

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

**Xây dựng website bán hàng thời trang sử dụng công
nghệ ReactJS và NodeJS**

Người hướng dẫn: **TS. LÊ VĂN MINH**

Sinh viên thực hiện: **NGUYỄN TIẾN NAM**

Mã số sinh viên: **2151220134**

Lớp: **21CT3**

Đà Nẵng,/2025

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

**Xây dựng website bán hàng thời trang sử dụng công
nghệ ReactJS và NodeJS**

Người hướng dẫn: **TS. LÊ VĂN MINH**
Sinh viên thực hiện: **NGUYỄN TIẾN NAM**
Mã số sinh viên: **2151220134**
Lớp: **21CT3**

Đà Nẵng,/2025

{Trang trắng này dùng để dán bản Nhận xét của người hướng dẫn, hoặc thay trang này bằng Nhận xét của người hướng dẫn}

{Trang trắng này dùng để dán bản Nhận xét của người phản biện, hoặc thay trang này bằng Nhận xét của người phản biện}

TÓM TẮT

Tên đề tài: Xây dựng website bán hàng thời trang sử dụng công nghệ ReactJS và Node.JS.

Sinh viên thực hiện: Nguyễn Tiến Nam

Mã số SV: 2151220134 Lớp: 21CT3

Đề tài “Xây dựng website bán hàng thời trang sử dụng công nghệ ReactJS và Node js” nhằm tạo ra một nền tảng thương mại điện tử hiện đại và tiện lợi cho việc mua sắm trực tuyến. Trong thời đại mà mua sắm online đang trở nên phổ biến, một website bán hàng thời trang sẽ đáp ứng được nhu cầu của người dùng, mang đến cho họ trải nghiệm thuận tiện khi có thể tìm kiếm, chọn lựa và mua sắm từ xa mà không cần đến trực tiếp cửa hàng. ReactJS – một thư viện JavaScript mạnh mẽ chuyên về xây dựng giao diện người dùng – được sử dụng để phát triển phần frontend nhờ vào khả năng tạo ra các giao diện tương tác, mượt mà và dễ bảo trì. Node.js – một môi trường chạy JavaScript phía máy chủ – được chọn để xây dựng phần backend vì hiệu suất cao, khả năng mở rộng tốt và hỗ trợ mạnh mẽ trong việc xử lý bất đồng bộ, rất phù hợp cho các ứng dụng web hiện đại. Sự kết hợp giữa ReactJS và Node.js sẽ giúp tạo ra một hệ thống toàn diện, dễ mở rộng và hiệu quả trong vận hành.

Mục tiêu chính của dự án là phát triển một website có giao diện người dùng thân thiện, dễ sử dụng và cung cấp đầy đủ các chức năng cơ bản như tìm kiếm, giỏ hàng và thanh toán trực tuyến an toàn. Website sẽ bao gồm cả phần quản trị viên, hỗ trợ quản lý đơn hàng và thông tin người dùng một cách dễ dàng và hiệu quả. Bên cạnh đó, kiến trúc hệ thống sẽ được thiết kế mở để có thể bổ sung các tính năng mở rộng như gợi ý, tích hợp đa phương thức thanh toán hoặc đánh giá sản phẩm trong tương lai.

Phạm vi của đồ án sẽ tập trung vào việc xây dựng các tính năng cốt lõi cho hệ thống, bao gồm giao diện người dùng và giao diện quản trị, quy trình thanh toán đơn giản, và cơ sở dữ liệu để quản lý sản phẩm, đơn hàng và thông tin khách hàng. Các công cụ như XAMPP và Visual Studio Code sẽ được sử dụng để hỗ trợ phát triển trên môi trường cục bộ. HTML, CSS và JavaScript sẽ được ứng dụng để tạo giao diện thân thiện, trực quan, cùng với ReactJs để nâng cao trải nghiệm người dùng và NodeJs kết hợp với cơ sở dữ liệu MYSQL để xây dựng phần backend mạnh mẽ.

LỜI CẢM ƠN

Trong quá trình thực hiện đồ án tốt nghiệp với đề tài “Xây dựng website bán hàng thời trang sử dụng công nghệ ReactJs và Node Js” em đã nhận được rất nhiều sự hỗ trợ từ nhiều phía, và em xin gửi lời cảm ơn chân thành đến những người đã đồng hành cùng em trong hành trình này.

Trước tiên, em xin chân thành cảm ơn TS. Lê Văn Minh, người đã tận tình hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện đồ án. Sự chỉ dẫn, kiến thức chuyên môn sâu sắc và những ý kiến phản hồi quý báu của cô đã giúp em định hướng đúng đắn trong quá trình nghiên cứu và phát triển hệ thống. Em rất biết ơn vì cô đã luôn sẵn sàng giải đáp mọi thắc mắc và truyền cảm hứng cho em để em có thể hoàn thành tốt nhất nhiệm vụ của mình.

Em cũng xin cảm ơn Khoa Công nghệ Thông tin, Trường Đại học Kiến trúc Đà Nẵng, nơi đã cung cấp môi trường học tập và nghiên cứu tốt nhất để em phát triển. Các thầy cô trong khoa đã tạo điều kiện thuận lợi cho em trong quá trình học tập và nghiên cứu.

Với điều kiện thời gian cũng như kinh nghiệm hạn chế của sinh viên nên bài báo cáo này không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của quý thầy cô cũng như lời nhận xét của hội đồng bảo vệ dự án để em có điều kiện bổ sung, nâng cao kiến thức của bản thân, phục vụ tốt hơn cho công tác sau này.

Chúng em xin chân thành cảm ơn !

CAM ĐOAN

Tôi, Nguyễn Tiến Nam, sinh viên lớp 21CT3, mã số sinh viên 2151220134, sinh viên khóa 21, Khoa Công nghệ Thông tin, Trường Đại học Kiến trúc Đà Nẵng, xin cam đoan rằng đề tài "Xây dựng website bán hàng thời trang bằng công nghệ ReactJS và NodeJS" là kết quả nghiên cứu và làm việc của riêng tôi, được thực hiện dưới sự hướng dẫn của Thầy Lê Văn Minh tại Trường. Tất cả các dữ liệu, kết quả và trích dẫn trong đồ án đều trung thực và có nguồn gốc rõ ràng. Tôi hoàn toàn chịu trách nhiệm về tính trung thực và chính xác của nội dung trong đồ án này.

Sinh viên thực hiện

MỤC LỤC

TÓM TẮT	I
LỜI CẢM ƠN	II
CAM ĐOAN.....	III
MỤC LỤC	IV
DANH MỤC CÁC BẢNG.....	VI
DANH MỤC CÁC HÌNH	VIII
DANH MỤC KÝ HIỆU, CHỮ VIẾT TẮT	XI
MỞ ĐẦU.....	XII
CHƯƠNG 1. ĐỐI TƯỢNG VÀ PHẠM VI.....	1
1.1. Tổng quan đề tài	1
1.2. Mục tiêu của đề tài.	2
1.3. Đối tượng nghiên cứu	3
1.4. Phạm vi nghiên cứu.	4
1.4.1. Về chức năng.....	4
1.4.2. Về công nghệ.....	4
1.4.3. Về sản phẩm.	4
1.4.4. Về giao diện.....	5
1.4.5. Về môi trường triển khai	5
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ PHÁT TRIỂN.....	6
2.1. Cơ sở lý thuyết.	6
2.1.1. Giới thiệu về ngôn ngữ JavaScript.	6
2.1.2. Giới thiệu về ReactJS.	8
2.1.3. Giới thiệu về ExpressJS.....	11
2.1.4. Giới thiệu về NodeJS.....	13
2.1.5. Mô hình MVC	15

2.1.6. Cơ sở dữ liệu MySQL.	17
2.1.7. Giới thiệu VNPay	18
2.1.8. Giới thiệu GeminiAI.....	19
2.2. Công cụ sử dụng.	20
2.2.1. Visual Studio Code.....	20
2.2.2. Xampp.	21
2.3. Các bước cài đặt công cụ.	22
2.3.1. Cài đặt Visual Studio code.	22
2.3.2. Cài đặt Xampp.....	23
CHƯƠNG 3. TRIỂN KHAI HỆ THỐNG.....	25
3.1. Giới thiệu chung.	25
3.1.1. Về phía khách hàng (User)	25
3.1.2. Về phía Quản trị viên (Administrator)	26
3.1.3. Các yêu cầu phi chức năng về hệ thống.	27
3.2. Mô hình UseCase.	28
3.2.1. Mô hình UseCase tổng quan.	28
3.2.2. Mô hình UseCase phân rã.....	28
3.2.3. Biểu đồ UseCase.....	31
3.2.4. Biểu đồ tuần tự	51
3.3. Thiết kế Cơ sở dữ liệu.	58
3.3.1. Mô hình ERD.	58
3.3.2. Cấu trúc bảng.....	60
CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC	73
4.1. Giao diện người dùng.....	73
4.2. Giao diện Admin.....	80
KẾT LUẬN	84
TÀI LIỆU THAM KHẢO.....	86

DANH MỤC CÁC BẢNG

Bảng 3.1. Yêu cầu phi chức năng.....	27
Bảng 3.2. Danh sách usecase.....	30
Bảng 3.3. Đặc tả usecase đăng nhập.....	32
Bảng 3.4. Đặc tả usecase đăng kí	33
Bảng 3.5. Đặc tả usecase đổi mật khẩu	35
Bảng 3.6. Đặc tả usecase xem sản phẩm	36
Bảng 3.7. Đặc tả usecase thêm sản phẩm vào giỏ hàng	38
Bảng 3.8. Bảng đặc tả usecase thanh toán.....	39
Bảng 3.9. Đặc tả usecase quản lý đơn hàng của user	41
Bảng 3.10. Đặc tả usecase cập nhật giỏ hàng.....	42
Bảng 3.11. Đặc tả usecase thêm sản phẩm	43
Bảng 3.12. Đặc tả usecase sửa sản phẩm	44
Bảng 3.13. Đặc tả usecase xóa sản phẩm	45
Bảng 3.14. Đặc tả usecase quản lý đơn hàng	47
Bảng 3.15. Đặc tả usecase quản lí người dùng.....	48
Bảng 3.16. Đặc tả usecase thống kê	50
Bảng 3.17. Cấu trúc dữ liệu bảng AddressUser	60
Bảng 3.18. Cấu trúc dữ liệu bảng Allcode	60
Bảng 3.19. Cấu trúc dữ liệu bảng Banner	61
Bảng 3.20. Cấu trúc dữ liệu bảng blog	61
Bảng 3.21. Cấu trúc dữ liệu bảng commnet	62
Bảng 3.22. Cấu trúc dữ liệu bảng Masssge	63
Bảng 3.23. Cấu trúc dữ liệu bảng OrderDetail	63
Bảng 3.24. Cấu trúc dữ liệu bảng OrderProduct	64
Bảng 3.25. Cấu trúc dữ liệu bảng ProductDetail.....	65
Bảng 3.26. Cấu trúc dữ liệu bảng ProductDetailSize	65

Bảng 3.27. Cấu trúc dữ liệu bảng ProductImage.....	66
Bảng 3.28. Cấu trúc dữ liệu bảng Product.....	66
Bảng 3.29. Cấu trúc dữ liệu bảng ShopCart.....	67
Bảng 3.30. Cấu trúc dữ liệu bảng User	68
Bảng 3.31. Cấu trúc dữ liệu bảng voucher	69
Bảng 3.32. Cấu trúc dữ liệu bảng RoomMessage	69
Bảng 3.33. Cấu trúc dữ liệu bảng TypeShip	70
Bảng 3.34. Cấu trúc dữ liệu bảng TypeVoucher	70
Bảng 3.35. Cấu trúc dữ liệu bảng VoucherUsed	71
Bảng 3.36. Cấu trúc dữ liệu bảng Supplier	71
Bảng 3.37. Cấu trúc dữ liệu bảng Receipt.....	72
Bảng 3.38. Cấu trúc dữ liệu bảng ReceiptDetail	72

DANH MỤC CÁC HÌNH

Hình 2.1. Logo Reactjs	8
Hình 2.2. Mô hình MVC	15
Hình 2.3. Bảng điều khiển của xampp	24
Hình 2.4. Localhost của xampp	24
Hình 3.1. Sơ đồ Usecase tổng quát.....	28
Hình 3.2. Sơ đồ User case khách hàng	29
Hình 3.3. Sơ đồ Use care Admin.....	30
Hình 3.4. Usecase đăng nhập	31
Hình 3.5. Usecase đăng kí	33
Hình 3.6. Ussecase đổi mật khẩu.....	35
Hình 3.7. Usecase xem sản phẩm.....	36
Hình 3.8. Usecase thêm sản phẩm vào giỏ hàng	38
Hình 3.9. Usecase thanh toán	39
Hình 3.10. Usecase quản lý đơn hàng của user	41
Hình 3.11. Usecase cập nhật giỏ hàng.....	42
Hình 3.12. Usecase thêm sản phẩm.....	43
Hình 3.13. Usecase sửa sản phẩm	44
Hình 3.14. Usecase xóa sản phẩm.....	45
Hình 3.15. Usecase quản lý đơn hàng của admin.....	47
Hình 3.16. Usecase quản lý người dùng.....	48
Hình 3.17. Usecase thống kê	50
Hình 3.18. Sơ đồ tuần tự đăng nhập	51
Hình 3.19. Sơ đồ tuần tự đăng kí.....	52
Hình 3.20. Sơ đồ tuần tự đổi mật khẩu.....	53
Hình 3.21. Sơ đồ tuần tự xem sản phẩm	53
Hình 3.22. Sơ đồ tuần tự thêm giỏ hàng.....	54

Hình 3.23. Sơ đồ tuần tự thanh toán.....	55
Hình 3.24. Sơ đồ tuần tự thêm sản phẩm	56
Hình 3.25. Sơ đồ tuần tự cập nhật sản phẩm.....	56
Hình 3.26. Sơ đồ tuần tự xoá sản phẩm.....	57
Hình 3.27. Sơ đồ cơ sở dữ liệu	58
Hình 3.28. Biểu đồ quan hệ	59
Hình 4.1. Giao diện trang đăng nhập.....	73
Hình 4.2. Giao diện trang đăng ký	73
Hình 4.3. Giao diện trang chủ	74
Hình 4.4. Giao diện trang cửa hàng.....	74
Hình 4.5. Giao diện trang tin tức	75
Hình 4.6. Giao diện trang mã giảm giá	75
Hình 4.7. Giao diện trang giới thiệu.....	76
Hình 4.8. Giao diện chi tiết sản phẩm	76
Hình 4.9. Giao diện trang giỏ hàng	77
Hình 4.10. Giao diện chi tiết trang thanh toán	77
Hình 4.11. Giao diện trang hồ sơ.....	78
Hình 4.12. Giao diện trang đổi mật khẩu	78
Hình 4.13. Giao diện trang đơn mua hàng	79
Hình 4.14. Giao diện trang kho voucher	79
Hình 4.15. Giao diện botchatAI	80
Hình 4.16. Giao diện trang chủ	80
Hình 4.17. Giao diện trang quản lý người dùng.....	81
Hình 4.18. Giao diện trang quản lí danh mục	81
Hình 4.19. Giao diện trang quản lý nhãn hàng.....	81
Hình 4.20. Giao diện trang quản lý sản phẩm	82
Hình 4.21. Giao diện trang quản lý bài đăng.....	82

Hình 4.22. Giao diện trang quản lý đơn hàng	82
Hình 4.23. Giao diện trang quản lý tin nhắn	83
Hình 4.24. Giao diện trang thống kê	83

DANH MỤC KÝ HIỆU, CHỮ VIẾT TẮT

STT	Kí hiệu viết tắt	Chữ viết tắt đầy đủ
1	HTML	Hyper Text Markup Language
2	CSS	Cascading Style Sheets
3	PHP	Personal Home Page
4	SQL	Structured Query Language
5	ORM	Object-Relational Mapping
6	JSON	JavaScript Object Notation
7	MVC	Model-View-Controller
8	AI	Artificial Intelligence
9	API	Application Programming Interface
10	CSRF	Cross-Site Request Forgery
11	XSS	Cross-Site Scripting
12	CMS	Content Management System

MỞ ĐẦU

Ngày nay, với sự phát triển mạnh mẽ của khoa học công nghệ, Công nghệ Thông tin là một trong những ngành có vị thế dẫn đầu và có vai trò rất lớn trong sự phát triển chung đó. Các ứng dụng của công nghệ thông tin được áp dụng trong mọi lĩnh vực nghiên cứu khoa học cũng như trong mọi lĩnh vực của đời sống. Là một phần của Công nghệ Thông tin, Công nghệ web đang có được sự phát triển mạnh mẽ và phổ biến rất nhanh bởi những lợi ích mà nó mang lại cho cộng đồng là rất lớn.

Bằng việc lựa chọn và thực hiện đề tài “Xây dựng website bán hàng thương mại điện tử”, em muốn tìm hiểu và đưa ra một giải pháp tốt nhằm giải quyết công việc bán hàng trực tuyến cho công ty. Website mang lại cho công ty rất nhiều lợi ích như: khả năng quảng cáo, phổ biến tên tuổi công ty rất nhanh trên môi trường mạng toàn cầu, việc kinh doanh sẽ mở 24/24, giảm thiểu chi phí nhân viên cho tiếp thị, quảng cáo, dễ dàng nhận phản hồi từ phía khách hàng, cơ hội mở rộng liên kết và hợp tác ở phạm vi quốc tế...v.v.

Website bán hàng giúp cho công ty cung cấp cho khách hàng những lựa chọn linh hoạt và tiện lợi trong việc tìm mua sản phẩm thông qua chức năng tìm kiếm và giỏ hàng.

Các thông tin về sản phẩm được hiển thị chi tiết với từng sản phẩm, từ đó khách hàng dễ dàng nhận biết và lựa chọn được thứ mình cần. Về phía Công ty, hệ thống mang lại sự tiện dụng trong việc cập nhập và quản lý các thông tin cho website, tạo khả năng liên lạc nhanh và dễ dàng với khách hàng.

CHƯƠNG 1. ĐỐI TƯỢNG VÀ PHẠM VI

1.1. Tổng quan đề tài

Hiện tại, Online là một thị trường lớn và đầy tiềm năng đối với kinh doanh bán lẻ. Trong khi đó, người tiêu dùng đang có xu hướng “online hóa” việc mua sắm để tiết kiệm thời gian, công sức, do đó, các doanh nghiệp bán lẻ cũng đang dần tiến sâu vào thị trường kinh doanh trực tuyến này với mong muốn tăng hiệu suất và doanh thu bán hàng.

Được biết, khi bán hàng theo phương pháp truyền thống rất bất cập trong vấn đề quảng cáo sản phẩm cũng như thống kê doanh thu. Việc thống kê báo cáo doanh thu hoàn toàn dựa trên phần mềm excel, việc này dẫn đến một số vấn đề khá khó khăn như thời gian xuất một báo cáo rất lâu, tốn nhiều thời gian và nhân lực, nhập liệu nếu có nhầm lẫn sẽ không được cảnh báo và xử lý, chẳng hạn như nếu xuất một sản phẩm mà trong kho số lượng tồn của sản phẩm đó không đủ để xuất hay sản phẩm đã hết hàng nhưng khách hàng không được thông báo về tình trạng này, nhân viên cũng không được nhắc để nhập thêm loại hàng ấy. Ngoài ra, khách hàng không chủ động được trong việc đặt trước và mua hàng của shop, mọi việc bán và mua phải diễn ra đồng thời giữa cả bên mua và bên bán.

Do đó, vấn đề xây dựng một shop bán hàng online rất cần thiết. Việc này giúp shop bán hàng quản lý hàng hóa tốt hơn, thống kê báo cáo doanh thu nhanh chóng, chính xác, giảm nhân lực, giảm chi phí và thời gian. Việc quảng cáo sản phẩm và thương hiệu cũng dễ dàng hơn. Về mặt khách hàng, một shop online sẽ giúp họ tiết kiệm thời gian mua sắm cũng như giúp “đặt chỗ” trước những sản phẩm mà họ yêu thích, không sợ bị mua trước hay hết hàng khi họ không đến shop, chủ động trong việc xem và đặt hàng.

Vì các vấn đề đã nêu, em chọn đề tài **“Xây dựng website bán hàng lĩnh vực thời trang”** nhằm giúp cửa hàng thời trang cải thiện được vấn đề quản lý, quảng cáo hàng hóa của mình. Đồng thời phù hợp với xu hướng phát triển của thế giới hiện nay.

Chương trình này xây dựng trên nền tảng ngôn ngữ Javascript. Với kiến thức còn hạn chế, chắc chắn không thể tránh khỏi những sai sót. Vì vậy, em rất mong nhận được mọi sự góp ý của quý thầy cô để em có thể hoàn thiện đề tài tốt hơn.

1.2. Mục tiêu của đề tài.

Đề tài “**Xây dựng website bán hàng lĩnh vực thời trang**” được thực hiện với mục tiêu ứng dụng công nghệ thông tin nhằm hỗ trợ hoạt động quản lý và kinh doanh trong lĩnh vực thời trang, đồng thời mang đến cho khách hàng một kênh mua sắm hiện đại, tiện lợi và phù hợp với xu hướng thương mại điện tử. Nghiên cứu tập trung vào việc tìm hiểu, lựa chọn và ứng dụng các công nghệ web tiên tiến để xây dựng một hệ thống bán hàng trực tuyến có tính khả thi cao, vừa đáp ứng nhu cầu thực tế của cửa hàng vừa phục vụ tốt trải nghiệm của người dùng.

Mục tiêu chính của đề tài là xây dựng một hệ thống website có thể quản lý đầy đủ các nghiệp vụ trong hoạt động bán hàng, bao gồm quản lý sản phẩm, theo dõi tình trạng kho, quản lý khách hàng và đơn hàng, cũng như thống kê và báo cáo doanh thu một cách nhanh chóng, chính xác. Bên cạnh đó, hệ thống còn được thiết kế để hỗ trợ hoạt động quảng bá sản phẩm, giúp cửa hàng dễ dàng tiếp cận khách hàng thông qua giao diện thân thiện, trực quan và dễ sử dụng.

Trong quá trình nghiên cứu và thực hiện, đề tài cũng hướng đến việc ứng dụng các công nghệ web hiện đại như ReactJS cho phần giao diện và Node.js cho phần xử lý phía máy chủ, kết hợp với hệ quản trị cơ sở dữ liệu để tạo ra một hệ thống ổn định, có khả năng mở rộng và đảm bảo hiệu suất hoạt động. Đây là cơ sở để khảo sát và ứng dụng các tiến bộ trong lĩnh vực công nghệ thông tin vào thực tiễn kinh doanh bán lẻ, đặc biệt là trong ngành thời trang.

Một mục tiêu quan trọng khác là nâng cao trải nghiệm của khách hàng. Người dùng có thể dễ dàng tìm kiếm, xem thông tin chi tiết sản phẩm, kiểm tra tình trạng còn hàng, đặt mua và theo dõi đơn hàng trực tuyến. Việc này không chỉ giúp tiết kiệm thời gian mà còn tạo cho khách hàng cảm giác chủ động trong việc mua sắm, tránh tình trạng hết hàng hoặc phải chờ đợi lâu.

Ngoài ra, hệ thống còn góp phần hỗ trợ chủ cửa hàng trong việc hoạch định chiến lược kinh doanh thông qua các chức năng thống kê, báo cáo doanh thu và quản lý dữ

liệu khách hàng. Đây sẽ là công cụ hữu ích giúp cửa hàng thời trang cải thiện hiệu quả hoạt động, giảm thiểu chi phí quản lý, nhân lực và dễ dàng mở rộng quy mô kinh doanh.

Về mặt ứng dụng thực tiễn, sản phẩm của đề tài có thể triển khai trực tiếp cho các cửa hàng thời trang nhỏ và vừa, đồng thời có khả năng phát triển để áp dụng cho nhiều lĩnh vực bán lẻ khác. Hơn nữa, đề tài cũng mở ra hướng nghiên cứu tiếp theo, chẳng hạn như tích hợp các công nghệ thanh toán trực tuyến, trí tuệ nhân tạo trong gợi ý sản phẩm, hoặc hệ thống quản lý quan hệ khách hàng (CRM). Như vậy, mục tiêu cuối cùng của đề tài không chỉ dừng lại ở việc xây dựng một website bán hàng thông thường, mà còn là bước đầu để ứng dụng công nghệ thông tin vào chuyển đổi số trong lĩnh vực kinh doanh thời trang.

1.3. Đối tượng nghiên cứu

Đề tài tập trung nghiên cứu và phát triển một website bán hàng trực tuyến, trong đó xác định hai đối tượng (actor) chính tham gia vào hệ thống bao gồm: Khách hàng và Người quản trị. Mỗi đối tượng đóng một vai trò khác nhau trong quá trình vận hành và sử dụng hệ thống.

➤ Đối với khách hàng

Là người sử dụng cuối của hệ thống với vai trò chính là mua sắm sản phẩm thời trang trên website. Các chức năng chính mà khách hàng có thể thực hiện bao gồm:

- Truy cập website để xem sản phẩm thời trang.
- Đăng nhập, đăng ký.
- Đổi mật khẩu.
- Xem sản phẩm.
- Đưa sản phẩm vào giỏ hàng.
- Đặt mua/thanh toán.
- Theo dõi đơn hàng và cập nhật thông tin cá nhân.
- ChatBOT hỗ trợ khách hàng.

➤ Đối với người quản trị

Là người chịu trách nhiệm quản lý và điều hành toàn bộ hoạt động của hệ thống. Các chức năng chính của người quản trị bao gồm:

- Quản lý danh mục (thêm, sửa, xóa).
- Quản lý thông tin sản phẩm.
- Quản lý đơn hàng, xác nhận và xử lý tình trạng đơn.
- Quản lý người dùng.
- Quản lý nhập hàng.
- Theo dõi doanh thu và thống kê.

1.4. Phạm vi nghiên cứu.

1.4.1. Về chức năng.

Tập trung vào các chức năng cơ bản của một website bán hàng trực tuyến, bao gồm: đăng nhập, đăng ký, quản lý tài khoản, xem sản phẩm, giỏ hàng, thanh toán, quản lý đơn hàng (cho cả khách hàng và quản trị viên), chatbot.

Các chức năng nâng cao như tích hợp mạng xã hội, chương trình khuyến mãi, blog,... có thể không được đề cập đến hoặc chỉ được nghiên cứu ở mức độ cơ bản.

1.4.2. Về công nghệ.

Sử dụng ReactJS và NodeJS, ExpressJs(phiên bản được chỉ định trong đề tài) để xây dựng website.

Cơ sở dữ liệu MySQL được sử dụng để lưu trữ dữ liệu.

Các công nghệ front-end chủ yếu là HTML, CSS và JavaScript.

Phạm vi nghiên cứu không bao gồm việc tìm hiểu sâu vào các công nghệ khác như AI, Machine Learning, Big Data,...

1.4.3. Về sản phẩm.

Website tập trung vào việc bán các sản phẩm thời trang của cửa hàng.

Các loại sản phẩm cụ thể có thể được giới hạn trong một số danh mục nhất định (ví dụ: quần áo, giày, đồng hồ...).

1.4.4. Về giao diện.

Nghiên cứu thiết kế giao diện website theo phong cách hiện đại, sang trọng, phù hợp với cửa hàng thời trang.

Phạm vi nghiên cứu có thể tập trung vào thiết kế giao diện cho các thiết bị desktop và mobile phổ biến.

1.4.5. Về môi trường triển khai

Website được thiết kế để triển khai trên môi trường web, sử dụng các công cụ và dịch vụ phổ biến (hosting, domain,...).

Phạm vi nghiên cứu không bao gồm việc triển khai website trên các nền tảng đặc thù khác.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ PHÁT TRIỂN

2.1. Cơ sở lý thuyết.

2.1.1. Giới thiệu về ngôn ngữ JavaScript.

JavaScript là một ngôn ngữ lập trình dạng kịch bản, đóng vai trò quan trọng trong việc tạo ra sự tương tác và sinh động cho các trang web. Ngôn ngữ này dựa trên mô hình hướng đối tượng, cho phép lập trình viên sử dụng các đối tượng có sẵn hoặc tự định nghĩa để xây dựng chức năng. Nhờ tính tiện lợi, khả năng tích hợp linh hoạt và hiệu quả cao trong phát triển giao diện người dùng, JavaScript ngày càng trở thành lựa chọn phổ biến và được ứng dụng rộng rãi trong các website hiện đại.

Việc sử dụng ngôn ngữ JS có thể ứng dụng cho mọi trình duyệt khác nhau, hiện được sử dụng phổ biến như Chrome, hay Firefox, ... Hơn nữa, đây còn là ngôn ngữ lập trình hoạt động hiệu quả, được hỗ trợ đầy đủ trên các trình duyệt của thiết bị di động. Bởi thế mà việc sử dụng đa dạng, có thể đáp ứng tốt cho nhiều nhu cầu, những đòi hỏi khác nhau của người dùng.

Nhìn chung, ngôn ngữ lập trình Javascript khi sử dụng được đánh giá dễ học, được phát triển bởi Netscape và sử dụng trong hơn 92% các website ngày nay. Việc sử dụng JS có thể gắn trong một element trong website, hoặc có thể trong một sự kiện của trang web tiện lợi và hiệu quả. Chính việc có thể hoạt động nhanh, nhẹ và mượt mà,... nên ngôn ngữ lập trình này càng được tin tưởng và ứng dụng nhiều hơn cho nhu cầu của người dùng. Chúng ta có thể thêm trực tiếp JS trực tiếp với HTML, cũng có thể lưu lại trên một files hoàn toàn riêng biệt và sử dụng khi cần thiết.

Những ưu điểm làm nên chất lượng của ngôn ngữ JS:

Sử dụng Javascript được thực hiện từ phía khách hàng: Điều này thức là các mã được tiến hành thực hiện thông qua bộ vi xử lý của chính người dùng, thay vì sử dụng trên máy chủ web. Chính đặc điểm này giúp việc tiết kiệm năng thông, đồng thời giảm những căng thẳng, hoạt động quá nhiều trên máy chủ web dễ dàng hơn.

Là ngôn ngữ lập trình dễ dàng sử dụng: Đơn giản, dễ dàng tìm hiểu và sử dụng là đặc điểm nổi bật của ngôn ngữ lập trình này. Với những cú pháp khá tương đồng với tiếng Anh thì việc sử dụng Javascript trở nên dễ dàng, dễ tiếp cận hơn rất nhiều. Thông

qua mô hình DOM được sử dụng, cung cấp nhiều tính năng hữu ích, được viết sẵn đem lại khả năng đáp ứng tốt cho những nhu cầu, những đòi hỏi khác nhau từ phía người dùng. Đánh giá cao ở độ nhanh chóng với người dùng: Việc sử dụng ngôn ngữ JS có khả năng đảm bảo khi mã được thực hiện trên thiết bị máy tính của người dùng

Thì quá trình xử lý, lần kết quả đều hoàn thành gần như ngay lập tức. Thời gian cụ thể phụ thuộc vào từng nhiệm vụ cụ thể mất thời gian nhất định, song thường nó rất nhanh có thể giúp vấn đề được giải quyết.

Sử dụng đa dạng trên nhiều trình duyệt: Một ưu điểm nổi bật giúp JS ngày càng được tin dùng, ứng dụng phong phú là ở khả năng sử dụng thông qua nhiều trình duyệt khác nhau. Từ các trình duyệt trên máy tính như Chrome, tới Firefox, ... hay những trình duyệt trên thiết bị di động, ... đều có thể sử dụng được đầy đủ và hiệu quả. Chính vì sự thích hợp với nhiều trình duyệt, nhiều nền tảng khác nhau giúp nó được tin dùng nhiều hơn để đáp ứng tốt cho đòi hỏi của con người.

Ví dụ:

```
// Hiển thị một thông báo chào mừng function
greetUser(name) { alert(`Xin chào, ${name}`!
    Chào mừng bạn đến với JavaScript.`; }

// Gọi hàm greetUser('Nam');
```

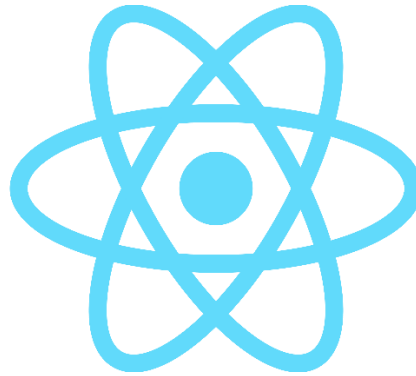
Lợi ích của Javascript:

- Tính năng tương tác (Interactivity): JavaScript giúp tạo các trang web tương tác, như hiển thị thông báo, xử lý sự kiện (click, hover), hoặc cập nhật nội dung trang mà không cần tải lại.
- Khả năng đa nền tảng: JavaScript có thể chạy trên mọi trình duyệt hiện đại và không cần phần mềm bổ sung.
- Thời gian thực (Real-time): JavaScript hỗ trợ xây dựng các ứng dụng thời gian thực như chat, thông báo, và các ứng dụng live-streaming.
- Hỗ trợ thư viện và framework mạnh mẽ: Các thư viện và framework phổ biến như React, Angular, và Vue.js giúp tăng tốc độ phát triển ứng dụng.

- Tích hợp dễ dàng với HTML/CSS: JavaScript hoạt động liền mạch với HTML và CSS, giúp tạo ra giao diện người dùng hấp dẫn và dễ sử dụng.
- Chạy trên cả phía client và server: Với Node.js, JavaScript không chỉ giới hạn ở phía client mà còn được sử dụng cho các ứng dụng server-side mạnh mẽ.

2.1.2. Giới thiệu về ReactJS.

ReactJS là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, dùng để xây dựng giao diện người dùng (UI) cho các ứng dụng web. ReactJS giúp các nhà phát triển tạo ra các ứng dụng web có khả năng tương tác cao và hiệu suất tốt bằng cách sử dụng khái niệm component-based architecture (kiến trúc dựa trên thành phần).



Hình 2.1. Logo Reactjs

Khi sử dụng ReactJs người dùng có thể khai thác được nhiều tính năng khác nhau, trong đó có:

- Xây dựng giao diện người dùng (UI) linh hoạt và tái sử dụng.
- Tối ưu hóa hiệu suất với Virtual DOM.
- Tích hợp dễ dàng với các công nghệ khác.
- Tính năng cập nhật nhanh chóng với ES6+.

Lý do ReactJs được nhiều doanh nghiệp phần mềm hướng tới trong thương mại điện tử:

- **Hiệu suất cao và tối ưu trải nghiệm người dùng:** React sử dụng Virtual DOM (DOM ảo) để tối ưu hóa quá trình cập nhật giao diện khi có sự thay đổi, giúp tăng tốc độ xử lý và cải thiện hiệu năng của ứng dụng. Điều này đặc biệt quan

trọng trong các trang thương mại điện tử, nơi người dùng thường xuyên thực hiện các thao tác như tìm kiếm sản phẩm, thêm vào giỏ hàng, thanh toán,... Việc phản hồi nhanh và giao diện mượt mà góp phần nâng cao trải nghiệm người dùng và cải thiện tỷ lệ chuyển đổi.

- **Lazy Loading (Tải lười):** React hỗ trợ tính năng lazy loading, cho phép tải nội dung chỉ khi thực sự cần thiết. Điều này giúp giảm thời gian tải ban đầu của trang web, đặc biệt hữu ích trong các ứng dụng thương mại điện tử có lượng nội dung lớn. Nhờ đó, tốc độ tải trang được cải thiện rõ rệt, đồng thời tối ưu hóa trải nghiệm người dùng trên cả thiết bị di động và máy tính.
- **Tính linh hoạt và tái sử dụng cao:** Component-based Architecture: Trong thương mại điện tử, nhiều phần của trang web có thể được sử dụng lại như các danh sách sản phẩm, biểu mẫu thanh toán, hoặc thanh điều hướng. React giúp tạo các component có thể tái sử dụng, từ đó giảm chi phí phát triển, bảo trì và cập nhật hệ thống. Điều này cũng giúp dễ dàng mở rộng quy mô ứng dụng khi thêm tính năng mới. Modular Development: Kiến trúc mô-đun của React cho phép đội ngũ phát triển làm việc song song trên các phần khác nhau của ứng dụng, tăng tốc độ triển khai và cải thiện tính linh hoạt trong phát triển.
- **Khả năng tối ưu hóa SEO:** Server-side Rendering (SSR): Mặc dù React là thư viện JavaScript, nó có thể được tích hợp với các framework như Next.js để hỗ trợ render phía máy chủ (Server-side Rendering), giúp tối ưu hóa SEO cho các trang thương mại điện tử. Điều này giúp cải thiện xếp hạng trang web trên các công cụ tìm kiếm, tăng lưu lượng truy cập và khả năng bán hàng. Improved SEO Performance: Các trang thương mại điện tử cần tối ưu hóa cho SEO để tăng khả năng hiển thị sản phẩm trên các công cụ tìm kiếm. React, khi được kết hợp với SSR hoặc prerendering, giúp cải thiện khả năng lập chỉ mục của trang web và làm tăng thứ hạng tìm kiếm.
- **Khả năng cá nhân hóa trải nghiệm người dùng:** Personalized Shopping Experience: React hỗ trợ việc tạo ra trải nghiệm mua sắm cá nhân hóa theo hành vi và sở thích của người dùng, từ đó tăng tỷ lệ chuyển đổi và doanh thu. Các tính năng như đề xuất sản phẩm, danh sách yêu thích, hoặc bộ lọc tìm kiếm sản phẩm được thực hiện dễ dàng và hiệu quả với React.

Ví dụ:

```
import React from 'react'; import
ReactDOM from 'react-dom';

// Tạo một thành phần React (React Component) function
Welcome(props) { return <h1>Xin chào,
{props.name}</h1>;

}

// Hiển thị thành phần lên giao diện
ReactDOM.render(

  <Welcome name="Nam" />,

  document.getElementById('root');
```

Lợi ích của ReactJs:

- **Tái sử dụng thành phần (Reusable Components):** React cho phép bạn xây dựng các thành phần UI độc lập, có thể tái sử dụng trong toàn bộ ứng dụng, giúp tiết kiệm thời gian phát triển.
- **Hiệu suất cao (High Performance):** Với Virtual DOM, React chỉ cập nhật những phần giao diện thay đổi thay vì làm mới toàn bộ trang, giúp tăng tốc độ xử lý.
- **Định hướng dữ liệu (Unidirectional Data Flow):** React quản lý dữ liệu theo hướng đơn chiều, giúp dễ dàng kiểm soát trạng thái và luồng dữ liệu trong ứng dụng.
- **Dễ bảo trì:** Cấu trúc thành phần của React giúp mã nguồn rõ ràng và dễ bảo trì, đặc biệt đối với các dự án lớn.
- **SEO thân thiện (SEO-Friendly):** React có thể sử dụng kèm với serverside rendering (SSR) để tăng cường khả năng SEO.
- **Hỗ trợ cộng đồng và công cụ mạnh mẽ:** React có một cộng đồng lớn và một hệ sinh thái mạnh mẽ, bao gồm các thư viện như React Router, Redux, và Next.js.

Khi nào nên sử dụng ReactJs:

- Khi bạn cần xây dựng một giao diện người dùng phức tạp và cần hiệu suất cao.
- Khi bạn muốn phát triển ứng dụng web với khả năng tái sử dụng mã nguồn.
- Khi bạn muốn tạo một ứng dụng single-page application (SPA) mượt mà.

2.1.3. Giới thiệu về ExpressJS.

Express.js là một framework web nhẹ và linh hoạt cho Node.js, được thiết kế để xây dựng các ứng dụng web và API một cách nhanh chóng và hiệu quả. Express.js đơn giản hóa quy trình phát triển ứng dụng phía server, cung cấp một bộ công cụ mạnh mẽ giúp các nhà phát triển dễ dàng xử lý yêu cầu HTTP, định tuyến, quản lý session, và nhiều tính năng khác mà không cần viết quá nhiều mã.

- **Dựa trên Node.js:** Express.js được xây dựng trên nền tảng của Node.js, sử dụng môi trường JavaScript phía server. Điều này giúp các nhà phát triển sử dụng JavaScript cho cả phía server và client, tạo nên tính đồng nhất và giảm thiểu thời gian học tập khi chỉ cần sử dụng một ngôn ngữ duy nhất.
- **Tối ưu hóa tốc độ và hiệu suất:** Với kiến trúc nhẹ và không dư thừa tính năng, Express.js giúp tối ưu hóa tốc độ xử lý của các ứng dụng. Điều này rất quan trọng đối với các ứng dụng web yêu cầu hiệu suất cao như thương mại điện tử, mạng xã hội, hoặc các dịch vụ trực tuyến có lượng truy cập lớn.
- **Middleware mạnh mẽ:** Middleware là một tính năng quan trọng trong Express.js, cho phép bạn thêm các lớp xử lý giữa yêu cầu và phản hồi. Các middleware có thể được sử dụng để xử lý xác thực, ghi log, quản lý session, phân tích yêu cầu (requestparsing), hoặc thực hiện bất kỳ chức năng nào trước khi gửi phản hồi tới client.
- **Hỗ trợ đầy đủ REST API:** Express.js là framework lý tưởng để xây dựng các API RESTful. Nó cung cấp các công cụ để xây dựng các endpoint, xử lý yêu cầu từ phía client, và trả về phản hồi theo chuẩn REST, phù hợp cho việc phát triển dịch vụ web và các ứng dụng di động.
- **Hệ sinh thái lớn:** Express.js có hệ sinh thái lớn và cộng đồng phát triển mạnh mẽ. Điều này có nghĩa là có rất nhiều module và middleware sẵn có mà bạn có

thể dễ dàng tích hợp vào ứng dụng của mình để tăng cường chức năng, chẳng hạn như bảo mật, xác thực, quản lý session, hoặc xử lý dữ liệu.

- **Hỗ trợ nhiều chế độ xem (view engine):** Express.js hỗ trợ nhiều engine cho phép tạo ra các tệp HTML động. Các engine phổ biến bao gồm Pug (trước đây là Jade), EJS, Handlebars, và nhiều lựa chọn khác. Điều này giúp dễ dàng render giao diện người dùng với dữ liệu động.
- **Cộng đồng lớn:** Express.js có một cộng đồng phát triển mạnh mẽ và rất nhiều tài liệu và ví dụ trực tuyến giúp bạn học và sử dụng nó dễ dàng.

Ví dụ:

```
const express = require('express');

const app = express();

// Định nghĩa một route cơ bản

app.get('/', (req, res) => {

  res.send('Hello, Express!');

});

// Khởi chạy server tại cổng 3000

app.listen(3000, () => {

  console.log('Server đang chạy tại http://localhost:3000');

});
```

Lợi ích của Express.js:

- Dễ học và sử dụng: Express.js có cú pháp đơn giản, phù hợp với cả người mới học lập trình.
- Xử lý nhanh: Nhờ sự kết hợp với Node.js, Express có thể xử lý hàng ngàn kết nối đồng thời.
- Tùy chỉnh linh hoạt: Bạn có thể thêm các middleware và module tùy chỉnh để đáp ứng yêu cầu dự án.

- Phát triển nhanh: Với hệ sinh thái module phong phú, bạn có thể dễ dàng thêm các tính năng như xác thực, ghi log, và cache.
- Hỗ trợ tốt cho API: Express.js là lựa chọn hàng đầu để xây dựng các API RESTful hiện đại.

Khi nào nên sử dụng Express.js:

- Khi bạn muốn xây dựng API nhanh chóng và hiệu quả.
- Khi cần tạo các ứng dụng web với cấu trúc linh hoạt.
- Khi làm việc với Node.js và cần một framework tối ưu.

2.1.4. Giới thiệu về NodeJS.

Node.js là một nền tảng chạy JavaScript phía server, giúp các nhà phát triển xây dựng các ứng dụng mạng nhanh chóng, hiệu quả và có khả năng mở rộng cao. Node.js được phát triển dựa trên V8 JavaScript Engine của Google, trình thông dịch JavaScript được sử dụng trong trình duyệt Chrome. Điểm đặc trưng của Node.js là kiến trúc hướng sự kiện (event-driven) và mô hình không đồng bộ (asynchronous), giúp xử lý các yêu cầu từ client nhanh chóng và hiệu quả.

JavaScript phía server: Node.js cho phép các nhà phát triển sử dụng JavaScript ở phía server. Điều này mang lại sự thống nhất về ngôn ngữ lập trình giữa client và server, giúp giảm bớt sự phức tạp và cải thiện năng suất phát triển.

Không đồng bộ và hướng sự kiện: Node.js sử dụng mô hình non-blocking I/O (nhập/xuất không chặn) và xử lý theo hướng sự kiện, giúp hệ thống có thể xử lý nhiều yêu cầu cùng lúc mà không bị chặn bởi các yêu cầu trước đó. Điều này giúp tối ưu hiệu suất, đặc biệt trong

Cross-platform: Node.js hoạt động trên nhiều hệ điều hành như Windows, macOS, Linux và thậm chí trên các thiết bị nhúng như Raspberry Pi, giúp nó trở thành một nền tảng linh hoạt cho các loại dự án khác nhau.

Xử lý đơn luồng với Event Loop: Mặc dù Node.js chỉ sử dụng một luồng để xử lý các yêu cầu, nó sử dụng Event Loop để quản lý nhiều kết nối cùng một lúc mà không cần nhiều luồng. Điều này giúp Node.js có thể xử lý nhiều yêu cầu đồng thời mà không

gặp các vấn đề thường gặp với các hệ thống đa luồng như cạnh tranh tài nguyên (resource contention).

Hỗ trợ WebSockets: Node.js hỗ trợ các kết nối WebSocket, giúp dễ dàng xây dựng các ứng dụng thời gian thực (real-time) như ứng dụng chat, các trò chơi nhiều người chơi, hoặc hệ thống theo dõi sự kiện.

Ví dụ:

```
// Import module HTTP có sẵn trong Node.js
const http = require('http');

// Tạo một server
const server = http.createServer((req, res) => { res.statusCode = 200;
    // Trả về mã trạng thái 200
    (thành công) res.setHeader('Content-Type', 'text/plain'); // Đặt
    kiểu dữ liệu trả về res.end('Hello, World!'); // Trả về nội
    dung
});

// Khởi động server tại cổng 3000
server.listen(3000, () => { console.log('Server
chạy tại http://localhost:3000/'); })
```

Ứng dụng của Node.js:

- API Backend: Node.js rất phổ biến để xây dựng các API RESTful với các thư viện như Express.js.
- Ứng dụng real-time: Node.js lý tưởng cho các ứng dụng thời gian thực như chat hoặc hệ thống thông báo.
- Xử lý file: Node.js có thể xử lý file (đọc, ghi, xóa) trên server một cách dễ dàng.
- Microservices: Node.js rất phù hợp cho các kiến trúc microservices nhờ khả năng nhẹ và hiệu suất cao.
- Ứng dụng IoT: Các thiết bị IoT thường sử dụng Node.js để xử lý dữ liệu cảm biến trong thời gian thực

Các thư viện phổ biến của Node.js:

- Express.js: Framework nhẹ, linh hoạt để xây dựng web server và API.

- Socket.io: Hỗ trợ giao tiếp real-time giữa client và server.
- JWT (jsonwebtoken): Dùng để xác thực người dùng qua token.
- Lodash: Cung cấp các hàm tiện ích để xử lý dữ liệu.

Lợi ích của Node.Js:

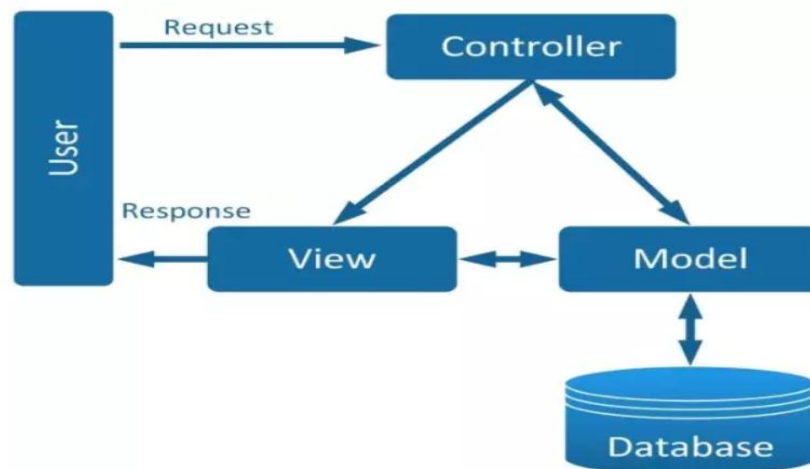
- Đơn giản hóa ngăn xếp công nghệ: Cả frontend và backend đều dùng JavaScript, giúp dễ dàng quản lý và đồng bộ mã nguồn.
- Hiệu suất cao: Xử lý hàng ngàn yêu cầu đồng thời mà không bị nghẽn.
- Cộng đồng lớn: Node.js có một cộng đồng nhà phát triển sôi động, luôn cập nhật công cụ và giải pháp mới.
- Tối ưu chi phí: Hiệu suất cao của Node.js giúp giảm tài nguyên phần cứng cần thiết, từ đó tiết kiệm chi phí.

Khi nào sử dụng Node.Js:

- Khi cần xử lý real-time (chat, game, thông báo).
- Khi cần xây dựng API backend nhanh chóng.
- Khi làm việc với các ứng dụng nhiều kết nối như IoT. Khi cần hiệu suất cao với tài nguyên tối thiểu.

2.1.5. Mô hình MVC

Mô hình MVC (Model-View-Controller) là một kiến trúc phần mềm phổ biến được sử dụng để phát triển các ứng dụng web, trong đó có website bán hàng. MVC chia ứng dụng thành ba thành phần logic riêng biệt, mỗi thành phần có trách nhiệm riêng:



Hình 2.2. Mô hình MVC

- **Model:** Đại diện cho dữ liệu của ứng dụng và các quy tắc nghiệp vụ liên quan đến dữ liệu đó. Model chịu trách nhiệm truy xuất, lưu trữ và xử lý dữ liệu. Trong website bán hàng, Model có thể bao gồm các đối tượng như Sản phẩm, Khách hàng, Đơn hàng,...
- **View:** Chịu trách nhiệm hiển thị dữ liệu cho người dùng. View nhận dữ liệu từ Model và hiển thị nó theo định dạng mà người dùng có thể hiểu được. Trong website bán hàng, View có thể là các trang web hiển thị danh sách sản phẩm, chi tiết sản phẩm, giỏ hàng,...
- **Controller:** Đóng vai trò trung gian giữa Model và View. Controller nhận yêu cầu từ người dùng, xử lý yêu cầu đó bằng cách tương tác với Model, và sau đó chọn View phù hợp để hiển thị kết quả.

Cách thức hoạt động:

1. Người dùng tương tác với View (ví dụ: click vào nút "Thêm vào giỏ hàng").
2. View gửi yêu cầu đến Controller.
3. Controller xử lý yêu cầu bằng cách tương tác với Model (ví dụ: thêm sản phẩm vào giỏ hàng trong database).
4. Controller chọn View phù hợp để hiển thị kết quả (ví dụ: hiển thị lại trang giỏ hàng với sản phẩm mới được thêm vào).

Ưu điểm của mô hình MVC:

- **Tổ chức code rõ ràng:** MVC giúp phân chia code thành các thành phần độc lập, dễ quản lý và bảo trì.
- **Khả năng tái sử dụng:** Các thành phần Model, View, Controller có thể được tái sử dụng ở nhiều nơi trong ứng dụng.
- **Dễ dàng phát triển và bảo trì:** Việc phân chia rõ ràng trách nhiệm giúp cho việc phát triển và bảo trì ứng dụng trở nên dễ dàng hơn.

Mô hình MVC trong Express Js (Model – View – Controller) là một mô hình để xây dựng các ứng dụng web được sử dụng rộng rãi, kết hợp với các framework Javascript front-end như ReactJs. Các ứng dụng di động trên IOS và Android cũng có thể dùng một biến thể của MVC.

2.1.6. Cơ sở dữ liệu MySQL.

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS) mã nguồn mở phổ biến, được sử dụng rộng rãi trong các ứng dụng web. Nó được biết đến với tính ổn định, hiệu năng cao và dễ sử dụng, đặc biệt là khi kết hợp với PHP.

Giới thiệu MySQL

MySQL được phát triển bởi công ty MySQL AB (nay thuộc sở hữu của Oracle) và được phát hành lần đầu tiên vào năm 1995. Nó là một thành phần quan trọng trong bộ công nghệ LAMP (Linux, Apache, MySQL, PHP), một trong những nền tảng phổ biến nhất để xây dựng các ứng dụng web.

Đặc điểm của MySQL:

- Mã nguồn mở: MySQL là phần mềm mã nguồn mở, có nghĩa là bạn có thể sử dụng, sửa đổi và phân phối nó một cách tự do.
- Quan hệ: MySQL là một hệ quản trị cơ sở dữ liệu quan hệ, dữ liệu được tổ chức thành các bảng với các mối quan hệ được xác định rõ ràng giữa chúng.
- Hiệu năng cao: MySQL được tối ưu hóa cho hiệu năng cao, có khả năng xử lý một lượng lớn dữ liệu và truy vấn phức tạp.
- Ổn định: MySQL được biết đến với tính ổn định và độ tin cậy cao, thích hợp cho các ứng dụng yêu cầu hoạt động liên tục.
- Dễ sử dụng: MySQL có cú pháp đơn giản, dễ học và dễ sử dụng, kể cả đối với những người mới bắt đầu.
- Hỗ trợ nhiều nền tảng: MySQL có thể chạy trên nhiều hệ điều hành khác nhau như Windows, Linux, macOS.
- Cộng đồng lớn: MySQL có một cộng đồng người dùng đông đảo và tích cực, luôn sẵn sàng hỗ trợ và chia sẻ kiến thức.

Ưu điểm của MySQL:

- Chi phí thấp: MySQL là phần mềm mã nguồn mở, bạn có thể sử dụng nó miễn phí.

- Dễ dàng cài đặt và sử dụng: Việc cài đặt và cấu hình MySQL tương đối đơn giản.
- Hiệu năng tốt: MySQL được tối ưu hóa cho tốc độ và hiệu năng.
- Bảo mật: MySQL cung cấp các tính năng bảo mật để bảo vệ dữ liệu.
- Khả năng mở rộng: MySQL có thể mở rộng để xử lý lượng dữ liệu lớn.

Ứng dụng của MySQL:

MySQL được sử dụng rộng rãi trong nhiều ứng dụng, bao gồm:

- Ứng dụng web: MySQL là một trong những lựa chọn phổ biến nhất cho các ứng dụng web, đặc biệt là các website thương mại điện tử, blog, forum,...
- Hệ thống quản lý nội dung (CMS): Nhiều hệ thống CMS phổ biến như WordPress, Drupal, Joomla! sử dụng MySQL để lưu trữ dữ liệu.
- Ứng dụng doanh nghiệp: MySQL được sử dụng trong các ứng dụng doanh nghiệp để quản lý dữ liệu khách hàng, nhân viên, sản phẩm,...
- Hệ thống nhúng: MySQL có thể được nhúng vào các hệ thống nhúng để lưu trữ dữ liệu.

Trong đồ án của bạn, MySQL sẽ được sử dụng để lưu trữ dữ liệu cho website bán hàng, bao gồm thông tin về sản phẩm, khách hàng, đơn hàng,...

2.1.7. Giới thiệu VNPAY

VNPAY là một nền tảng thanh toán điện tử hàng đầu tại Việt Nam, cung cấp các dịch vụ thanh toán trực tuyến và QR code nhằm tạo điều kiện cho người dùng thực hiện thanh toán dễ dàng, an toàn và nhanh chóng. VNPAY hỗ trợ đa dạng các phương thức thanh toán như thanh toán qua ứng dụng di động, ngân hàng điện tử, hoặc quét mã QR. Nền tảng này giúp các doanh nghiệp, đặc biệt trong lĩnh vực thương mại điện tử, tối ưu hóa quy trình thanh toán và nâng cao trải nghiệm khách hàng.

Ưu điểm của VNPAY:

- Bảo mật cao: VNPAY tuân thủ các tiêu chuẩn bảo mật quốc tế, đảm bảo an toàn cho thông tin thanh toán của khách hàng.

- **Tiện lợi:** Hỗ trợ thanh toán qua nhiều phương thức, bao gồm mã QR, ví điện tử, và liên kết với nhiều ngân hàng nội địa.
- **Tích hợp dễ dàng:** API của VNPAY cho phép tích hợp dễ dàng vào các hệ thống website hoặc ứng dụng di động.
- **Mạng lưới rộng rãi:** VNPAY liên kết với nhiều ngân hàng và đối tác, tạo điều kiện thuận lợi cho người dùng.

Nhược điểm của VNPay:

- **Chi phí giao dịch:** Có thể có phí dịch vụ đối với các giao dịch qua VNPAY, đặc biệt là với các giao dịch giá trị lớn.
- **Tính cạnh tranh với các nền tảng khác:** VNPAY phải cạnh tranh với các ví điện tử và nền tảng thanh toán lớn khác như Momo, ZaloPay, và các nền tảng thanh toán quốc tế, khiến khách hàng có nhiều lựa chọn thay thế.

2.1.8. Giới thiệu GeminiAI.

GeminiAI là nền tảng trí tuệ nhân tạo do Google phát triển, hỗ trợ tạo lập nội dung, xử lý ngôn ngữ tự nhiên (NLP), và xây dựng chatbot tự động tương tác với người dùng trên website. Dựa trên mô hình ngôn ngữ tiên tiến, GeminiAI có khả năng phân tích ngữ cảnh, hiểu câu hỏi và đưa ra câu trả lời tự nhiên, chính xác. Nhờ đó, nền tảng này trở thành công cụ mạnh mẽ giúp doanh nghiệp nâng cao trải nghiệm khách hàng, tối ưu hóa quy trình hỗ trợ và tiết kiệm chi phí vận hành.

Ưu điểm của GeminiAI:

- **Khả năng hiểu ngôn ngữ tự nhiên mạnh mẽ:** GeminiAI sử dụng mô hình AI mới nhất, cho phép chatbot trả lời tự nhiên, chính xác và đa dạng tình huống.
- **Tích hợp linh hoạt:** Dễ dàng tích hợp vào website hoặc ứng dụng thông qua API.
- **Tùy chỉnh linh hoạt:** Hỗ trợ tùy chỉnh kịch bản hội thoại, dữ liệu huấn luyện và ngữ cảnh theo nhu cầu ứng dụng thực tế.
- **Hỗ trợ đa ngôn ngữ:** Có khả năng xử lý và giao tiếp bằng nhiều ngôn ngữ khác nhau, phù hợp với lượng khách du lịch đa dạng.

Nhược điểm của GeminiAI:

- **Phụ thuộc vào Internet:** Để hoạt động hiệu quả, GeminiAI yêu cầu kết nối mạng ổn định.

- Chi phí nâng cao: Một số tính năng nâng cao hoặc mức sử dụng lớn có thể phát sinh chi phí tùy theo gói dịch vụ của Google.

2.2. Công cụ sử dụng.

2.2.1. Visual Studio Code.

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mạnh mẽ và miễn phí, được phát triển bởi Microsoft. VS Code hỗ trợ nhiều ngôn ngữ lập trình như PHP, HTML, CSS, JavaScript và có thể mở rộng thông qua các plugin, đặc biệt là các tiện ích hỗ trợ Laravel và PHP, giúp lập trình viên làm việc hiệu quả hơn.

VS Code cung cấp các tính năng mạnh mẽ như kiểm tra lỗi, gợi ý mã, tích hợp Git, và khả năng chạy debug trực tiếp trong trình soạn thảo. Những tính năng này giúp lập trình viên dễ dàng phát hiện và sửa lỗi trong mã nguồn, đồng thời giúp quản lý phiên bản dự án dễ dàng hơn. Với giao diện thân thiện và các công cụ hỗ trợ như IntelliSense, Code Snippets, VS Code giúp tiết kiệm thời gian khi lập trình và làm việc với các dự án web phức tạp như website bán sách. Ngoài ra, VS Code còn hỗ trợ Live Server, cho phép xem kết quả ngay lập tức mà không cần phải tải lại trang web, giúp quá trình phát triển giao diện trở nên nhanh chóng và mượt mà hơn.

Ưu điểm:

- Miễn phí và đa nền tảng: Hoạt động trên Windows, macOS, Linux và hoàn toàn miễn phí.
- Tích hợp mạnh mẽ với plugin: Hỗ trợ nhiều plugin cho PHP, HTML, CSS, JavaScript, giúp nâng cao hiệu suất làm việc và có các tiện ích như IntelliSense, Code Snippets và các tính năng tự động hoàn thành mã, tiết kiệm thời gian.
- Tích hợp Git: Quản lý phiên bản dễ dàng ngay trong giao diện, hỗ trợ lập trình viên theo dõi và kiểm soát mã nguồn hiệu quả.
- Hỗ trợ Live Server: Cho phép hiển thị kết quả lập tức khi sửa mã, giúp quá trình phát triển giao diện nhanh chóng.
- Cộng đồng lớn: Có nhiều tài liệu, hướng dẫn và hỗ trợ từ cộng đồng lập trình viên.

Nhược điểm:

- Tiêu tốn tài nguyên: Với các dự án lớn, VS Code có thể gây chậm hoặc tiêu hao nhiều RAM, đặc biệt trên máy cấu hình thấp.
- Phụ thuộc vào plugin: Để đạt hiệu quả cao nhất, cần cài đặt và cấu hình nhiều plugin, điều này có thể gây phức tạp với người mới.
- Không phải IDE đầy đủ: So với các IDE như PHPStorm, VS Code thiếu một số tính năng nâng cao như phân tích mã chuyên sâu, hỗ trợ framework tự động hơn

2.2.2. Xampp.

XAMPP là phần mềm mã nguồn mở, cung cấp môi trường phát triển web cục bộ. Nó bao gồm Apache (web server), MySQL (quản lý cơ sở dữ liệu), PHP và Perl.

XAMPP giúp tạo ra một môi trường phát triển web hoàn chỉnh trên máy tính cá nhân mà không cần kết nối với server trực tuyến. Việc cài đặt và cấu hình XAMPP rất đơn giản, giúp lập trình viên nhanh chóng triển khai và kiểm thử ứng dụng web.

Trong dự án bán sách trực tuyến, XAMPP sẽ đóng vai trò là nền tảng phát triển, cho phép chạy mã PHP và kết nối với cơ sở dữ liệu MySQL, từ đó xử lý các yêu cầu người dùng như thêm sản phẩm vào giỏ hàng, thanh toán đơn hàng, hay truy vấn thông tin sách. Apache trong XAMPP chịu trách nhiệm xử lý các yêu cầu HTTP và trả về các trang HTML hoặc dữ liệu từ cơ sở dữ liệu. MySQL giúp lưu trữ các thông tin quan trọng như danh mục sách, thông tin khách hàng và lịch sử đơn hàng. Nhờ vào XAMPP, các lập trình viên có thể phát triển và thử nghiệm ứng dụng web mà không cần kết nối internet hoặc thuê server ngoài.

Ưu điểm:

- Dễ cài đặt và sử dụng: Chỉ cần tải về, chạy trình cài đặt, là bạn đã có môi trường phát triển web hoàn chỉnh mà không cần cấu hình phức tạp và giao diện thân thiện, dễ sử dụng ngay cả với người mới bắt đầu.
- Tích hợp đầy đủ các công cụ cần thiết: Cung cấp Apache (máy chủ web), MySQL (cơ sở dữ liệu), PHP và Perl trong một gói duy nhất và bao gồm cả phpMyAdmin để quản lý cơ sở dữ liệu MySQL thông qua giao diện đồ họa.

- Hoàn toàn miễn phí và mã nguồn mở: Không cần trả phí, có thể sử dụng để phát triển ứng dụng cá nhân hoặc học tập.
- Thích hợp cho phát triển cục bộ: Cho phép phát triển và kiểm thử ứng dụng web ngay trên máy tính cá nhân mà không cần kết nối internet và có thể mô phỏng gần như hoàn chỉnh môi trường server thực tế.
- Đa nền tảng: Có thể cài đặt trên Windows, macOS, và Linux.

Nhược điểm

- Hiệu năng không cao: XAMPP được thiết kế để phát triển và kiểm thử, không tối ưu cho môi trường sản xuất và không hỗ trợ tải cao hoặc xử lý nhiều yêu cầu đồng thời như các server chuyên dụng.
- Bảo mật thấp: Vì XAMPP không tập trung vào bảo mật, nếu sử dụng trên môi trường thực tế, ứng dụng có thể dễ bị tấn công và tất cả dịch vụ trong XAMPP đều được bật mặc định, dễ bị khai thác nếu không cấu hình kỹ lưỡng.
- Hạn chế trong tùy chỉnh nâng cao: Không thích hợp cho các ứng dụng web phức tạp hoặc có yêu cầu đặc biệt về cấu hình và khi cần cập nhật phiên bản mới của PHP hoặc MySQL, phải tải và cài đặt toàn bộ gói XAMPP mới.
- Không chuyên biệt: XAMPP là một giải pháp chung, không tối ưu cho từng mục đích riêng như LAMP (Linux, Apache, MySQL, PHP) hoặc Docker.

2.3. Các bước cài đặt công cụ.

2.3.1. Cài đặt Visual Studio code.

Để cài đặt Visual Studio Code (VS Code), thực hiện theo các bước sau:

Bước 1: Tải xuống trình cài đặt

Truy cập trang chủ chính thức: <https://code.visualstudio.com>

Chọn hệ điều hành tương ứng (Windows, macOS hoặc Linux), sau đó tải về.

Bước 2: Cài đặt (trên Windows)

Mở tệp VSCodeSetup.exe vừa tải về.

Nhấn Next liên tục để tiếp tục các bước cài đặt.

Tick chọn các tùy chọn sau để sử dụng tiện lợi hơn:

“Add to PATH” – để có thể mở VS Code từ Command Line.

“Register Code as an editor for supported file types.”

“Add ‘Open with Code’ action to Windows Explorer file context menu.”

Nhấn Install → chờ quá trình cài đặt hoàn tất.

Nhấn Finish để khởi động VS Code

Bước 3: Cài đặt tiện ích mở rộng (Extension)

Mở VS Code → nhấn biểu tượng Extensions (hình ô vuông) bên trái.

Tìm và cài một số extension hữu ích:

Live Server: Chạy ứng dụng web ngay trong trình duyệt.

Prettier: Tự động định dạng mã nguồn.

ESLint: Kiểm tra lỗi và chuẩn hóa mã JavaScript.

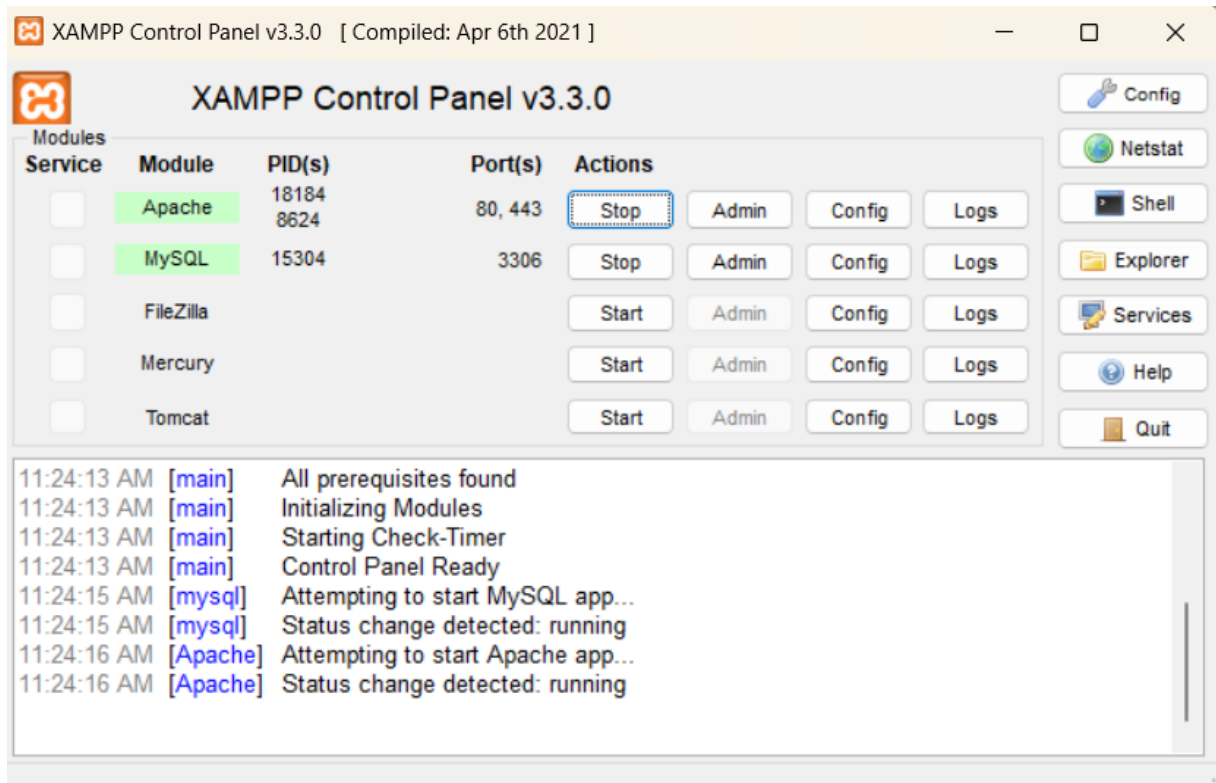
Tailwind CSS IntelliSense (nếu dùng Tailwind).

React snippets, Clerk VS Code Extension, v.v.

2.3.2. Cài đặt Xampp.

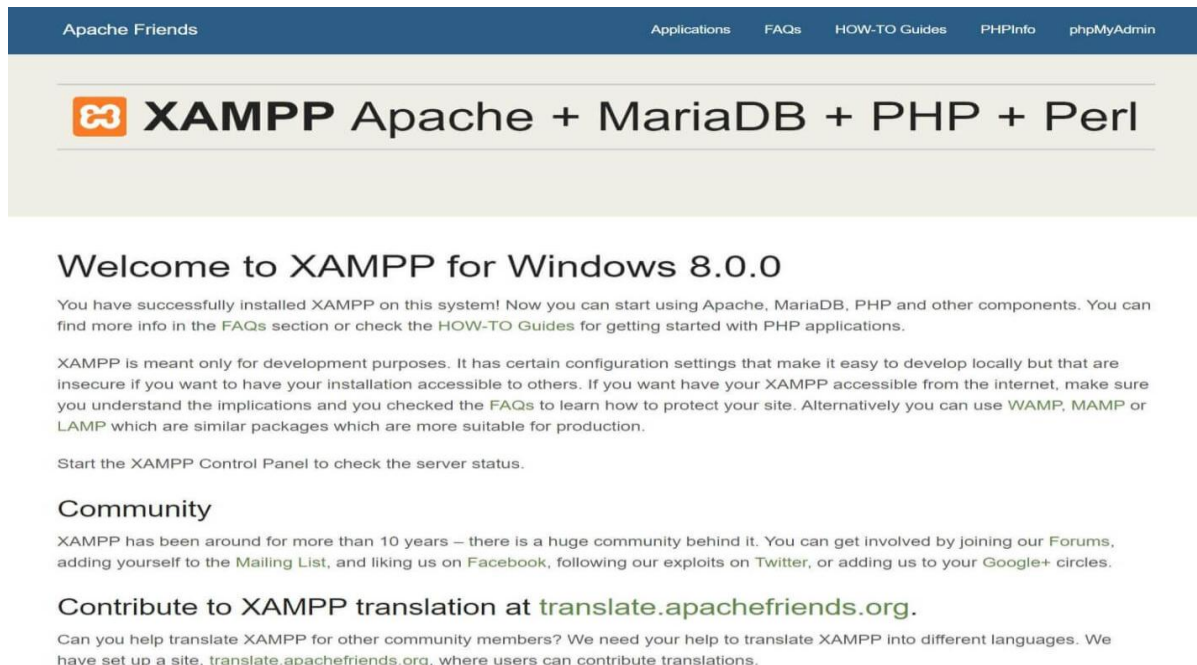
Bước 1: Tải XAMPP từ trang <https://www.apachefriends.org/download.html> và cài đặt như các phần mềm thông dụng khác.

Bước 2: Mở XAMPP control panel và khởi động Apache cùng MySQL để bắt đầu sử dụng.



Hình 2.3. Bảng điều khiển của xampp

Bước 3: Nhập địa chỉ localhost vào trình duyệt. Nếu màn hình dưới đây xuất hiện, điều đó có nghĩa là đã cài đặt thành công.



Hình 2.4. Localhost của xampp

CHƯƠNG 3. TRIỂN KHAI HỆ THỐNG

3.1. Giới thiệu chung.

3.1.1. Về phía khách hàng (User)

Xem danh sách sản phẩm: Khi người dùng truy cập vào website thì hệ thống sẽ tự động hiển thị một số sản phẩm nổi bật. Khi người dùng chọn vào một danh mục sản phẩm cụ thể nào của cửa hàng thì tất cả sản phẩm của danh mục đó sẽ hiển thị lên giao diện

Xem chi tiết sản phẩm: Sau khi tìm thấy sản phẩm cần, người dùng có thể chọn vào sản phẩm để xem thông tin chi tiết của sản phẩm đó.

Tìm kiếm: Khi khách hàng cần tìm một sản phẩm nào đó thì có thể nhập thông tin sản phẩm vào ô tìm kiếm để thực hiện chức năng tìm kiếm. Nếu sản phẩm đó có tồn tại trong dữ liệu thì sẽ hiện thông tin sản phẩm được tìm thấy.

Đăng ký: Khi khách hàng ghé thăm muốn mua hàng thì có thể thực hiện chức năng đăng ký để trở thành thành viên của website và thực hiện các chức năng mua hàng.

Đưa sản phẩm vào giỏ hàng: Sau khi khách hàng tìm thấy sản phẩm đáp ứng nhu cầu của mình thì có thể chọn đưa sản phẩm vào giỏ hàng.

Cập nhật giỏ hàng: Khi người dùng đã lựa chọn một sản phẩm nhưng tìm được một sản phẩm hợp lý hơn thì có thể xóa sản phẩm đã có trong giỏ hàng vào chọn lại sản phẩm mới. Ngoài ra, khi cần mua sản phẩm đó với một số lượng thì người dùng cũng có thể cập nhật trong giỏ hàng.

Thanh toán: Khi đã quyết định chắc chắn mua sản phẩm, người dùng có thể chọn chức năng thanh toán để thực hiện xác nhận đơn hàng cho mình. Muốn thực hiện thanh toán thì yêu cầu người dùng phải đăng nhập vào hệ thống. Trong quá trình thực hiện chức năng thanh toán thì người dùng có thể thấy được sản phẩm mà mình đặt mua trong giỏ hàng. Người dùng cần điền một số thông tin như chọn đơn vị vận chuyển

(giao hàng tiết kiệm, giao hàng nhanh, hỏa tốc), sau đó đi đến thanh toán chọn phương thức thanh toán (Thanh toán online, thanh toán khi nhận hàng,). Sau khi thực hiện xong các bước thì người dùng xác nhận đặt đơn hàng.

Xem danh sách sản phẩm: Khách hàng có thể xem lại các sản phẩm mà mình đã và đang đặt mua trên website.

Quản lý đơn hàng: Sau khi đặt mua khách hàng có thể cập nhật được trạng thái đơn hàng đơn hàng.

Chatbot: Hỗ trợ khách hàng trò chuyện trực tiếp với hệ thống để được gợi ý giá và các size áo quần, thêm các sản phẩm ưng ý của khách hàng vào giỏ hàng..

3.1.2. Về phía Quản trị viên (Administrator)

Quản lý sản phẩm, đơn hàng, khách hàng, một số thông tin khác: Để thực hiện chức năng quản lý thì yêu cầu nhân viên phải đăng nhập hệ thống. Nhân viên chọn mục mình cần thực hiện quản lý như sản phẩm. Trong quản lý sản phẩm có xem, thêm, sửa, xóa sản phẩm. Chọn thao tác cần thực hiện, điền đầy đủ thông tin trong quá trình thực hiện rồi xác nhận thao tác. Các thao tác quản lý cần có quyền hạn tương ứng để thực hiện, nếu không đủ quyền hạn sẽ không thực hiện được.

Người quản trị : Người có toàn quyền trong hệ thống, có tất cả chức năng của nhân viên, người dùng.

Từ các yêu cầu trên website xây dựng phải có các yêu cầu sau:

Các trang này cần có các chức năng xem, thêm, sửa, xóa, tìm kiếm, phân trang một cách hợp lý,...

- Quản lý danh mục.
- Quản lý sản phẩm .
- Quản lý nhãn hàng.
- Quản lý băng rôn.
- Quản lý về chủ đề.
- Quản lý bài đăng.
- Quản lý loại ship
- Quản lý nhập hàng.
- Quản lý đơn hàng
- Quản lý tin nhắn
- Quản lý người dùng.

- Quản lý nhà cung cấp.
- Quản lý thông kê(doanh thu, lợi nhuận, tồn kho).

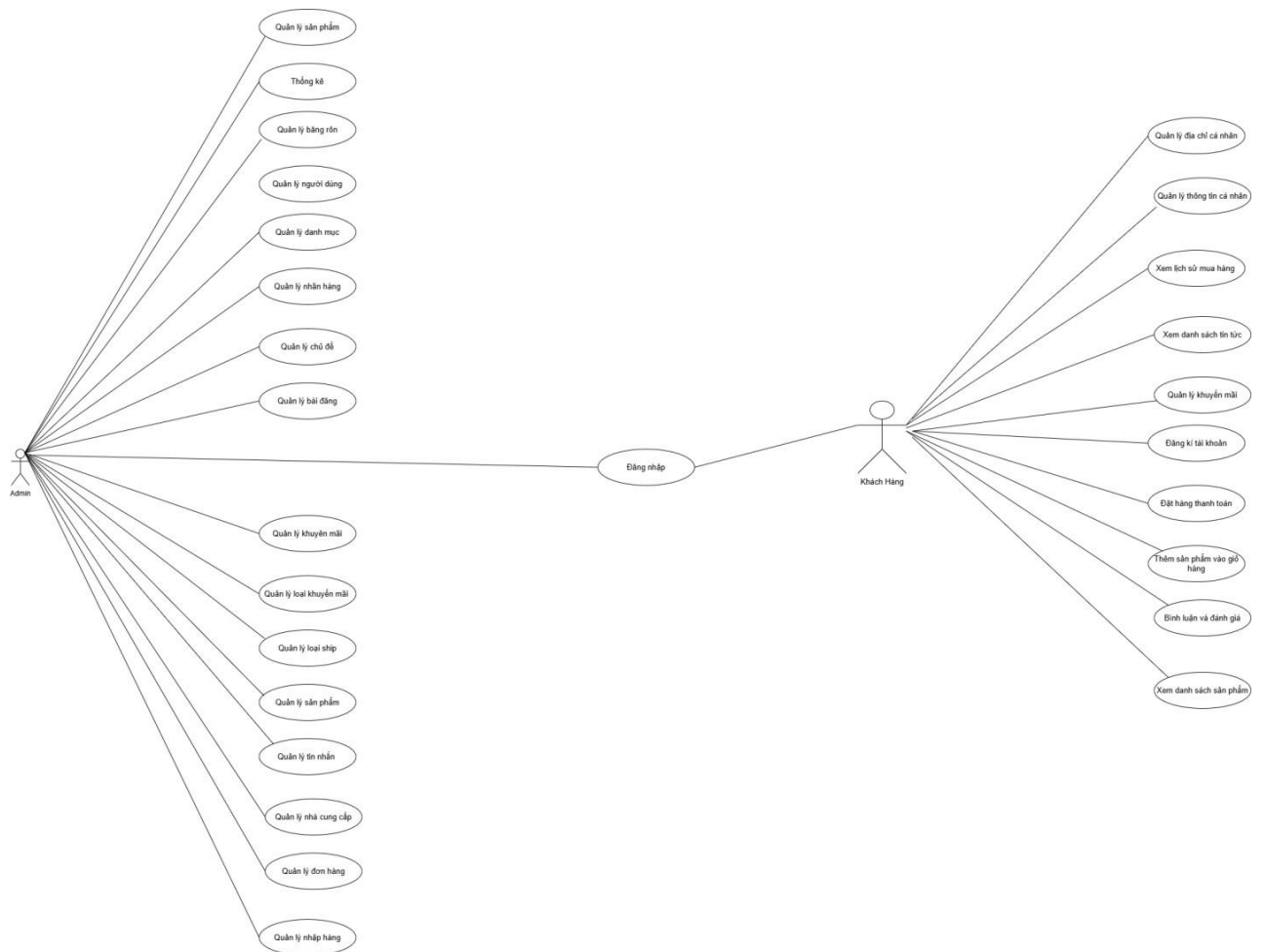
3.1.3. Các yêu cầu phi chức năng về hệ thống.

Bảng 3.1. Yêu cầu phi chức năng

STT	Tên yêu cầu	Mô tả yêu cầu
1	Giao diện	Giao diện hệ thống phải dễ sử dụng, trực quan, thân thiện với người dùng.
2	Tốc độ xử lý	Hệ thống phải xử lý nhanh chóng và chính xác. Dung lượng website vừa phải, tốc độ truy xuất nhanh.
3	Bảo mật	Tính bảo mật và độ an toàn cao.
4	Tương thích	Tương thích với đa phần các trình duyệt web hiện tại.

3.2. Mô hình UseCase.

3.2.1. Mô hình UseCase tổng quan.



Hình 3.1. Sơ đồ Usecase tổng quát

3.2.2. Mô hình UseCase phân rã.

- Khách hàng:



Hình 3.2. Sơ đồ User case khách hàng

- Người quản trị:



Hình 3.3.Sơ đồ Use care Admin

Danh sách các usecase:

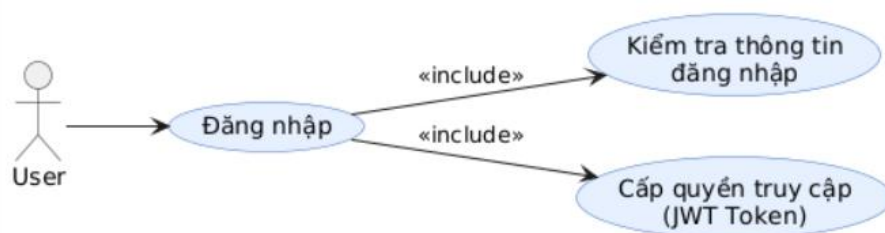
Bảng 3.2. Danh sách usecase

ID	Tên Usecase	Actor Sử Dụng
1	Xem danh sách sản phẩm	Khách hàng ghé thăm, khách hàng thành viên, nhân viên, người quản trị
2	Xem thông tin sản phẩm	Khách hàng ghé thăm, nhân viên, người quản trị
3	Đăng ký	Khách hàng ghé thăm, khách hàng thành viên, nhân viên, người quản trị
4	Tìm kiếm	Khách hàng ghé thăm, khách hàng thành viên, nhân viên, người quản trị
5	Xem tin tức, khuyến mãi	Khách hàng ghé thăm, khách hàng thành viên, nhân viên, người quản trị

6	Đưa sản phẩm vào giỏ hàng	Khách hàng thành viên,
7	Đăng nhập	Khách hàng thành viên, nhân viên, người quản trị
8	Đăng xuất	Khách hàng thành viên, nhân viên, người quản trị
9	Đánh giá	Khách hàng thành viên, nhân viên, người quản trị
10	Thông tin tài khoản	Khách hàng thành viên, nhân viên, người quản trị
11	Đặt mua, thanh toán	Khách hàng thành viên
12	Quản lý đơn hàng	Người quản trị
13	Quản lý danh mục	Người quản trị
14	Quản lý nhà cung cấp	Người quản trị
15	Quản lý nhập hàng	Người quản trị
16	Quản lý bài đăng, chủ đề	Người quản trị
17	Quản lý sản phẩm	Người quản trị
18	Quản lý khách hàng	Người quản trị
19	Quản lý bài viết	Người quản trị
20	Quản lý bình luận	Người quản trị
21	Thống kê, báo cáo	Người quản trị

3.2.3. Biểu đồ UseCase.

3.2.3.1. Biểu đồ usecase đăng nhập.



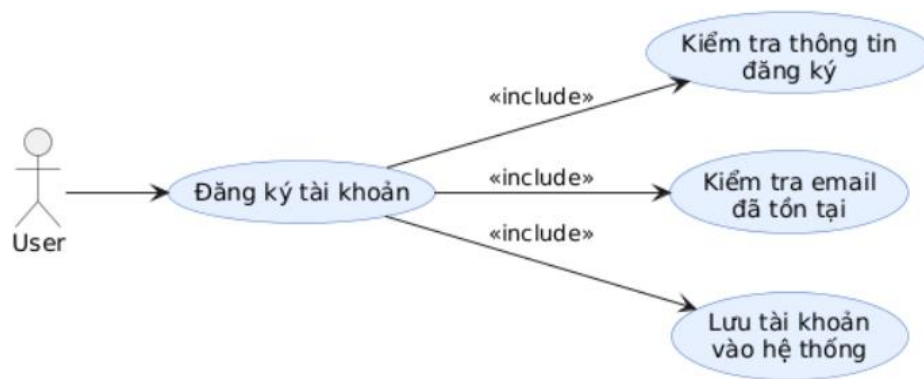
Hình 3.4. Usecase đăng nhập

Bảng 3.3. Đặc tả usecase đăng nhập

Tác nhân	User (Khách hàng), Admin
Mô tả	Use case cho phép người dùng nhập thông tin tài khoản để đăng nhập vào hệ thống. Nếu đăng nhập thành công, hệ thống cấp quyền truy cập dưới dạng JWT Token và cho phép người dùng sử dụng những chức năng phù hợp với vai trò của mình.
Điều kiện trước	Người dùng đã có tài khoản hợp lệ trong hệ thống. Tài khoản chưa bị khóa. Hệ thống hoạt động bình thường.
Luồng sự kiện chính	Hệ thống hiển thị giao diện đăng nhập. Người dùng nhập email và mật khẩu. Người dùng nhấn nút Đăng nhập . Hệ thống kiểm tra thông tin đăng nhập (email, mật khẩu). Nếu thông tin hợp lệ: - Hệ thống sinh JWT Token chứa thông tin người dùng và vai trò. - JWT Token được gửi về phía người dùng. Người dùng đăng nhập thành công và được điều hướng đến giao diện phù hợp với vai trò.
Luồng sự kiện phụ	1. Email hoặc mật khẩu sai: →Hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại. 2. Tài khoản bị khóa: →Hệ thống từ chối đăng nhập và thông báo cho người dùng. 3. Token hết hạn hoặc không hợp lệ trong các lần truy cập sau:

	→ Hệ thống yêu cầu người dùng đăng nhập lại.
Điều kiện sau	Người dùng được cấp JWT Access Token hợp lệ. Người dùng có thể truy cập các tài nguyên và chức năng phù hợp với quyền hạn. Hệ thống ghi nhận trạng thái đăng nhập (qua Token).
Kết quả mong đợi	Người dùng đăng nhập thành công và truy cập được các chức năng theo role. Trường hợp thất bại, hệ thống thông báo lỗi rõ ràng.

3.2.3.2. Biểu đồ usecase đăng kí.



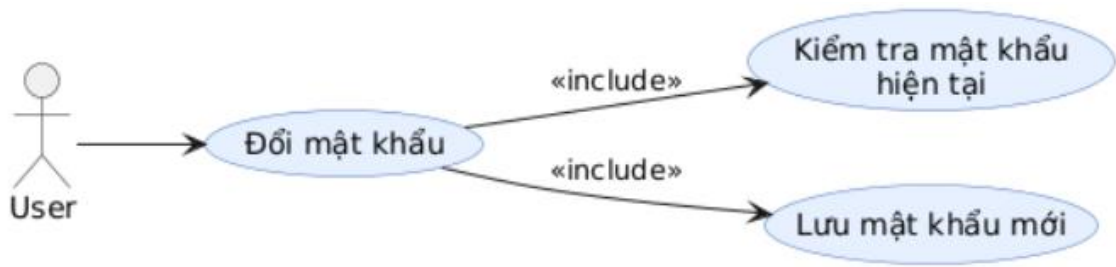
Hình 3.5. Usecase đăng kí

Bảng 3.4. Đặc tả usecase đăng kí

Tác nhân	User (Khách hàng)
Mô tả	Use case cho phép khách hàng tạo tài khoản mới trên hệ thống bằng cách cung cấp thông tin cá nhân cơ bản. Sau khi đăng ký thành công, người dùng có thể sử dụng tài khoản để đăng nhập và thực hiện các chức năng như đặt hàng, xem giỏ hàng, theo dõi đơn hàng,...
Điều kiện trước	Người dùng chưa có tài khoản trên hệ thống. Email đăng ký chưa tồn tại trong database.

	Hệ thống hoạt động bình thường.
Luồng sự kiện chính	1. Người dùng truy cập giao diện Đăng ký. 2. Người dùng nhập thông tin đăng ký: họ tên, email, mật khẩu, số điện thoại (nếu có). 3. Người dùng nhấn nút Đăng ký. 4. Hệ thống kiểm tra thông tin nhập vào (đủ dữ liệu, định dạng hợp lệ). 5. Hệ thống kiểm tra xem email đã tồn tại chưa. 6. Nếu hợp lệ, hệ thống lưu tài khoản mới vào database. 7. Hệ thống thông báo Đăng ký thành công và điều hướng người dùng sang trang đăng nhập.
Luồng sự kiện phụ	1. Email đã tồn tại: → Hệ thống báo lỗi “Email đã được sử dụng”, yêu cầu nhập email khác. 2. Mật khẩu không hợp lệ (ngắn quá, không đúng định dạng...): → Hệ thống yêu cầu nhập lại mật khẩu. 3. Người dùng bỏ trống dữ liệu: → Hiện thị thông báo lỗi và yêu cầu nhập lại. 4. Lỗi hệ thống khi lưu dữ liệu: → Hiện thị “Đăng ký thất bại, vui lòng thử lại”.
Điều kiện sau	Một tài khoản mới được tạo trong hệ thống. Người dùng có thể đăng nhập bằng tài khoản mới.
Kết quả mong đợi	Hệ thống tạo được tài khoản mới. Người dùng nhận thông báo đăng ký thành công.

3.2.3.3. Biểu đồ usecase đổi mật khẩu.



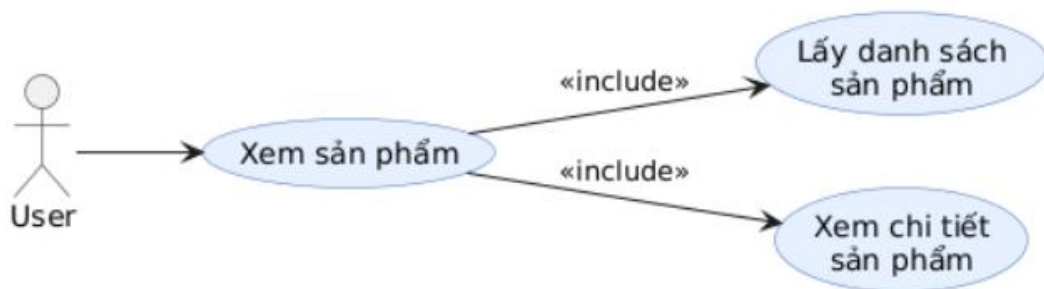
Hình 3.6. Ussecase đổi mật khẩu

Bảng 3.5. Đặc tả usecase đổi mật khẩu

Tác nhân	User (Khách hàng)
Mô tả	Use case cho phép người dùng đang đăng nhập thay đổi mật khẩu hiện tại sang mật khẩu mới để đảm bảo an toàn tài khoản. Người dùng phải nhập đúng mật khẩu cũ trước khi được phép đổi.
Điều kiện trước	Người dùng đã đăng nhập và có JWT Token hợp lệ. Người dùng phải cung cấp đúng mật khẩu cũ. Hệ thống hoạt động bình thường.
Luồng sự kiện chính	Người dùng mở giao diện đổi mật khẩu. Người dùng nhập mật khẩu hiện tại, mật khẩu mới và xác nhận mật khẩu mới. Người dùng nhấn Xác nhận đổi mật khẩu. Hệ thống kiểm tra tính hợp lệ của dữ liệu nhập vào: - Mật khẩu cũ có đúng không? - Mật khẩu mới có hợp lệ theo quy tắc không? Nếu hợp lệ, hệ thống cập nhật mật khẩu mới vào cơ sở dữ liệu. Hệ thống thông báo đổi mật khẩu thành công.
Luồng sự kiện phụ	1. Mật khẩu cũ không đúng:

	→ Hệ thống báo lỗi: “Mật khẩu hiện tại không chính xác. 2. Mật khẩu mới không hợp lệ (quá ngắn, không khớp xác nhận,...): → Hệ thống yêu cầu nhập lại. 3. Token hết hạn hoặc không hợp lệ: → Hệ thống yêu cầu người dùng đăng nhập lại rồi mới được đổi mật khẩu. 4. Lỗi hệ thống trong quá trình cập nhật: → Báo lỗi và yêu cầu thử lại sau.
Điều kiện sau	Mật khẩu mới được lưu thành công trong cơ sở dữ liệu. Người dùng sẽ sử dụng mật khẩu mới cho các lần đăng nhập sau. Trong một số hệ thống, token cũ có thể bị vô hiệu hóa và yêu cầu đăng nhập lại.
Kết quả mong đợi	Người dùng đổi mật khẩu thành công và được thông báo rõ ràng. Nếu thất bại, hệ thống đưa ra thông báo lỗi phù hợp.

3.2.3.4. Biểu đồ usecase xem sản phẩm.



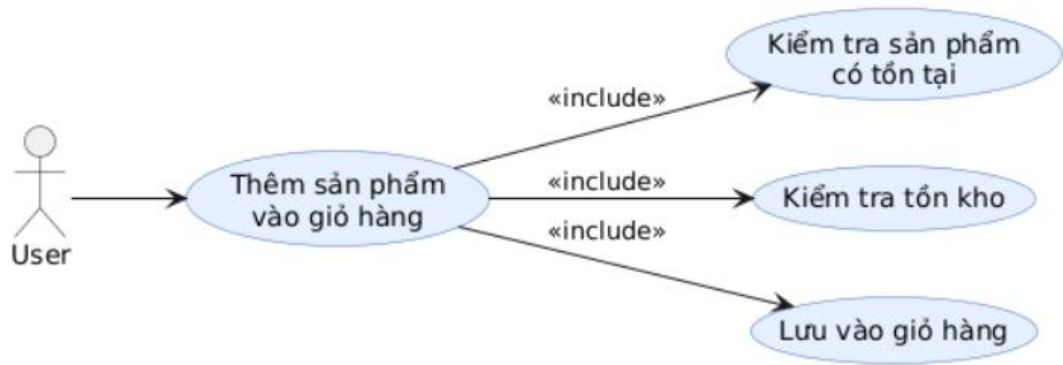
Hình 3.7. Usecase xem sản phẩm

Bảng 3.6. Đặc tả usecase xem sản phẩm

Tác nhân	User (Khách hàng)
-----------------	-------------------

Mô tả	Use case cho phép khách hàng xem danh sách sản phẩm đang bán trên hệ thống. Khách hàng có thể xem đầy đủ thông tin sản phẩm và xem chi tiết một sản phẩm cụ thể.
Điều kiện trước	Hệ thống có dữ liệu sản phẩm trong cơ sở dữ liệu. Kết nối hệ thống hoạt động bình thường.
Luồng sự kiện chính	1. Người dùng truy cập trang danh sách sản phẩm. 2. Hệ thống truy vấn và hiển thị danh sách sản phẩm (hình ảnh, giá, mô tả ngắn). 3. Người dùng chọn một sản phẩm để xem chi tiết. 4. Hệ thống hiển thị thông tin chi tiết của sản phẩm được chọn.
Luồng sự kiện phụ	1. Không có sản phẩm trong hệ thống: → Hệ thống hiển thị thông báo: “Hiện chưa có sản phẩm nào.” 2. Người dùng nhập từ khóa tìm kiếm (nếu có chức năng tìm kiếm): → Hệ thống lọc và hiển thị danh sách sản phẩm phù hợp. 3. Mất kết nối hoặc lỗi truy vấn: → Hệ thống báo lỗi và đề nghị thử lại.
Điều kiện sau	Danh sách sản phẩm đã được hiển thị cho người dùng. Người dùng có thể tiếp tục chọn sản phẩm khác hoặc xem chi tiết sản phẩm.
Kết quả mong đợi	Người dùng xem được danh sách sản phẩm và thông tin chi tiết khi chọn sản phẩm.

3.2.3.5. Biểu đồ usecase thêm sản phẩm vào giỏ hàng.



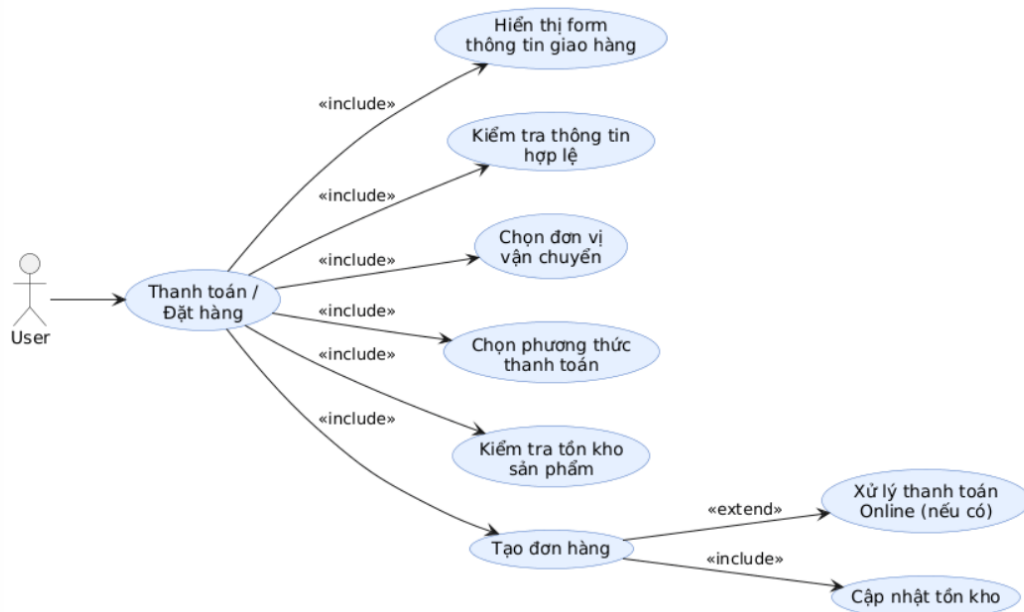
Hình 3.8. Usecase thêm sản phẩm vào giỏ hàng

Bảng 3.7. Đặc tả usecase thêm sản phẩm vào giỏ hàng

Tác nhân	User (Khách hàng)
Mô tả	Use case cho phép người dùng chọn sản phẩm mong muốn và thêm vào giỏ hàng để chuẩn bị cho quá trình đặt hàng.
Điều kiện trước	Người dùng đang xem danh sách sản phẩm hoặc chi tiết sản phẩm. Sản phẩm còn hàng (stock > 0). Hệ thống hoạt động bình thường.
Luồng sự kiện chính	1. Người dùng chọn một sản phẩm từ danh sách hoặc trang chi tiết sản phẩm. 2. Người dùng nhấn nút “Thêm vào giỏ hàng” . 3. Hệ thống kiểm tra: • Sản phẩm có tồn tại không. • Sản phẩm còn đủ số lượng không. 4. Hệ thống thêm sản phẩm vào giỏ hàng (tạo mới hoặc tăng số lượng nếu đã có trong giỏ). 5. Hệ thống hiển thị thông báo: “Đã thêm vào giỏ hàng”.
Luồng sự kiện phụ	1. Sản phẩm hết hàng: → Hệ thống hiển thị thông báo “Sản phẩm đã hết hàng”.

	2. Người dùng chưa đăng nhập (nếu yêu cầu): → Hệ thống yêu cầu đăng nhập để sử dụng giỏ hàng. 3. Lỗi hệ thống khi thêm vào giỏ: → Hiện thị thông báo lỗi và yêu cầu thử lại.
Điều kiện sau	Giỏ hàng đã được cập nhật với sản phẩm người dùng chọn. Người dùng có thể tiếp tục mua sắm hoặc truy cập trang giỏ hàng.
Kết quả mong đợi	Sản phẩm được thêm thành công vào giỏ hàng và phản ánh đúng số lượng trong giỏ.

3.2.3.6. Biểu đồ usecase thanh toán.



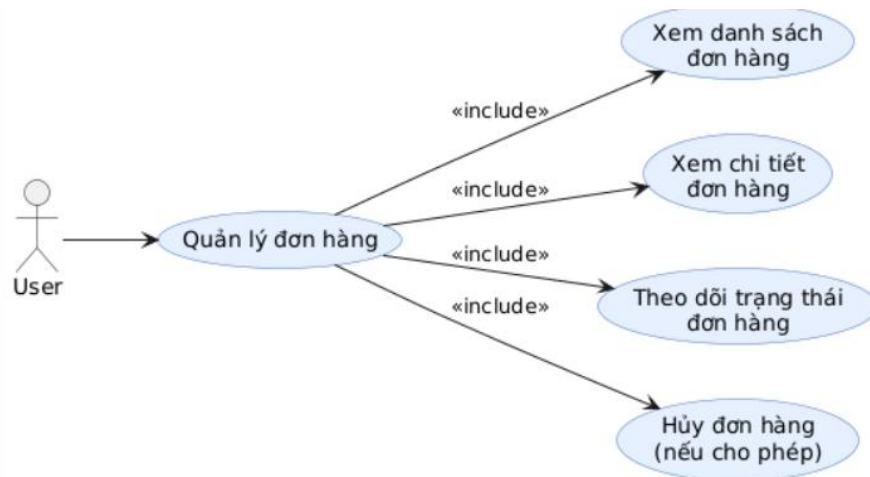
Hình 3.9. Usecase thanh toán

Bảng 3.8. Bảng đặc tả usecase thanh toán

Tác nhân	User (Khách hàng)
Mô tả	Use case cho phép người dùng xác nhận đơn hàng từ giỏ hàng, nhập thông tin giao hàng và hoàn tất quá trình mua hàng.

Điều kiện trước	Người dùng đã có sản phẩm trong giỏ hàng.. Người dùng đã đăng nhập (nếu hệ thống yêu cầu). Sản phẩm trong giỏ vẫn còn tồn kho.
Luồng sự kiện chính	1. Người dùng mở trang giỏ hàng và chọn Thanh toán . 2. Hệ thống hiển thị form thông tin giao hàng (tên, số điện thoại, địa chỉ). 3. Người dùng nhập thông tin giao hàng. 4. Hệ thống kiểm tra tính hợp lệ của thông tin. 5. Hệ thống kiểm tra tồn kho cho từng sản phẩm trong giỏ. 6. Nếu hợp lệ, hệ thống tạo đơn hàng (Order) . 7. Hệ thống trừ số lượng tồn kho của sản phẩm. 8. Hệ thống thông báo đặt hàng thành công.
Luồng sự kiện phụ	1. Giỏ hàng trống: → Hệ thống thông báo “Giỏ hàng của bạn đang trống.” 2. Sản phẩm hết hàng hoặc không đủ số lượng: → Hệ thống báo lỗi và yêu cầu cập nhật giỏ hàng. 3. Thông tin giao hàng không hợp lệ: → Yêu cầu người dùng nhập lại. 4. Lỗi hệ thống khi tạo đơn hàng: → Hệ thống báo “Đặt hàng thất bại, vui lòng thử lại.”
Điều kiện sau	Một đơn hàng mới được tạo trong hệ thống. Tồn kho sản phẩm được cập nhật. Giỏ hàng của người dùng được làm mới (xoá sản phẩm đã thanh toán).
Kết quả mong đợi	Người dùng thanh toán thành công và thấy thông báo "Đặt hàng thành công".

3.2.3.7. Biểu đồ usecase quản lý đơn hàng (User)



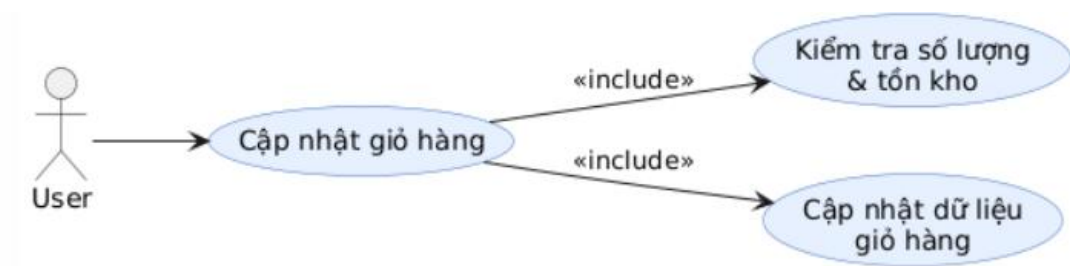
Hình 3.10. Usecase quản lý đơn hàng của user

Bảng 3.9. Đặc tả usecase quản lý đơn hàng của user

Tác nhân	User (Khách hàng)
Mô tả	Use Case cho phép khách hàng xem danh sách đơn hàng đã mua, xem chi tiết từng đơn hàng, kiểm tra trạng thái và thực hiện hủy đơn nếu đơn hàng chưa được xử lý.
Điều kiện trước	Người dùng đã đăng nhập hệ thống. Người dùng đã có ít nhất một đơn hàng.
Luồng sự kiện chính	1. User mở trang “Đơn hàng của tôi”. 2. Hệ thống hiển thị danh sách đơn hàng của User. 3. User chọn một đơn hàng để xem chi tiết. 4. User có thể xem trạng thái, xem sản phẩm trong đơn. 5. Nếu đơn hàng đang ở trạng thái “Chờ xác nhận”, User có thể chọn hủy đơn. 6. Hệ thống cập nhật trạng thái đơn hàng và thông báo thành công.
Luồng sự kiện phụ	Nếu đơn hàng đã được xử lý (Đang giao / Đã giao) → không cho phép hủy. Nếu không có đơn hàng → hiển thị “Bạn chưa có đơn hàng nào”.

	Nếu user chưa đăng nhập → yêu cầu đăng nhập.
Điều kiện sau	Danh sách đơn hàng hiển thị đúng thông tin. Nếu hủy đơn → trạng thái đơn hàng được cập nhật.
Kết quả mong đợi	Người dùng theo dõi được tình trạng đơn hàng của mình. Người dùng có thể hủy đơn dễ dàng khi cần.

3.2.3.8. Biểu đồ usecase cập nhật giỏ hàng.



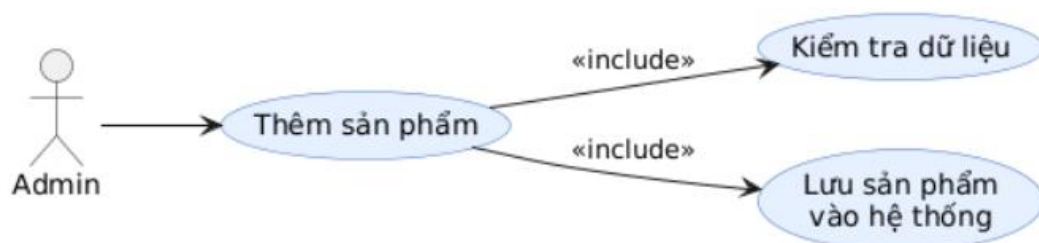
Hình 3.11. Usecase cập nhật giỏ hàng

Bảng 3.10. Đặc tả usecase cập nhật giỏ hàng

Tác nhân	User (Khách hàng)
Mô tả	Use case cho phép người dùng thay đổi số lượng sản phẩm trong giỏ hàng hoặc xóa sản phẩm khỏi giỏ trước khi tiến hành thanh toán.
Điều kiện trước	Người dùng đã có ít nhất 1 sản phẩm trong giỏ hàng.. Hệ thống phải xác định được giỏ hàng của người dùng (qua session hoặc token). Các sản phẩm trong giỏ vẫn còn tồn kho.
Luồng sự kiện chính	1. Người dùng truy cập vào trang giỏ hàng. 2. Người dùng thay đổi số lượng sản phẩm (tăng/giảm) hoặc xóa sản phẩm khỏi giỏ. 3. Hệ thống kiểm tra: • Số lượng mới có hợp lệ không (≥ 1).

	• Có đủ tồn kho hay không. 4. Hệ thống cập nhật giỏ hàng trong cơ sở dữ liệu hoặc session. 5. Hệ thống tính lại tổng tiền đơn hàng. 6. Hệ thống hiển thị giỏ hàng đã cập nhật cho người dùng.
Luồng sự kiện phụ	1. Số lượng thay đổi vượt quá tồn kho: → Hệ thống báo “Không đủ số lượng sản phẩm”. 2. Số lượng nhập không hợp lệ (0 hoặc âm): → Hệ thống báo lỗi và yêu cầu nhập lại. 3. Người dùng xóa sản phẩm khỏi giỏ: → Hệ thống cập nhật giỏ và tính lại tổng tiền. 4. Lỗi hệ thống khi cập nhật giỏ: → Thông báo lỗi và yêu cầu thử lại sau.
Điều kiện sau	Giỏ hàng của người dùng được cập nhật đúng với thao tác thực hiện. Tổng tiền được tính lại đúng với số lượng mới.
Kết quả mong đợi	Giỏ hàng phản ánh chính xác các thay đổi người dùng thực hiện.

3.2.3.9. Biểu đồ usecase thêm sản phẩm.

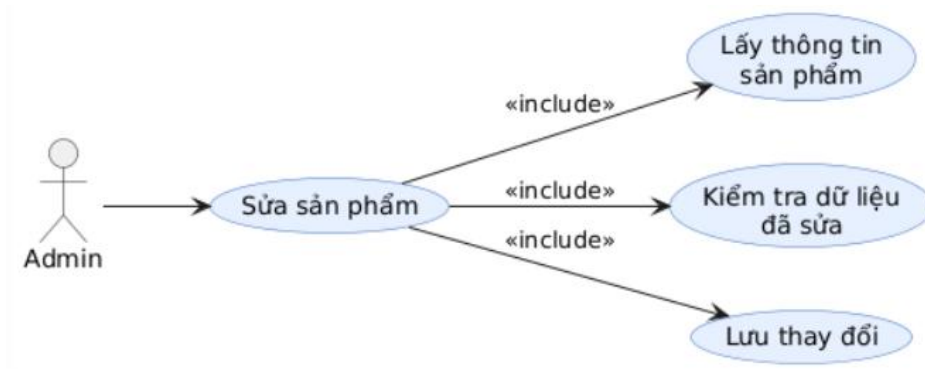


Hình 3.12. Usecase thêm sản phẩm

Bảng 3.11. Đặc tả usecase thêm sản phẩm

Tác nhân	Admin
Mô tả	Admin thêm sản phẩm mới vào hệ thống với thông tin như tên, giá, mô tả, danh mục, số lượng tồn kho.
Điều kiện trước	Admin đã đăng nhập. Admin có quyền thêm sản phẩm.
Luồng sự kiện chính	1. Admin mở trang thêm sản phẩm. 2. Admin nhập thông tin sản phẩm. 3. Hệ thống kiểm tra tính hợp lệ dữ liệu. 4. Hệ thống lưu sản phẩm mới vào database. 5. Hệ thống thông báo thêm thành công.
Luồng sự kiện phụ	Dữ liệu không hợp lệ → báo lỗi. Lỗi lưu database → thông báo lỗi.
Điều kiện sau	Sản phẩm mới được tạo trong hệ thống.
Kết quả mong đợi	Sản phẩm mới được thêm thành công vào hệ thống. Tất cả thông tin nhập vào được lưu đúng và hiển thị trong danh sách sản phẩm. Admin nhận được thông báo “Thêm sản phẩm thành công

3.2.3.10. Biểu đồ usecase sửa sản phẩm.

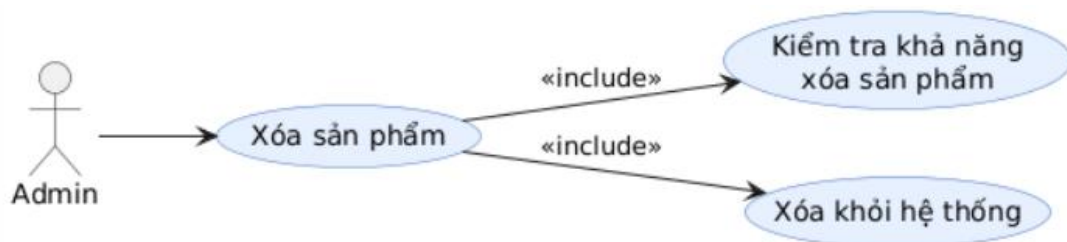


Hình 3.13. Usecase sửa sản phẩm

Bảng 3.12. Đặc tả usecase sửa sản phẩm

Tác nhân	Admin
Mô tả	Admin chỉnh sửa thông tin sản phẩm đã có trong hệ thống.
Điều kiện trước	Admin đã đăng nhập. Sản phẩm tồn tại.
Luồng sự kiện chính	1. Admin chọn một sản phẩm để sửa. 2. Hệ thống hiển thị thông tin chi tiết sản phẩm. 3. Admin cập nhật thông tin. 4. Hệ thống kiểm tra dữ liệu mới. 5. Hệ thống lưu thông tin đã sửa. 6. Thông báo sửa thành công.
Luồng sự kiện phụ	Sản phẩm không tồn tại → báo lỗi. Dữ liệu không hợp lệ → yêu cầu nhập lại.
Điều kiện sau	Thông tin sản phẩm được cập nhật.
Kết quả mong đợi	Thông tin sản phẩm được cập nhật đúng theo yêu cầu. Dữ liệu hiển thị mới phản ánh thay đổi ngay lập tức. Admin nhận được thông báo “Cập nhật sản phẩm thành công”.

3.2.3.11. Biểu đồ usecase xóa sản phẩm.

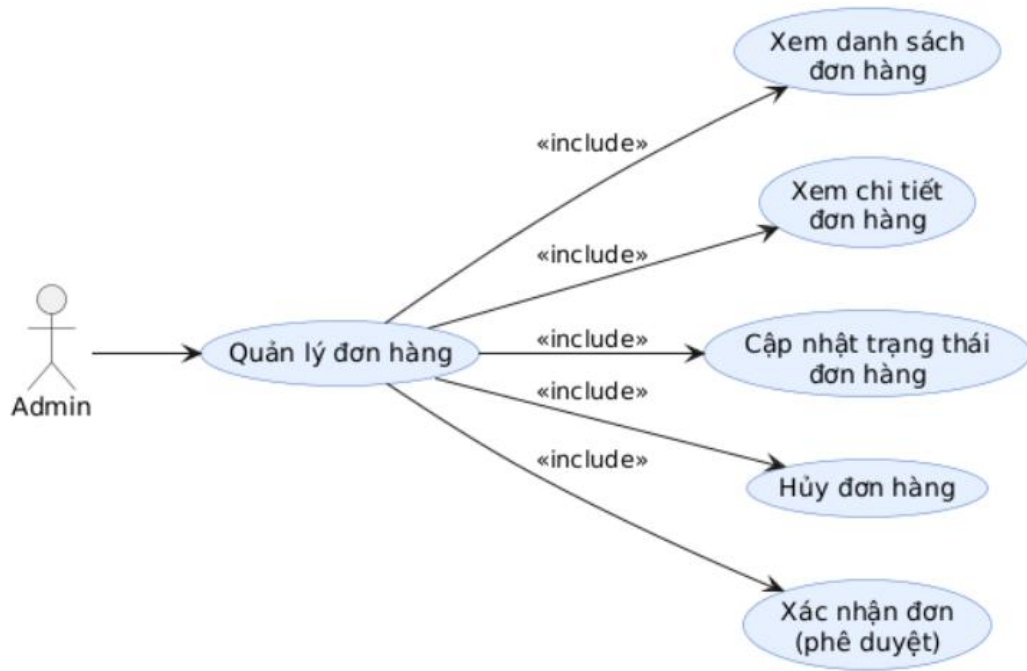


Hình 3.14. Usecase xóa sản phẩm

Bảng 3.13. Đặc tả usecase xóa sản phẩm

Tác nhân	Admin
Mô tả	Admin chỉnh sửa thông tin sản phẩm đã có trong hệ thống.
Điều kiện trước	Admin đã đăng nhập. Sản phẩm tồn tại.
Luồng sự kiện chính	1. Admin chọn sản phẩm muốn xóa. 2. Hệ thống xác nhận yêu cầu xóa. 3. Hệ thống kiểm tra sản phẩm có thể xóa hay không. 4. Hệ thống xóa sản phẩm khỏi database. 5. Hệ thống lưu thông tin đã sửa. 6. Hệ thống thông báo xóa thành công.
Luồng sự kiện phụ	Sản phẩm không tồn tại → báo lỗi. Sản phẩm có ràng buộc đơn hàng → không cho xóa.
Điều kiện sau	Sản phẩm bị loại khỏi hệ thống.
Kết quả mong đợi	Sản phẩm được xóa khỏi danh sách sản phẩm trong hệ thống. Danh sách sản phẩm hiển thị lại không còn sản phẩm đã xóa. Admin nhận được thông báo “Xóa sản phẩm thành công”.

3.2.3.12. Biểu đồ usecase quản lý đơn hàng (Admin).



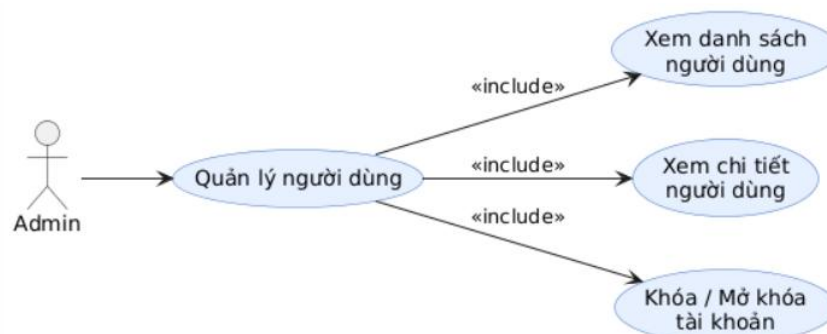
Hình 3.15. Usecase quản lý đơn hàng của admin

Bảng 3.14. Đặc tả usecase quản lý đơn hàng

Tác nhân	Admin
Mô tả	Use case cho phép Admin xem danh sách đơn hàng, xem chi tiết từng đơn hàng và cập nhật trạng thái đơn hàng trong quá trình xử lý như: Chờ xác nhận → Đang giao → Hoàn thành hoặc Hủy đơn.
Điều kiện trước	Admin đã đăng nhập và có quyền quản lý đơn hàng. Có đơn hàng trong hệ thống.
Luồng sự kiện chính	1. Admin truy cập trang quản lý đơn hàng. 2. Hệ thống hiển thị danh sách đơn hàng (mã đơn hàng, khách hàng, tổng tiền, trạng thái,...). 3. Admin chọn một đơn hàng để xem chi tiết. 4. Hệ thống hiển thị đầy đủ thông tin chi tiết đơn hàng (sản phẩm, số lượng, giá, địa chỉ giao hàng...). 5. Admin có thể cập nhật trạng thái đơn hàng: • Chờ xác nhận

	• Đang giao • Hoàn thành • Hủy đơn 6. Hệ thống lưu trạng thái mới vào database. 7. Hệ thống thông báo cập nhật thành công.
Luồng sự kiện phụ	Không có đơn hàng trong hệ thống: → Hiển thị thông báo “Không có đơn hàng nào.” Admin cập nhật trạng thái không hợp lệ: → Hệ thống báo lỗi. Lỗi khi lưu dữ liệu: → Thông báo lỗi và yêu cầu thử lại.
Điều kiện sau	Danh sách đơn hàng phản ánh chính xác trạng thái cập nhật. Thông tin đơn hàng được lưu lại đầy đủ trong hệ thống.
Kết quả mong đợi	Admin xem được tất cả đơn hàng và trạng thái. Admin cập nhật thành công trạng thái đơn hàng. Dữ liệu đơn hàng luôn được đồng bộ và chính xác.

3.2.3.13. Biểu đồ usecase quản lý người dùng

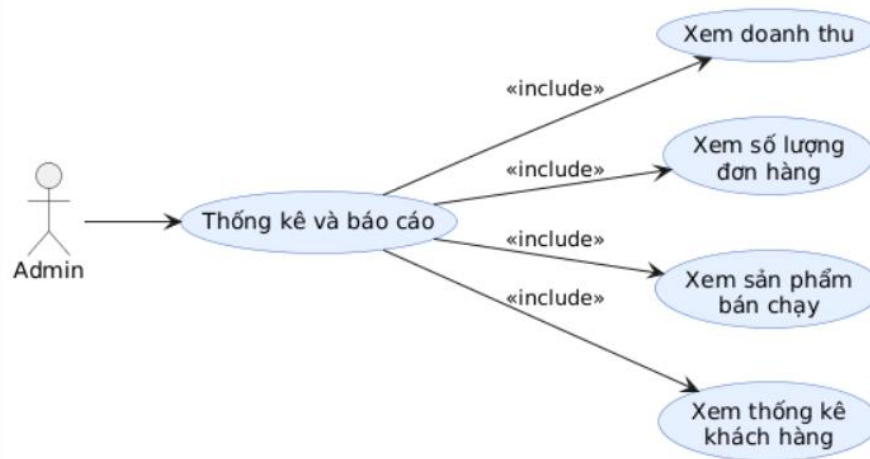


Hình 3.16. Usecase quản lý người dùng

Bảng 3.15. Đặc tả usecase quản lý người dùng

Tác nhân	Admin
Mô tả	Use case cho phép Admin xem danh sách người dùng trong hệ thống, xem thông tin chi tiết của từng người dùng và cập nhật trạng thái tài khoản (hoạt động / khóa).
Điều kiện trước	Admin đã đăng nhập vào hệ thống. Admin có quyền quản lý khách hàng. Hệ thống hoạt động bình thường.
Luồng sự kiện chính	1. Admin truy cập trang Quản lý khách hàng. 2. Hệ thống hiển thị danh sách tất cả người dùng (id, tên, email, trạng thái...). 3. Admin chọn một người dùng để xem chi tiết. 4. Hệ thống hiển thị: thông tin tài khoản, lịch sử hoạt động (nếu có), trạng thái 5. Admin có thể thực hiện: • Khóa tài khoản • Mở khóa tài khoản 6. Hệ thống lưu thay đổi vào database. 7. Hệ thống thông báo cập nhật thành công.
Luồng sự kiện phụ	Người dùng không tồn tại → Báo lỗi Hệ thống lỗi khi lưu dữ liệu → Báo “Cập nhật thất bại, thử lại sau” Không đủ quyền (ví dụ nhân viên truy cập) → Hiển thị “Không có quyền truy cập”
Điều kiện sau	Trạng thái tài khoản người dùng được cập nhật thành công. Danh sách khách hàng hiển thị đúng trạng thái mới.
Kết quả mong đợi	Admin quản lý tài khoản khách hàng thuận lợi. Hệ thống phản ánh chính xác thông tin cập nhật.

3.2.3.14. Biểu đồ usecase thống kê



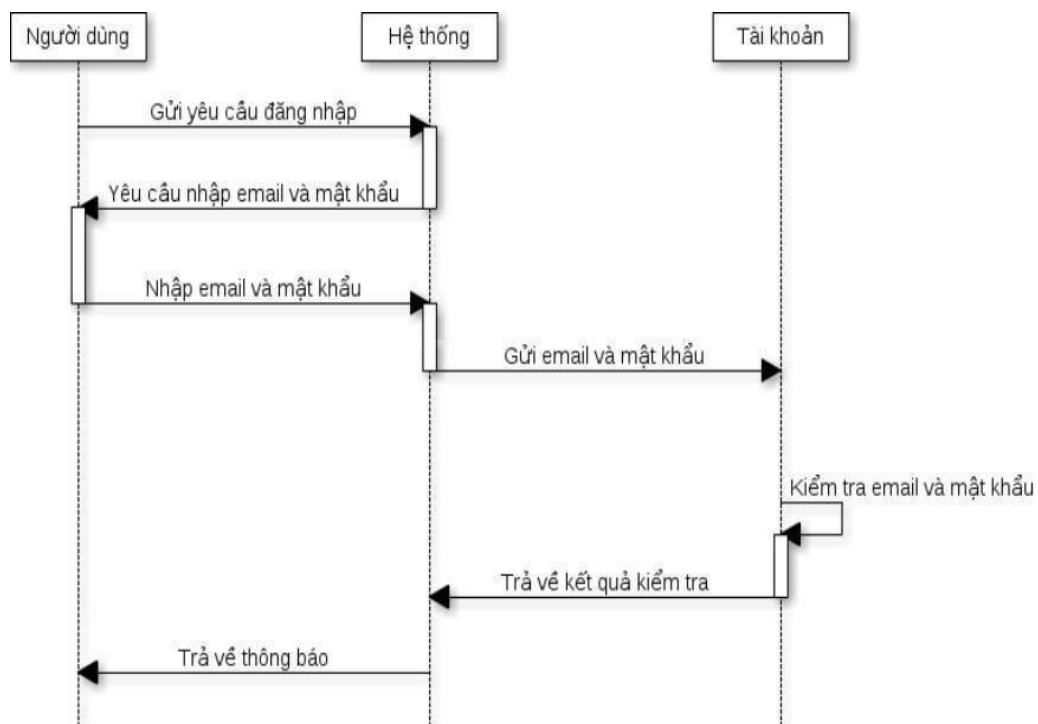
Hình 3.17. Usecase thống kê

Bảng 3.16. Đặc tả usecase thống kê

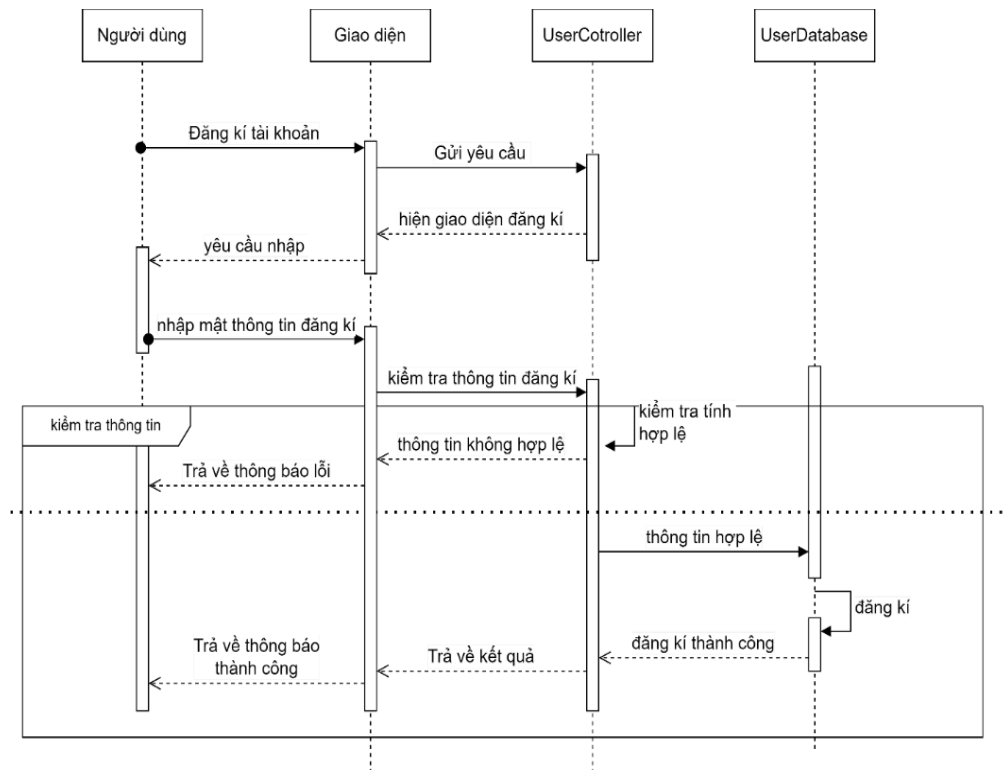
Tác nhân	Admin
Mô tả	Use case cho phép Admin xem các báo cáo thống kê liên quan đến hoạt động bán hàng bao gồm: • Doanh thu theo ngày / tháng / năm • Số lượng đơn hàng • Top sản phẩm bán chạy • Số lượng khách hàng mới • Tình trạng tồn kho Các thống kê này giúp Admin đánh giá hiệu quả kinh doanh và ra quyết định.
Điều kiện trước	Admin đã đăng nhập. Cơ sở dữ liệu có dữ liệu đơn hàng, sản phẩm, khách hàng... Hệ thống hoạt động bình thường.
Luồng sự kiện chính	1. Admin truy cập mục Thống kê – Báo cáo. 2. Hệ thống hiển thị bảng lựa chọn loại báo cáo: doanh thu, đơn hàng, sản phẩm, khách hàng,... 3. Admin chọn loại báo cáo muốn xem.

	4. Admin chọn khoảng thời gian (nếu cần). 5. Hệ thống truy vấn dữ liệu theo yêu cầu. 6. Hệ thống hiển thị biểu đồ hoặc bảng thống kê. 7. Admin có thể xuất báo cáo (PDF / Excel) nếu được hỗ trợ.
Luồng sự kiện phụ	Không có dữ liệu trong khoảng thời gian → hiển thị “Không có dữ liệu thống kê”. Hệ thống lỗi khi truy vấn → thông báo “Không thể tải dữ liệu”. Admin nhập khoảng thời gian không hợp lệ → yêu cầu nhập lại.
Điều kiện sau	Thống kê được hiển thị theo đúng yêu cầu. Dữ liệu được cập nhật theo thời gian hiện tại..
Kết quả mong đợi	Admin xem được báo cáo chính xác, trực quan. Báo cáo hỗ trợ quyết định kinh doanh..

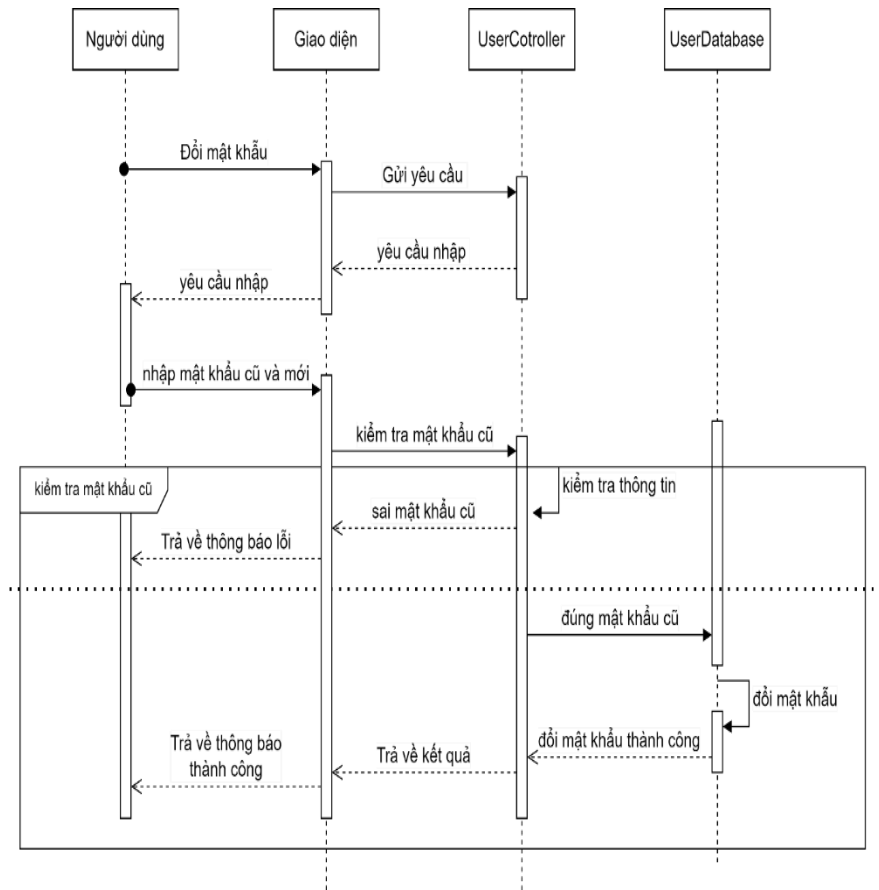
3.2.4. Biểu đồ tuần tự



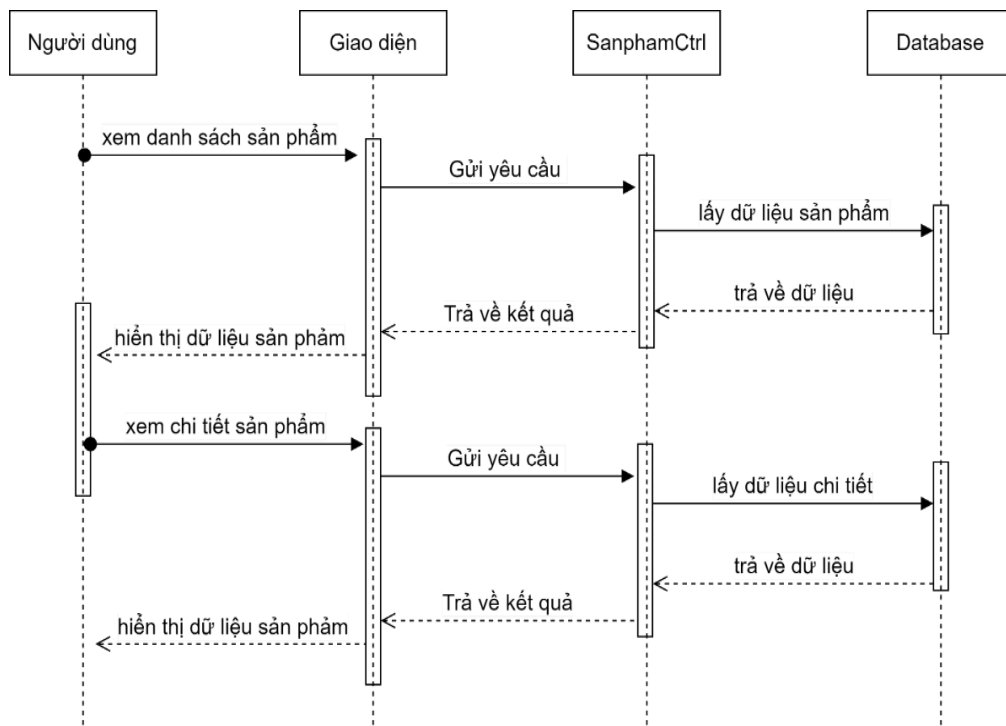
Hình 3.18. Sơ đồ tuần tự đăng nhập



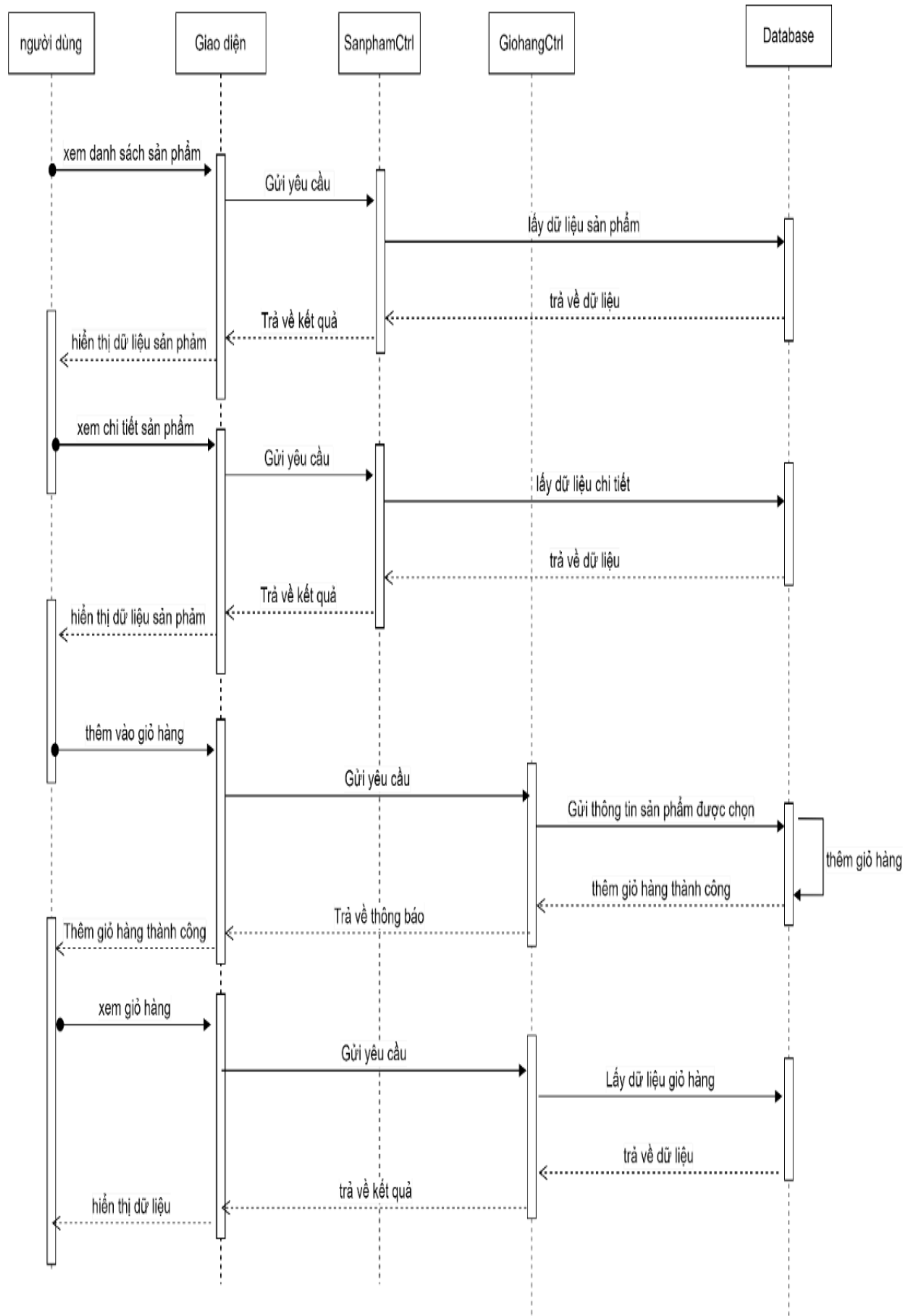
Hình 3.19. Sơ đồ tuần tự đăng kí



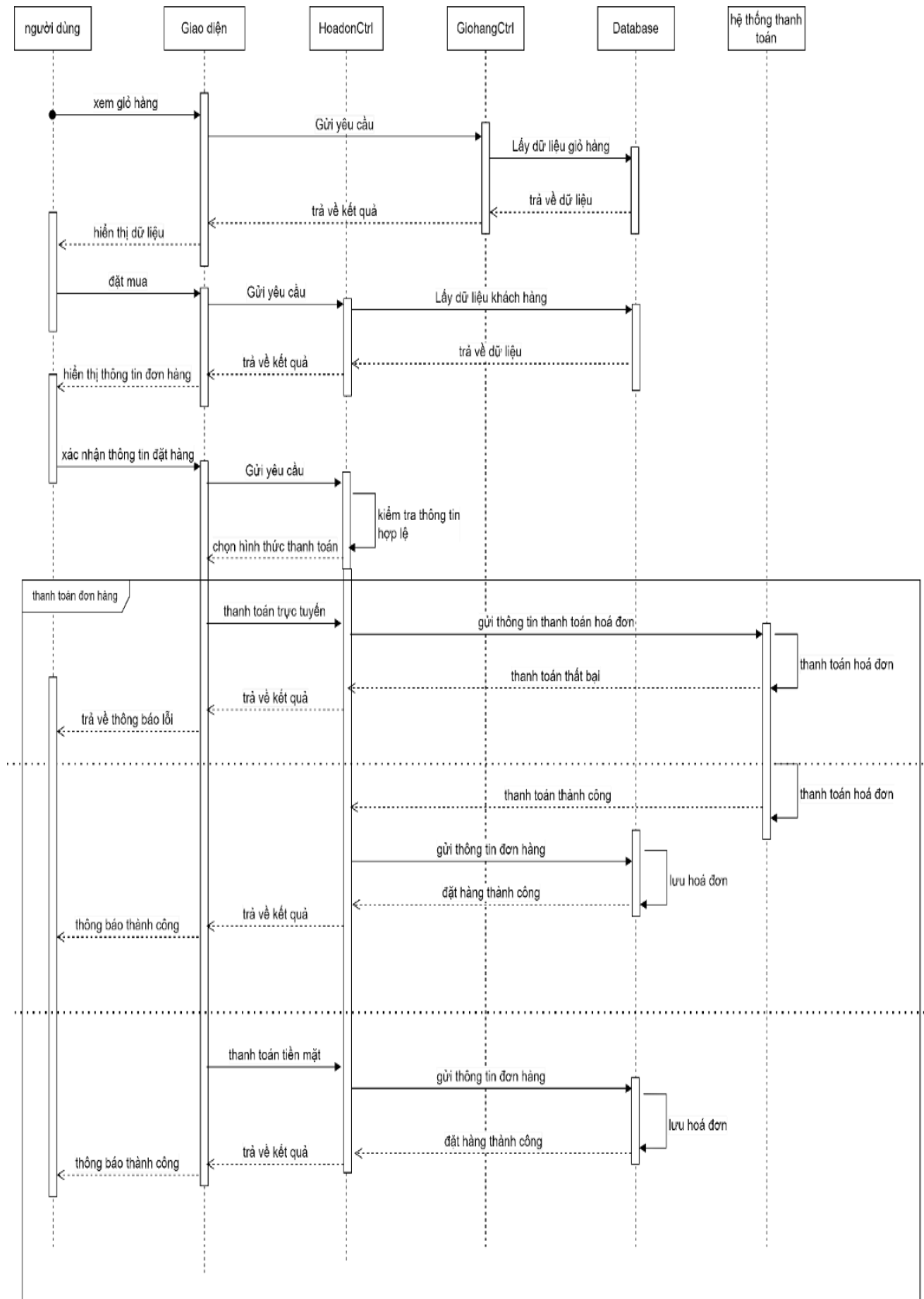
Hình 3.20. Sơ đồ tuần tự đổi mật khẩu



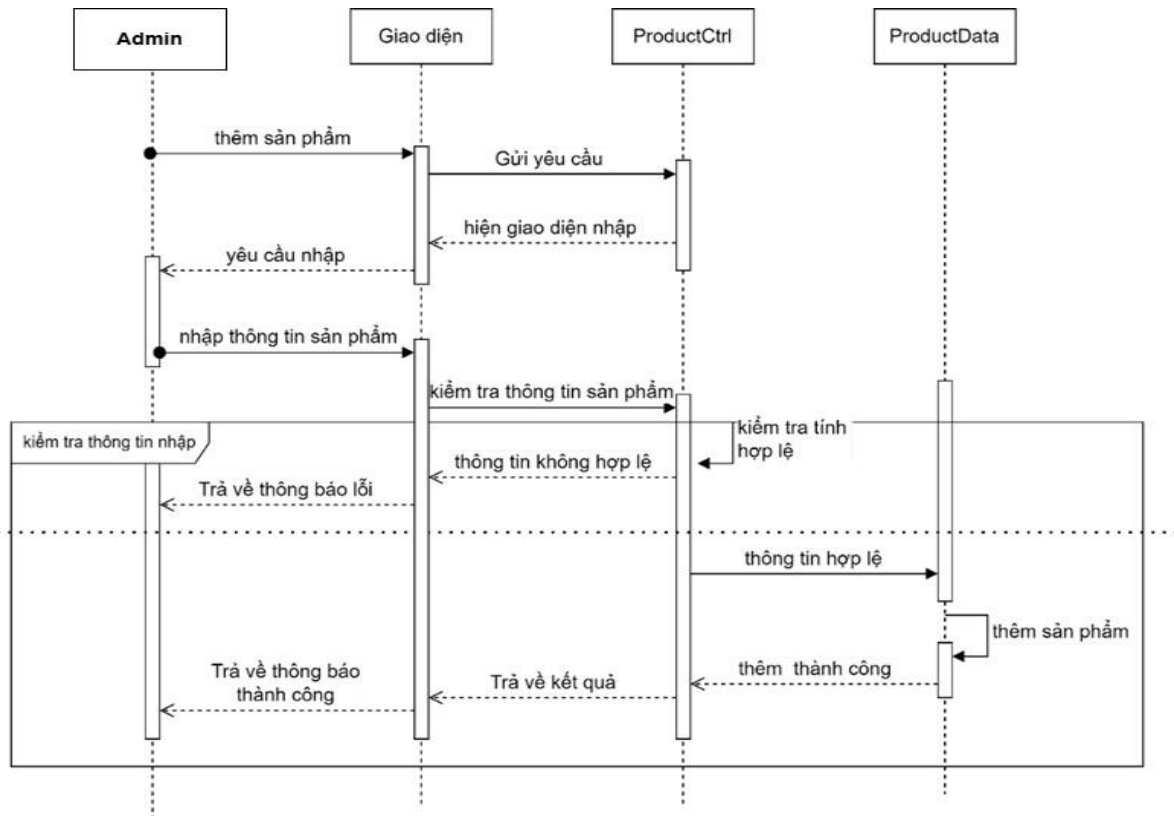
Hình 3.21. Sơ đồ tuần tự xem sản phẩm



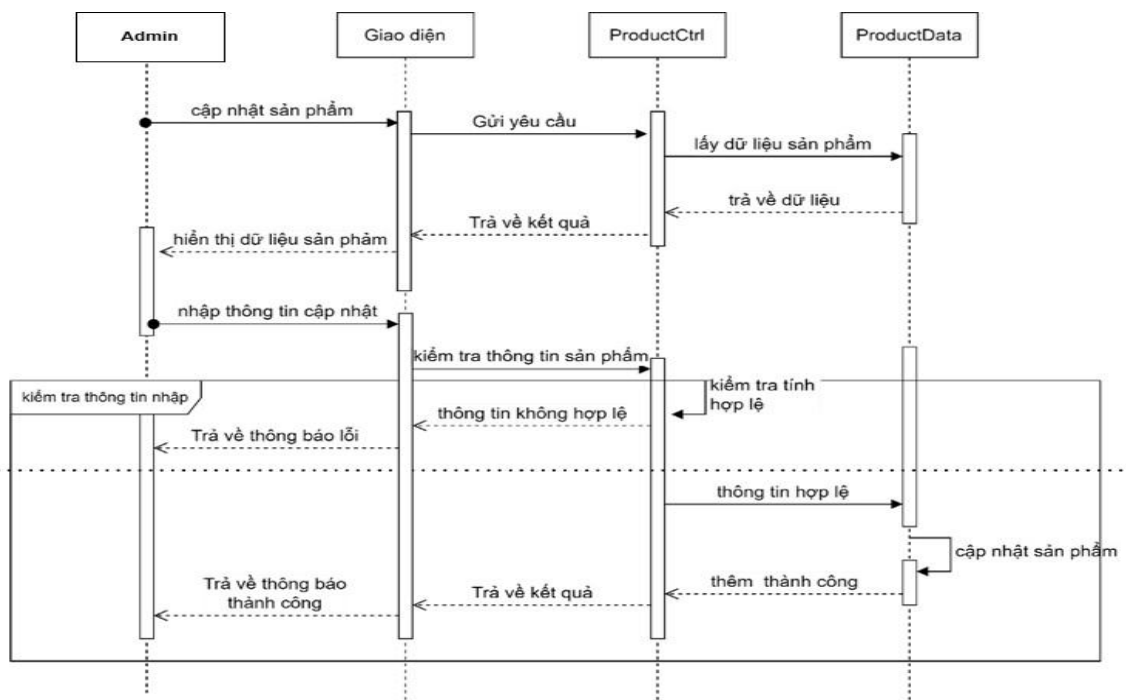
Hình 3.22. Sơ đồ tuần tự thêm giỏ hàng



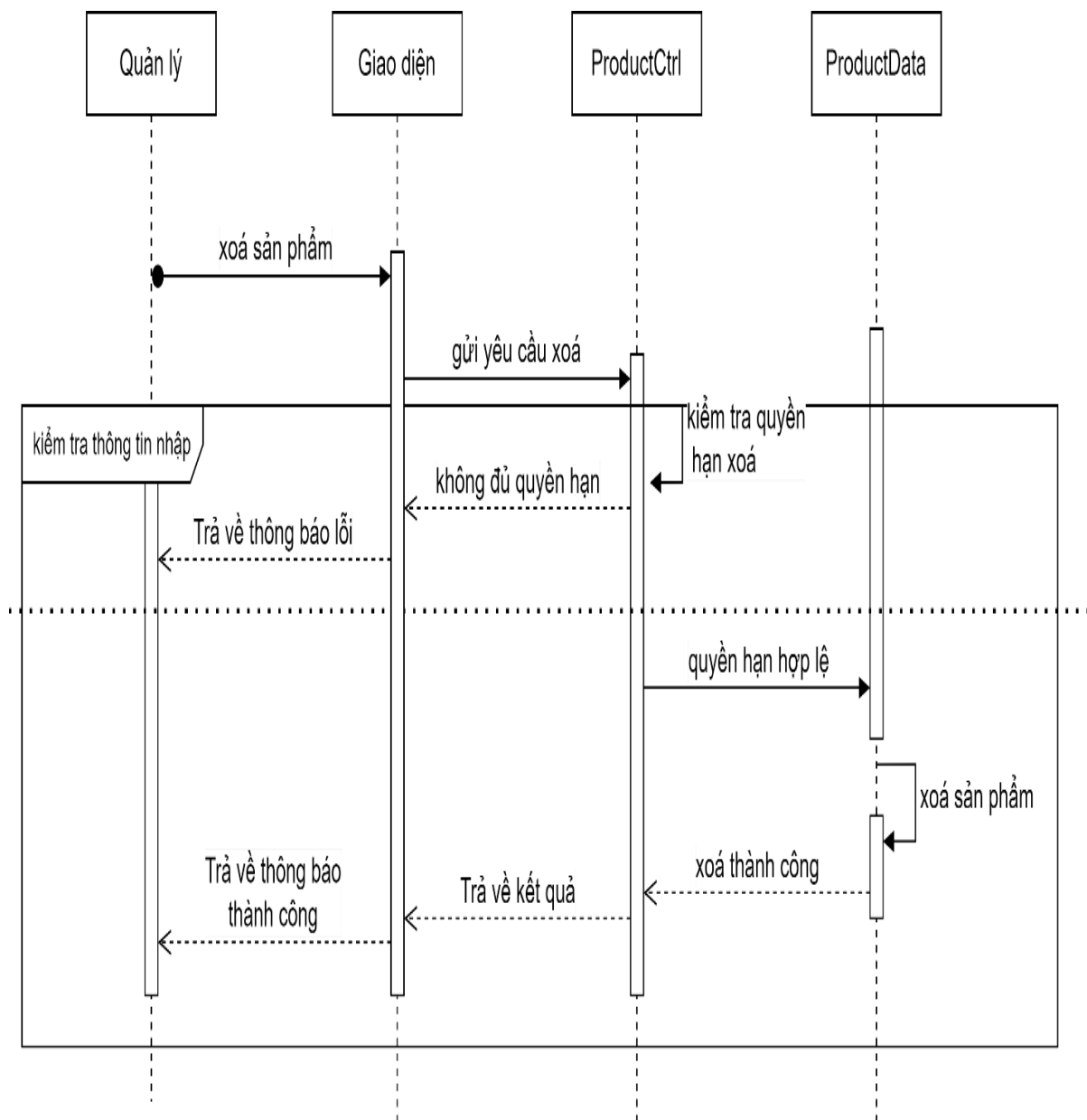
Hình 3.23. Sơ đồ tuần tự thanh toán



Hình 3.24. Sơ đồ tuần tự thêm sản phẩm



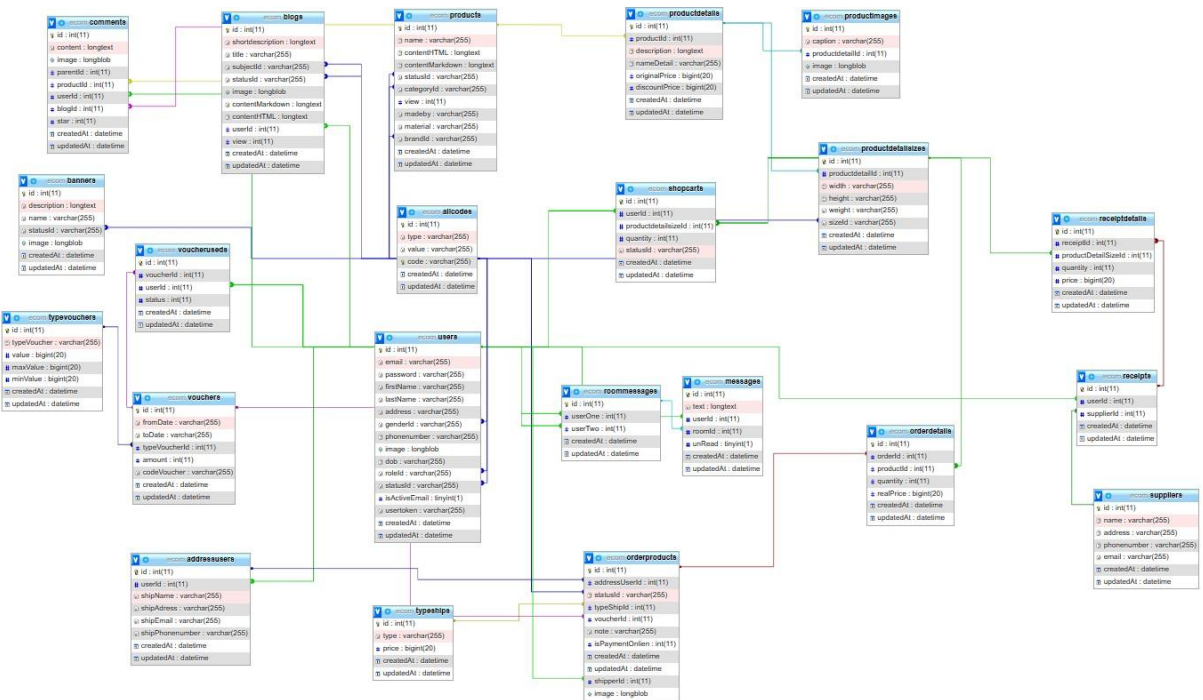
Hình 3.25. Sơ đồ tuần tự cập nhật sản phẩm



Hình 3.26. Sơ đồ tuần tự xóa sản phẩm

3.3. Thiết kế Cơ sở dữ liệu.

3.3.1. Mô hình ERD.



Hình 3.27. Sơ đồ cơ sở dữ liệu

Mô hình ERD (Entity Relationship Diagram) là một biểu đồ thể hiện mối quan hệ giữa các thực thể trong cơ sở dữ liệu. ERD giúp hình dung cấu trúc dữ liệu một cách trực quan, từ đó thiết kế cơ sở dữ liệu hiệu quả.

Các thực thể trong website bán hàng:

- **Users:** Lưu trữ thông tin về người dùng (khách hàng, quản trị viên), bao gồm: id, name, email, password, address, phone,...
- **Products:** Lưu trữ thông tin về sản phẩm, bao gồm: id, name, description, price, image, category_id,...
- **Orders:** Lưu trữ thông tin về đơn hàng, bao gồm: id, user_id, voucher_id, address_id, status,...
- **Order_details:** Lưu trữ thông tin chi tiết của từng đơn hàng, bao gồm: id, order_id, product_id, quantity, price,...
- **ProductDetails:** Lưu chi tiết thông tin về sản phẩm, bao gồm: id, product_id, contentMarkdown, contentHTML, description, createdAt, updateAt,...

- Biểu đồ quan hệ:**



3.3.2. Cấu trúc bảng.

Bảng 3.17. Cấu trúc dữ liệu bảng AddressUser

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
id	Int(11)	Không	Null	Primary key, auto_increment	Mã địa chỉ tài khoản
userid	Int(11)	Có	Null	Foreign key	Mã khoản tài
shipName	Varchar(225)	Có	Null		Tên người nhận
shipAdress	Varchar(225)	Có	Null		Địa chỉ người nhận
shipEmail	Varchar(225)	Có	Null		Địa chỉ mail người nhận
shipPhonenumber	Varchar(225)	Có	Null		Số điện thoại người nhận
createdAt	datetime	Không			Ngày tạo
updateAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.18. Cấu trúc dữ liệu bảng Allcode

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
Id	int(11)	Không		Primary key, auto_increment	Mã định danh
type	varchar(255)	Có	NULL		Loại mã
value	varchar(255)	Có	NULL		Giá trị
code	varchar(255)	Có	NULL	Unique	Mã code
createdAt	datetime	Không			Ngày tạo

updatedAt	datetime	Không			Ngày chỉnh sửa
-----------	----------	-------	--	--	-------------------

Bảng 3.19. Cấu trúc dữ liệu bảng Banner

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
id	int(11)	Không		Primary key, auto_increment	Mã bảng rôn
description	longtext	Có	NULL		Mô tả
name	varchar(255)	Có	NULL		Tên bảng rôn
statusId	varchar(255)	Có	NULL	Foreign Key	Mã trạng thái
image	longblob	Có	NULL		Hình ảnh
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.20. Cấu trúc dữ liệu bảng blog

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
Id	int(11)	Không		Primary key, auto_increment	Foreign Key
shortdescription	longtext	Có			Mô tả ngắn
title	varchar(255)	Có	NULL		Tiêu đề
subjectId	varchar(255)	Có	NULL		Mã chủ đề
statusId	varchar(255)	Có	NULL		Mã trạng thái
image	longblob	Có	NULL		Hình ảnh
contentMarkdown	longtext	Có	NULL		Nội dung markdown dạng text

contentHTML	longtext	Có	NULL		Nội dung markdown dạng html
userId	int(11)	Có	NULL	Foreign Key	Mã người dùng
view	int(11)	Có	NULL		Lượt xem
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.21. Cấu trúc dữ liệu bảng comment

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
Id	int(11)	Không		Primary key, auto_increment	Mã bình luận / đánh giá
content	longtext	Có	NULL		Nội dung
image	longblob	Có	NULL		Hình ảnh
parentId	int(11)	Có	NULL		Mã bình luận / đánh giá cấp cha con
productId	int(11)	Có	NULL	Foreign Key	Mã sản phẩm
userId	int(11)	Có	NULL	Foreign Key	Mã người dùng
blogId	int(11)	Có	NULL	Foreign Key	Mã bài đăng
Star	int(11)	Có	NULL		Số sao
createdAt	datetime	Có			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.22. Cấu trúc dữ liệu bảng Masssge

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
Id	int(11)	Không		Primary key, auto_increment	Mã tin nhắn
text	longtext	Có	NULL		Nội dung
userId	int(11)	Có	NULL	Foreign Key	Mã người dùng
roomId	int(11)	Có	NULL	Foreign Key	Mã phòng
unRead	tinyint(1)	Có	NULL		Trạng thái đọc nhắn
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.23. Cấu trúc dữ liệu bảng OrderDetail

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
Id	int(11)	Không		Primary key, auto_increment	Mã chi tiết đơn hàng
orderId	int(11)	Có	NULL	Foreign Key	Mã hàng đơn
productId	int(11)	Có	NULL	Foreign Key	Mã kích thước sản phẩm
quantity	int(11)	Có	NULL		Số lượng
realPrice	bigint(20)	Có	NULL		Giá thực tế
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.24. Cấu trúc dữ liệu bảng OrderProduct

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
Id	int(11)	Không		Primary key, auto_increment	Mã đơn hàng
addressUserId	int(11)	Có	NULL	Foreign Key	Mã địa chỉ người dùng
statusId	varchar(255)	Có	NULL	Foreign Key	Mã trạng thái
typeShipId	int(11)	Có	NULL	Foreign Key	Mã loại vận chuyển
voucherId	int(11)	Có	NULL	Foreign Key	Mã khuyến mã
note	varchar(255)	Có	NULL		Ghi chú
isPaymentOnline	int(11)	Có	NULL		Trạng thái thanh toán
shipperId	int(11)	Có	NULL		Mã shipper
image	longblob	Có	NULL		Hình ảnh
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.25. Cấu trúc dữ liệu bảng ProductDetail

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
id	int(11)	Không		Primary key, auto_increment	Mã loại sản phẩm
productId	int(11)	Có	NULL	Foreign Key	Mã sản phẩm
description	longtext	Có	NULL		Mô tả
nameDetail	varchar(255)	Có	NULL		Tên loại sản phẩm
originalPrice	bigint(20)	Có	NULL		Giá gốc
discountPrice	bigint(20)	Có	NULL		Giá khuyến mãi
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.26. Cấu trúc dữ liệu bảng ProductDetailSize

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
id	int(11)	Không		Primary key, auto_increment	Mã kích thước loại sản phẩm
productdetailId	int(11)	Có	NULL	Foreign Key	Mã loại sản phẩm
width	varchar(255)	Có	NULL		Chiều rộng
height	varchar(255)	Có	NULL		Chiều cao
weight	varchar(255)	Có	NULL		Cân nặng
sizeId	varchar(255)		NULL	Foreign Key	Mã kích thước
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.27. Cấu trúc dữ liệu bảng ProductImage

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
id	int(11)	Không		Primary key, auto_increment	Mã hình ảnh loại sản phẩm
caption	varchar(255)	Có	NULL		Mô tả
productdetailId	int(11)	Có	NULL	Foreign Key	Mã loại sản phẩm
image	longblob	Có	NULL		Hình ảnh
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.28. Cấu trúc dữ liệu bảng Product

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
id	int(11)	Không		Primary key, auto_increment	Mã sản phẩm
name	varchar(255)	Có	NULL		Tên sản phẩm
contentHTML	longtext	Có	NULL		Nội dung markdown dạng html
contentMarkdown	longtext	Có	NULL		Nội dung markdown dạng text
statusId	varchar(255)	Có	NULL	Foreign Key	Mã trạng thái
categoryId	varchar(255)	Có	NULL	Foreign Key	Mã danh mục
view	int(11)	Có	NULL		Lượt xem

madeby	varchar(255)	Có	NULL		Làm/thiết kế bởi
material	varchar(255)	Có	NULL		Chất liệu
brandId	varchar(255)	Có	NULL	Foreign Key	Mã nhãn hàng
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.29. Cấu trúc dữ liệu bảng ShopCart

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
id	int(11)	Không		Primary key, auto_increment	Mã giỏ hàng
userId	int(11)	Có	NULL	Foreign Key	Mã người dùng
productdetailsId	int(11)	Có	NULL	Foreign Key	Mã kích thước loại sản phẩm
quantity	int(11)	Có	NULL		Số lượng
statusId	varchar(255)	Có	NULL		Trạng thái giỏ hàng
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.30. Cấu trúc dữ liệu bảng User

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
id	int(11)	Không		Primary key, auto_increment	Mã người dùng
email	varchar(255)	Có	Null		Địa chỉ email
password	varchar(255)	Có	Null		Mật khẩu
firstName	varchar(255)	Có	Null		Họ
lastName	varchar(255)	Có	Null		Tên
address	varchar(255)	Có	Null		Địa chỉ
genderId	varchar(255)	Có	Null	Foreign Key	Mã giới tính
phonenumber	varchar(255)	Có	Null		Số điện thoại
image	longblob	Có	Null		Hình ảnh
dob	varchar(255)	Có	Null		Ngày sinh
roleId	varchar(255)	Có	Null	Foreign Key	Mã vai trò
statusId	varchar(255)	Có	Null	Foreign Key	Mã trạng thái
isActiveEmail	tinyint(1)	Có	Null		Trạng thái kích hoạt email
usertoken	varchar(255)	Có	Null		Token người dùng
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.31. Cấu trúc dữ liệu bảng voucher

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
id	int(11)	Không		Primary key, auto_increment	Mã khuyến mãi
fromDate	varchar(255)	Có	NULL		Ngày bắt đầu
toDate	varchar(255)	Có	NULL		Ngày kết thúc
typeVoucherId	int(11)	Có	NULL	Foreign Key	Mã loại khuyến mãi
amount	int(11)	Có	NULL		Số lượng
codeVoucher	varchar(255)	Có	NULL		Mã code khuyến mãi
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.32. Cấu trúc dữ liệu bảng RoomMessage

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
id	int(11)	Không		Primary key, auto_increment	Mã phòng
userOne	int(11)	Có	NULL	Foreign Key	Mã người tham gia thứ nhất
userTwo	int(11)	Có	NULL	Foreign Key	Mã người tham gia thứ hai
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.33. Cấu trúc dữ liệu bảng TypeShip

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
id	int(11)	Không		Primary key, auto_increment	Mã loại vận chuyển
type	varchar(255)	Có	NULL		Tên loại
price	bigint(20)	Có	NULL		Giá tiền
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.34. Cấu trúc dữ liệu bảng TypeVoucher

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
id	int(11)	Không		Primary key, auto_increment	Mã loại khuyến mãi
typeVoucher	varchar(255)	Có	NULL		Tên loại
value	bigint(20)	Có	NULL		Giá trị
maxValue	bigint(20)	Có	NULL		Giá trị tối đa
minValue	bigint(20)	Có	NULL		Giá trị tối thiểu
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.35. Cấu trúc dữ liệu bảng VoucherUsed

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
id	int(11)	Không		Primary key, auto_increment	Mã id
voucherId	int(11)	Không		Foreign Key	Mã voucher
userId	int(11)	Không			Mã người dùng
status	int(11)	Không			Trạng thái
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.36. Cấu trúc dữ liệu bảng Supplier

Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
id	int(11)	Không		Primary key, auto_increment	Mã nhà cung cấp
name	varchar(255)	Có	NULL		Tên nhà cung cấp
address	varchar(255)	Có	NULL		Địa chỉ
phonenumber	varchar(255)	Có	NULL		Số điện thoại
email	varchar(255)	Có	NULL		Địa chỉ email
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.37. Cấu trúc dữ liệu bảng Receipt

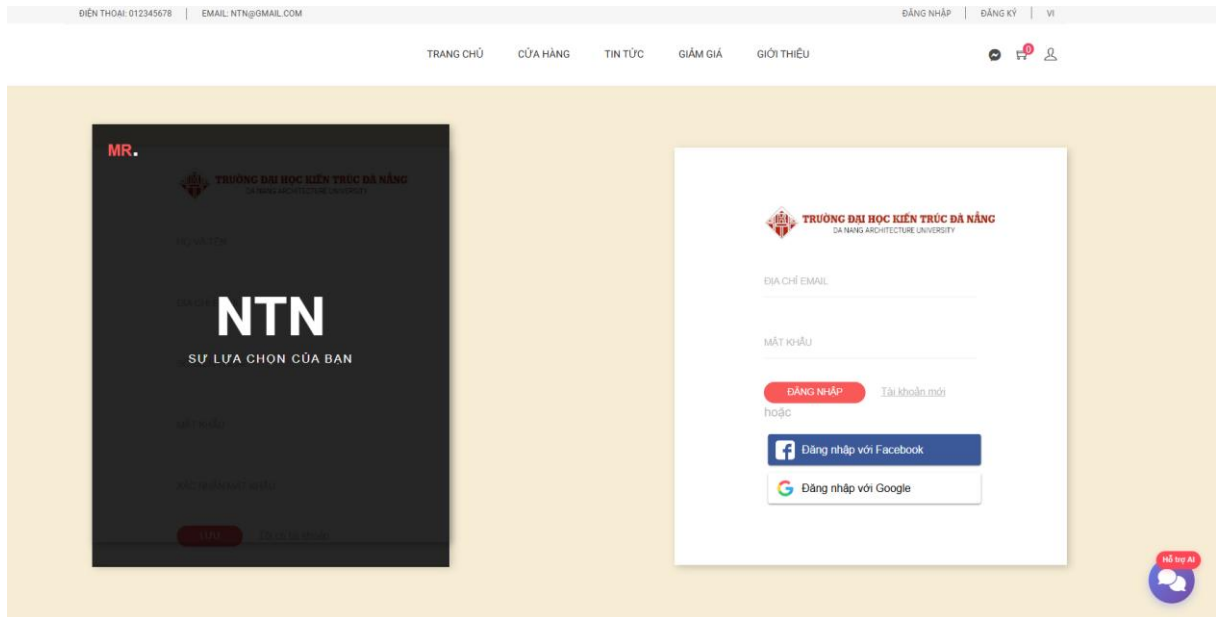
Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
id	int(11)	Không		Primary key, auto_increment	Mã nhập hàng
userId	int(11)	Có	NULL	Foreign Key	Mã nhân viên
supplierId	int(11)	Có	NULL	Foreign Key	Mã nhà cung cấp
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

Bảng 3.38. Cấu trúc dữ liệu bảng ReceiptDetail

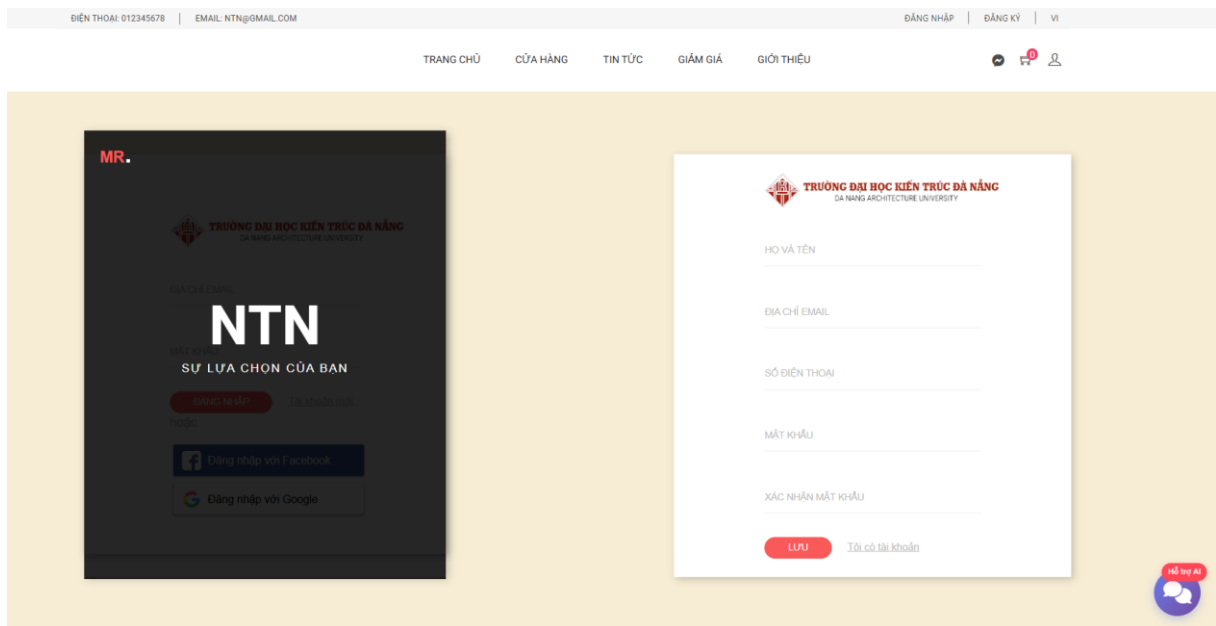
Cột	Kiểu	Null	Mặc định	Ràng buộc	Mô tả
id	int(11)	Không		Primary key, auto_increment	Mã chi tiết nhập hàng
receiptId	int(11)	Có	NULL	Foreign Key	Mã nhập hàng
productDetailSizeId	int(11)	Có	NULL	Foreign Key	Mã khách thước sản phẩm
quantity	int(11)	Có	NULL		Số lượng
price	bigint(20)	Có	NULL		Đơn giá
createdAt	datetime	Không			Ngày tạo
updatedAt	datetime	Không			Ngày chỉnh sửa

CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC

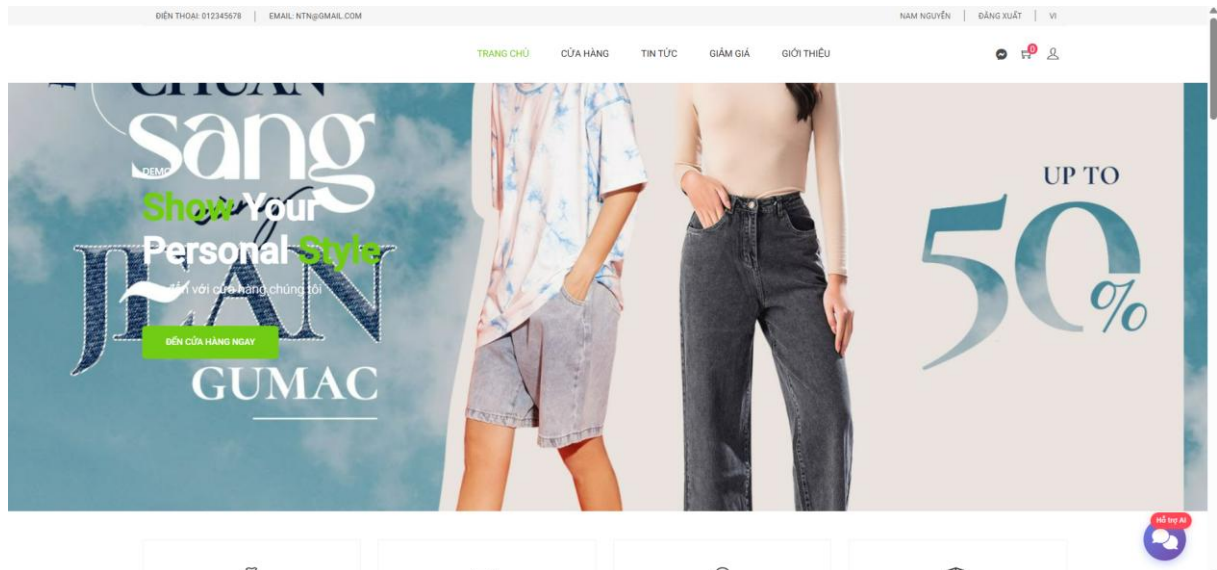
4.1. Giao diện người dùng.



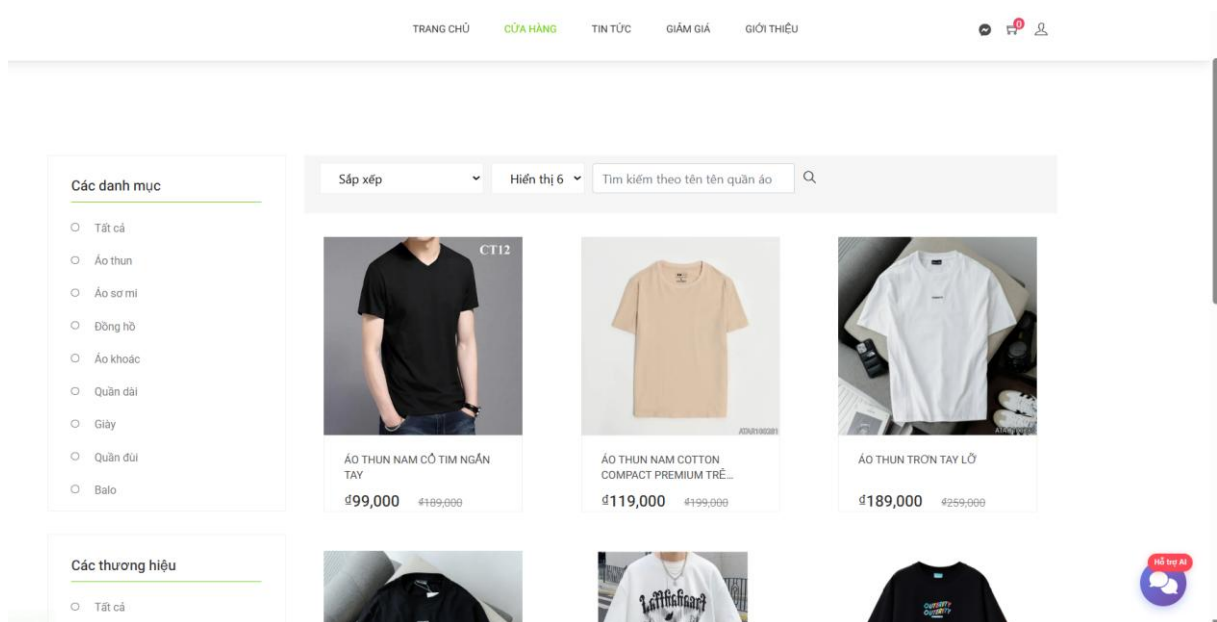
Hình 4.1. Giao diện trang đăng nhập



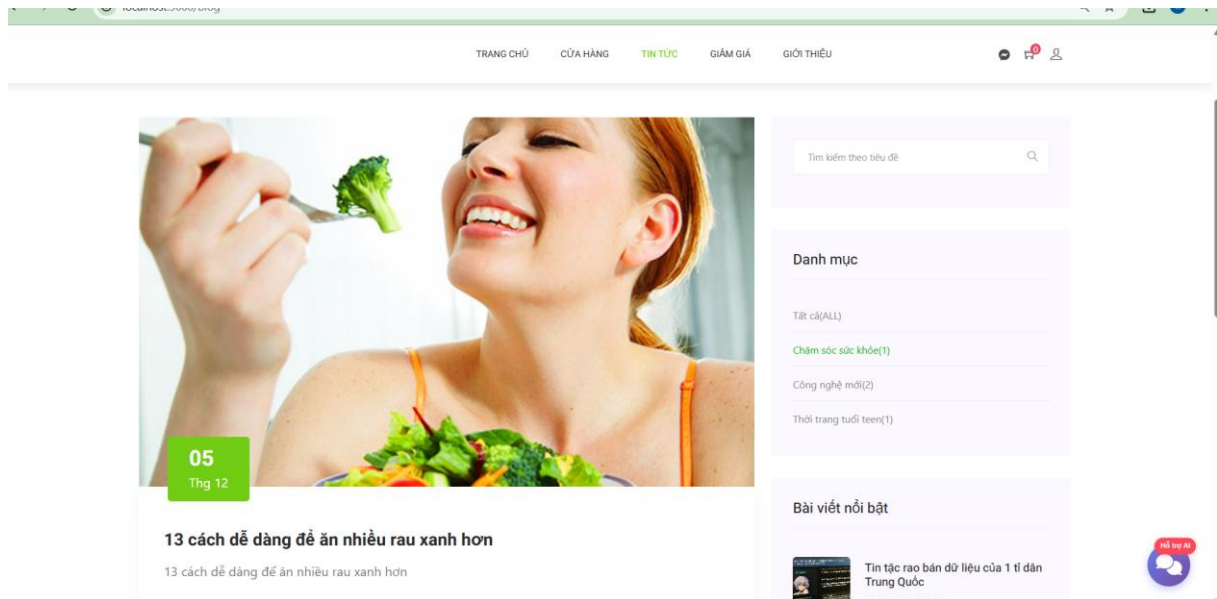
Hình 4.2. Giao diện trang đăng ký



Hình 4.3. Giao diện trang chủ



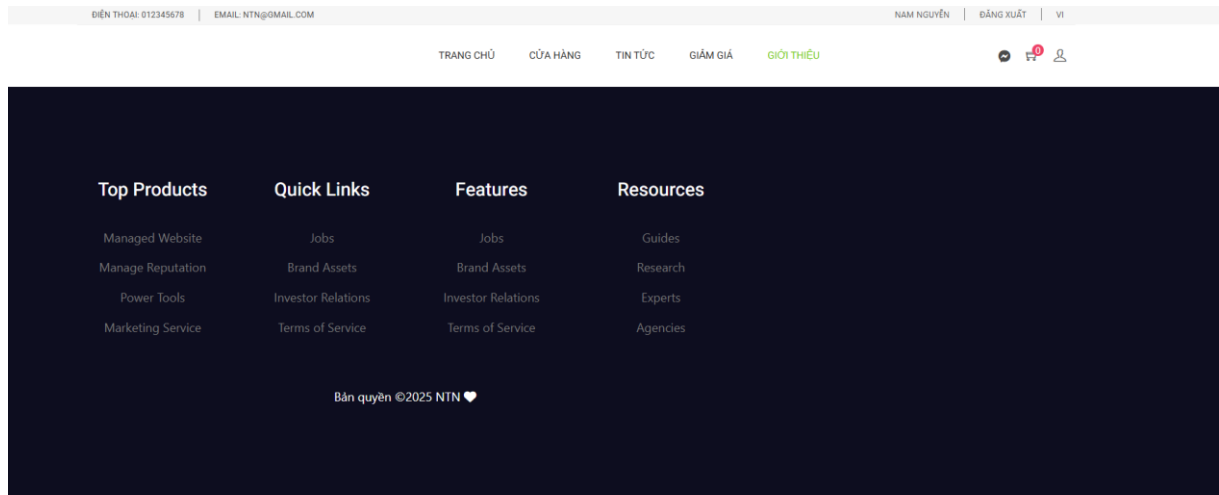
Hình 4.4. Giao diện trang cửa hàng



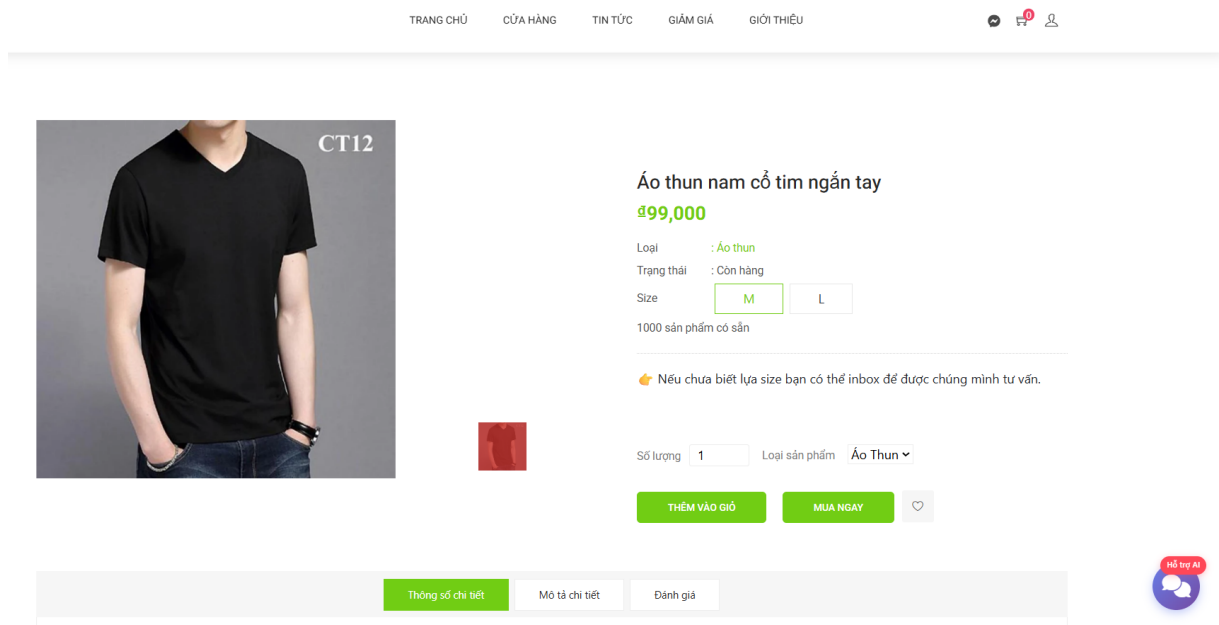
Hình 4.5. Giao diện trang tin tức



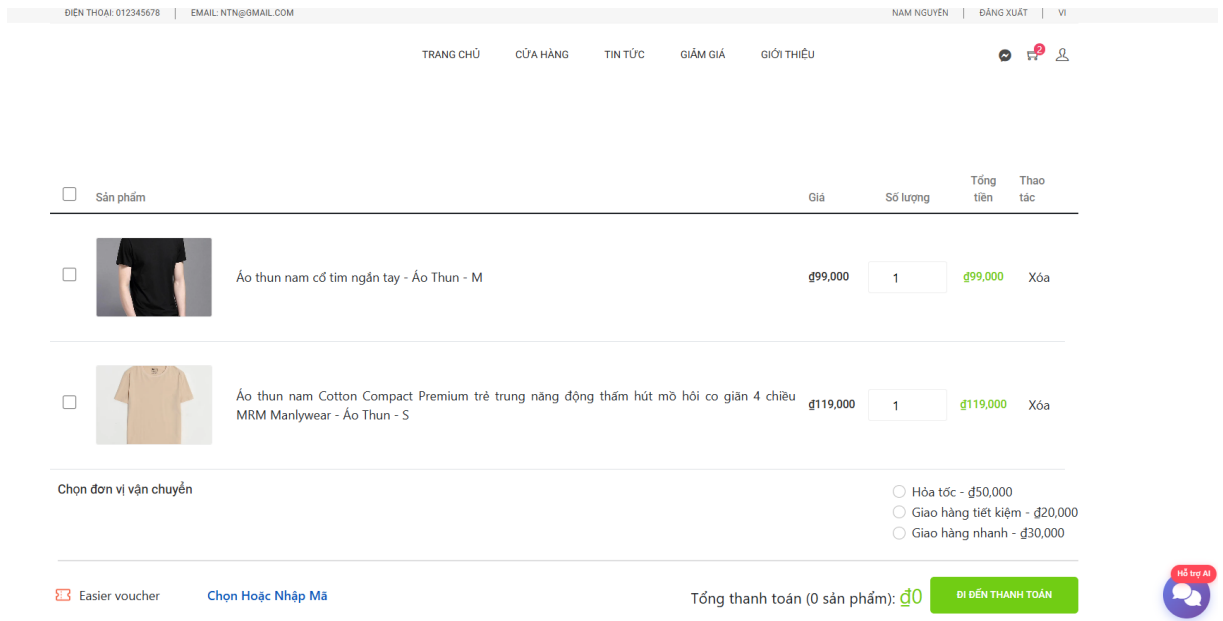
Hình 4.6. Giao diện trang mã giảm giá



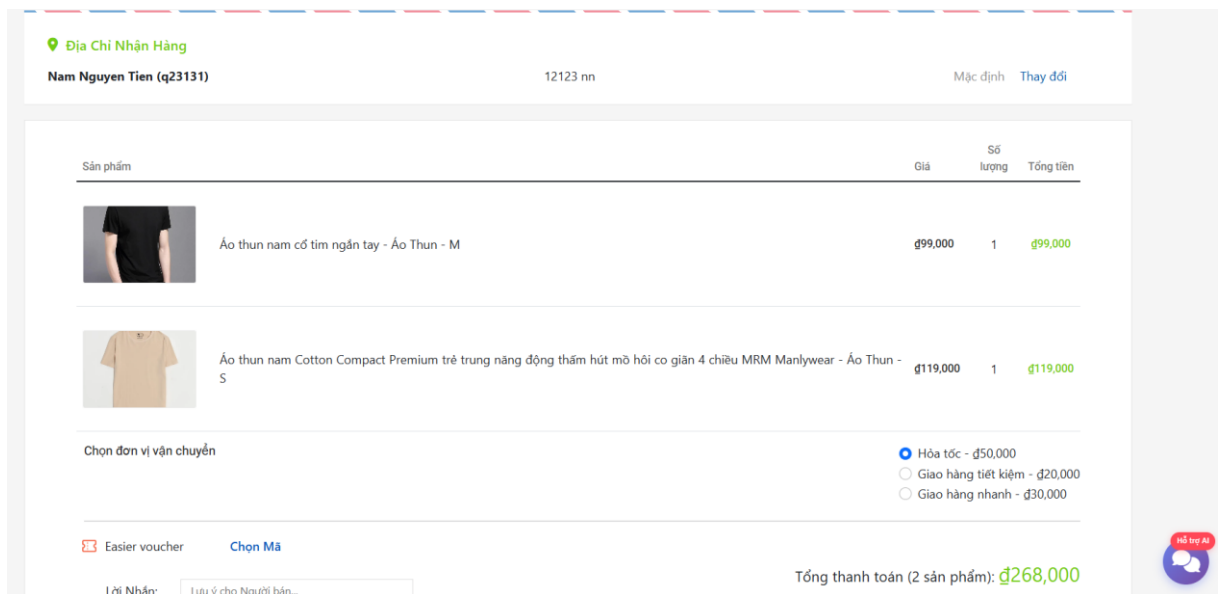
Hình 4.7. Giao diện trang giới thiệu



Hình 4.8. Giao diện chi tiết sản phẩm



Hình 4.9. Giao diện trang giỏ hàng



Hình 4.10. Giao diện chi tiết trang thanh toán

TRANG CHỦCỬA HÀNGTIN TỨCGIẢM GIÁGIỚI THIỆU

Nguyễn

admin@gmail.com

xác thực

Thông tin cá nhân

Họ

Nam

Tên

Nguyễn

Số điện thoại

0886862474

Địa chỉ

Quảng Trị

Giới tính

Nam

Ngày sinh

23/09/2003

Chọn ảnh

Tải ảnh

Lưu thông tin

Danh mục

Tài khoản của tôi

Hồ sơ

Địa chỉ

Đổi mật khẩu

Đơn mua

Kho voucher

Hình 4.11. Giao diện trang hồ sơ

TRANG CHỦCỬA HÀNGTIN TỨCGIẢM GIÁGIỚI THIỆU

Thay đổi thông tin tài khoản

Đổi mật khẩu

Mật khẩu cũ

Mật khẩu mới

Nhập lại mật khẩu

Lưu thông tin

Danh mục

Tài khoản của tôi

Hồ sơ

Địa chỉ

Đổi mật khẩu

Đơn mua

Kho voucher

Top ProductsQuick LinksFeaturesResources

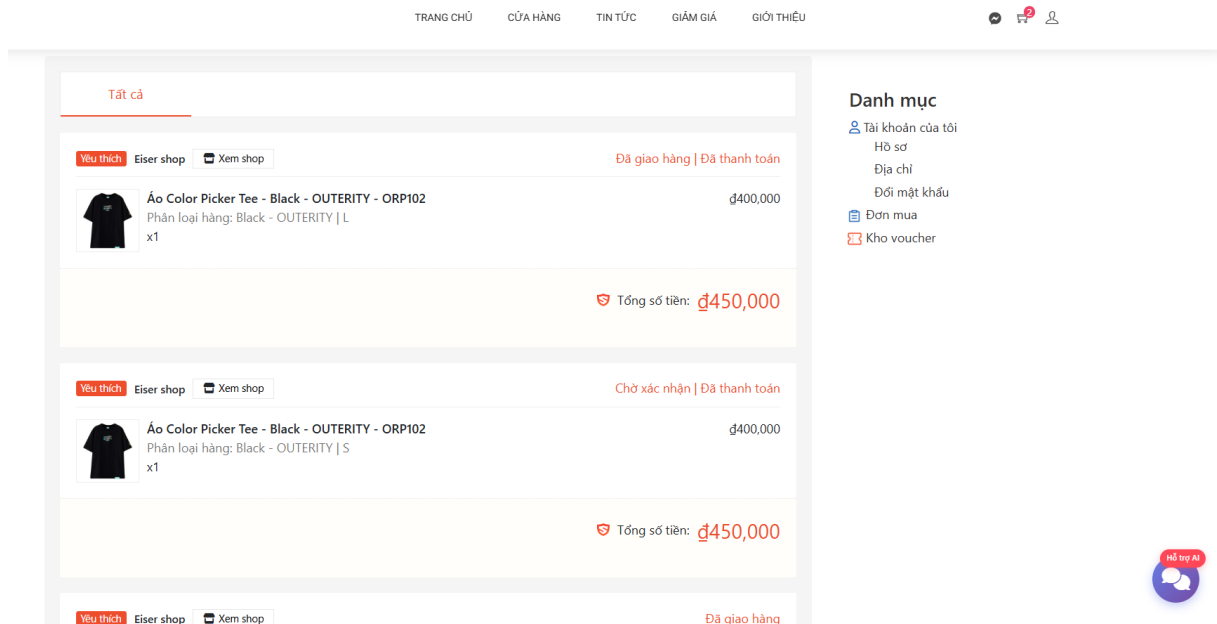
Hỗ trợ AI

Hình 4.12. Giao diện trang đổi mật khẩu

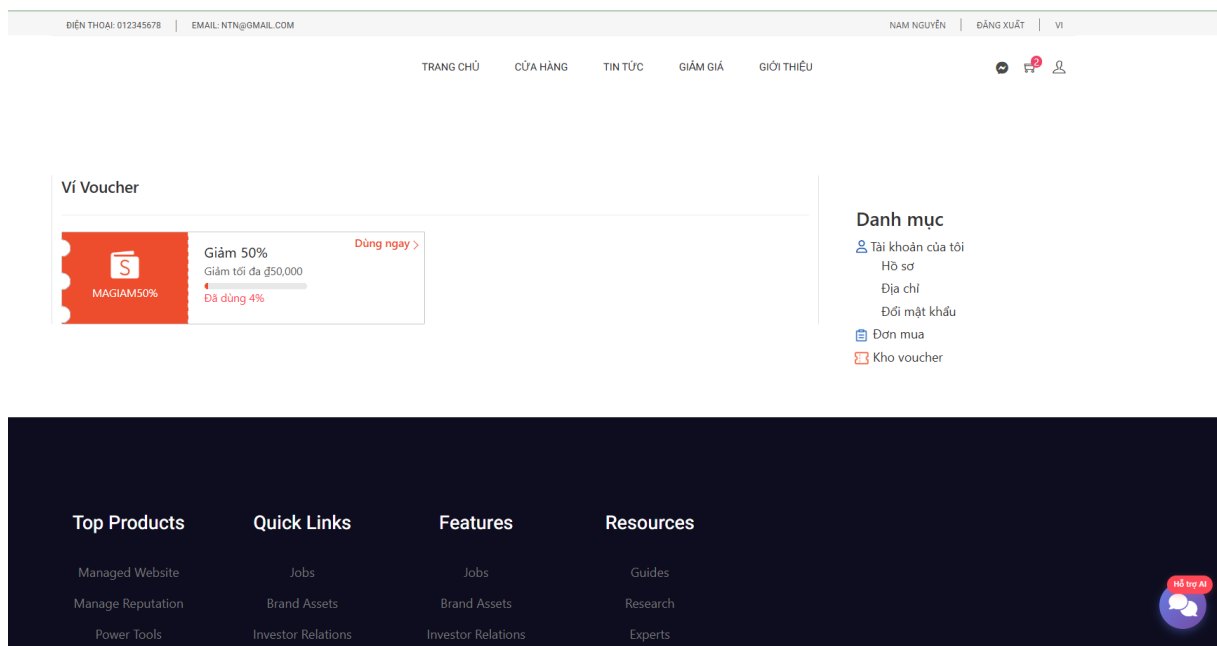
Sinh viên thực hiện: Nguyễn Tiến Nam

Hướng dẫn: TS. Lê Văn Minh

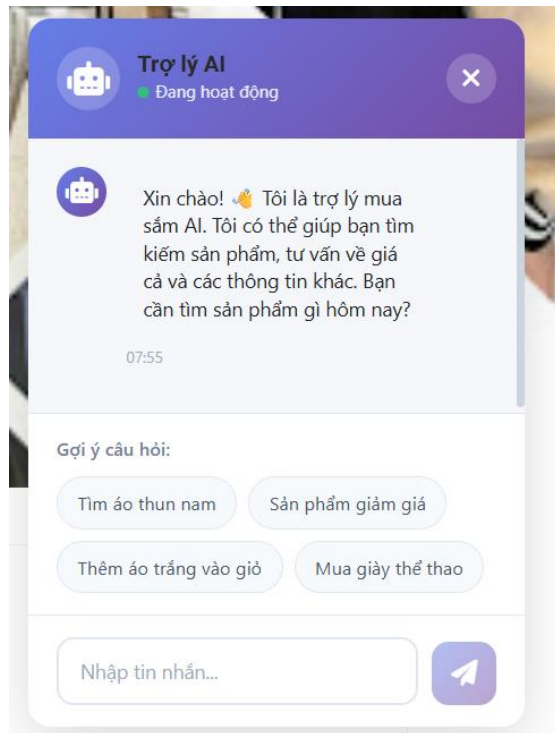
78



Hình 4.13. Giao diện trang đơn mua hàng

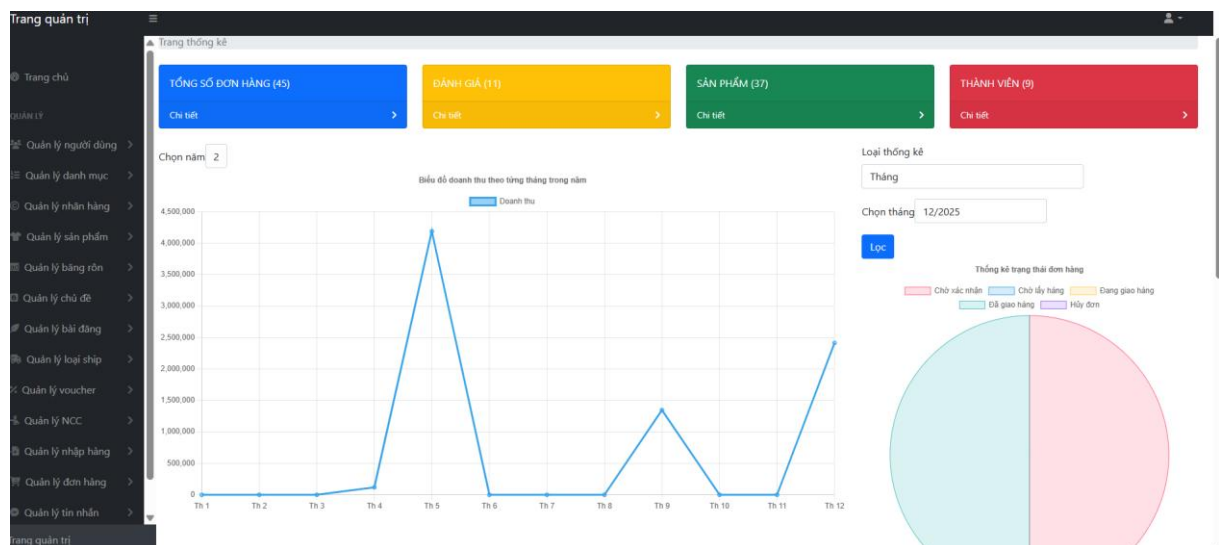


Hình 4.14. Giao diện trang kho voucher

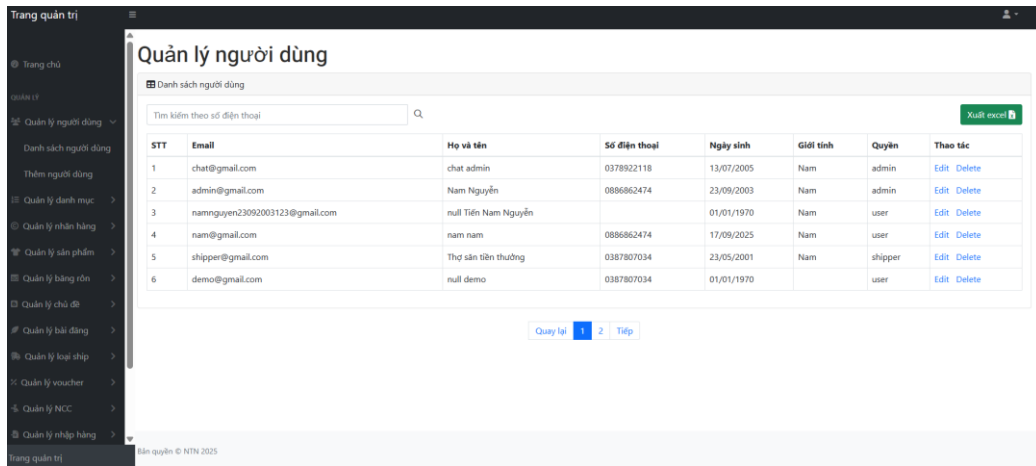


Hình 4.15. Giao diện botchatAI

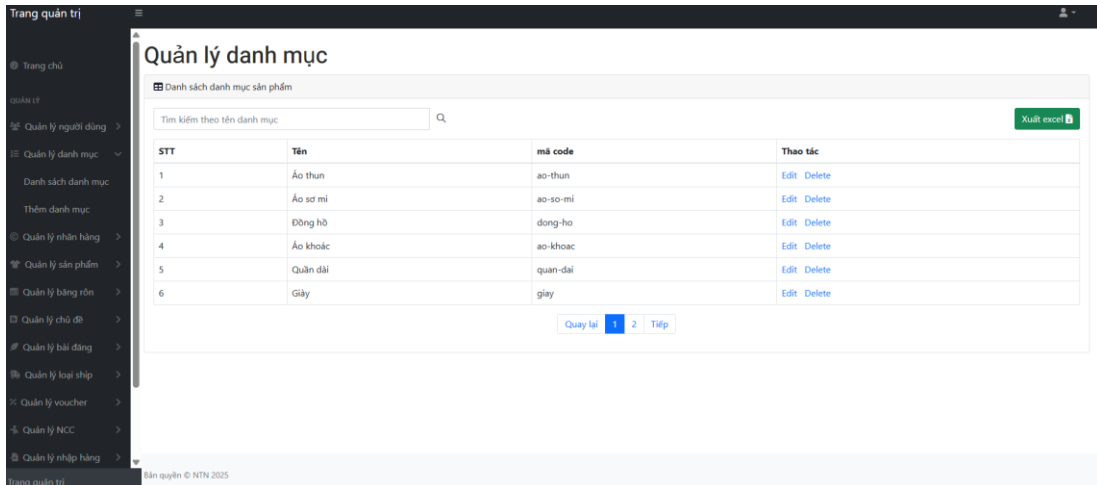
4.2. Giao diện Admin.



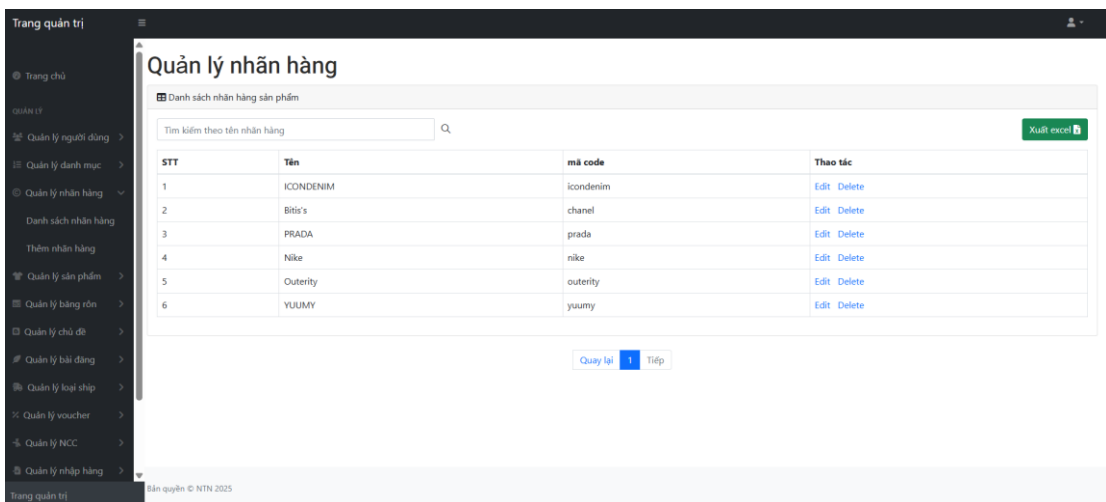
Hình 4.16. Giao diện trang chủ



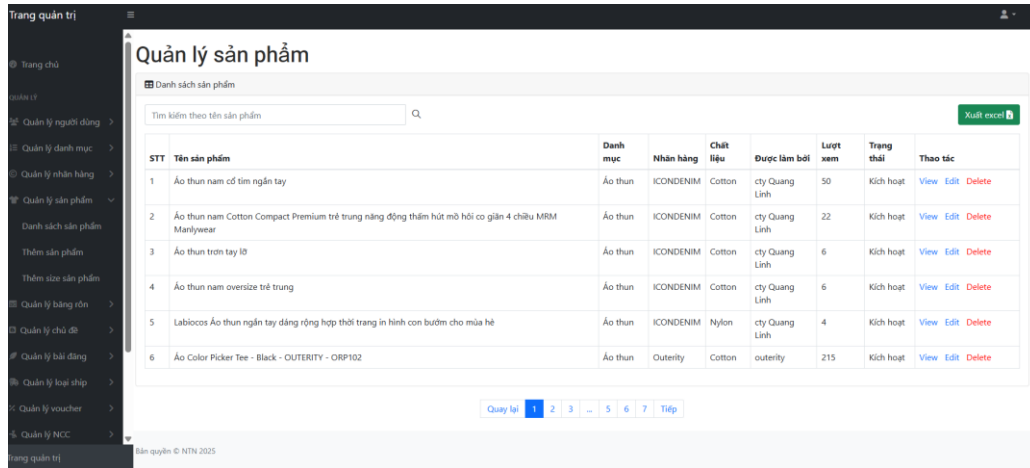
Hình 4.17. Giao diện trang quản lý người dùng



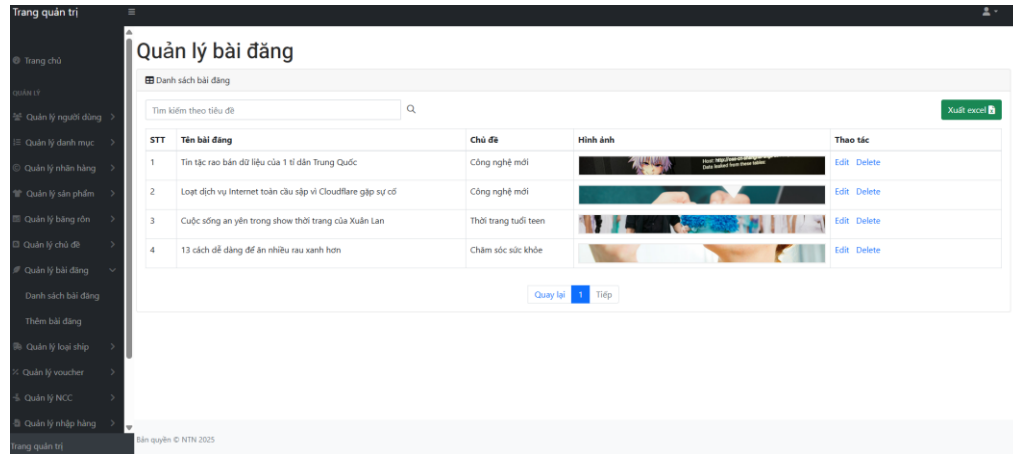
Hình 4.18. Giao diện trang quản lý danh mục



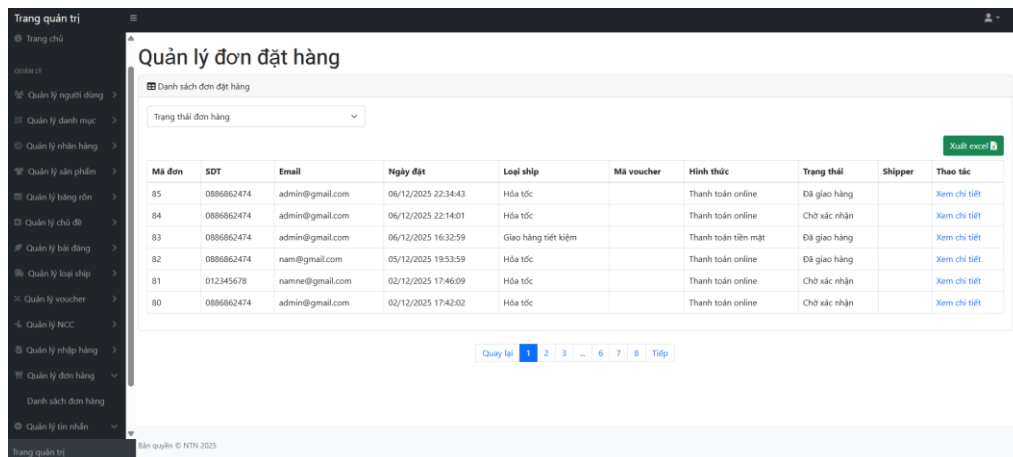
Hình 4.19. Giao diện trang quản lý nhãn hàng



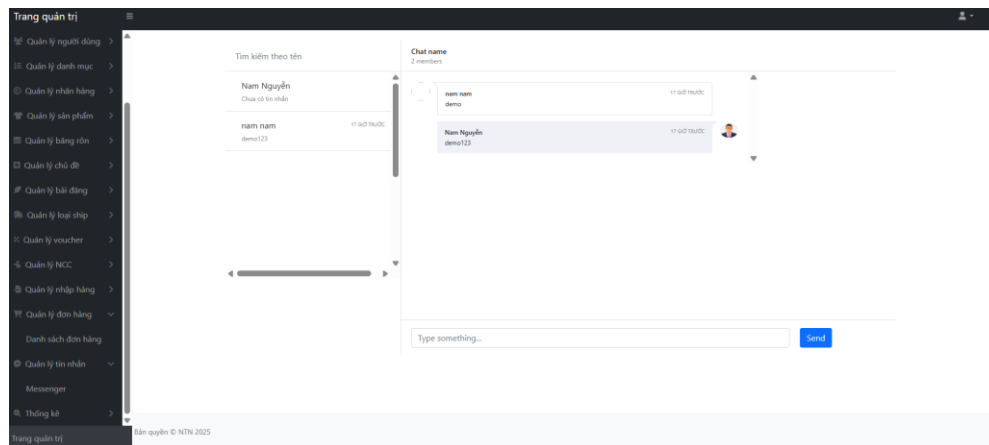
Hình 4.20. Giao diện trang quản lý sản phẩm



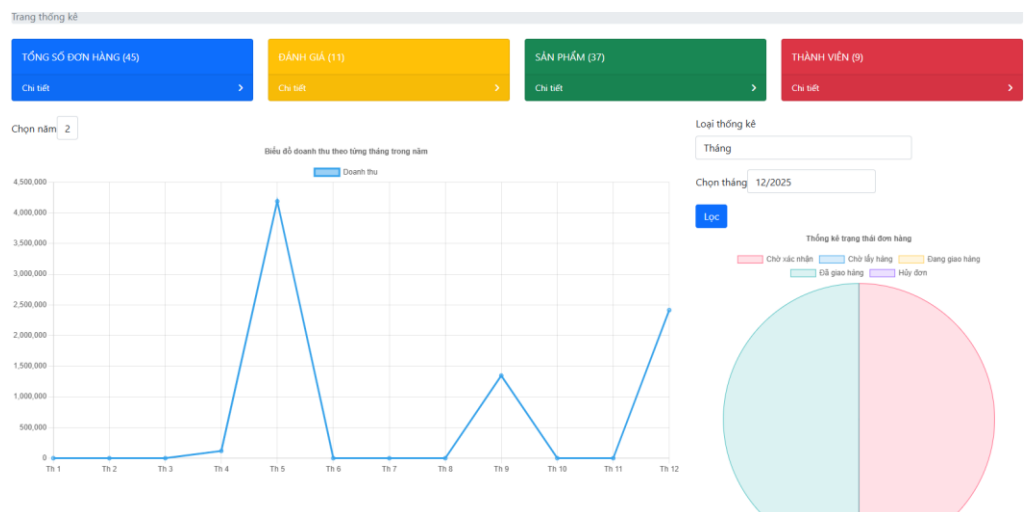
Hình 4.21. Giao diện trang quản lý bài đăng



Hình 4.22. Giao diện trang quản lý đơn hàng



Hình 4.23. Giao diện trang quản lý tin nhắn



Hình 4.24. Giao diện trang thống kê

KẾT LUẬN

Đề tài “Xây dựng website bán hàng thời trang sử dụng công nghệ ReactJS và NodeJS” đã được hoàn thành và đáp ứng các mục tiêu nghiên cứu đã đề ra. Trong suốt quá trình triển khai, hệ thống được xây dựng hướng đến sự tiện lợi cho người dùng, tính linh hoạt cho người quản trị và phù hợp với xu hướng thương mại điện tử hiện nay.

1. Thành tựu đạt được:

- Xây dựng thành công hệ thống website bán hàng thời trang hoàn chỉnh, hỗ trợ các chức năng cốt lõi như xem sản phẩm, tìm kiếm, đưa vào giỏ hàng, thanh toán, quản lý tài khoản, quản lý đơn hàng và quản trị hệ thống.
- Áp dụng ReactJS để xây dựng giao diện hiện đại, tương tác tốt, mượt mà và tối ưu trải nghiệm người dùng.
- Sử dụng NodeJS và ExpressJS cho backend giúp hệ thống xử lý nhanh, ổn định và dễ mở rộng.
- Hệ thống phân quyền rõ ràng, đảm bảo người dùng và quản trị viên được sử dụng đúng chức năng.
- Tích hợp ChatBot sử dụng GeminiAI, mang lại khả năng hỗ trợ người dùng tức thời như tư vấn size, gợi ý sản phẩm và hỗ trợ thao tác mua sắm.
- Cơ sở dữ liệu MySQL được thiết kế đầy đủ, đảm bảo lưu trữ và xử lý thông tin hiệu quả cho toàn bộ hệ thống.

2. Đóng góp và ý nghĩa:

- Đối với học thuật: Đề tài giúp củng cố và mở rộng kiến thức về lập trình web hiện đại, từ đó minh chứng rõ ràng khả năng ứng dụng ReactJS và NodeJS trong việc xây dựng hệ thống thương mại điện tử. Hơn nữa, quy trình thiết kế – phân tích – xây dựng – kiểm thử được áp dụng bài bản, tạo tiền đề tốt cho các dự án lớn trong tương lai.
- Đối với thực tiễn: Hệ thống mang tính ứng dụng cao, phù hợp cho các cửa hàng thời trang đang muốn chuyển đổi số. Website giúp đơn giản hóa quy trình bán hàng, mở rộng tiếp cận khách hàng và nâng cao hiệu quả quản lý sản phẩm – đơn

hàng – khách hàng. Đồng thời, chatbot và các tính năng tự động hóa góp phần cải thiện trải nghiệm mua sắm trực tuyến.

3. Hạn chế và bài học kinh nghiệm:

- Hạn chế: Một số tính năng nâng cao như gợi ý sản phẩm thông minh, phân tích hành vi người dùng hoặc báo cáo chuyên sâu chưa được triển khai do giới hạn thời gian, một số thao tác trên giao diện cần tối ưu thêm để trải nghiệm tốt hơn trên thiết bị di động.
- Bài học kinh nghiệm: Nâng cao kỹ năng lập trình full-stack với JavaScript, ReactJS và NodeJS, hiểu rõ tầm quan trọng của việc phân tích yêu cầu, thiết kế kiến trúc và quản lý thời gian, rèn luyện khả năng xử lý lỗi, kiểm thử và triển khai hệ thống một cách khoa học.

4. Hướng phát triển:

- Hệ thống có thể gợi ý sản phẩm dựa trên lịch sử xem, hành vi mua sắm hoặc xu hướng thời trang. Chatbot cũng có thể được nâng cấp để tư vấn thông minh hơn, hỗ trợ tự động hóa chăm sóc khách hàng.
- Xây dựng phiên bản Android/iOS giúp người dùng truy cập nhanh chóng, đồng thời tăng khả năng mua hàng mọi lúc, mọi nơi.
- Tối ưu giao diện trên mọi thiết bị, cải thiện tốc độ tải trang, bổ sung hiệu ứng tương tác nhằm mang đến trải nghiệm mua sắm chuyên nghiệp và hiện đại hơn.
- Hướng tới mô hình microservices hoặc Docker hóa toàn bộ hệ thống để dễ dàng mở rộng khi lượng truy cập tăng cao. Bổ sung cơ chế cache để nâng cao tốc độ xử lý và khả năng chịu tải

TÀI LIỆU THAM KHẢO

- [1] "ExpressJS" [Online]. Available: <https://expressjs.com/>.
- [2] "MySQL" [Online]. Available: <https://www.mysql.com/>.
- [3] "NodeJS" [Online]. Available: <https://nodejs.org/en/docs/>.
- [4] Tài liệu JavaScript: <https://javascript.info/37>
- [5] Tài liệu React: <https://reactjs.org/docs/getting-started.html>
- [6] StackOverFlow: <https://stackoverflow.com/>