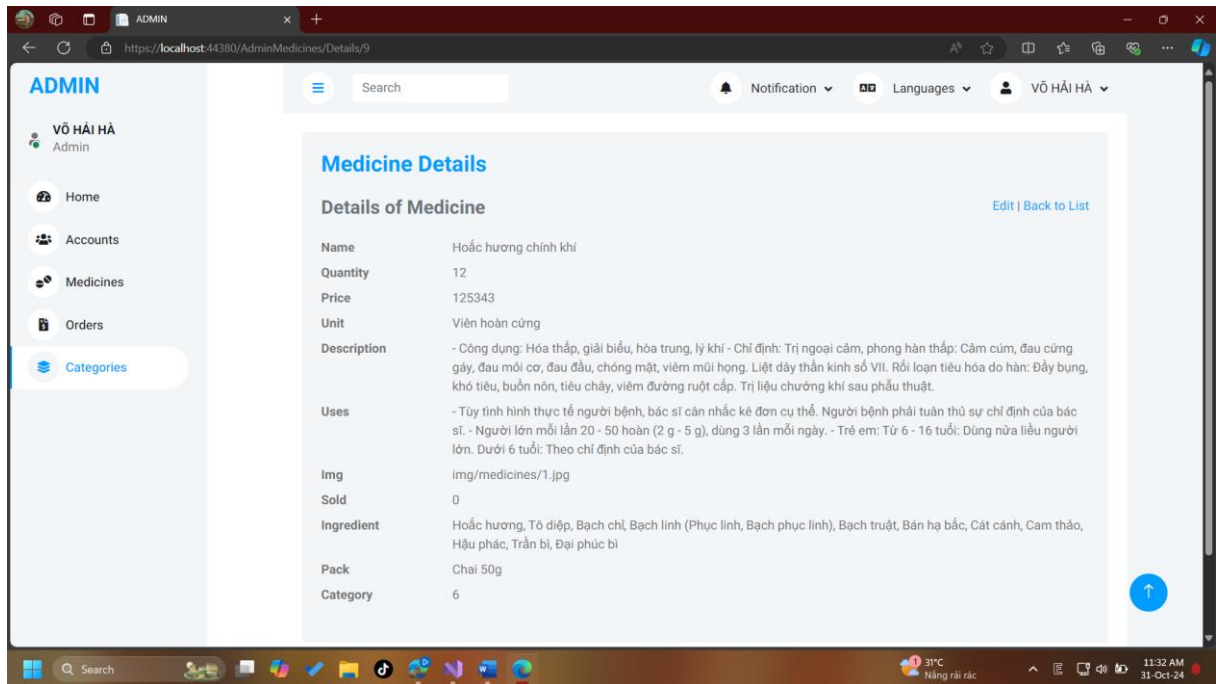
}
<div class="container-fluid pt-4 px-4">
  <div class="row g-4">
    <div class="col-12">
      <div class="bg-light rounded h-100 p-4">
        <h3 class="mb-4 text-primary">Edit Medicine</h3>
        <div class="d-flex justify-content-between">
          <h4>Edit Details</h4>
          <a asp-action="Index">Back to List</a>
        </div>
        <form asp-action="EditMedicine">
          <div class="col-sm-12 col-xl-12">
            <div class="bg-light rounded h-100 p-4">
              <div asp-validation-summary="ModelOnly" class="text-danger"></div>
              <input type="hidden" asp-for="Id" />
              <div class="form-floating mb-3">
                <input type="text" class="form-control" asp-for="Name" placeholder="Name" />
                <label asp-for="Name">Name</label>
                <span asp-validation-for="Name" class="text-danger"></span>
              </div>
              <div class="form-floating mb-3">
                <select asp-for="CategoryId" class="form-select" asp-items="ViewBag.CategoryId">
                  <option value="">Select Category</option>
                </select>
                <label asp-for="CategoryId">Category</label>
                <span asp-validation-for="CategoryId" class="text-danger"></span>
              </div>
              <div class="form-floating mb-3">
                <input type="number" class="form-control" asp-for="Quantity" placeholder="Quantity" />
                <label asp-for="Quantity">Quantity</label>
                <span asp-validation-for="Quantity" class="text-danger"></span>
              </div>
              <div class="form-floating mb-3">
                <input type="number" class="form-control" asp-for="Price" placeholder="Price" />
                <label asp-for="Price">Price</label>
                <span asp-validation-for="Price" class="text-danger"></span>
              </div>
              <div class="form-floating mb-3">
                <input type="text" class="form-control" asp-for="Unit" placeholder="Unit" />
                <label asp-for="Unit">Unit</label>
                <span asp-validation-for="Unit" class="text-danger"></span>
              </div>
              <div class="form-floating mb-3">

```

```

        <input type="text" class="form-control" asp-for="Description" placeholder="Description" />
        <label asp-for="Description">Description</label>
        <span asp-validation-for="Description" class="text-danger"></span>
    </div>
    <div class="form-floating mb-3">
        <input type="text" class="form-control" asp-for="Uses" placeholder="Uses" />
        <label asp-for="Uses">Uses</label>
        <span asp-validation-for="Uses" class="text-danger"></span>
    </div>
    <div class="form-floating mb-3">
        <input type="text" class="form-control" asp-for="Img" placeholder="Image URL" />
        <label asp-for="Img">Image URL</label>
        <span asp-validation-for="Img" class="text-danger"></span>
    </div>
    <div class="form-floating mb-3">
        <input type="number" class="form-control" asp-for="Sold" placeholder="Sold" />
        <label asp-for="Sold">Sold</label>
        <span asp-validation-for="Sold" class="text-danger"></span>
    </div>
    <div class="form-floating mb-3">
        <input type="text" class="form-control" asp-for="Ingredient" placeholder="Ingredient" />
        <label asp-for="Ingredient">Ingredient</label>
        <span asp-validation-for="Ingredient" class="text-danger"></span>
    </div>
    <div class="form-floating mb-3">
        <input type="text" class="form-control" asp-for="Pack" placeholder="Pack" />
        <label asp-for="Pack">Pack</label>
        <span asp-validation-for="Pack" class="text-danger"></span>
    </div>
    <input type="submit" value="Save" class="btn btn-primary" />
</div>
</div>
@section Scripts {
    @{
        await Html.RenderPartialAsync("_ValidationScriptsPartial");
    }
}
</form>

```

Hình 21: Giao diện chi tiết sản phẩm

```
@model ThucTap_ThuongMaiDienTu.Models.Medicine
```

```
@using ThucTap_ThuongMaiDienTu.Resources
```

```
@{
```

```
    ViewData["Title"] = Resource.CategoriesM;
```

```
    ViewData["Active"] = "Categories";
```

```
    Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
```

```
}
```

```
<div class="container-fluid pt-4 px-4">
```

```
    <div class="row g-4">
```

```
        <div class="col-12">
```

```
            <div class="bg-light rounded h-100 p-4">
```

```
                <h3 class="mb-4 text-primary">Medicine Details</h3>
```

```
                <div class="d-flex justify-content-between">
```

```
                    <h4 class="mb-4 text-secondary">Details of Medicine</h4>
```

```
                <div>
```

```
                    <a asp-action="Edit" asp-route-id="@Model?.Id">Edit</a> |
```

```
                    <a asp-action="Index">Back to List</a>
```

```
                </div>
```

```
            </div>
```

```
        </div>
```

```
        <dl class="row">
```

```
            <dt class="col-sm-2">
```

```
                @Html.DisplayNameFor(model => model.Name)
```

```
            </dt>
```

```
            <dd class="col-sm-10">
```

```
                @Html.DisplayFor(model => model.Name)
```

```

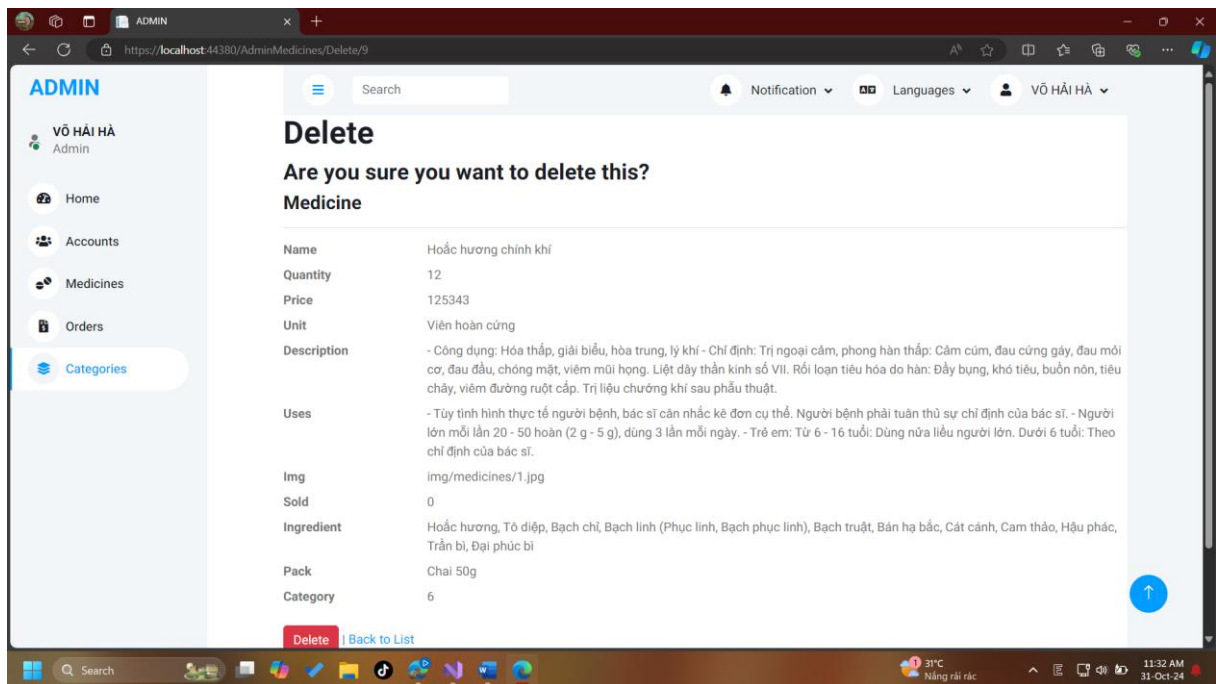
</dd>
<dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Quantity)
</dt>
<dd class="col-sm-10">
    @Html.DisplayFor(model => model.Quantity)
</dd>
<dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Price)
</dt>
<dd class="col-sm-10">
    @Html.DisplayFor(model => model.Price)
</dd>
<dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Unit)
</dt>
<dd class="col-sm-10">
    @Html.DisplayFor(model => model.Unit)
</dd>
<dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Description)
</dt>
<dd class="col-sm-10">
    @Html.DisplayFor(model => model.Description)
</dd>
<dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Uses)
</dt>
<dd class="col-sm-10">
    @Html.DisplayFor(model => model.Uses)
</dd>
<dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Image)
</dt>
<dd class="col-sm-10">
    @Html.DisplayFor(model => model.Image)
</dd>
<dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Sold)
</dt>
<dd class="col-sm-10">
    @Html.DisplayFor(model => model.Sold)
</dd>

```

```

<dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Ingredient)
</dt>
<dd class="col-sm-10">
    @Html.DisplayFor(model => model.Ingredient)
</dd>
<dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Pack)
</dt>
<dd class="col-sm-10">
    @Html.DisplayFor(model => model.Pack)
</dd>
<dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Category)
</dt>
<dd class="col-sm-10">
    @Html.DisplayFor(model => model.Category.Id)
</dd>
</dl>

```



Hình 22: Giao diện xác nhận xóa sản phẩm

```

@model ThucTap_ThuongMaiDienTu.Models.Medicine
@using ThucTap_ThuongMaiDienTu.Resources
@{
    ViewData["Title"] = Resource.CategoriesM;
    ViewData["Active"] = "Categories";
    Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
}

```

```

}
<h1>Delete</h1>
<h3>Are you sure you want to delete this?</h3>
<div>
  <h4>Medicine</h4>
  <hr />
  <dl class="row">
    <dt class="col-sm-2">
      @Html.DisplayNameFor(model => model.Name)
    </dt>
    <dd class="col-sm-10">
      @Html.DisplayFor(model => model.Name)
    </dd>
    <dt class="col-sm-2">
      @Html.DisplayNameFor(model => model.Quantity)
    </dt>
    <dd class="col-sm-10">
      @Html.DisplayFor(model => model.Quantity)
    </dd>
    <dt class="col-sm-2">
      @Html.DisplayNameFor(model => model.Price)
    </dt>
    <dd class="col-sm-10">
      @Html.DisplayFor(model => model.Price)
    </dd>
    <dt class="col-sm-2">
      @Html.DisplayNameFor(model => model.Unit)
    </dt>
    <dd class="col-sm-10">
      @Html.DisplayFor(model => model.Unit)
    </dd>
    <dt class="col-sm-2">
      @Html.DisplayNameFor(model => model.Description)
    </dt>
    <dd class="col-sm-10">
      @Html.DisplayFor(model => model.Description)
    </dd>
    <dt class="col-sm-2">
      @Html.DisplayNameFor(model => model.Uses)
    </dt>
    <dd class="col-sm-10">
      @Html.DisplayFor(model => model.Uses)
    </dd>
  </dl>

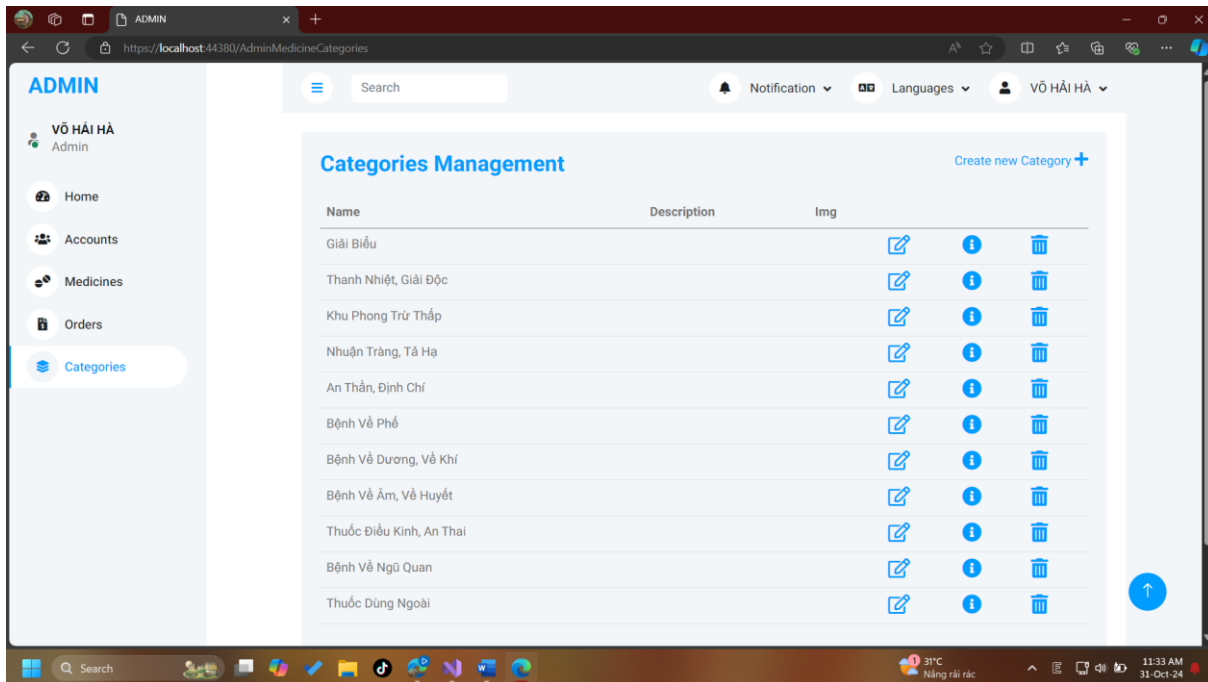
```

```

<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Image)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Image)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Sold)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Sold)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Ingredient)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Ingredient)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Pack)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Pack)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Category)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Category.Id)
</dd>
</div>
<form asp-action="Delete">
    <input type="hidden" asp-for="Id" />
    <input type="submit" value="Delete" class="btn btn-danger" /> |
    <a asp-action="Index">Back to List</a>
</form>
</div>

```

4.1.11. Quản lý danh mục thuốc



Hình 23: Giao diện danh sách danh mục

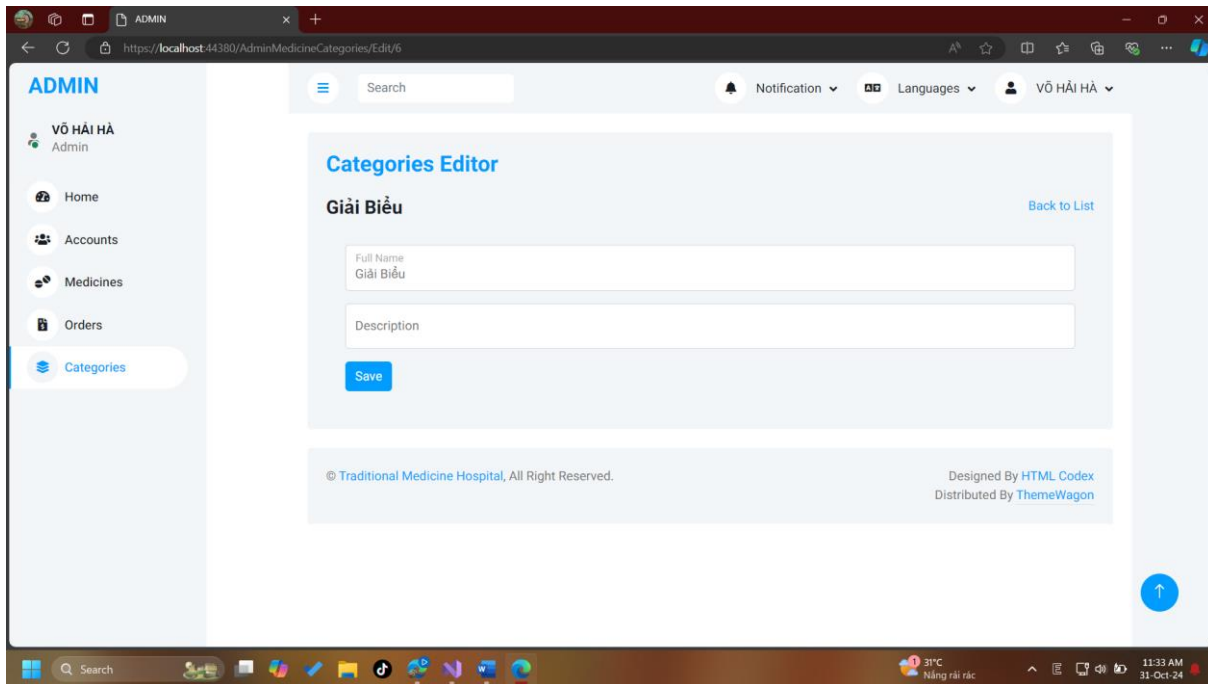
```
@model IEnumerable<ThucTap_ThuongMaiDienTu.Models.MedicineCategory>
@using ThucTap_ThuongMaiDienTu.Resources
@{
    ViewData["Title"] = Resource.CategoriesM;
    ViewData["Active"] = "Categories";
    Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
}

<!-- Table Start -->
<div class="container-fluid pt-4 px-4">
    <div class="row g-4">
        <div class="col-12">
            <div class="bg-light rounded h-100 p-4">
                <div class="d-flex justify-content-between">
                    <h3 class="mb-4 text-primary">@Resource.CategoriesM</h3>
                    <a asp-action="Create">@Resource.CategoriesN <i class="fa fa-plus" style="font-size: 20px;"></i></a>
                </div>
                <div class="table-responsive">
                    <table class="table">
                        <thead>
                            <tr>
                                <th>
                                    @Html.DisplayNameFor(model => model.Name)
                                </th>
                                <th>
```

```

        @Html.DisplayNameFor(model => model.Description)
    </th>
    <th>
        @Html.DisplayNameFor(model => model.Image)
    </th>
    <th></th>
    <th></th>
    <th></th>
</tr>
</thead>
<tbody>
    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.Name)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Description)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Image)
            </td>
            <td>
                <a asp-action="Edit" asp-route-id="@item.Id"><i class="fas fa-edit" style="font-size: 25px;"></i></a>
            </td>
            <td>
                <a asp-action="Details" asp-route-id="@item.Id"><i class="fas fa-info-circle" style="font-size:
25px;"></i></a>
            </td>
            <td>
                <a asp-action="Delete" asp-route-id="@item.Id"><i class="fas fa-trash-alt" style="font-size:
25px;"></i></a>
            </td>
        </tr>
    }
</tbody>
</table>

```



Hình 24: Giao diện chỉnh sửa danh mục

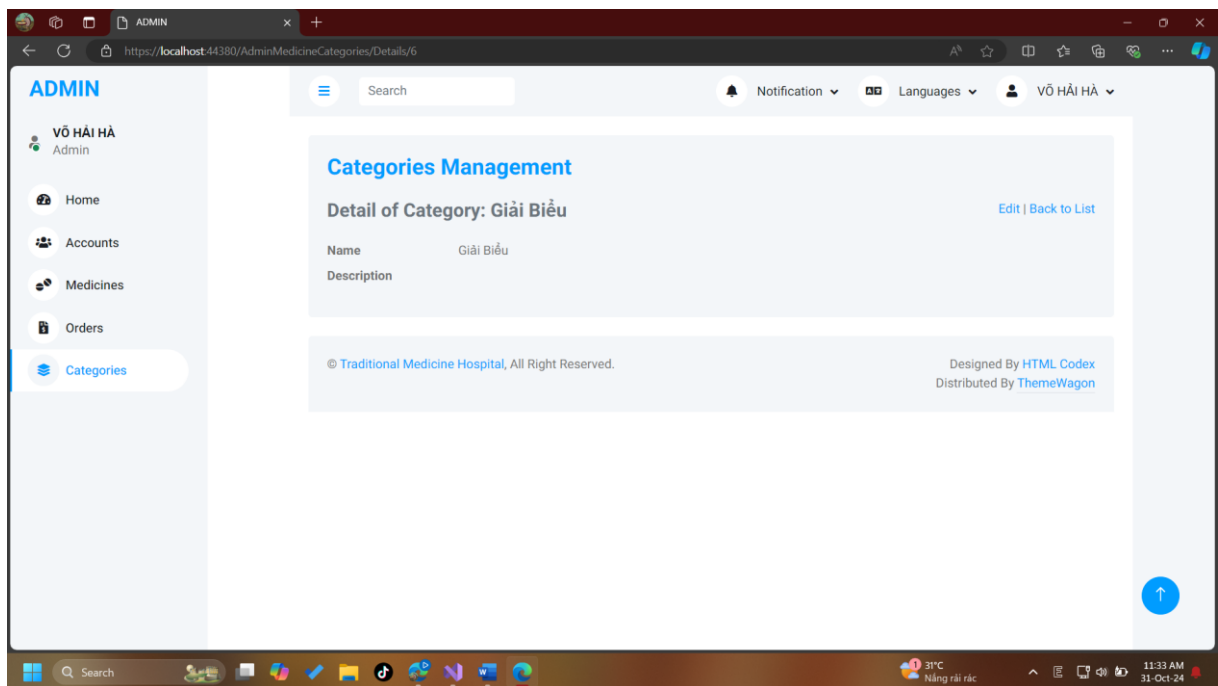
```
@model ThucTap_ThuongMaiDienTu.Models.MedicineCategory
@using ThucTap_ThuongMaiDienTu.Resources
@{
    ViewData["Title"] = @Resource.CategoriesM;
    ViewData["Active"] = "Categories";
    Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
}
<!-- Table Start -->
<div class="container-fluid pt-4 px-4">
    <div class="row g-4">
        <div class="col-12">
            <div class="bg-light rounded h-100 p-4">
                <h3 class="mb-4 text-primary">@Resource.CategoriesE</h3>
                <div class="d-flex justify-content-between">
                    <h4>@Html.DisplayFor(model => model.Name)</h4>
                    <a asp-action="Index">Back to List</a>
                </div>
                <form asp-action="Edit">
                    <div class="col-sm-12 col-xl-12">
                        <div class="bg-light rounded h-100 p-4">
                            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
                            <input type="hidden" asp-for="Id" />
                            <input type="hidden" asp-for="Img" />
                            <div class="form-floating mb-3">
```



```

        <input type="text" class="form-control" id="floatingInputEmail" asp-for="Name"
placeholder="Name" />
        <label asp-for="Name">@Resource.Name</label>
        <span asp-validation-for="Name" class="text-danger"></span>
    </div>
    <div class="form-floating mb-3">
        <input type="text" class="form-control" id="floatingInputName" asp-for="Description"
placeholder="Description" />
        <label asp-for="Description">@Resource.Description</label>
        <span asp-validation-for="Description" class="text-danger"></span>
    </div>
    <input type="submit" value="Save" class="btn btn-primary" />
</div>
</div>
@section Scripts {
    @ {
        await Html.RenderPartialAsync("_ValidationScriptsPartial");
    }
}
</form>

```



Hình 25: Giao diện chi tiết danh mục

```
@model ThucTap_ThuongMaiDienTu.Models.MedicineCategory
```

```
@using ThucTap_ThuongMaiDienTu.Resources
```

```
@ {
```

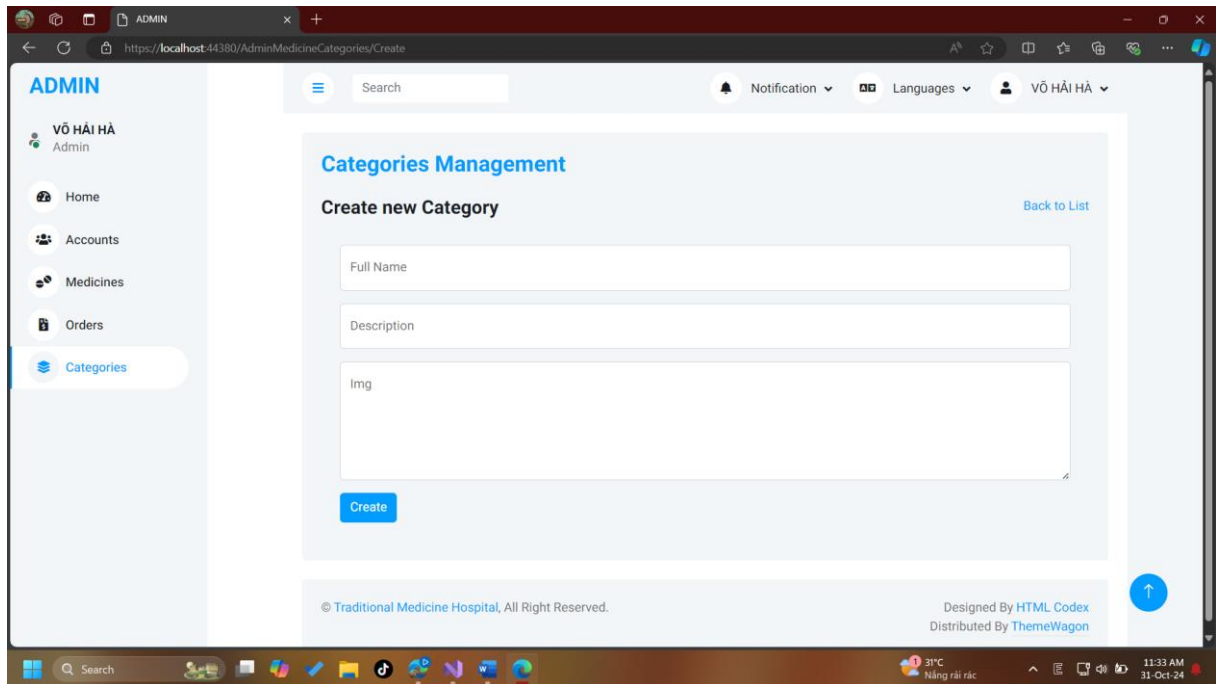
```
    ViewData["Title"] = Resource.CategoriesM;
```

```
    ViewData["Active"] = "Categories";
```

```

Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
}
<!-- Table Start -->
<div class="container-fluid pt-4 px-4">
  <div class="row g-4">
    <div class="col-12">
      <div class="bg-light rounded h-100 p-4">
        <h3 class="mb-4 text-primary">
          @Resource.CategoriesM
        </h3>
        <div class="d-flex justify-content-between">
          <h4 class="mb-4 text-secondary">
            @Resource.DetailsOfCategories
            @Html.DisplayFor(model => model.Name)
          </h4>
          <div>
            <a asp-action="Edit" asp-route-id="@Model?.Id">Edit</a> |
            <a asp-action="Index">Back to List</a>
          </div>
        </div>
      </div>
    </div>
    <dl class="row">
      <dt class="col-sm-2">
        @Html.DisplayNameFor(model => model.Name)
      </dt>
      <dd class="col-sm-10">
        @Html.DisplayFor(model => model.Name)
      </dd>
      <dt class="col-sm-2">
        @Html.DisplayNameFor(model => model.Description)
      </dt>
      <dd class="col-sm-10">
        @Html.DisplayFor(model => model.Description)
      </dd>
    </dl>
  </div>
</div>

```



Hình 26: Giao diện khởi tạo danh mục mới

```
@model ThucTap_ThuongMaiDienTu.Models.MedicineCategory
```

```
@using ThucTap_ThuongMaiDienTu.Resources
```

```
@{
```

```
    ViewData["Title"] = Resource.CategoriesM;
```

```
    ViewData["Active"] = "Categories";
```

```
    Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
```

```
}
```

```
<!-- Table Start -->
```

```
<div class="container-fluid pt-4 px-4">
```

```
    <div class="row g-4">
```

```
        <div class="col-12">
```

```
            <div class="bg-light rounded h-100 p-4">
```

```
                <h3 class="mb-4 text-primary">@Resource.CategoriesM</h3>
```

```
                <div class="d-flex justify-content-between">
```

```
                    <h4>@Resource.CategoriesN</h4>
```

```
                    <a asp-action="Index">@Resource.BackToList</a>
```

```
                </div>
```

```
                <form asp-action="Create">
```

```
                    <div class="col-sm-12 col-xl-12">
```

```
                        <div class="bg-light rounded h-100 p-4">
```

```
                            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
```

```
                            <div class="form-floating mb-3">
```

```
                                <input type="text" class="form-control" id="floatingInputEmail" asp-for="Name"
```

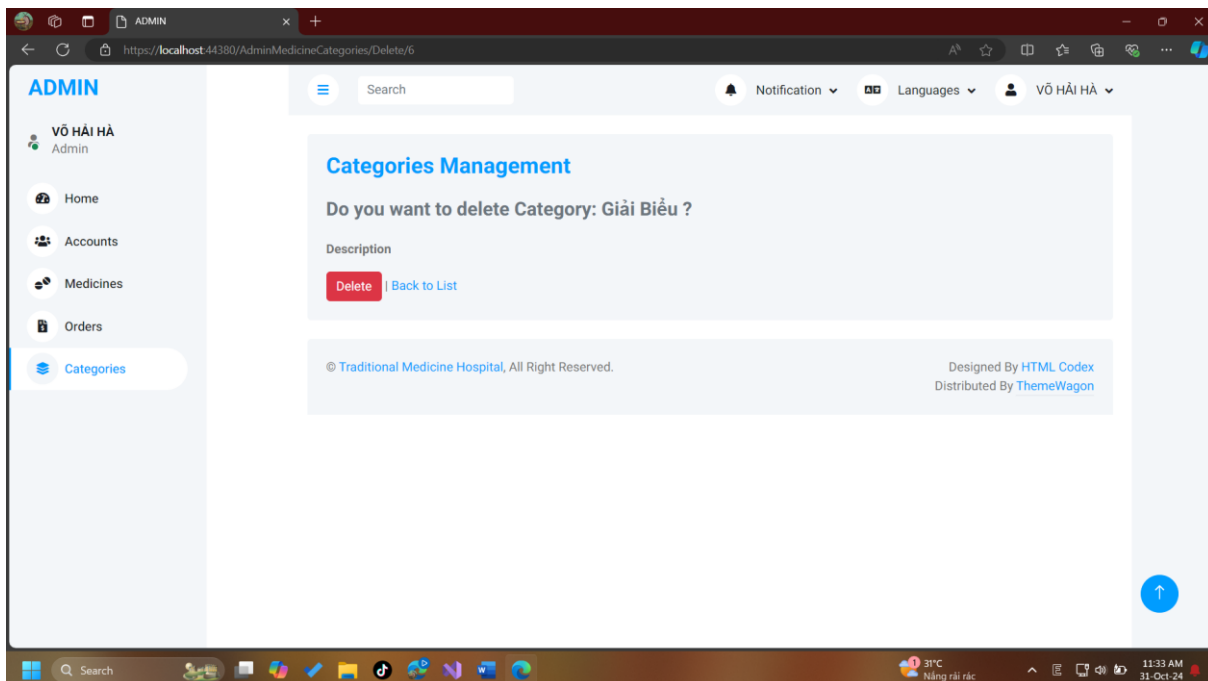
```
                                placeholder="@Resource.Name" />
```

```
                                <label asp-for="Name">@Resource.Name</label>
```

```

        <span asp-validation-for="Name" class="text-danger"></span>
    </div>
    <div class="form-floating mb-3">
        <input type="text" class="form-control" id="floatingInputUsername" asp-for="Description"
placeholder="@Resource.Description" />
        <label asp-for="Description">@Resource.Description</label>
        <span asp-validation-for="Description" class="text-danger"></span>
    </div>
    <div class="form-floating mb-3">
        <textarea class="form-control" placeholder="Address" asp-for="Img" id="floatingTextarea"
style="height: 150px;"></textarea>
        <label asp-for="Img">Img</label>
        <span asp-validation-for="Img" class="text-danger"></span>
    </div>
    <input type="submit" value="Create" class="btn btn-primary" />
</div>
</div>
@section Scripts {
    @{
        await Html.RenderPartialAsync("_ValidationScriptsPartial");
    }
}
</form>

```



Hình 27: Giao diện xác nhận xóa danh mục

@model ThucTap_ThuongMaiDienTu.Models.MedicineCategory

@using ThucTap_ThuongMaiDienTu.Resources

```

@{
    ViewData["Title"] = Resource.CategoriesM;
    ViewData["Active"] = "Categories";
    Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
}
<!-- Table Start -->
<div class="container-fluid pt-4 px-4">
    <div class="row g-4">
        <div class="col-12">
            <div class="bg-light rounded h-100 p-4">
                <h3 class="mb-4 text-primary">
                    @Resource.CategoriesM
                </h3>
                <h4 class="mb-4 text-secondary">
                    @Resource.DeleteCategories
                    @Html.DisplayFor(model => model.Name) ?
                </h4>
                <dl class="row">
                    <dt class="col-sm-2">
                        @Html.DisplayNameFor(model => model.Description)
                    </dt>
                    <dd class="col-sm-10">
                        @Html.DisplayFor(model => model.Description)
                    </dd>
                </dl>
                <form asp-action="Delete">
                    <input type="hidden" asp-for="Id" />
                    <input type="submit" value="Delete" class="btn btn-danger" /> |
                    <a asp-action="Index">Back to List</a>
                </form>
            </div>
        </div>
    </div>
</div>

```

4.2. XÂY DỰNG MODEL DỮ LIỆU

4.2.1. Model Tài khoản

```

public partial class Account
{
    public int Id { get; set; }
    public string? Name { get; set; }
    public string Username { get; set; } = null!;
    public string Password { get; set; } = null!;
    public string AccountType { get; set; } = null!;
    public string? Address { get; set; }
    public string Code { get; set; } = null!;
}

```

```

public bool Active { get; set; }
public string Email { get; set; } = null!;
public virtual ICollection<Cart> Carts { get; set; } = new List<Cart>();
public virtual ICollection<Order> Orders { get; set; } = new List<Order>();
public virtual ICollection<Rating> Ratings { get; set; } = new List<Rating>();
}

```

4.2.2. Model Giỏ hàng

```

public partial class Cart
{
    public int Id { get; set; }
    public int AccountId { get; set; }
    public int? Total { get; set; }
    public virtual Account Account { get; set; } = null!;
    public virtual ICollection<CartDetail> CartDetails { get; set; } = new List<CartDetail>();
}

```

4.2.3. Model Chi tiết giỏ hàng

```

public partial class CartDetail
{
    public int CartId { get; set; }
    public int MedicineId { get; set; }
    public int? Amount { get; set; }
    public int? Total { get; set; }
    public virtual Cart Cart { get; set; } = null!;
    public virtual Medicine Medicine { get; set; } = null!;
}

```

4.2.4. Model Đăng nhập

```

public partial class Login
{
    public string Username { get; set; } = null!;
    public string Password { get; set; } = null!;
}

```

4.2.5. Model Đăng kí

```

public partial class Logup
{
    public string? Name { get; set; }
    public string Username { get; set; } = null!;
    public string Password { get; set; } = null!;
    public string CFPASSWORD { get; set; } = null!;
    public string? Address { get; set; }
    public string Email { get; set; } = null!;
}

```

4.2.6. Model Thuốc

```
public partial class Medicine
{
    public int Id { get; set; }
    public string Name { get; set; } = null!;
    public int CategoryId { get; set; }
    public int? Quantity { get; set; }
    public int? Price { get; set; }
    public string? Unit { get; set; }
    public string? Description { get; set; }
    public string? Uses { get; set; }
    public string? Img { get; set; }
    public int? Sold { get; set; }
    public string? Ingredient { get; set; }
    public string? Pack { get; set; }
    public virtual ICollection<CartDetail> CartDetails { get; set; } = new List<CartDetail>();
    public virtual MedicineCategory Category { get; set; } = null!;
    public virtual ICollection<OrderDetail> OrderDetails { get; set; } = new List<OrderDetail>();
    public virtual ICollection<Rating> Ratings { get; set; } = new List<Rating>();
}
```

4.2.7. Model Danh mục thuốc

```
public partial class MedicineCategory
{
    public int Id { get; set; }
    public string Name { get; set; } = null!;
    public string? Img { get; set; }
    public string? Description { get; set; }
    public virtual ICollection<Medicine> Medicines { get; set; } = new List<Medicine>();
}
```

4.2.8. Model Đơn hàng

```
public partial class Order
{
    public int Id { get; set; }
    public int AccountId { set; set; }
    public DateTime Date { get; set; }
    public int? Total { get; set; }
    public string? Address { get; set; }
    public bool Status { get; set; }
    public virtual Account Account { get; set; } = null!;
    public virtual ICollection<OrderDetail> OrderDetails { get; set; } = new List<OrderDetail>();
}
```

4.2.9. Model Chi tiết đơn hàng

```
public partial class OrderDetail
{
    public int OrderId { get; set; }
    public int MedicineId { get; set; }
    public int? Amount { get; set; }
    public int? MedicinePrice { get; set; }
    public int? SumPrice { get; set; }
    public virtual Medicine Medicine { get; set; } = null!;
    public virtual Order Order { get; set; } = null!;
}
```

4.2.10. Model Đánh giá

```
public partial class Rating
{
    public int MedicineId { get; set; }
    public int AccountId { get; set; }
    public int Stars { get; set; }
    public string? Comment { get; set; }
    public string? Reply { get; set; }
    public virtual Account Account { get; set; } = null!;
    public virtual Medicine Medicine { get; set; } = null!;
}
```

4.2.11. ViewModel Danh mục thuốc

```
public partial class MedicineCategoryVM
{
    public int Id { get; set; }
    public string? Name { get; set; }
    public int Number { get; set; }
}
```

4.2.12. ViewModel Khởi tạo thuốc

```
public class MedicineCreateVM
{
    public int Id { get; set; }
    [Required(ErrorMessage = "Name is required.")]
    public string Name { get; set; } = null!;
    [Required(ErrorMessage = "Category is required.")]
    public int CategoryId { get; set; }
    [Range(0, int.MaxValue, ErrorMessage = "Quantity must be a non-negative number.")]
    public int? Quantity { get; set; }
    [Range(0, int.MaxValue, ErrorMessage = "Price must be a non-negative number.")]
    public int? Price { get; set; }
```



```

    public string? Unit { get; set; }
    public string? Description { get; set; }
    public string? Uses { get; set; }
    public string? Img { get; set; }
    public int? Sold { get; set; }
    public string? Ingredient { get; set; }
    public string? Pack { get; set; }
}
}

```

4.2.13. Viewmodel Danh sách thuốc

```

public partial class MedicineVM
{
    public int Id { get; set; }
    public string? Img { get; set; }
    public string Category { get; set; }
    public int CategoryID { get; set; }
    public int Price { get; set; }
    public string? Description { get; set; }
    public string? Uses { get; set; }
    public string? Unit { get; set; }
    public string? Name { get; set; }
    public double AverageRating { get; set; }
    public int RatingsCount { get; set; }
    public string? Ingredient { get; set; }
    public string? Pack { get; set; }
    public List<RatingVM> Ratings { get; set; }
}

```

4.2.14. Viewmodel Đánh giá thuốc

```

public class RatingVM
{
    public int MedicineId { get; set; }
    public string? Account { get; set; }
    public int Stars { get; set; } // Rating value (e.g., out of 5)
    public string? Comment { get; set; } // The actual comment
}

```

4.3. XÂY DỰNG CHỨC NĂNG PHẦN MỀM

4.3.1. Xây dựng chức năng xác thực tài khoản

4.3.1.1. Khởi tạo mã JWT xác thực tài khoản

```

public static string GenerateToken(Account user)

```

```

{
    var config = new ConfigurationBuilder()
        .AddJsonFile(Path)
        .Build()
        .GetSection("Jwt");
    var key = Encoding.UTF8.GetBytes(config["SecretKey"]);
    var issuer = config["Issuer"];
    var audience = config["Audience"];
    if (string.IsNullOrEmpty(issuer) || string.IsNullOrEmpty(audience) || key.Length < 32)
        throw new InvalidOperationException("JWT configuration is missing or invalid.");
    var claims = new[]
    {
        new Claim(ClaimTypes.Name, user.Name),
        new Claim(ClaimTypes.Role, user.AccountType),
        new Claim("UserId", user.Id.ToString())
    };
    var tokenDescriptor = new SecurityTokenDescriptor
    {
        Subject = new ClaimsIdentity(claims),
        Expires = DateTime.UtcNow.AddHours(1), // Set token expiration
        Issuer = issuer, // Set Issuer
        Audience = audience, // Set Audience
        SigningCredentials = new SigningCredentials(
            new SymmetricSecurityKey(key), SecurityAlgorithms.HmacSha256Signature)
    };
    var tokenHandler = new JwtSecurityTokenHandler();
    var token = tokenHandler.CreateToken(tokenDescriptor);
    return tokenHandler.WriteToken(token);
}

```

4.3.1.2. *Xác thực mã JWT*

```

public static ClaimsPrincipal? ValidateTokenFromRequest(HttpRequest request, out SecurityToken validatedToken)
{
    validatedToken = null;
    var jwtToken = request.Cookies["jwtToken"];
    if (string.IsNullOrEmpty(jwtToken))
    {
        Console.WriteLine("JWT token not found.");
        return null;
    }
    var secretKey = new ConfigurationBuilder()
        .AddJsonFile(Path)
        .Build().GetSection("Jwt")["SecretKey"];

```

```

        return ValidateToken(jwtToken, secretKey, out validatedToken);
    }
    public static ClaimsPrincipal? ValidateToken(string token, string secretKey, out SecurityToken validatedToken)
    {
        var tokenHandler = new JwtSecurityTokenHandler();
        var key = Encoding.UTF8.GetBytes(secretKey);

        try
        {
            var claimsPrincipal = tokenHandler.ValidateToken(token, new TokenValidationParameters
            {
                ValidateIssuerSigningKey = true,
                IssuerSigningKey = new SymmetricSecurityKey(key),
                ValidateIssuer = false, // Set to `true` if you validate the issuer
                ValidateAudience = false, // Set to `true` if you validate the audience
                ValidateLifetime = true,
                ClockSkew = TimeSpan.Zero // Optional: adjust for server time differences
            }, out validatedToken);
            return claimsPrincipal;
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Token validation failed: {ex.Message}");
            validatedToken = null;
            return null;
        }
    }
    public static string? GetClaimValue(ClaimsPrincipal principal, string claimType)
    {
        return principal?.FindFirst(claimType)?.Value;
    }
    public static bool TryAuthenticateUser(HttpRequest request, HttpContext context, out ClaimsPrincipal? principal)
    {
        principal = ValidateTokenFromRequest(request, out SecurityToken validatedToken);
        if (principal == null)
        {
            Console.WriteLine("Invalid JWT token.");
            return false;
        }
        context.User = principal;
        return true;
    }
}

```

4.3.1.3. Đăng nhập

[HttpGet]

```
public IActionResult Login()
{
    // Check if the user is already authenticated via JWT
    if (JwtTokenHelper.TryAuthenticateUser(Request, HttpContext, out var principal))
    {
        return RedirectToAction("Index"); // Redirect to Index if already logged in
    }
    return View(); // Show login view for unauthenticated users
}
```

[HttpPost]

[ValidateAntiForgeryToken]

```
public IActionResult Login(Login login)
{
    var user = db.Accounts.SingleOrDefault(u => u.Username == login.Username);
    if (user == null || !VerifyPassword(login.Password, user.Password))
    {
        ModelState.AddModelError(string.Empty, "Invalid username or password.");
        return View(); // Return the same view with error message
    }
    if (!user.Active)
    {
        ModelState.AddModelError(string.Empty, "Your account is not activated. Please check your email for the activation code.");
        TempData["Username"] = user.Username; // Store username temporarily
        return RedirectToAction("Activate"); // Redirect to the activation page
    }
    var token = JwtTokenHelper.GenerateToken(user);
    Response.Cookies.Append("jwtToken", token, new CookieOptions
    {
        HttpOnly = true,
        Expires = DateTime.UtcNow.AddHours(1) // Set token expiry
    });
    HttpContext.Session.SetString("UserAccount", Newtonsoft.Json.JsonConvert.SerializeObject(user));
    return user.AccountType == "admin" ? RedirectToAction("Index", "AdminHome") : RedirectToAction("Index");
}
```

4.3.1.4. Đăng kí

```
public IActionResult Logup()
```

```
{
    if (JwtTokenHelper.TryAuthenticateUser(Request, HttpContext, out var principal))
    {
```

```

        return RedirectToAction("Index");
    }
    return View();
}
[HttpPost]
public IActionResult Logup(Logup logup)
{
    if (logup.Password != logup.CFPassword)
    {
        ModelState.AddModelError(string.Empty, "Passwords do not match.");
        return View();
    }
    var existingUser = db.Accounts
        .SingleOrDefault(u => u.Username == logup.Username || u.Email == logup.Email);
    if (existingUser != null)
    {
        ModelState.AddModelError(string.Empty, "Username or Email is already exists");
        return View();
    }
    var Code = GenerateRandomCode();
    var newUser = new Account
    {
        Username = logup.Username,
        Password = HashPassword(logup.Password), // Store hashed password
        Name = logup.Name,
        Address = logup.Address,
        Email = logup.Email,
        AccountType = "customer", // Set a default role
        Active = false,
        Code = Code,
    };
    db.Accounts.Add(newUser);
    db.SaveChanges();
    var subject = "Account Activation";
    var body = $"<p>Welcome, {logup.Name}!</p>" +
        $"<p>Your activation code is: <strong>{Code}</strong></p>" +
        "<p>Please enter this code to activate your account.</p>";
    EmailHelper.SendEmail(logup.Email, subject, body);
    TempData["Username"] = newUser.Username; // Pass username to the view
    return RedirectToAction("Activate");
}

```

4.3.1.5. Xác thực tài khoản

```
[HttpGet]
public IActionResult Activate()
{
    if (JwtTokenHelper.TryAuthenticateUser(Request, HttpContext, out var principal))
    {
        return RedirectToAction("Index");
    }
    return View();
}

[HttpPost]
public IActionResult Activate(string Username, string Code)
{
    var user = db.Accounts.SingleOrDefault(u => u.Username == Username && u.Code == Code);
    if (user == null)
    {
        ModelState.AddModelError(string.Empty, "Invalid activation code.");
        return View();
    }
    user.Active = true; // Activate the user
    db.SaveChanges();
    return RedirectToAction("Login");
}
```

4.3.1.6. Mã hoá mật khẩu

```
public static string HashPassword(string password)
{
    var salt = new ConfigurationBuilder()
        .AddJsonFile("appsettings.json")
        .Build()
        .GetSection("Security")["GlobalSalt"];
    return Convert.ToBase64String(KeyDerivation.Pbkdf2(
        password: password + salt,
        salt: Encoding.UTF8.GetBytes(salt), // Optional, adds more entropy
        prf: KeyDerivationPrf.HMACSHA256,
        iterationCount: 10000,
        numBytesRequested: 32));
}

private bool VerifyPassword(string enteredPassword, string storedHash)
{
    var salt = new ConfigurationBuilder()
        .AddJsonFile("appsettings.json")
        .Build()
```

```

        .GetSection("Security")["GlobalSalt"]);
var enteredHash = Convert.ToBase64String(KeyDerivation.Pbkdf2(
    password: enteredPassword + salt,
    salt: Encoding.UTF8.GetBytes(salt),
    prf: KeyDerivationPrf.HMACSHA256,
    iterationCount: 10000,
    numBytesRequested: 32));
return storedHash == enteredHash;
}

```

4.3.1.7. *Tạo mã xác thực tài khoản đăng kí*

```

private string GenerateRandomCode(int length = 8)
{
    const string chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
    var random = new Random();
    return new string(Enumerable.Repeat(chars, length)
        .Select(s => s[random.Next(s.Length)]).ToArray());
}

public IActionResult Logout()
{
    Response.Cookies.Delete("jwtToken");
    HttpContext.Session.Remove("UserAccount");
    return RedirectToAction("Login");
}

```

4.3.2. **Xây dựng chức năng sản phẩm**

4.3.2.1. *Hiển thị danh sách sản phẩm*

```

public IActionResult Index(int? category)
{
    if (!JwtTokenHelper.TryAuthenticateUser(Request, HttpContext, out var principal))
    {
        return RedirectToAction("Login", "Dashboard");
    }
    var Medicines = db.Medicines.AsQueryable();
    if (category.HasValue)
    {
        Medicines = Medicines.Where(m => m.CategoryId == category.Value);
    }
    var Result = Medicines.Select(m => new MedicineVM
    {
        Id = m.Id,
        Name = m.Name,
        Description = m.Description,

```

```

        Price = m.Price ?? 0,
        Pack = m.Pack,
        Img = m.Img,
        Category = m.Category.Name,
        CategoryID = m.CategoryId
    });
    return View(Result);
}

```

4.3.2.2. *Tìm kiếm sản phẩm*

```

public IActionResult Search(String? query)
{
    if (!JwtTokenHelper.TryAuthenticateUser(Request, HttpContext, out var principal))
    {
        return RedirectToAction("Login", "Dashboard");
    }
    var Medicines = db.Medicines.AsQueryable();
    if (query != null)
    {
        Medicines = Medicines.Where(m => m.Name.Contains(query));
    }
    var Result = Medicines.Select(m => new MedicineVM
    {
        Id = m.Id,
        Name = m.Name,
        Description = m.Description,
        Price = m.Price ?? 0,
        Pack = m.Pack,
        Img = m.Img,
        Category = m.Category.Name
    });
    return View(Result);
}

```

4.3.2.3. *Chi tiết sản phẩm*

```

public IActionResult ShopDetail(int? id)
{
    if (!JwtTokenHelper.TryAuthenticateUser(Request, HttpContext, out var principal))
    {
        return RedirectToAction("Login", "Dashboard");
    }
    var data = db.Medicines
        .Include(m => m.Category)
        .Include(m => m.Ratings)

```



```

        .ThenInclude(r => r.Account) // Include Account while loading Ratings
        .SingleOrDefault(m => m.Id == id);
if (data == null)
{
    TempData["Message"] = $"This medicine {id} is not found";
    return Redirect("/404");
}
var ratingsList = data.Ratings.Select(r => new RatingVM
{
    Account = r.Account.Name,
    Stars = r.Stars,
    Comment = r.Comment,
}).ToList();
var result = new MedicineVM
{
    Id = data.Id,
    Name = data.Name,
    Description = data.Description,
    Uses = data.Uses,
    Price = data.Price ?? 0,
    Unit = data.Unit,
    Ingredient = data.Ingredient,
    Pack = data.Pack,
    Img = data.Img,
    Category = data.Category.Name,
    CategoryID = data.Category.Id,
    Ratings = ratingsList, // Pass the list of ratings
    AverageRating = ratingsList.Any() ? ratingsList.Average(r => r.Stars) : 0, // Calculate average rating
    RatingsCount = ratingsList.Count // Count of ratings
};
return View(result);
}

```

4.3.3. Xây dựng chức năng mua hàng

4.3.3.1. Thêm sản phẩm vào giỏ hàng

[HttpPost]

```
public IActionResult AddToCart(int productId, int quantity, int price)
```

```

{
    var userIdClaim = User.Claims.FirstOrDefault(c => c.Type == "UserId")?.Value;
    if (!int.TryParse(userIdClaim, out int userId))
    {
        TempData["ErrorMessage"] = "User not authenticated.";
        return RedirectToAction("ProductDetail", new { id = productId }); // Redirect back if user is not authenticated
    }
}

```

```

    }
    CreateCartIfNotExists(userId);
    var cart = db.Carts.SingleOrDefault(c => c.AccountId == userId);
    if (quantity <= 0)
        quantity = 1;
    var cartDetail = db.CartDetails.SingleOrDefault(cd => cd.CartId == cart.Id && cd.MedicineId == productId);
    if (cartDetail == null)
    {
        cartDetail = new CartDetail
        {
            CartId = cart.Id, // Use the newly created or existing cart ID
            MedicineId = productId,
            Amount = quantity,
            Total = quantity * price
        };
        db.CartDetails.Add(cartDetail);
    }
    else
    {
        cartDetail.Amount += quantity;
        cartDetail.Total = cartDetail.Amount * price;
        db.CartDetails.Update(cartDetail);
    }

    db.Carts.Update(cart); // Save the cart total change if modified
    db.SaveChanges(); // Save changes to the database
    TempData["SuccessMessage"] = "Product added to cart successfully!";
    return RedirectToAction("ShopDetail", "Shop", new { id = productId });
}

```

4.3.3.2. *Hiển thị giỏ hàng*

```

public IActionResult Index()
{
    if (!JwtTokenHelper.TryAuthenticateUser(Request, HttpContext, out var principal))
    {
        return RedirectToAction("Login", "Dashboard");
    }
    var userIdClaim = User.Claims.FirstOrDefault(c => c.Type == "UserId")?.Value;
    if (!int.TryParse(userIdClaim, out int userId))
    {
        return Unauthorized(); // Handle unauthorized access
    }
    CreateCartIfNotExists(userId);
}

```

```

var cart = db.Carts
    .Include(c => c.Account) // Eager load the associated account
    .Include(c => c.CartDetails)
    .ThenInclude(cd => cd.Medicine)
    .FirstOrDefault(c => c.AccountId == userId);
if (cart == null)
{
    ViewBag.Message = "Your cart is empty.";
    return View();
}
cart.Total = cart.CartDetails.Sum(cd => cd.Total);
db.Carts.Update(cart);
db.SaveChanges();
return View(cart);
}

```

4.3.3.3. *Mua hàng*

```

public IActionResult Checkout(string address)
{
    if (!JwtTokenHelper.TryAuthenticateUser(Request, HttpContext, out var principal))
    {
        return RedirectToAction("Login", "Dashboard");
    }
    var userIdClaim = User.Claims.FirstOrDefault(c => c.Type == "UserId")?.Value;
    if (!int.TryParse(userIdClaim, out int userId))
    {
        return Unauthorized(); // Handle unauthorized access
    }
    var cart = db.Carts
        .Include(c => c.CartDetails)
        .ThenInclude(cd => cd.Medicine)
        .FirstOrDefault(c => c.AccountId == userId);
    if (cart == null || !cart.CartDetails.Any())
    {
        TempData["ErrorMessage"] = "Your cart is empty.";
        return RedirectToAction("Index", "Cart");
    }
    var newOrder = new Order
    {
        Status = false,
        Address = address,
        AccountId = userId,
        Date = DateTime.Now,

```

```

    Total = cart.Total,
    OrderDetails = cart.CartDetails.Select(cd => new OrderDetail
    {
        MedicineId = cd.MedicineId,
        Amount = cd.Amount,
        MedicinePrice = cd.Medicine.Price,
        SumPrice = cd.Total
    }).ToList()
};
// Save the new order to the database
db.Orders.Add(newOrder);
db.SaveChanges();
db.CartDetails.RemoveRange(cart.CartDetails);
db.Carts.Remove(cart);
db.SaveChanges();
TempData["SuccessMessage"] = "Order placed successfully!";
return RedirectToAction("OrderDetail", "Buy", new { id = newOrder.Id });
}

```

4.3.3.4. *Hiển thị danh sách đơn hàng*

```

public async Task<IActionResult> OrderList()
{
    if (!JwtTokenHelper.TryAuthenticateUser(Request, HttpContext, out var principal))
    {
        return RedirectToAction("Login", "Dashboard");
    }
    var userIdClaim = (principal as ClaimsPrincipal)?.Claims.FirstOrDefault(c => c.Type == "UserId")?.Value;
    if (!int.TryParse(userIdClaim, out int userId))
    {
        return Unauthorized(); // Handle unauthorized access
    }
    var userOrders = await db.Orders
        .Where(o => o.AccountId == userId) // Filter by user ID
        .OrderByDescending(o => o.Date) // Optional: Order by most recent
        .ToListAsync();
    return View(userOrders); // Pass the list to the view
}

```

4.3.3.5. *Chi tiết đơn hàng*

```

public async Task<IActionResult> OrderDetail(int? id)
{
    if (!JwtTokenHelper.TryAuthenticateUser(Request, HttpContext, out var principal))
    {
        return RedirectToAction("Login", "Dashboard");
    }
}

```

```

    }
    if (id == null || db.Orders == null)
    {
        return NotFound();
    }
    var userIdClaim = (principal as ClaimsPrincipal)?.Claims.FirstOrDefault(c => c.Type == "UserId")?.Value;
    if (!int.TryParse(userIdClaim, out int userId))
    {
        return Unauthorized(); // Handle unauthorized access
    }
    var order = await db.Orders
        .Include(o => o.OrderDetails)
        .ThenInclude(od => od.Medicine) // Include medicine details
        .FirstOrDefaultAsync(o => o.Id == id);
    if (order == null || order.AccountId != userId)
    {
        TempData["ErrorMessage"] = "You are not authorized to view this order.";
        return RedirectToAction("OrderList"); // Redirect to the list of orders
    }
    return View(order);
}

```

4.3.4. Xây dựng chức năng quản lý tài khoản

4.3.4.1. *Hiển thị danh sách tài khoản*

```

public async Task<IActionResult> Index()
{
    return _context.Accounts != null ?
        View(await _context.Accounts.ToListAsync()) :
        Problem("Entity set 'HisContext.Accounts' is null.");
}

```

4.3.4.2. *Thêm tài khoản mới*

```

public IActionResult Create()
{
    return View();
}

[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Create(Account account)
{
    var existingUser = _context.Accounts
        .SingleOrDefault(u => u.Username == account.Username || u.Email == account.Email);
    if (existingUser != null)

```

```

{
    ModelState.AddModelError(string.Empty, "Username or Email is already exists");
    return View();
}
if (!string.IsNullOrEmpty(account.Username) && !Regex.IsMatch(account.Username, @"^\S+$"))
{
    ModelState.AddModelError(nameof(account.Username), "Username cannot contain spaces. (controller noti)");
}
account.Password = DashboardController.HashPassword("123456"); // Example password
account.Code = "0"; // Example password
account.Active = true; // Example password
ModelState.Remove("Password");
ModelState.Remove("Code");
if (ModelState.IsValid)
{
    _context.Add(account);
    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}
return View(account);
}

```

4.3.4.3. *Sửa tài khoản*

```

public async Task<IActionResult> Edit(int? id)
{
    if (id == null || _context.Accounts == null)
    {
        return NotFound();
    }
    var account = await _context.Accounts.FindAsync(id);
    if (account == null)
    {
        return NotFound();
    }
    return View(account);
}

[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Edit(int id, Account account)
{
    if (id != account.Id)
    {
        return NotFound();
    }
}

```

```

    }
    if (ModelState.IsValid)
    {
        try
        {
            _context.Update(account);
            await _context.SaveChangesAsync();
        }
        catch (DbUpdateConcurrencyException)
        {
            if (!AccountExists(account.Id))
            {
                return NotFound();
            }
        }
        return RedirectToAction(nameof(Index));
    }
    return View(account);
}

```

4.3.4.4. *Chi tiết tài khoản*

```

public async Task<IActionResult> Details(int? id)
{
    if (id == null || _context.Accounts == null)
    {
        return NotFound();
    }
    var account = await _context.Accounts
        .FirstOrDefaultAsync(m => m.Id == id);
    if (account == null)
    {
        return NotFound();
    }
    return View(account);
}

```

4.3.5. Xây dựng chức năng quản lý sản phẩm

4.3.5.1. *Hiển thị danh sách sản phẩm*

```

public async Task<IActionResult> Index()
{
    var hisContext = _context.Medicines.Include(m => m.Category);
    return View(await hisContext.ToListAsync());
}

```

4.3.5.2. *Thêm sản phẩm*

```
public IActionResult Create()
{
    ViewData["CategoryId"] = new SelectList(_context.MedicineCategories, "Id", "Name");
    return View();
}

[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Create(MedicineCreateVM model)
{
    var category = await _context.MedicineCategories.FindAsync(model.CategoryId);
    if (category == null)
    {
        ModelState.AddModelError("CategoryId", "The selected category does not exist.");
        ViewData["CategoryId"] = new SelectList(_context.MedicineCategories, "Id", "Name", model.CategoryId);
        return View(model);
    }
    var medicine = new Medicine
    {
        Name = model.Name,
        CategoryId = model.CategoryId,
        Quantity = model.Quantity,
        Price = model.Price,
        Unit = model.Unit,
        Description = model.Description,
        Uses = model.Uses,
        Img = model.Img,
        Sold = model.Sold,
        Ingredient = model.Ingredient,
        Pack = model.Pack
    };
    if (ModelState.IsValid)
    {
        _context.Add(medicine);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    ViewData["CategoryId"] = new SelectList(_context.MedicineCategories, "Id", "Name", model.CategoryId);
    return View(model);
}
```


4.3.5.3. *Xoá sản phẩm*

```
public async Task<IActionResult> Delete(int? id)
{
    if (id == null || _context.Medicines == null)
    {
        return NotFound();
    }
    var medicine = await _context.Medicines
        .Include(m => m.Category)
        .FirstOrDefaultAsync(m => m.Id == id);
    if (medicine == null)
    {
        return NotFound();
    }
    return View(medicine);
}
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<IActionResult> DeleteConfirmed(int id)
{
    if (_context.Medicines == null)
    {
        return Problem("Entity set 'HisContext.Medicines' is null.");
    }
    var medicine = await _context.Medicines.FindAsync(id);
    if (medicine != null)
    {
        _context.Medicines.Remove(medicine);
    }
    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}
```

4.3.5.4. *Sửa sản phẩm*

```
public async Task<IActionResult> Edit(int? id)
{
    if (id == null || _context.Medicines == null)
    {
        return NotFound();
    }
    var medicine = await _context.Medicines.FindAsync(id);
    if (medicine == null)
    {

```

```

        return NotFound();
    }
    var model = new MedicineCreateVM
    {
        Id = medicine.Id,
        Name = medicine.Name,
        CategoryId = medicine.CategoryId,
        Quantity = medicine.Quantity,
        Price = medicine.Price,
        Unit = medicine.Unit,
        Description = medicine.Description,
        Uses = medicine.Uses,
        Img = medicine.Img,
        Sold = medicine.Sold,
        Ingredient = medicine.Ingredient,
        Pack = medicine.Pack
    };
    ViewData["CategoryId"] = new SelectList(_context.MedicineCategories, "Id", "Name", medicine.CategoryId);
    return View(model);
}

[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> EditMedicine(int id, MedicineCreateVM model)
{
    if (id != model.Id)
    {
        return NotFound();
    }
    var category = await _context.MedicineCategories.FindAsync(model.CategoryId);
    if (category == null)
    {
        ModelState.AddModelError("CategoryId", "The selected category does not exist.");
        ViewData["CategoryId"] = new SelectList(_context.MedicineCategories, "Id", "Name", model.CategoryId);
        return View(model);
    }
    var medicine = new Medicine
    {
        Id = model.Id,
        Name = model.Name,
        CategoryId = model.CategoryId,
        Quantity = model.Quantity,
        Price = model.Price,
        Unit = model.Unit,

```

```

        Description = model.Description,
        Uses = model.Uses,
        Img = model.Img,
        Sold = model.Sold,
        Ingredient = model.Ingredient,
        Pack = model.Pack
    };
    if (ModelState.IsValid)
    {
        try
        {
            _context.Update(medicine);
            await _context.SaveChangesAsync();
        }
        catch (DbUpdateConcurrencyException)
        {
            if (!MedicineExists(medicine.Id))
            {
                return NotFound();
            }
        }
        return RedirectToAction(nameof(Index));
    }
    ViewData["CategoryId"] = new SelectList(_context.MedicineCategories, "Id", "Name", model.CategoryId);
    return View(model);
}

```

4.3.5.5. *Chi tiết sản phẩm*

```

public async Task<IActionResult> Details(int? id)
{
    if (id == null || _context.Medicines == null)
    {
        return NotFound();
    }
    var medicine = await _context.Medicines
        .Include(m => m.Category)
        .FirstOrDefaultAsync(m => m.Id == id);
    if (medicine == null)
    {
        return NotFound();
    }
    return View(medicine);
}

```

4.3.6. Xây dựng chức năng quản lý danh mục

4.3.6.1. *Hiển thị danh sách danh mục*

```
public async Task<IActionResult> Index()
{
    return _context.MedicineCategories != null ?
        View(await _context.MedicineCategories.ToListAsync()) :
        Problem("Entity set 'HisContext.MedicineCategories' is null.");
}
```

4.3.6.2. *Thêm danh mục*

```
public IActionResult Create()
{
    return View();
}
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Create([Bind("Id,Name,Img,Description")] MedicineCategory medicineCategory)
{
    if (ModelState.IsValid)
    {
        _context.Add(medicineCategory);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    return View(medicineCategory);
}
```

4.3.6.3. *Sửa danh mục*

```
public async Task<IActionResult> Edit(int? id)
{
    if (id == null || _context.MedicineCategories == null)
    {
        return NotFound();
    }
    var medicineCategory = await _context.MedicineCategories.FindAsync(id);
    if (medicineCategory == null)
    {
        return NotFound();
    }
    return View(medicineCategory);
}
[HttpPost]
[ValidateAntiForgeryToken]
```

```

public async Task<IActionResult> Edit(int id, [Bind("Id,Name,Img,Description")] MedicineCategory
medicineCategory)
{
    if (id != medicineCategory.Id)
    {
        return NotFound();
    }
    if (ModelState.IsValid)
    {
        try
        {
            _context.Update(medicineCategory);
            await _context.SaveChangesAsync();
        }
        catch (DbUpdateConcurrencyException)
        {
            if (!MedicineCategoryExists(medicineCategory.Id))
            {
                return NotFound();
            }
            else
            {
                throw;
            }
        }
        return RedirectToAction(nameof(Index));
    }
    return View(medicineCategory);
}

```

4.3.6.4. *Xoá danh mục*

```

public async Task<IActionResult> Delete(int? id)
{
    if (id == null || _context.MedicineCategories == null)
    {
        return NotFound();
    }
    var medicineCategory = await _context.MedicineCategories
        .FirstOrDefaultAsync(m => m.Id == id);
    if (medicineCategory == null)
    {
        return NotFound();
    }
}

```

```

    return View(medicineCategory);
}
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<IActionResult> DeleteConfirmed(int id)
{
    if (_context.MedicineCategories == null)
    {
        return Problem("Entity set 'HisContext.MedicineCategories' is null.");
    }
    var medicineCategory = await _context.MedicineCategories.FindAsync(id);
    if (medicineCategory != null)
    {
        _context.MedicineCategories.Remove(medicineCategory);
    }
    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}

```

4.3.6.5. *Chi tiết danh mục*

```

public async Task<IActionResult> Details(int? id)
{
    if (id == null || _context.MedicineCategories == null)
    {
        return NotFound();
    }
    var medicineCategory = await _context.MedicineCategories
        .FirstOrDefaultAsync(m => m.Id == id);
    if (medicineCategory == null)
    {
        return NotFound();
    }
    return View(medicineCategory);
}

```

4.3.7. Xây dựng chức năng quản lý đơn hàng

4.3.7.1. *Danh sách đơn hàng (toàn bộ, đã duyệt và chưa duyệt)*

```

public async Task<IActionResult> Index()
{
    var hisContext = _context.Orders.Include(o => o.Account);
    ViewBag.ListType = "All";
    return View(await hisContext.ToListAsync());
}

```

```

public async Task<IActionResult> ShippingOrders()
{
    var hisContext = _context.Orders
        .Include(o => o.Account)
        .Where(o => o.Status); // Filters orders with Status = true
    ViewBag.ListType = "Shipping";
    return View("Index", await hisContext.ToListAsync());
}

public async Task<IActionResult> PendingOrders()
{
    var hisContext = _context.Orders
        .Include(o => o.Account)
        .Where(o => o.Status == false); // Filters orders with Status = true
    ViewBag.ListType = "Pending";
    return View("Index", await hisContext.ToListAsync());
}

public async Task<IActionResult> NewPendingOrders()
{
    var oneWeekAgo = DateTime.Now.AddDays(-7); // Calculate the date one week ago
    var hisContext = _context.Orders
        .Include(o => o.Account)
        .Where(o => o.Status == false && o.Date >= oneWeekAgo); // Add date filter
    return View("Index", await hisContext.ToListAsync());
}

```

4.3.7.2. Chi tiết đơn hàng

```

public async Task<IActionResult> Details(int? id)
{
    if (!JwtTokenHelper.TryAuthenticateUser(Request, HttpContext, out var principal))
    {
        return RedirectToAction("Login", "Dashboard");
    }
    if (id == null || _context.Orders == null)
    {
        return NotFound();
    }
    var userIdClaim = (principal as ClaimsPrincipal)?.Claims.FirstOrDefault(c => c.Type == "UserId")?.Value;
    if (!int.TryParse(userIdClaim, out int userId))
    {
        return Unauthorized(); // Handle unauthorized access
    }
    var order = await _context.Orders
        .Include(o => o.OrderDetails)

```

```

        .ThenInclude(od => od.Medicine) // Include medicine details
        .FirstOrDefaultAsync(o => o.Id == id);
if (order == null || order.AccountId != userId)
{
    TempData["ErrorMessage"] = "You are not authorized to view this order.";
    return RedirectToAction("OrderList"); // Redirect to the list of orders
}
return View(order);
}

```

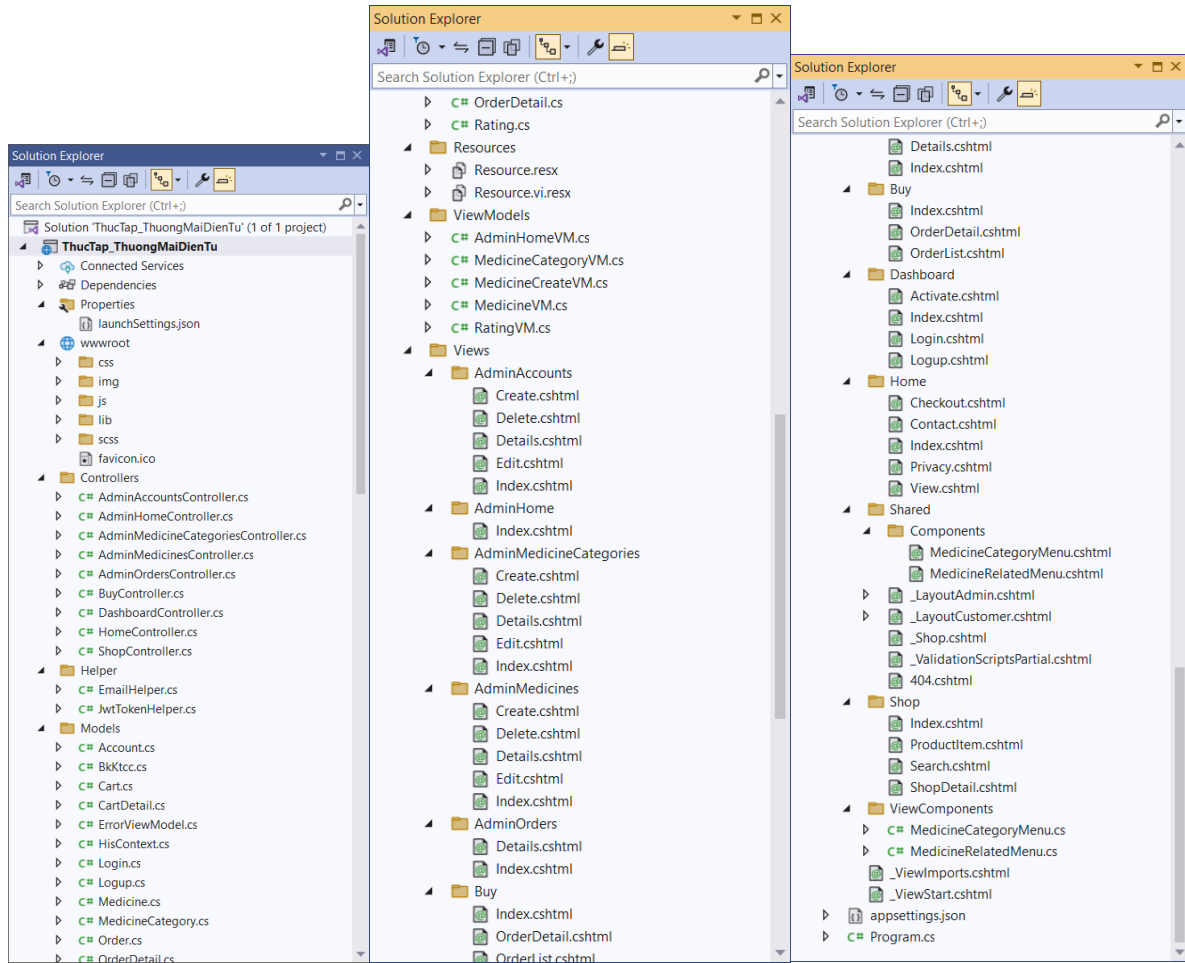
4.3.7.3. *Xác nhận đơn hàng*

```

[HttpPost]
public async Task<IActionResult> Approve(int orderId)
{
    // Retrieve the order from the database using the provided orderId
    var order = await _context.Orders.FindAsync(orderId);
    if (order == null)
    {
        return NotFound();
    }
    order.Status = true;
    _context.Update(order);
    await _context.SaveChangesAsync();
    return Json(new { reload = true });
}

```


4.4. CẤU TRÚC DỰ ÁN



CHƯƠNG 5: KẾT QUẢ THỰC TẬP

Trong thời gian thực tập tại bệnh viện, em đã có cơ hội học hỏi và phát triển nhiều kỹ năng quan trọng trong vai trò của một kỹ sư phần mềm. Công việc tại đây không chỉ giúp tôi củng cố kiến thức chuyên môn mà còn mở rộng hiểu biết về lĩnh vực y tế và công nghệ thông tin y tế. Dưới đây là những kết quả tôi đã đạt được sau kỳ thực tập:

5.1. PHÁT TRIỂN KỸ NĂNG CHUYÊN MÔN

Kỹ năng lập trình: Em đã có cơ hội sử dụng các ngôn ngữ lập trình và công nghệ như C#, ASP.NET, và SQL Server để phát triển, xây dựng dự án phần mềm thương mại thuốc men cho Viện. Mặc dù phần mềm chưa thật sự đáp ứng đầy đủ yêu cầu thực tế, nhưng nó có thể là tiền đề để em có thể phát triển phần mềm đầy đủ chức năng và đáp ứng các yêu cầu thực tiễn, từ đó mang lại giá trị cho Viện.

Kỹ năng phân tích và thiết kế hệ thống: Tham gia vào quy trình phân tích yêu cầu người dùng và thiết kế hệ thống giúp em hiểu rõ hơn về các yêu cầu đặc thù của phần mềm y tế, cũng như cách tối ưu hóa hệ thống cho hiệu suất và độ bảo mật cao.

Kỹ năng giải quyết vấn đề: Trong Viện luôn tồn tại nhiều vấn đề về phần mềm cũng như phần cứng. Việc xử lý các lỗi, khắc phục vấn đề trong thời gian ngắn và đảm bảo hệ thống hoạt động trơn tru là những trải nghiệm quý báu giúp tôi rèn luyện kỹ năng giải quyết vấn đề.

5.2. HIỂU BIẾT VỀ NGÀNH Y TẾ

Thông qua các dự án phần mềm, tôi đã hiểu sâu hơn về quy trình làm việc trong bệnh viện và nhu cầu quản lý dữ liệu bệnh nhân an toàn và hiệu quả. Em nhận ra vai trò quan trọng của công nghệ thông tin trong việc hỗ trợ nhân viên y tế cung cấp dịch vụ chăm sóc tốt nhất cho bệnh nhân.

Việc tiếp xúc và làm việc cùng các bác sĩ, y tá đã giúp tôi hiểu rõ hơn về cách công nghệ có thể giúp tối ưu hóa công việc của họ, đồng thời nâng cao độ chính xác và hiệu quả trong quản lý y tế.

5.3. KINH NGHIỆM LÀM VIỆC NHÓM

Thực tập tại bệnh viện cho em cơ hội làm việc cùng một đội ngũ đa ngành gồm bác sĩ, kỹ thuật viên và các chuyên gia CNTT. Điều này giúp tôi phát triển kỹ năng giao tiếp,

học cách lắng nghe và hiểu nhu cầu của người dùng cuối, từ đó có thể phát triển những giải pháp phù hợp.

Qua quá trình làm việc nhóm, em học được cách phối hợp công việc, chia sẻ ý tưởng, và hỗ trợ lẫn nhau để đạt được mục tiêu chung của dự án.

5.4. TƯ DUY PHÁT TRIỂN BẢN THÂN

Thực tập tại bệnh viện không chỉ là công việc chuyên môn mà còn là cơ hội để em hoàn thiện bản thân. Em học được cách chịu trách nhiệm cho công việc của mình, biết tự đặt ra các tiêu chuẩn cao hơn cho bản thân và liên tục cải thiện trong từng nhiệm vụ nhỏ nhất.

Kết quả thực tập giúp em định hướng rõ hơn về con đường sự nghiệp, đặc biệt là việc tiếp tục phát triển chuyên môn trong lĩnh vực công nghệ y tế. Em nhận thấy giá trị to lớn mà công việc của mình có thể đem lại cho cộng đồng và có thêm động lực để theo đuổi lĩnh vực này lâu dài.

Tp. Hồ Chí Minh, ngày ... tháng năm 20...

BẢNG GHI NHẬN KẾT QUẢ THỰC TẬP HÀNG TUẦN

Một số thông tin liên hệ

Họ và tên: Võ Hải Hà
Ngày sinh: 14/08/2002
Mã số sinh viên: 3120410002
Lớp: DCT1201
Ngành học: Công nghệ thông tin
Email: voha308@gmail.com
Điện thoại: 0333.464.563

Chuyên gia cơ quan: Chế Hồng Trung
Email: chetrung@gmail.com
Điện thoại: 0976.767.939

Giảng viên hướng dẫn: Nguyễn Trung Tín
Email: tinnt@sgu.edu.vn
Điện thoại: 0909.236.008

Tuần	Nội dung thực tập (do chuyên gia của cơ quan giao)	Kết quả thực tập (do chuyên gia của cơ quan đánh giá)
1 Từ ngày 1/10/2024 đến ngày 6/10/2024	<ul style="list-style-type: none">Xử lý sự cố mạng tại phòng Dinh dưỡng và Vật lý trị liệuThiết kế chức năng xuất file pdf.	
2 Từ ngày 7/10/2024 đến ngày 13/10/2024	<ul style="list-style-type: none">Thực hiện dự án cá nhânTrực phòng Hội nghị Khoa Học tại Viện	
3 Từ ngày 14/10/2024	<ul style="list-style-type: none">Thực hiện dự án cá nhânTìm hiểu về JWT và chức năng đăng nhập	

đến ngày 20/10/2024	- Trục phòng Hội nghị Khoa Học tại Viện	
4 Từ ngày 21/10/2024 đến ngày 27/10/2024	- Thực hiện dự án cá nhân - Tìm hiểu về đa ngôn ngữ trong ASP.NET MVC	
5 Từ ngày 28/10/2024 đến ngày 3/11/2024	- Thực hiện dự án cá nhân - Bảo trì sự cố máy tính tại phòng Chuẩn đoán hình ảnh	
6 Từ ngày 4/11/2024 đến ngày 10/11/2024	- Thực hiện dự án cá nhân - Bảo trì sự cố tại phòng Bán Thuốc	
7 Từ ngày 11/11/2024 đến ngày 17/11/2024	- Thực hiện dự án cá nhân - Bảo trì sự cố và nối thêm dây mạng tại phòng Bảo Hiểm Y Tế	
8 Từ ngày 18/11/2024 đến ngày 24/11/2024	- Thực hiện dự án cá nhân - Viết báo cáo - Tham gia kiểm kê tài sản thuộc phòng CNTT tại Viện	

Chuyên gia cơ quan hướng dẫn thực tập

(Ký tên và ghi họ tên)

Ghi chú:

-**Chuyên gia doanh nghiệp** ghi nhận kết quả thực tập của **sinh viên** theo tuần và gửi qua email cho **giảng viên hướng dẫn** khi kết thúc các tuần 3,6 của đợt thực tập. BẢNG GHI NHẬN KẾT QUẢ THỰC TẬP TỐT NGHIỆP HÀNG TUẦN này là một trong những hồ sơ kèm theo quyền báo cáo thực tập tốt nghiệp.

-Cột **Kết quả thực tập**, chuyên gia doanh nghiệp có thể ghi *Hoàn thành tốt, Hoàn thành, Không đạt* hoặc có thể ghi nhận chi tiết hơn.

KHOA CNTT

BẢNG ĐÁNH GIÁ QUÁ TRÌNH THỰC TẬP TỐT NGHIỆP
(do chuyên gia doanh nghiệp đánh giá).

Họ và tên sinh viên: Võ Hải Hà
Ngày sinh: 14/08/2002
Mã số sinh viên: 3120410002
Lớp : DCT1201
Thời gian thực tập: 30/9/2024 – 23/11/2024
Cơ quan thực tập: Viện Y Dược Học Dân Tộc
Địa chỉ cơ quan: 273-275 Nguyễn Văn Trỗi, P.10, Q.Phú Nhuận, TP. Hồ Chí Minh
Chuyên gia cơ quan hướng dẫn: Chế Hồng Trung

I. ĐÁNH GIÁ VỀ QUÁ TRÌNH THỰC TẬP

STT	Nội dung đánh giá	ĐIỂM		
		0	0.5	1
1	Khả năng thực hành			
2	Khả năng làm việc nhóm			
3	Tính thân thiện			
4	Tính năng động			
5	Tính thân sáng tạo			
6	Chấp hành nội quy cơ quan			
7	Giờ giấc làm việc			
8	Phương pháp làm việc			
9	Khối lượng công việc			
10	Báo cáo thực tập tốt nghiệp			

(theo thang điểm 10).

II. CÁC ĐÁNH GIÁ KHÁC:

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

III. KẾT QUẢ TỔNG HỢP:

Điểm tổng cộng :

XÁC NHẬN CỦA CƠ QUAN
(đóng mộc tròn của cơ quan, họ tên, ký tên)

Chuyên gia hướng dẫn
(Ký và ghi họ tên)

KHOA CNTT

PHIẾU ĐÁNH GIÁ KẾT QUẢ THỰC TẬP TỐT NGHIỆP
(do giảng viên hướng dẫn đánh giá)

Họ và tên sinh viên: Võ Hải Hà

Ngày sinh: 14/08/2002

Mã số sinh viên: 3120410002

Lớp : DCT1201

Thời gian thực tập: 30/9/2024 – 23/11/2024

Cơ quan thực tập: Viện Y Dược Học Dân Tộc

Địa chỉ cơ quan: 273-275 Nguyễn Văn Trỗi P.10, Q.Phú Nhuận, TP. Hồ Chí Minh

Chuyên gia cơ quan hướng dẫn: Chế Hồng Trung

I. ĐIỂM CỦA CHUYÊN GIA DOANH NGHIỆP :

(thang điểm 10)

II. ĐIỂM CỦA GIẢNG VIÊN HƯỚNG DẪN:

(thang điểm 10)

III. ĐIỂM TỔNG KẾT:

(trung bình cộng 2 cột điểm trên, thang điểm 10)

Xếp loại :

TP Hồ Chí Minh, ngày tháng năm

Giảng viên hướng dẫn

GHI CHÚ :

Xếp loại được tính như sau:

Xếp loại	Xuất sắc	nếu điểm tổng hợp từ 9 đến 10
Xếp loại	Giỏi	nếu điểm tổng hợp từ 8 đến cận 9
Xếp loại	Khá	nếu điểm tổng hợp từ 7 đến cận 8
Xếp loại	TB Khá	nếu điểm tổng hợp từ 6 đến cận 7
Xếp loại	TB	nếu điểm tổng hợp từ 5 đến cận 6
Xếp loại	Không đạt	nếu điểm tổng hợp từ 0 đến cận 5

CHƯƠNG 6. KẾT LUẬN VÀ KIẾN NGHỊ

6.1. KẾT LUẬN

Trong thời gian thực tập tại bệnh viện, em đã phát triển nhiều kỹ năng quan trọng như lập trình (C#, ASP.NET, SQL Server), phân tích và thiết kế hệ thống, và giải quyết vấn đề thực tiễn liên quan đến công nghệ y tế. Bên cạnh đó, em hiểu rõ hơn về vai trò của công nghệ thông tin trong quản lý dữ liệu và tối ưu hóa công việc của nhân viên y tế, đồng thời trau dồi kinh nghiệm làm việc nhóm với đội ngũ đa ngành. Quá trình thực tập không chỉ củng cố chuyên môn mà còn giúp em định hướng rõ ràng về sự nghiệp, nhận thức giá trị của công nghệ y tế và quyết tâm phát triển trong lĩnh vực này để mang lại lợi ích cho cộng đồng.

6.2. KIẾN NGHỊ

Để nâng cao chất lượng và hiệu quả của website bán thuốc được xây dựng bằng ASP.NET MVC và SQL, em đề xuất một số kiến nghị như sau: Thứ nhất, cần bổ sung các tính năng nâng cao như hệ thống gợi ý sản phẩm dựa trên lịch sử mua hàng hoặc tích hợp chatbot hỗ trợ khách hàng, nhằm cải thiện trải nghiệm người dùng. Thứ hai, tăng cường các biện pháp bảo mật như mã hóa dữ liệu, xác thực hai lớp, và kiểm tra lỗ hổng để đảm bảo an toàn thông tin, đặc biệt với dữ liệu nhạy cảm như thông tin khách hàng và đơn hàng. Thứ ba, tối ưu hóa hiệu suất hệ thống bằng cách sử dụng cơ sở dữ liệu thứ hai là PostgreSQL. Cuối cùng, có thể mở rộng chức năng quản trị, cho phép các nhà quản lý dễ dàng theo dõi doanh số, tồn kho, và báo cáo chi tiết để hỗ trợ quá trình ra quyết định. Các kiến nghị này không chỉ nâng cao chất lượng dự án mà còn giúp website đáp ứng tốt hơn nhu cầu thực tế của người dùng và doanh nghiệp.

TÀI LIỆU THAM KHẢO

- [1]. <https://www.vienydhdt.gov.vn/gioi-thieu/lich-su-hinh-thanh-va-phat-trien>
- [2]. https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-9.0&WT.mc_id=dotnet-35129-website&tabs=visual-studio
- [3]. <https://www.oshyn.com/blog/asp-net-mvc-guide>
- [4]. <https://codegym.vn/blog/c-la-gi-tim-hieu-ve-ngon-ngu-lap-trinh-c/>
- [5]. <https://itnavi.com.vn/blog/aspnet-core-la-gi>
- [6]. [https://200lab.io/blog/sql-server-la-gi/#:~:text=SQL%20Server%20\(hay%20Microsoft%20SQL,t%E1%BB%AB%20nhi%E1%BB%81u%20ngu%E1%BB%93n%20kh%C3%A1c%20nhau](https://200lab.io/blog/sql-server-la-gi/#:~:text=SQL%20Server%20(hay%20Microsoft%20SQL,t%E1%BB%AB%20nhi%E1%BB%81u%20ngu%E1%BB%93n%20kh%C3%A1c%20nhau)
- [7]. <https://itviec.com/blog/mvc-la-gi/#:~:text=MVC%20l%C3%A0%20vi%E1%BA%BFt%20t%E1%BA%AFt%20c%E1%BB%A7a,th%E1%BB%83%20c%E1%BB%A7a%20m%E1%BB%99t%20%E1%BB%A9ng%20d%E1%BB%A5ng>
- [8]. <https://wiki.matbao.net/bootstrap-la-gi-cai-dat-bootstrap-web-chuan-responsive/>