

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 CÔNG NGHỆ

Người hướng dẫn: **ThS. NGUYỄN THỊ XUÂN THỦY**

Sinh viên thực hiện: **NGUYỄN HẢI HƯNG**

Mã số sinh viên: **2051220176**

Lớp: **20CT3**

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 CÔNG NGHỆ

Người hướng dẫn: **ThS. NGUYỄN THỊ XUÂN THỦY**

Sinh viên thực hiện: **NGUYỄN HẢI HÙNG**

Mã số sinh viên: **2051220176**

Lớp: **20CT3**

Đà Nẵng,/2025

TÓM TẮT

Tên đề tài: ***Xây dựng website bán hàng công nghệ.***

Sinh viên thực hiện : ***Nguyễn Hải Hưng .***

Mã số Sinh Viên: 2051220176.

Đề tài: “Xây dựng website bán hàng Công nghệ ” 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 công nghệ 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 sản phẩm (AI recommendation), Chatbot tư vấn sản phẩm, 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 mongoDB, Firebase 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 NÓI ĐẦU

Trong quá trình thực hiện đề án tốt nghiệp với đề tài “Xây dựng website bán hàng công nghệ ” 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 Th.S Nguyễn Thị Xuân Thủy, người đã tận tình hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện đề 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.

Em xin chân thành cảm ơn !

CAM ĐOAN

Tôi xin cam đoan rằng đồ án tốt nghiệp với đề tài "Xây dựng website bán hàng công nghệ" là kết quả nghiên cứu và làm việc của riêng tôi dưới sự hướng dẫn của Th.S Nguyễn Thị Xuân Thủy.

Tất cả nội dung trong đồ án, bao gồm các phần nghiên cứu lý thuyết, phân tích hệ thống, thiết kế và triển khai đều do tôi tự thực hiện. Những tài liệu, số liệu tham khảo từ các nguồn khác đều được trích dẫn rõ ràng và đúng quy định.

Tôi hoàn toàn chịu trách nhiệm về tính trung thực và chính xác của các nội dung trong đồ án này. Nếu có bất kỳ sai phạm nào, tôi xin chịu hoàn toàn trách nhiệm trước hội đồng và khoa CNTT của trường nhà trường.

Sinh viên thực hiện

MỤC LỤC

TÓM TẮT	1
LỜI NÓI ĐẦU	I
CAM ĐOAN	II
MỤC LỤC	III
DANH MỤC BẢNG BIỂU	VI
DANH MỤC HÌNH ẢNH	VII
DANH SÁCH KÝ HIỆU, CHỮ VIẾT TẮT	IX
CHƯƠNG 1. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU	1
1.1. Tổng quan đề tài	1
1.2. Đối tượng nghiên cứu	1
1.3. Phạm vi nghiên cứu	2
1.3.1. Về chức năng	3
1.3.2. Về công nghệ	3
1.3.3. Về sản phẩm	3
1.3.4. Về giao diện	3
1.3.5. Về môi trường triển khai	3
1.4. Yêu cầu khách hàng.....	4
1.5. Yêu cầu người quản trị.....	5
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ XÂY DỰNG HỆ THỐNG	7
2.1. Cơ sở lý thuyết	7
2.1.1. Giới thiệu về ngôn ngữ JavaScript	7
2.1.2. Giới thiệu về ReactJs	7
2.1.3. Giới thiệu về NodeJs	9
2.1.4. Giới thiệu về MongoDB	11

2.1.5. Giới thiệu về Firebase	12
2.1.6. Mô hình MVC	13
2.2. Công cụ và cài đặt	14
2.2.1. Visual studio code	14
2.2.2. MongoDB và Firebase.....	15
CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG THÔNG TIN	17
3.1. Giới thiệu	17
3.1.1. Khảo sát thực tế	17
3.1.2. Về khách hàng	18
3.1.3. Về phía Quản trị (Administrator)	19
3.1.4. Các yêu cầu phi chức năng	20
3.2. Biểu đồ Usecase.	20
3.2.1. Biểu đồ Usecase tổng quát.	20
3.2.2. Mô hình UseCase phân rã.....	22
3.2.3. Mô tả Usecase chi tiết	25
3.2.3.1. Usecase đăng kí	25
3.2.3.2. Usecase đăng nhập	26
3.2.3.3. Usecase quản lí sản phẩm	27
3.2.3.4. Usecase giỏ hàng	28
3.2.3.5. Usecase quản lí đơn hàng	30
3.3. Biểu đồ tuần tự	31
3.3.1. Biểu đồ tuần tự của User	31
3.3.2. Biểu đồ tuần tự của Admin	37
3.4. Thiết kế cơ sở dữ liệu	40
3.4.1. Quan hệ giữa các collection.	40
3.4.2. Cấu trúc các collection thực thể trong Website	43
CHƯƠNG 4. CẤU HÌNH, TRIỂN KHAI HỆ THỐNG VÀ KẾT QUẢ	53

4.1. Cấu hình hệ thống.	53
4.2. Triển khai hệ thống.	53
4.2.1. Cài đặt môi trường.....	53
4.2.2. Cài đặt CSDL.	54
4.2.3. Khởi động dự án	55
4.3. Kết quả đạt được.	56
4.3.1. Giao diện người dùng.	56
4.3.2. Giao diện người quản trị.	62
KẾT LUẬN	65
TÀI LIỆU THAM KHẢO.	67

DANH MỤC BẢNG BIỂU

Bảng 3.1. Bảng yêu cầu phi chức năng.	20
Bảng 3.2. Đặc tả Usecase tổng quát	21
Bảng 3.3. Đặc tả Usecase khách hàng.	22
Bảng 3.4. Đặc tả Usecase người quản trị.	24
Bảng 3.5. Đặc tả Uscase đăng kí.	25
Bảng 3.6. Đặc tả Usecase đăng nhập.	26
Bảng 3.7. Đặc tả Usecase quản lí sản phẩm.	27
Bảng 3.8. Đặc tả Usecase giỏ hàng.	28
Bảng 3.9. Bảng quan hệ collection products.	40
Bảng 3.10. Bảng quan hệ collection cart.	40
Bảng 3.11. Bảng quan hệ collection catalogs.	41
Bảng 3.12. Bảng quan hệ collection notifications.	41
Bảng 3.13. Bảng quan hệ collection user_voucher	41
Bảng 3.14. Bảng quan hệ collection users.	42
Bảng 3.15 Bảng quan hệ collection vouchers.	42
Bảng 3.16 Bảng quan hệ collection orders.	43
Bảng 3.17. Mô tả collection AdressUser.	44
Bảng 3.18. Mô tả collection Products.	45
Bảng 3.19. Mô tả collection catalogs.	45
Bảng 3.20. Mô tả collection Cart	46
Bảng 3.21. Mô tả collection Voucher.	47
Bảng 3.22. Mô tả collection UserVoucher.	48
Bảng 3.23. Mô tả collection Notifications.....	49
Bảng 3.24. Mô tả collection Order.	50
Bảng 3.25. Mô tả collection Userbehaviors.	52

DANH MỤC HÌNH ẢNH

Hình 3.1. Biểu đồ Usecase tổng quát.	20
Hình 3.2. Biểu đồ Usecase khách hàng.	22
Hình 3.3. Biểu đồ Usecase người quản trị.	23
Hình 3.4. Uscase đăng kí.	25
Hình 3.5. Usecase đăng nhập.	26
Hình 3.6. Usecase quản lí sản phẩm.	27
Hình 3.7. Usecase giỏ hàng.	28
Hình 3.8. Usecase quản lí đơn hàng.	30
Hình 3.9. Biểu đồ tuần tự đăng kí.	31
Hình 3.10. Biểu đồ tuần tự đăng nhập.	32
Hình 3.11. Biểu đồ tuần tự xem sản phẩm.	33
Hình 3.12. Biểu đồ tuần tự thêm sản phẩm vào giỏ hàng.	34
Hình 3.13 Biểu đồ tuần tự thanh toán.	36
Hình 3.14. Biểu đồ tuần tự thêm sản phẩm.	37
Hình 3.15. Biểu đồ tuần tự xóa sản phẩm.	38
Hình 3.16 Biểu đồ tuần tự cập nhập sản phẩm.	39
Hình 3.18. Minh họa collection User	43
Hình 3.19. Minh họa collection Products	44
Hình 3.20. Minh họa collection catalogs.	45
Hình 3.21. Minh họa collection Cart.	46
Hình 3.22. minh họa collection Voucher.	47
Hình 3.23. minh họa collection User Voucher.	48
Hình 3.24. Minh họa collection Notifications	49
Hình 3.25. Minh họa collection Order.....	50
Hình 3.26. Minh họa collection Userbehaviors.	52

Hình 4.1 Các collection được thêm vào Firebase trong Admin.	54
Hình 4.2 Các collection được thêm vào Mongoddb trong Admin.	54
Hình 4.3 Khởi tạo và chạy server.	55
Hình 4.4 Khởi tạo và kết nối với các api và cơ sở dữ liệu.	55
Hình 4.5. Trang giao diện Website.	56
Hình 4.6. Giao diện voucher.	57
Hình 4.7. Giao diện AIrecommendation.....	57
Hình 4.8. Trang đăng kí User.	58
Hình 4.9. Trang đăng nhập User.	58
Hình 4.10. Trang chi tiết sản phẩm.....	59
Hình 4.11. Trang giỏ hàng.	59
Hình 4.12. Trang lịch sử mua hàng.	60
Hình 4.13. Trang thanh toán trực tuyến Zalopay.	60
Hình 4.14. Trang phương thức thanh toán.	61
Hình 4.15. Trang Chatbot hỗ trợ khách hàng.	61
Hình 4.16 Trang thông báo cho người dùng.	62
Hình 4.17. Trang quản trị Admin.	62
Hình 4.18. Trang quản lí sản phẩm.....	63
Hình 4.19. Trang thêm sản phẩm.	63
Hình 4.20. Trang quản lí đơn hàng.	64
Hình 4.21. Trang quản lí Voucher.	64

DANH SÁCH KÝ HIỆU, CHỮ VIẾT TẮT

<i>STT</i>	<i>Kí hiệu viết tắt</i>	<i>Chữ viết tắt đầy đủ</i>
1	HTML	HyperText Markup Language
2	CSS	Cascading Style Sheets
3	PHP	Hypertext Preprocessor
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	IDE	Integrated Development Environment

CHƯƠNG 1. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

1.1. Tổng quan đề tài

Trong bối cảnh ***công nghệ thông tin và Internet phát triển mạnh mẽ*** thì môi trường 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. 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 các shop bán đồ công nghệ nên có thêm một website để có thể bán lẻ các đồ công nghệ để có thể quản lý hàng hóa tốt hơn, báo cáo doanh thu nhanh chóng, giảm nhân lực, chi phí và thời gian, quảng cáo sản phẩm đến tay người dùng cũng dễ dàng hơn.

Vì các vấn đề đã nêu, em chọn đề tài “Xây dựng website bán hàng công nghệ” nhằm giúp cửa hàng cải thiện được vấn đề quản lý, quảng cáo hàng hóa của mình. Đồng thời phù hợp với xu hướng phát triển của công nghệ trên 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. 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.

Khách hàng (người mua hàng): Nghiên cứu nhu cầu và trải nghiệm của khách hàng khi mua sắm các sản phẩm trực tuyến trên website, là người sử dụng cuối

cùng của hệ thống vai trò chính của nhóm người dùng này là mua sắm sản phẩm trên website. Các chức năng chính mà người dùng có thể sử dụng trong website bao gồm: - Truy cập website để xem các sản phẩm công nghệ.

- Đăng nhập, đăng kí.
- Xem sản phẩm.
- Thu thập voucher.
- Đưa sản phẩm vào giỏ hàng, thanh toán, đặt mua.
- Nếu chưa có lựa chọn có thể nhờ chobot tư vấn.
- Theo dõi đơn hàng, trạng thái đơn hàng.

Người quản trị/Chủ cửa hàng: Tìm hiểu nhu cầu quản lý sản phẩm, quản lý đơn hàng, quản lý khách hàng và phân quyền người dùng để vận hành hệ thống bán hàng trực tuyến hiệu quả. Cũng là người chịu trách nhiệm quản lý của người quản lý và điều hành hoạt động của hệ thống. Các chức năng chính của người quản trị bao gồm:

- Quản lý sản phẩm (thêm, sửa, xóa, cập nhật)
- Quản lý danh mục.
- Quản lý thông tin sản phẩm.
 - Quản lý đơn hàng, xác nhận tình trạng đơn hàng.
 - Quản lý người dùng.
 - Quản lý số lượng tồn kho.
 - Quản lý voucher.
 - Theo dõi doanh thu và thống kê.

1.3. Phạm vi nghiên cứu

Dự án sẽ tập trung vào việc nghiên cứu và ứng dụng công nghệ xây dựng một website bán hàng công nghệ dựa trên các công cụ và ngôn ngữ lập trình như:

NodeJS để xây dựng phần backend của hệ thống.

MongoDB và Firebase: Lưu trữ và quản lý dữ liệu người dùng và thông tin thi cử.

JavaScript và HTML/CSS: Xây dựng giao diện người dùng trực quan và thân thiện.

Phát triển ứng dụng: Xây dựng Website bán hàng công nghệ, bao gồm các tính năng như tạo sản phẩm, quản lý sản phẩm, và một số chức năng khác.

Phạm vi nghiên cứu sẽ tập trung vào việc thiết kế và phát triển một hệ thống bán hàng trực tuyến toàn diện, thân thiện với người dùng, đảm bảo tính bảo mật và linh hoạt cho người sử dụng

1.3.1. 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).

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

1.3.2. 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: Firebase và MongoDB để lưu cơ sở dữ liệu.

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

Có tìm hiểu thêm về các bootstrap.

1.3.3. Về sản phẩm

Website tập trung vào việc bán các sản phẩm công nghệ của cửa hàng.

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

1.3.4. Về giao diện

Nghiên cứu thiết kế giao diện website theo phong cách hiện đại, thông minh, phù hợp với cửa hàng công nghệ.

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

1.3.5. 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.

1.4. Yêu cầu khách hàng

Khách hàng là người sử dụng cuối cùng của hệ thống - những người truy cập website để xem, lựa chọn và mua sản phẩm công nghệ.

Các yêu cầu của khách hàng được tập trung vào trải nghiệm sử dụng, tính tiện lợi và sự an toàn khi mua hàng trực tuyến.

Các yêu cầu chức năng:

Đăng ký và đăng nhập tài khoản:

Khách hàng có thể tạo tài khoản mới hoặc đăng nhập bằng email, Google thông qua hệ thống Firebase Authentication.

Xem danh mục sản phẩm:

Người dùng có thể xem các loại sản phẩm thời trang được phân loại rõ ràng như: ***điện thoại, máy tính, laptop, đồng hồ, đồ gia dụng,...***

Tìm kiếm sản phẩm:

Hệ thống hỗ trợ tìm kiếm nhanh theo tên sản phẩm, danh mục hoặc từ khóa liên quan.

Xem chi tiết sản phẩm:

Khách hàng có thể xem thông tin chi tiết của từng sản phẩm như hình ảnh, giá, mô tả, kích thước, màu sắc và tình trạng còn hàng.

Thêm sản phẩm vào giỏ hàng:

Cho phép người dùng chọn và lưu các sản phẩm mong muốn trước khi thanh toán.

Cập nhật giỏ hàng:

Khách hàng có thể thay đổi số lượng, xóa hoặc thêm sản phẩm mới trong giỏ hàng.

Chatbot:

Khách hàng có thể nhờ chatbot tư vấn sản phẩm hoặc hỏi về giá trị hoặc hiệu số của sản phẩm.

Thanh toán đơn hàng:

Hệ thống hỗ trợ khách hàng nhập thông tin giao hàng và xác nhận đơn hàng ship COD và thanh toán trực tuyến thông qua zalopay hoặc các tranh thanh toán tiền tự động khác.

Xem lịch sử đơn hàng:

Cho phép người dùng xem lại các đơn hàng đã mua, tình trạng đơn hàng và thời gian giao hàng.

Giao diện thân thiện:

Website được thiết kế **responsive**, hiển thị tốt trên cả máy tính.

Các yêu cầu phi chức năng:

Hệ thống hoạt động ổn định, tốc độ nhanh, và phản hồi tức thì.

Giao diện trực quan, dễ sử dụng, màu sắc phù hợp với phong cách thời trang.

Dữ liệu người dùng được bảo mật, đặc biệt là thông tin đăng nhập và đơn hàng

Hệ thống có khả năng xử lý đồng thời nhiều truy cập mà không bị gián đoạn

1.5. Yêu cầu người quản trị

Người quản trị (Admin) là người chịu trách nhiệm quản lý và vận hành toàn bộ hệ thống website bán hàng. Các yêu cầu của quản trị viên tập trung vào ***quản lý dữ liệu sản phẩm, đơn hàng, và người dùng*** để đảm bảo hoạt động của website diễn ra hiệu quả.

Các yêu cầu chức năng:

Đăng nhập quản trị:

Quản lý sản phẩm:

Quản lý danh mục:

Quản lý đơn hàng:

Quản lý người dùng:

Thống kê và báo cáo:

Các yêu cầu phi chức năng:

Giao diện quản trị trực quan, dễ thao tác.

Hạn chế truy cập: chỉ có tài khoản của quản trị mới được truy cập vào trang Admin, tài khoản khác thì phải chờ quyền chấp nhận truy cập web của Admin.

- ☐ **Bảo mật dữ liệu:** Các thao tác xóa, sửa hoặc thay đổi trạng thái đơn hàng phải được xác nhận.
- ☐ **Hiệu năng tốt:** Hệ thống có khả năng xử lý dữ liệu lớn mà không gây chậm trễ.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ XÂY DỰNG HỆ THỐNG

2.1. Cơ sở lý thuyết

2.1.1. Giới thiệu về ngôn ngữ JavaScript

JavaScript là 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.

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).

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.

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 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.

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.

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 server side 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 phát triển ứng dụng web với khả năng tái sử dụng mã nguồn.

2.1.3. Giới thiệu về NodeJs

NodeJS 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.

Ứng dụng của NodeJS:

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 NodeJS:

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 NodeJS:

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 nên sử dụng NodeJS:

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.

2.1.4. Giới thiệu về MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu phi quan hệ (NoSQL) mã nguồn mở, được phát triển bởi công ty MongoDB Inc. và phát hành lần đầu vào năm **2009**.

Khác với cơ sở dữ liệu quan hệ truyền thống (như MySQL), MongoDB lưu trữ dữ liệu dưới dạng **document (tài liệu)** theo cấu trúc **JSON/BSON** linh hoạt, cho phép mở rộng và thay đổi cấu trúc dữ liệu dễ dàng mà không cần định nghĩa bảng hoặc cột cố định.

MongoDB đặc biệt phù hợp với các ứng dụng web hiện đại sử dụng công nghệ **Node.js** và **ReactJS**, nhờ khả năng lưu trữ và truy xuất dữ liệu nhanh chóng, linh hoạt, cùng với khả năng tích hợp tự nhiên với JavaScript.

Đặc điểm nổi bật của MongoDB:

Phi quan hệ: Không sử dụng bảng hay dòng như trong cơ sở dữ liệu quan hệ, mà tổ chức dữ liệu dưới dạng tài liệu JSON.

Tính linh hoạt cao: Không yêu cầu cấu trúc dữ liệu cố định (schema-less), dễ dàng mở rộng hoặc thay đổi mô hình dữ liệu.

Hiệu năng tốt: Hỗ trợ truy vấn nhanh, tối ưu hóa cho xử lý dữ liệu lớn và thời gian thực.

Mở rộng dễ dàng: Hỗ trợ sharding (phân mảnh dữ liệu) và replication (sao lưu dữ liệu) giúp mở rộng quy mô lưu trữ và tăng độ tin cậy.

- ☐ Mã nguồn mở và đa nền tảng: Có thể chạy trên nhiều hệ điều hành như Windows, Linux, macOS.

Ứng dụng trong dự án:

Trong đồ án xây dựng website bán hàng công nghệ, MongoDB được sử dụng làm cơ sở dữ liệu chính để lưu trữ thông tin sản phẩm, khách hàng, đơn hàng và giỏ hàng. Nhờ cấu trúc dữ liệu linh hoạt và khả năng tích hợp tốt với Node.js, MongoDB giúp hệ thống có thể dễ dàng thêm mới, chỉnh sửa, và quản lý dữ liệu một cách nhanh chóng và hiệu quả.

2.1.5. Giới thiệu về Firebase

Firestore là nền tảng phát triển ứng dụng web và di động do Google phát triển, cung cấp một hệ sinh thái gồm nhiều công cụ và dịch vụ giúp lập trình viên xây dựng, triển khai và quản lý ứng dụng một cách dễ dàng.

Firestore được giới thiệu lần đầu vào năm **2011**, và đến nay đã trở thành một trong những nền tảng phổ biến nhất trong phát triển web hiện đại, đặc biệt khi kết hợp với ReactJS và Node.js.

Firestore hoạt động trên nền điện toán đám mây (Cloud), cho phép ứng dụng hoạt động và đồng bộ dữ liệu theo thời gian thực, đồng thời hỗ trợ nhiều tính năng như xác thực người dùng, lưu trữ tệp, phân tích dữ liệu, và triển khai website.

Các dịch vụ của Firebase:

Firebase Authentication: Quản lý đăng nhập, đăng ký, và xác thực người dùng bằng email, Google, Facebook,...

Cloud Firestore/Realtime Database: Cơ sở dữ liệu NoSQL dạng tài liệu, hỗ trợ đồng bộ dữ liệu real-time giữa client và server.

Firebase Storage: Lưu trữ và quản lý hình ảnh hoặc tệp tải lên.

Firebase Cloud Functions: Cho phép viết các hàm xử lý phía server mà không cần quản lý máy chủ vật lý.

Ứng dụng trong dự án:

Trong đồ án xây dựng website bán hàng công nghệ, Firebase được sử dụng để:

Quản lý xác thực người dùng (đăng nhập, đăng ký, đăng xuất).

Lưu trữ hình ảnh sản phẩm trong Firebase Storage.

Thêm sản phẩm.

Nhờ Firebase, nhóm phát triển có thể rút ngắn thời gian triển khai, đảm bảo bảo mật và dễ dàng mở rộng hệ thống, mà không cần xây dựng hạ tầng server phức tạp.

2.1.6. 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:

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:

Bước 1: Người dùng tương tác với View (ví dụ: click vào nút "Thêm vào giỏ hàng").

Bước 2: View gửi yêu cầu đến Controller.

Bước 3: Controller xử lý yêu cầu bằng cách tương tác với Model (ví dụ: thêm sản phẩm vào giỏ hàng trong database).

Bước 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.

Đễ 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.

2.2. Công cụ và cài đặt

Việc xây dựng một website hoàn chỉnh, đặc biệt là website bán hàng với nhiều chức năng phức tạp như website bán hàng công nghệ, đòi hỏi sự kết hợp của nhiều công cụ hỗ trợ khác nhau. Các công cụ này giúp đơn giản hóa quy trình phát triển, nâng cao năng suất làm việc, và tạo ra sản phẩm chất lượng cao.

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. *MongoDB và Firebase*

Firebase và MongoDB là hai nền tảng mạnh mẽ hỗ trợ phát triển ứng dụng web hiện đại, thay thế cho các giải pháp truyền thống như XAMPP và MySQL.

Firebase là nền tảng phát triển ứng dụng web và di động do **Google** cung cấp, hỗ trợ nhiều dịch vụ như xác thực người dùng (Authentication), lưu trữ dữ liệu thời gian thực (Realtime Database hoặc Firestore), lưu trữ hình ảnh (Storage), và triển khai web (Hosting).

Trong khi đó, MongoDB là hệ quản trị cơ sở dữ liệu phi quan hệ (NoSQL) mã nguồn mở, được tối ưu hóa cho việc lưu trữ dữ liệu linh hoạt dưới dạng document JSON, phù hợp với các ứng dụng web sử dụng Node.js.

Firebase và MongoDB kết hợp tạo nên một môi trường phát triển mạnh mẽ, cho phép các lập trình viên **xây dựng, quản lý, và mở rộng ứng dụng web** một cách nhanh chóng mà không cần cài đặt server thủ công như trong XAMPP.

Vai trò trong dự án:

Trong dự án xây dựng website bán hàng công nghệ, Firebase và MongoDB đóng vai trò là nền tảng phát triển và lưu trữ dữ liệu chính.

Firebase được sử dụng để:

Quản lý xác thực người dùng (đăng ký, đăng nhập).

Lưu trữ và quản lý hình ảnh sản phẩm trên Firebase Storage.

Triển khai (hosting) website lên môi trường trực tuyến nhanh chóng.

MongoDB được sử dụng để:

Lưu trữ thông tin sản phẩm, đơn hàng, khách hàng dưới dạng các tài liệu (documents) linh hoạt.

Giao tiếp với Node.js backend thông qua thư viện Mongoose, giúp thao tác dữ liệu hiệu quả:

Nhờ sự kết hợp này, lập trình viên có thể ***xây dựng toàn bộ hệ thống web động***, bao gồm cả phần giao diện người dùng (ReactJS), máy chủ (Node.js), và cơ sở dữ liệu (MongoDB, Firebase) chỉ trong một hệ thống thống nhất.

Ưu điểm:

Không cần cài đặt phức tạp: Firebase và MongoDB Atlas hoạt động trực tuyến, cho phép triển khai nhanh mà không cần cấu hình server như XAMPP.

Thời gian thực: Firebase hỗ trợ đồng bộ dữ liệu real-time, giúp phản hồi tức thì khi người dùng thêm hoặc chỉnh sửa dữ liệu.

Lưu trữ linh hoạt: MongoDB không cần định nghĩa bảng cố định, dễ dàng mở rộng và thay đổi cấu trúc dữ liệu.

Bảo mật cao: Firebase cung cấp cơ chế xác thực và phân quyền người dùng tích hợp sẵn; MongoDB hỗ trợ mã hóa và xác thực truy cập.

Khả năng mở rộng mạnh: Cả Firebase và MongoDB đều hỗ trợ mở rộng trên nền tảng đám mây, phù hợp cho các ứng dụng thương mại điện tử.

Tích hợp tốt với React và Node.js: Dữ liệu dạng JSON giúp trao đổi dễ dàng giữa front-end và back-end.

Nhược điểm:

Phụ thuộc vào internet: Vì hoạt động chủ yếu trên nền tảng đám mây, hệ thống cần kết nối mạng ổn định.

Giới hạn miễn phí: Firebase có giới hạn dung lượng và lượt truy cập ở gói miễn phí, cần nâng cấp khi hệ thống mở rộng.

Khó quản lý dữ liệu phức tạp: Với MongoDB, việc thiết kế cơ sở dữ liệu yêu cầu lập trình viên hiểu rõ mô hình NoSQL.

Chi phí khi mở rộng: Khi lượng người dùng và dữ liệu tăng cao, chi phí duy trì Firebase và MongoDB Atlas có thể lớn hơn so với server cục bộ.

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG THÔNG TIN

3.1. Giới thiệu

3.1.1. Khảo sát thực tế

Trong những năm gần đây, cùng với sự phát triển mạnh mẽ của thương mại điện tử, nhu cầu mua sắm trực tuyến các sản phẩm công nghệ tại Việt Nam tăng rõ rệt. người tiêu dùng có xu hướng tìm kiếm các lựa chọn các thiết bị thông minh như: điện thoại, laptop, đồng hồ, phụ kiện thông minh,... Thông qua các website hay các trang thương mại điện tử thay vì đến trực tiếp các cửa hàng để lựa chọn sản phẩm theo kiểu truyền thống nữa. Các yếu tố được quan tâm hàng đầu giữa các sản phẩm, giá cả, chi tiết sản phẩm, chất lượng, bảo hành rõ ràng và yếu tố quan trọng là có thể đặt hàng mọi lúc. Tuy nhiên hiện nay có nhiều website thương mại vẫn còn nhiều hạn chế như giao diện chưa tối ưu, tốc độ chậm, thông tin không nhất quán, và đặc biệt thiếu đi tương tác người dùng.

Xuất phát từ thực tế đó, hệ thống website bán hàng công nghệ của em được xây dựng với mục tiêu là:

- Đáp ứng nhu cầu cũng như xu hướng mua sắm trực tuyến hiện đại và thân thiện người dùng.
- Tích hợp bộ lọc sản phẩm theo giá, nhu cầu sử dụng, màu sắc,...
- Hiện thị thông tin về sản phẩm rõ ràng, chất lượng hình ảnh chi tiết.

Cho phép đăng nhập phân quyền giữa người dùng và quản trị viên.

- Hỗ trợ quản lý sản phẩm nhanh chóng: thêm, sửa, xóa, cập nhật,...
- Tạo đề án mở để tương lai có thể cập nhật thêm các tính năng hiện đại.

So với các website truyền thống thì hệ thống của em mang lại các ưu điểm như sau:

- Trải nghiệm mượt mà nhờ sử dụng công nghệ hiện đại và khả năng xử lý linh hoạt.
- Quản lý nội dung dễ dàng dành cho cả người không chuyên kỹ thuật.
- Truy cập và tra cứu sản phẩm với tốc độ nhanh.
- Chatbot tư vấn cho khách hàng về nhu cầu người dùng thay vì có người trực tuyến trên website như truyền thống nữa thay vào đó thì chatbot sẽ tư vấn trực tiếp dành cho khách hàng tham khảo website khi có thắc mắc gì.
- Thanh toán online.
- Gợi ý sản phẩm thông minh, phân tích hành vi khách hàng.

3.1.2. Về khách hàng

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 tìm, người dùng có thể chọn vào sản phẩm để xem thông tin chi tiết của sản phẩm đó, yêu thích sản phẩm tiện lợi cho lần tiếp theo khi vào website để xem sản phẩm.

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 tham khảo web, 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 và 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.

Chatbot: tư vấn cho khách hàng về các loại mà khách hàng muốn mua, về giá cả.

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 địa chỉ nhận hàng, sau đó đi đến thanh toán chọn phương thức thanh toán ‘Thanh toán online, thanh toán khi nhận hàng (COD)’, Sau khi thực hiện xong các bước thì người dùng xác nhận đặt đơn hàng và chờ 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, và có thể hủy đơn khi đơn chưa trong trạng thái đang ship đi.

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

3.1.3. Về phía Quản trị (Administrator)

Quản lý sản phẩm, đơn hàng, khách hàng, một số thông tin khác: Để thực hiện chức năng quản lý thì phải đăng nhập hệ thống bằng gmail riêng biệt hoặc có thể phân hiệu cho người dùng khác để quản lý website. Chọn mục mình cần thực hiện quản lý như sản phẩm. Trong quản lý sản phẩm có xem, thêm, sửa, xóa sản phẩm, cập nhập lại số lượng, giảm giá cho 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 cần có phân quyền tương ứng để thực hiện, nếu không đủ quyền hạn sẽ không thực hiện được.

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

Các trang quản lý cơ bản

- Quản lý người dùng
- Quản lý sản phẩm
- Quản lý danh mục
- Quản lý giá sản phẩm
- Quản lý bài đăng sản phẩm
- Quản lý nhập hàng
- Quản lý đơn hàng

Quản lý thống kê (doanh thu, lợi nhuận, tồn kho)

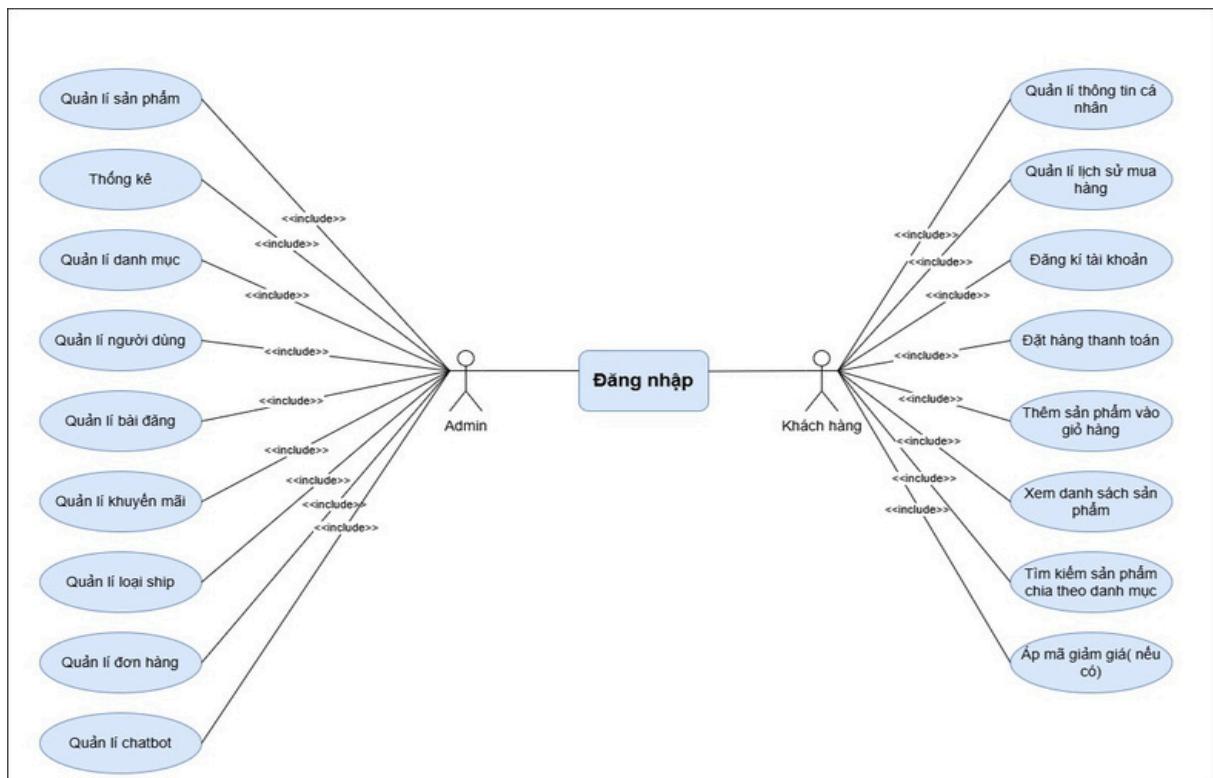
3.1.4. Các yêu cầu phi chức năng

Bảng 3.1. Bảng yêu cầu phi chức năng.

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

3.2. Biểu đồ Usecase.

3.2.1. Biểu đồ Usecase tổng quát.



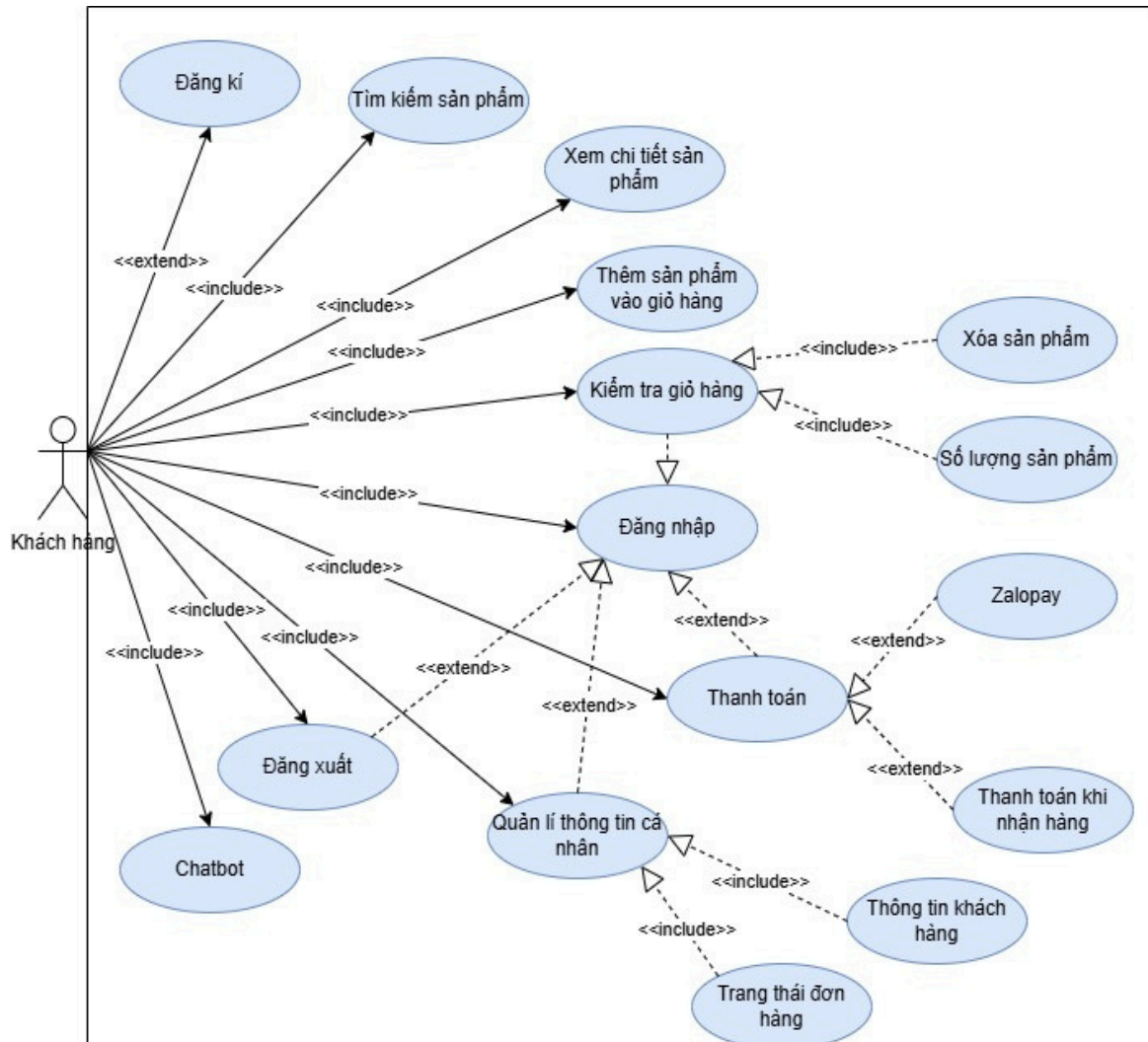
Hình 3.1. Biểu đồ Usecase tổng quát.

Bảng 3.2. Đặc tả Usecase tổng quát

<i>Thuộc tính</i>	<i>Mô tả</i>
Uscase	Sử dụng Hệ thống Bán hàng Điện tử.
Mục tiêu	Thực hiện tất cả các nghiệp vụ chính của một hệ thống bán hàng trực tuyến, bao gồm Quản trị và Giao dịch mua bán.
Tác nhân	Admin, Khách hàng.
Loại	Use Case Ngõu cảnh.
UseCase Chính	- Admin: Quản lý sản phẩm, Quản lý đơn hàng, Quản lý người dùng, Thống kê, v.v. - Khách hàng: Đăng kí tài khoản, Đặt hàng thanh toán, Thêm sản phẩm vào giỏ hàng, v.v.
Điều kiện tiên quyết	Người dùng truy cập vào trang chủ của hệ thống.
Luồng hoạt động	Khách hàng: - Khách hàng thực hiện tìm kiếm, xem danh sách sản phẩm. - Khách hàng chọn và thêm sản phẩm vào giỏ hàng. - Khách hàng thực hiện đăng kí, đăng nhập. - Khách hàng thực hiện Đặt hàng thanh toán Admin: - Admin thực hiện Đăng nhập vào khu vực quản trị. - Admin thực hiện Quản lý sản phẩm, Quản lý đơn hàng, Quản lý người dùng, v.v.
Điều kiện	Khách hàng hoàn tất giao dịch mua hàng. Hoặc Admin hoàn tất các tác vụ quản trị, dữ liệu hệ thống được cập nhập.

3.2.2. Mô hình UseCase phân rã

Khách hàng:



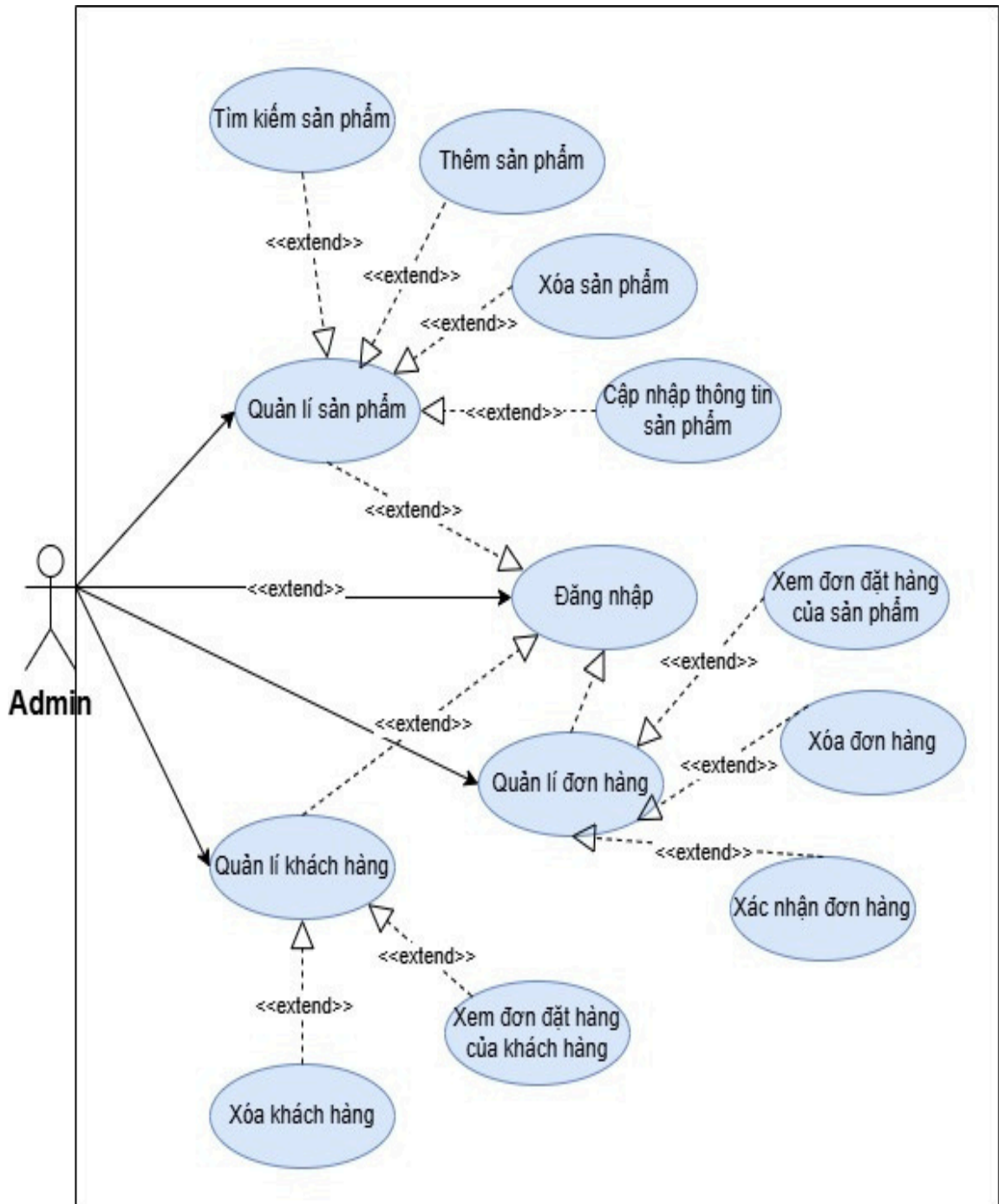
Hình 3.2. Biểu đồ Usecase khách hàng.

Bảng 3.3. Đặc tả Usecase khách hàng.

Thuộc tính	Mô tả
Tác nhân	Khách hàng
Mục đích	Thực hiện hành vi mua sắm, giao dịch trực tuyến, và tương tác với các tính năng cá nhân trên hệ thống.
Mục tiêu	Tìm kiếm sản phẩm, đặt hàng thành công, và quản lý thông tin tài khoản cá nhân

Phạm vi	Giao diện người dùng (Frontend) của hệ thống bán hàng điện tử.
Điều kiện	Khách hàng đã Đăng nhập.

Người quản trị (Admin)



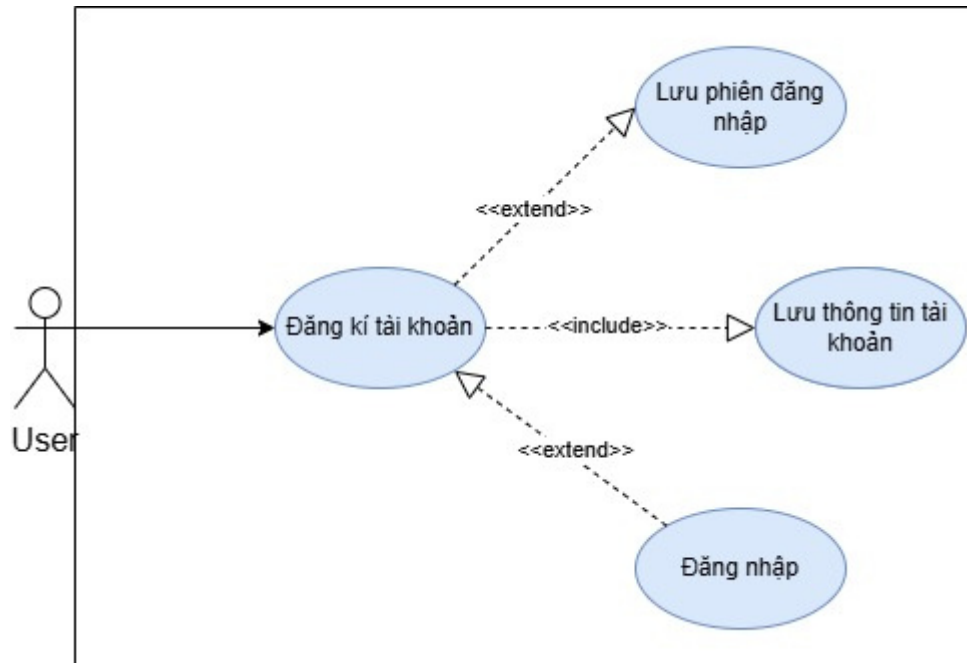
Hình3.3. Biểu đồ Usecase người quản trị.

Bảng 3.4. Đặc tả Usecase người quản trị.

<i>Thuộc tính</i>	<i>Mô tả</i>
Tác nhân	Admin (Người Quản trị).
Mục đích	Quản lý và duy trì hoạt động của hệ thống thương mại điện tử/bán hàng.
Phạm vi	Đảm bảo tính chính xác của dữ liệu, quản lý kho hàng, xử lý đơn hàng, và quản lý người dùng.
Usecase tương tác	Quản lý sản phẩm: Tìm kiếm, Thêm, Xóa, Cập nhật sản phẩm. Đăng nhập: Truy cập an toàn vào khu vực quản trị. Quản lý đơn hàng: Xem, Xác nhận, Xóa đơn hàng, Xem đơn hàng theo sản phẩm. Quản lý khách hàng: Xem đơn hàng theo khách hàng, Xóa khách hàng.
Quyền hạn	Quyền hạn cao nhất trong hệ thống, có thể thực hiện các thao tác thay đổi dữ liệu cốt lõi trên các thực thể chính.
Mô tả chi tiết	Admin là người dùng có tài khoản đặc biệt, được cấp quyền sau khi Đăng nhập thành công, có khả năng điều hành và giám sát toàn bộ các nghiệp vụ kinh doanh của hệ thống.
Điều kiện tiên quyết	Phải có tài khoản Admin được tạo ra trong quá trình khởi tạo hệ thống

3.2.3. Mô tả Usecase chi tiết

3.2.3.1. Usecase đăng kí

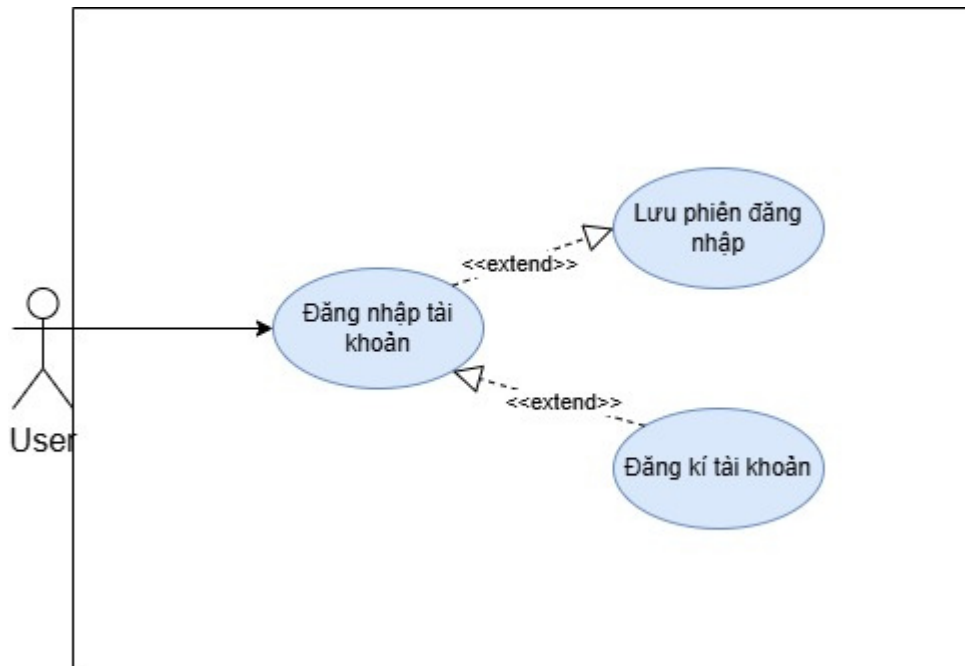


Hình 3.4. Usecase đăng kí.

Bảng 3.5. Đặc tả Usecase đăng kí.

<i>Thuộc tính</i>	<i>Mô tả</i>
Usecase	Đăng kí tài khoản.
Mục tiêu	Cho phép người dùng tạo một tài khoản mới trên hệ thống để sử dụng các chức năng cá nhân và giao dịch
Tác nhân	User
Loại	Use Case Chính
Điều kiện sau	Đăng nhập thành công chuyển hướng tới trang chủ và thực hiện các chức năng.
Yêu cầu đặc biệt	Không có.

3.2.3.2. Usecase đăng nhập

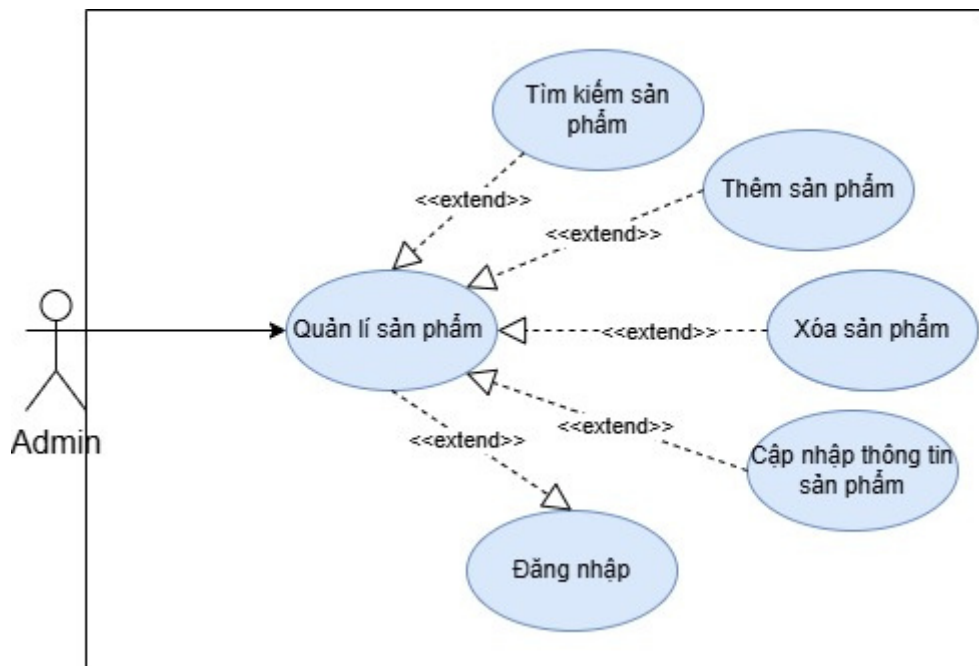


Hình 3.5. Usecase đăng nhập.

Bảng 3.6. Đặc tả Usecase đăng nhập.

Thuộc tính	Mô tả
Usecase	Đăng nhập.
Mục tiêu	Người dùng đăng nhập vào hệ thống để thực hiện các tác vụ.
Điều kiện trước	Phải đăng kí thành công tài khoản của hệ thống được lưu trong database.
Luồng chính	- Hiển thị form đăng nhập. - Nhập thông tin. - Đăng nhập. - Kiểm tra thông tin. - Lưu phiên đăng nhập.
Điều kiện sau	Đăng nhập thành công thì chuyển hướng tới trang chủ, cho phép thực hiện các tác vụ có yêu cầu đăng nhập.
Yêu cầu đặc biệt	Không có.

3.2.3.3. Usecase quản lí sản phẩm



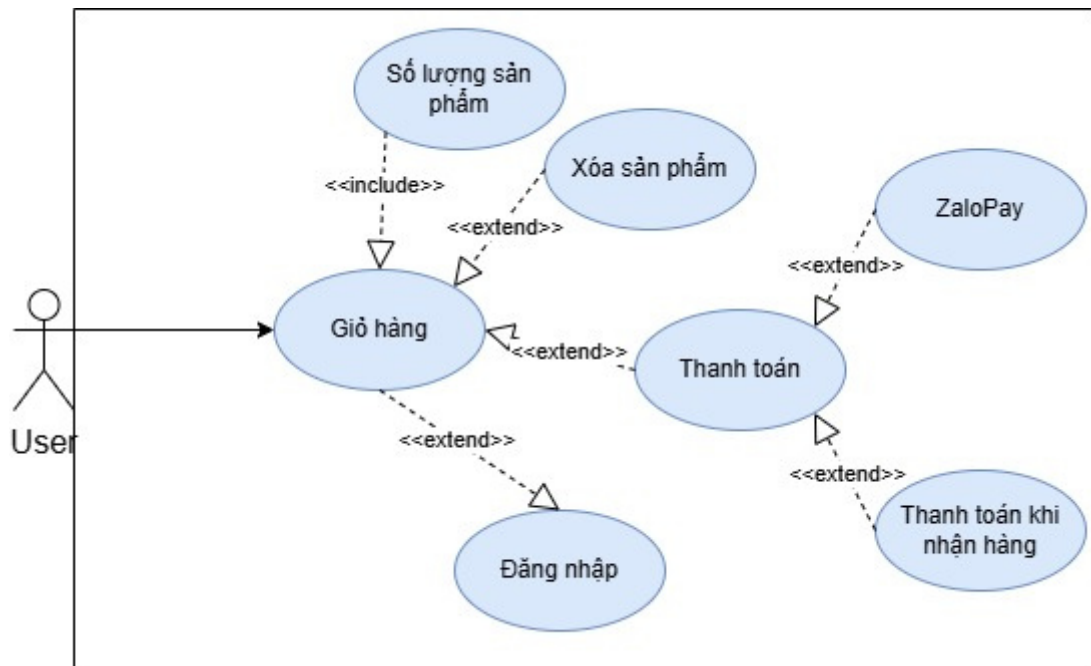
Hình 3.6. Usecase quản lí sản phẩm.

Bảng 3.7. Đặc tả Usecase quản lí sản phẩm.

Thuộc tính	Mô tả
Use Case	Quản lí sản phẩm.
Mục tiêu	Cho phép Admin duy trì và kiểm soát toàn bộ danh mục sản phẩm của hệ thống (Thêm, Sửa, Xóa).
Tác nhân	Admin
Loại	Use Case Bao gồm
Các Use Case Mở rộng	Tìm kiếm sản phẩm, Thêm sản phẩm, Xóa sản phẩm, Cập nhật thông tin sản phẩm (Sửa).
Điều kiện trước	Admin đã Đăng nhập thành công vào hệ thống.
Điều kiện sau	Dữ liệu sản phẩm trên hệ thống được cập nhật chính xác.
Luồng hoạt động	1. Thêm sản phẩm - Admin nhập thông tin sản phẩm vào form. - Hệ thống xác thực và tạo mới sản phẩm trong CSDL.

	2. Sửa/ cập nhập sản phẩm - Admin chọn sản phẩm và chỉnh sửa thông tin. - Hệ thống xác thực và cập nhật dữ liệu sản phẩm trong CSDL. 3. Xóa sản phẩm - Admin chọn sản phẩm và xác nhận xóa. - Hệ thống xóa sản phẩm khỏi CSDL.
Luồng Ngoại lệ	Dữ liệu không hợp lệ: Thông tin nhập vào thiếu hoặc sai định dạng (ví dụ: giá không phải là số). Hệ thống hiển thị lỗi và yêu cầu sửa lại.
Yêu cầu đặc biệt	Không có.

3.2.3.4. Usecase giỏ hàng



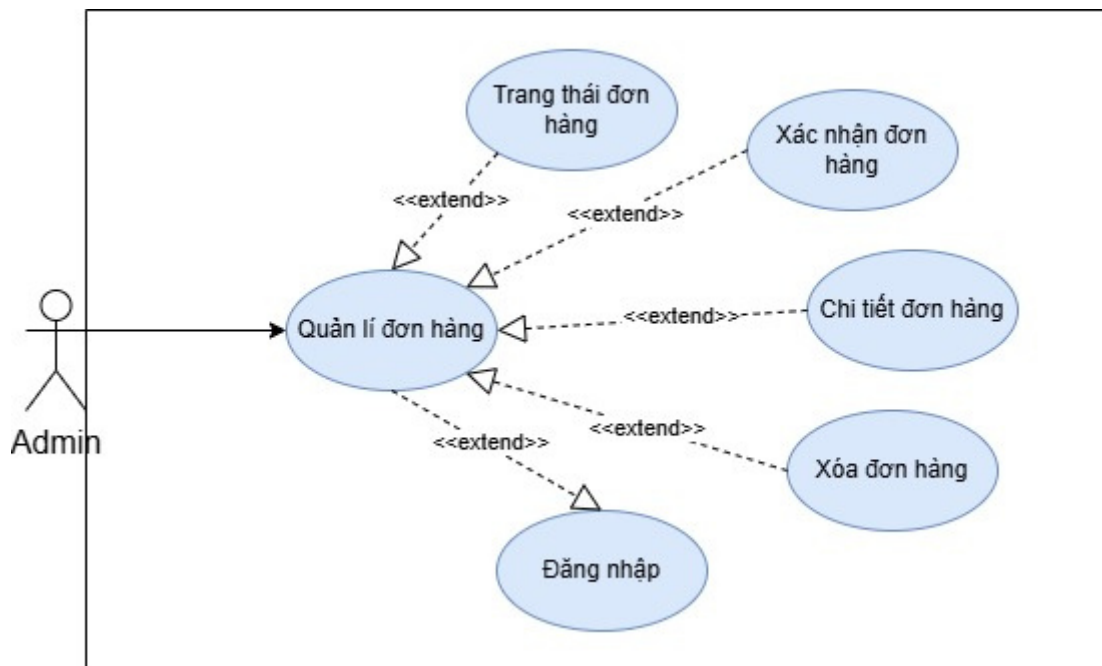
Hình 3.7. Usecase giỏ hàng.

Bảng 3.8. Đặc tả Usecase giỏ hàng.

Thuộc tính	Mô tả
Usecase	Giỏ hàng

Mục tiêu	Cho phép người dùng (Khách hàng) xem lại các sản phẩm đã chọn, điều chỉnh số lượng, xóa sản phẩm và bắt đầu quy trình Thanh toán.
Tác nhân	Khách hàng (User).
Loại	Use Case Chính/Bao gồm (Base Use Case)
Các Use Case Liên quan	Xóa sản phẩm, Số lượng sản phẩm. Thanh toán, Đăng nhập.
Điều kiện trước	Người dùng đã thêm ít nhất một sản phẩm vào giỏ hàng.
Điều kiện sau	Giỏ hàng được cập nhật, hoặc quy trình Thanh toán được khởi tạo.
Luồng	- Người dùng truy cập vào trang Giỏ hàng. - Hệ thống hiển thị danh sách các sản phẩm đang có, bao gồm tên sản phẩm, giá, số lượng, và tổng tiền tạm tính. - Người dùng thực hiện các thao tác điều chỉnh: Cập nhật số lượng: Người dùng thay đổi số lượng và hệ thống thực hiện Số lượng sản phẩm, Xóa sản phẩm: Người dùng chọn xóa một sản phẩm và hệ thống thực hiện Xóa sản phẩm. - Người dùng nhấn nút "Tiến hành Thanh toán". - Hệ thống kiểm tra điều kiện (ví dụ: yêu cầu Đăng nhập nếu chưa đăng nhập) và chuyển sang Use Case Thanh toán.
Luồng mở rộng	Đăng nhập: Nếu người dùng chưa đăng nhập, hệ thống yêu cầu thực hiện Đăng nhập trước khi tiến hành Thanh toán. Sau khi đăng nhập xong, hệ thống tiếp tục Use Case Giỏ hàng/Thanh toán.
Luồng ngoại lệ	Số lượng không khả dụng: Người dùng yêu cầu số lượng sản phẩm lớn hơn số lượng tồn kho. Hệ thống hiển thị cảnh báo và giới hạn số lượng tối đa.
Yêu cầu đặc biệt	Không có.

3.2.3.5. Usecase quản lí đơn hàng



Hình 3.8. Usecase quản lí đơn hàng.

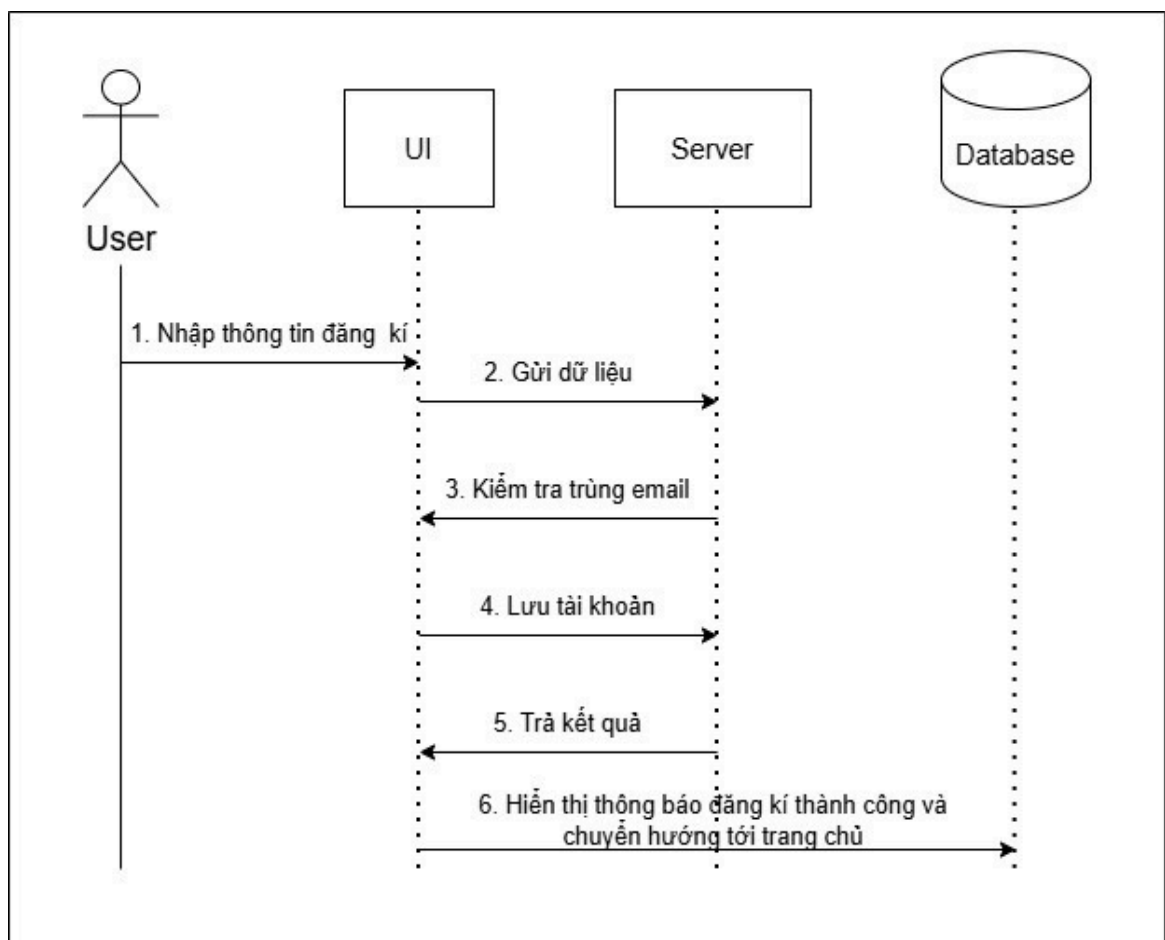
Thuộc tính	Mô tả
Usecase	Quản lí đơn hàng.
Mục tiêu	Cho phép Admin xem, theo dõi, xử lý và duy trì toàn bộ các đơn đặt hàng trong hệ thống
Tác nhân	Admin
Loại	Use Case Bao gồm (Base/Abstract)
Use Case Mở rộng	Xác nhận đơn hàng, Chi tiết đơn hàng, Xóa đơn hàng, Trạng thái đơn hàng, Đăng nhập.
Điều kiện trước	Admin đã Đăng nhập thành công vào hệ thống.
Điều kiện sau	Đơn hàng được xử lý, trạng thái được cập nhật chính xác.
Luồng	- Admin truy cập vào chức năng Quản lí đơn hàng. - Hệ thống hiển thị danh sách tất cả đơn hàng, kèm trạng thái và thông tin cơ bản.

	- Admin có thể thực hiện các tác vụ sau: xem chi tiết đơn hàng, hệ thống cần hiển thị (sản phẩm, tổng tiền, thông tin khách hàng). Xác nhận đơn hàng mới, hệ thống thực hiện xác nhận đơn hàng. Xóa đơn hàng hệ thống thực hiện xoas đơn hàng sau khi xác nhận. Theo dõi đơn hàng xem trạng thái đơn hàng đang ở giao đoạn nào (đã xác nhận, đang giao, đã giao)
Luồng ngoại lệ	Đơn hàng không tồn tại: Admin cố gắng xem chi tiết một đơn hàng đã bị xóa hoặc không tìm thấy. Hệ thống hiển thị thông báo lỗi.
Yêu cầu đặc biệt	Không có.

3.3. Biểu đồ tuần tự

3.3.1. Biểu đồ tuần tự của User

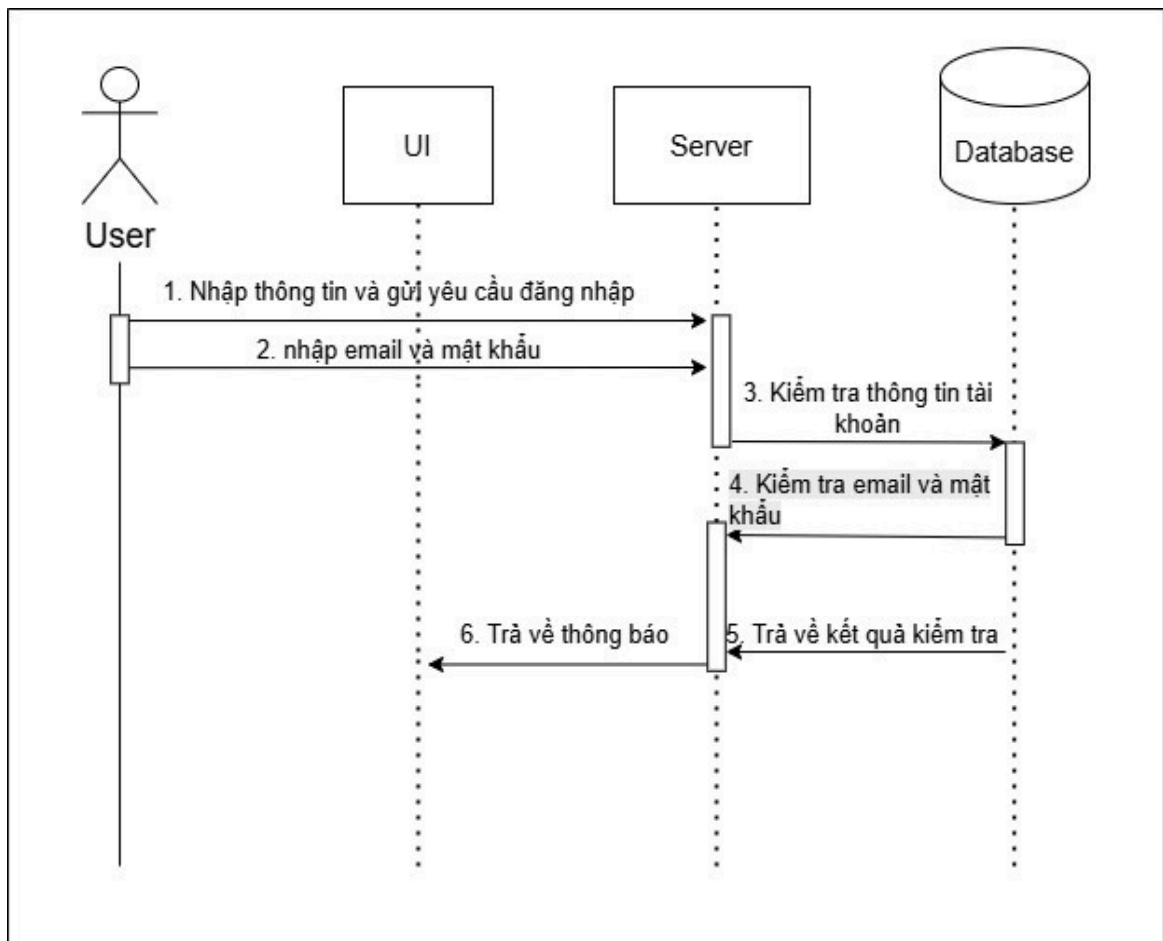
Biểu đồ tuần tự đăng kí:



Hình 3.9. Biểu đồ tuần tự đăng kí.

- (1) Người dùng đăng kí thông tin.
- (2) Giao diện người dùng UI gửi dữ liệu đăng kí đến back-end.
- (3) Server truy vấn cơ sở dữ liệu (MongoDB) để đảm bảo email đăng ký chưa được sử dụng.
- (4) Nếu email hợp lệ, Server mã hóa mật khẩu và lưu thông tin tài khoản.
- (5) Server trả về thông báo xác nhận thành công.
- (6) Giao diện người dùng đang hiển thị thông báo Đăng ký thành công và chuyển hướng người dùng tới trủ.

Biểu đồ tuần tự đăng nhập:

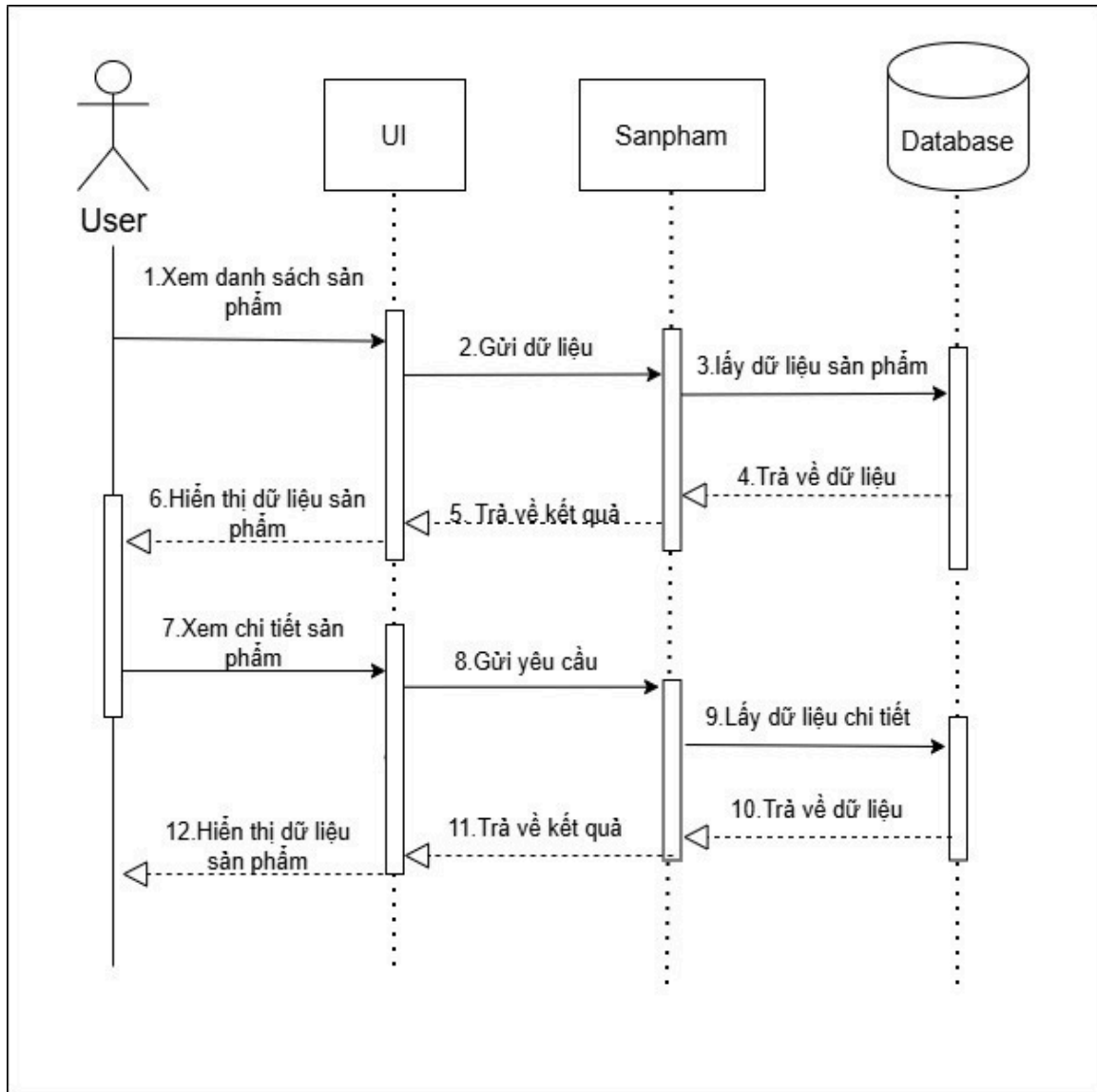


Hình 3.10. Biểu đồ tuần tự đăng nhập.

- (1) Người dùng mở trang đăng nhập gửi yêu cầu đăng nhập.
- (2) Người dùng nhập email và mật khẩu.
- (3) Giao diện người dùng (UI) gửi dữ liệu đến Back-end (Server). Server nhận yêu cầu và bắt đầu quá trình kiểm tra.
- (4) Server truy vấn cơ sở dữ liệu (Database) để kiểm tra tính hợp lệ của tài khoản.
- (5) Database trả về kết quả cho Server.

(6) Server xử lý kết quả: (Thành công) thì Giao diện người dùng hiển thị thông báo và chuyển hướng người dùng đến trang chủ/Dashboard. Tạo và gửi về token/session đăng nhập. (Thất bại): Gửi thông báo lỗi (ví dụ: "Tài khoản hoặc mật khẩu không đúng").

Biểu đồ tuần tự xem sản phẩm:

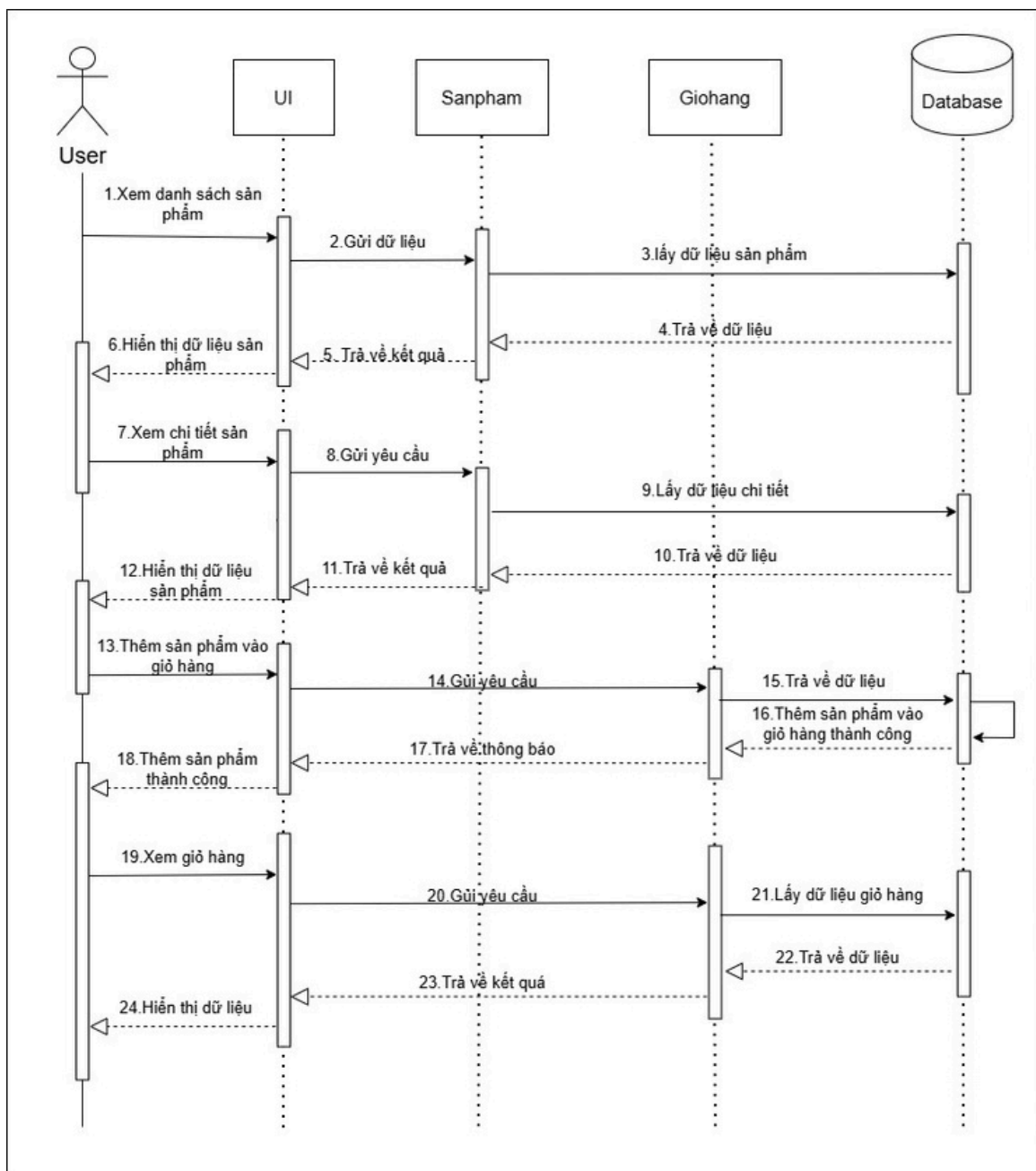


Hình 3.11. Biểu đồ tuần tự xem sản phẩm.

- (1) Người dùng truy cập vào trang danh sách sản phẩm.
- (2) Giao diện (UI) gửi yêu cầu lấy danh sách sản phẩm đến Server.
- (3) Server truy vấn Database (MongoDB) để lấy danh sách sản phẩm.
- (4) Database trả về danh sách sản phẩm cho Server.
- (5) Server gửi dữ liệu danh sách sản phẩm đến giao diện.
- (6) Giao diện hiển thị danh sách sản phẩm cho người dùng.

- (7) Người dùng chọn một sản phẩm từ danh sách và nhấp vào để xem chi tiết.
- (8) Giao diện gửi yêu cầu lấy chi tiết sản phẩm cụ thể đến Server, kèm theo ID sản phẩm..
- (9) Server truy vấn Database để lấy thông tin chi tiết của sản phẩm đó.
- (10) Database trả về dữ liệu chi tiết sản phẩm cho Server.
- (11) Server gửi dữ liệu chi tiết sản phẩm đến giao diện.
- (12) Giao diện hiển thị trang chi tiết sản phẩm cho người dùng.

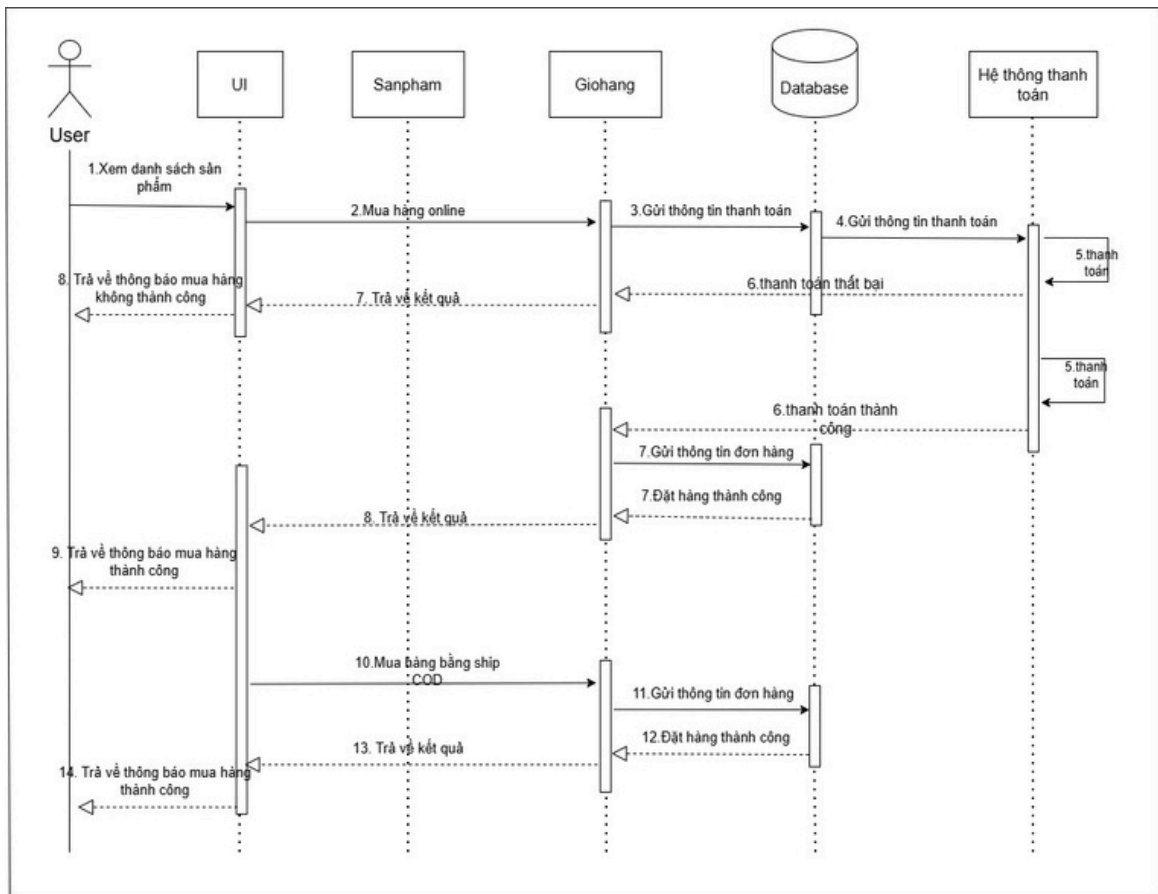
Biểu đồ tuần tự thêm sản phẩm vào giỏ hàng:



Hình 3.12. Biểu đồ tuần tự thêm sản phẩm vào giỏ hàng.

- (1) Người dùng đã ở trang chi tiết sản phẩm.
- (2) Giao diện (UI) gửi yêu cầu và dữ liệu sản phẩm đến module.
- (3) Module Giỏ hàng lưu hoặc cập nhật thông tin sản phẩm vào giỏ hàng của Khách hàng trong Database.
- (4) Database trả về xác nhận thành công (hoặc lỗi) cho module Giỏ hàng.
- (5) Module Giỏ hàng gửi phản hồi về trạng thái thực hiện cho giao diện người dùng.
- (6) Giao diện hiển thị thông báo xác nhận cho người dùng.
- (7) Người dùng nhấp vào một sản phẩm để xem thông tin chi tiết.
- (8) Giao diện gửi yêu cầu lấy chi tiết sản phẩm (kèm ID) đến Server.
- (9) Server truy vấn Database để lấy toàn bộ thông tin chi tiết của sản phẩm đó.
- (10) Database trả về dữ liệu chi tiết sản phẩm.
- (11) Server gửi dữ liệu chi tiết sản phẩm đến giao diện.
- (12) Giao diện hiển thị trang chi tiết sản phẩm cho người dùng.
- (13) Người dùng nhấn nút "Thêm sản phẩm vào giỏ hàng".
- (14) Giao diện gửi yêu cầu đến module.
- (15) Database trả về xác nhận/dữ liệu cho module.
- (16) Module Giỏ hàng thực hiện lưu/cập nhật thông tin vào Database.
- (17) Module Giỏ hàng gửi phản hồi về trạng thái thực hiện cho giao diện.
- (18) Giao diện hiển thị thông báo xác nhận thành công cho người dùng.
- (19) Người dùng nhấp vào biểu tượng giỏ hàng hoặc liên kết Giỏ hàng.
- (20) Giao diện gửi yêu cầu lấy thông tin giỏ hàng đến module.
- (21) Module Giỏ hàng truy vấn Database để lấy tất cả các mặt hàng đã được lưu.
- (22) Database trả về danh sách sản phẩm, số lượng, tổng tiền trong giỏ hàng.
- (23) Module Giỏ hàng gửi dữ liệu về giao diện người dùng.
- (24) Giao diện hiển thị đầy đủ chi tiết giỏ hàng cho người dùng.

Biểu đồ tuần tự thanh toán:



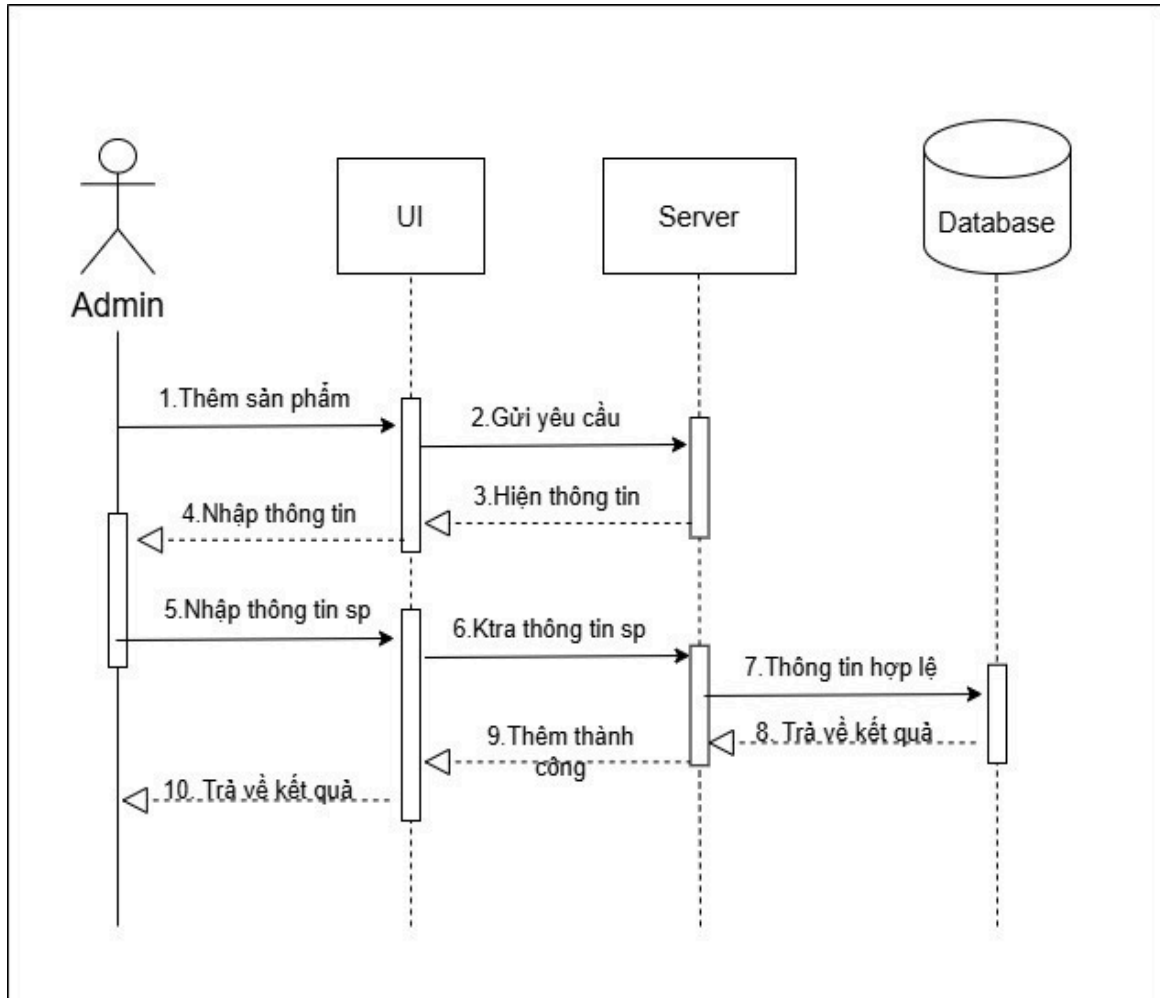
Hình 3.13 Biểu đồ tuần tự thanh toán.

- (1) Người dùng chọn phương thức thanh toán online.
- (2) Giao diện gửi dữ liệu giỏ hàng, thông tin vận chuyển đến Server.
- (3) Server gửi thông tin đơn hàng đến module.
- (4) Tương tác với Hệ thống thanh toán bên ngoài.
- (5) Hệ thống thanh toán xử lý giao dịch của người dùng.
- (6) Hệ thống thanh toán trả kết quả về module.
- (7) xử lý kết quả: (Thành công): Ghi nhận đơn hàng vào Database, gửi thông tin đặt hàng thành công. (Thất bại): Gửi thông báo thanh toán thất bại.
- (8) Giao diện hiển thị kết quả cuối cùng cho người dùng.
- (9) Người dùng chọn phương thức Thanh toán khi nhận hàng (COD) và nhấn nút Đặt hàng.
- (10) Giao diện gửi dữ liệu giỏ hàng, thông tin vận chuyển đến Server.
- (11) Server gửi thông tin đơn hàng đến module.
- (12) Ghi nhận đơn hàng (Trạng thái chờ xác nhận) vào Database.
- (13) Gửi xác nhận đặt hàng thành công về giao diện.

(14) Giao diện hiển thị thông báo xác nhận đã đặt hàng thành công.

3.3.2. Biểu đồ tuần tự của Admin

Biểu đồ tuần tự thêm sản phẩm:

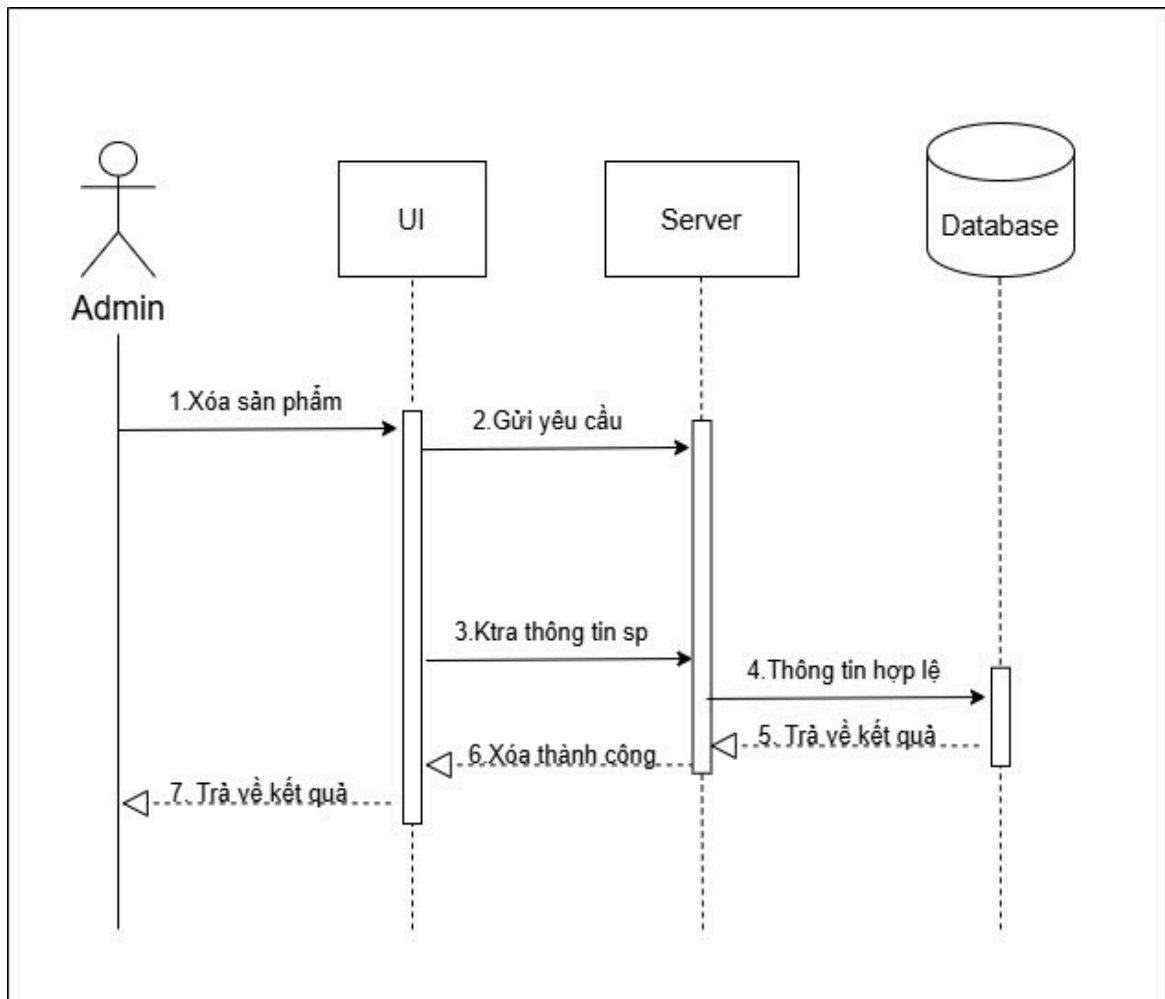


Hình 3.14. Biểu đồ tuần tự thêm sản phẩm.

- (1) Admin truy cập vào màn hình Quản lý sản phẩm.
- (2) Giao diện gửi yêu cầu mở form/màn hình thêm sản phẩm đến Server.
- (3) Server trả về giao diện form để Admin nhập thông tin sản phẩm.
- (4) Admin nhập thông tin chi tiết của sản phẩm.
- (5) Admin nhấn nút "Lưu" hoặc "Thêm" để gửi dữ liệu sản phẩm đã nhập lên Server.
- (6) Giao diện gửi dữ liệu đến Server. Server bắt đầu quá trình kiểm tra tính hợp lệ của dữ liệu.
- (7) Nếu dữ liệu hợp lệ, Server gửi lệnh lưu thông tin sản phẩm mới vào Database.
- (8) Database trả về xác nhận đã lưu thành công hoặc báo lỗi.
- (9) Server gửi thông báo xác nhận đã thêm sản phẩm thành công đến giao diện.

(10) Giao diện hiển thị thông báo "Thêm sản phẩm thành công" và có thể chuyển hướng Admin về trang danh sách sản phẩm đã cập nhật.

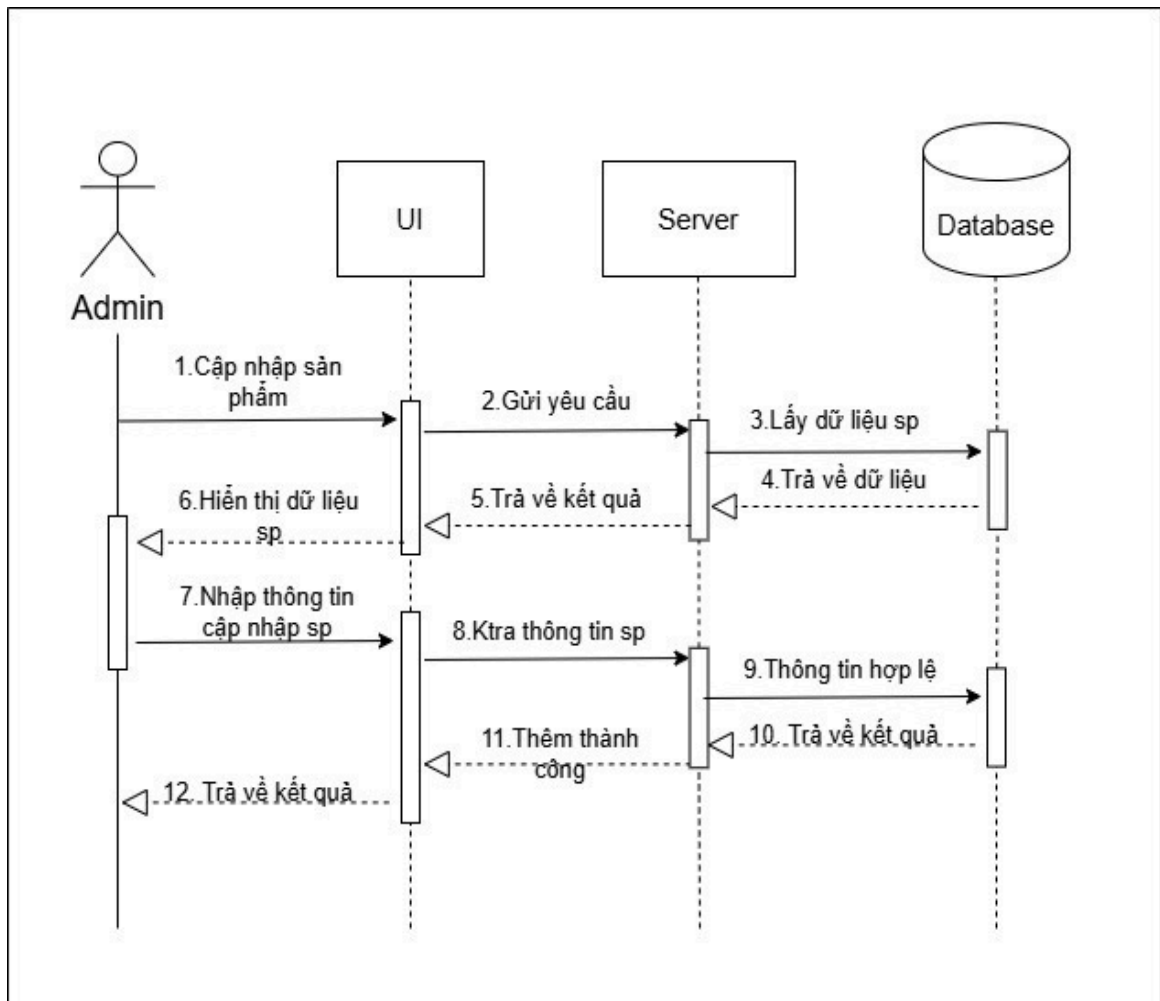
Biểu đồ tuần tự xóa sản phẩm:



Hình 3.15. Biểu đồ tuần tự xóa sản phẩm.

- (1) Admin truy cập vào màn hình Quản lý sản phẩm, chọn sản phẩm cần xóa và nhấn nút "Xóa".
- (2) Giao diện gửi yêu cầu xóa sản phẩm (kèm ID sản phẩm) đến Server.
- (3) Server kiểm tra tình trạng hợp lệ của yêu cầu xóa.
- (4) Nếu yêu cầu hợp lệ, Server gửi lệnh xóa sản phẩm khỏi Database.
- (5) Database trả về xác nhận đã xóa thành công hoặc báo lỗi.
- (6) Server gửi thông báo xác nhận đã xóa sản phẩm thành công đến giao diện.
- (7) Giao diện hiển thị thông báo "Xóa sản phẩm thành công" và cập nhật lại danh sách sản phẩm.

Biểu đồ tuần tự cập nhập sản phẩm:



Hình 3.16 Biểu đồ tuần tự cập nhập sản phẩm.

- (1) Admin truy cập màn hình Quản lý sản phẩm, chọn sản phẩm cần chỉnh sửa, và nhấn nút "Cập nhật" hoặc "Chỉnh sửa".
- (2) Giao diện gửi yêu cầu lấy thông tin sản phẩm hiện tại.
- (3) Giao diện gửi yêu cầu lấy thông tin sản phẩm hiện tại.
- (4) Database trả về dữ liệu sản phẩm hiện tại cho Server.
- (5) Server gửi dữ liệu sản phẩm hiện tại đến giao diện để điền vào form chỉnh sửa.
- (6) Giao diện hiển thị form chỉnh sửa đã được điền sẵn thông tin sản phẩm.
- (7) Admin tiến hành thay đổi các trường thông tin cần thiết trên form.
- (8) Giao diện gửi dữ liệu đã chỉnh sửa lên Server. Server bắt đầu quá trình kiểm tra tính hợp lệ của dữ liệu mới.
- (9) Nếu dữ liệu hợp lệ, Server gửi lệnh cập nhật thông tin sản phẩm trong Database.
- (10) Database trả về xác nhận đã cập nhật thành công hoặc báo lỗi.
- (11) Server gửi thông báo xác nhận đã cập nhật thành công đến giao diện

(12) Giao diện hiển thị thông báo "Cập nhật thành công" và có thể chuyển Admin về trang danh sách sản phẩm.

3.4. Thiết kế cơ sở dữ liệu

3.4.1. Quan hệ giữa các collection.

Bảng 3.9. Bảng quan hệ collection products.

<i>Collection chính</i>	<i>Quan hệ</i>	<i>Collection liên kết</i>
products	1 – n	users
products	n - n	catalogs
products	n - n	tags
products	n - n	colors
products	n - n	sizes

- User: Mỗi người dùng (user) có thể đăng nhiều sản phẩm. Trường uid trong product đóng vai trò là khóa tham chiếu
- Catalogs: Một sản phẩm có thể thuộc nhiều danh mục. Một danh mục cũng có thể chứa nhiều sản phẩm.
- Tags: Một sản phẩm có thể có nhiều thẻ tag để hỗ trợ tìm kiếm. Một tag cũng có thể được dùng cho nhiều sản phẩm khác
- Colors: Một sản phẩm có thể có nhiều tùy chọn màu sắc. Mỗi màu sắc có thể được dùng bởi nhiều sản phẩm khác.
- Sizes: Một sản phẩm có thể có nhiều kích cỡ. Kích cỡ cũng có thể dùng cho nhiều loại sản phẩm khác.

Bảng 3.10. Bảng quan hệ collection cart.

<i>Collection chính</i>	<i>Quan hệ</i>	<i>Collection liên kết</i>
cart	1 – n	users
cart	n - n	products
cart	(phụ thuộc)	images (nằm trong product)

- Users: Một user có thể có nhiều món hàng trong giỏ → userId là khóa tham chiếu đến users
- Products: Mỗi item trong giỏ chỉ liên kết tới 1 sản phẩm duy nhất → productId trở tới products.
- Images: Ảnh sản phẩm lấy từ sản phẩm, không tạo quan hệ riêng.

Bảng 3.11. Bảng quan hệ collection catalogs.

<i>Collection chính</i>	<i>Quan hệ</i>	<i>Collection liên kết</i>
catalogs	n - n	products

- Products: Một catalog (danh mục) có thể chứa nhiều sản phẩm, và một sản phẩm có thể thuộc nhiều catalog.

Bảng 3.12. Bảng quan hệ collection notifications.

<i>Collection chính</i>	<i>Quan hệ</i>	<i>Collection liên kết</i>
notifications	1 – n	users
notifications	1 - 1	vouchers
notifications	1 - 1	orders
notifications	1 - 1	products

- Users: Một user có thể nhận nhiều thông báo. userId là khóa tham chiếu.
- Vouchers: Nếu thông báo liên quan voucher, voucherId sẽ trở tới voucher đó.
- Orders: Nếu thông báo liên quan đơn hàng, orderId sẽ trở tới đơn hàng.
- Products: Nếu thông báo liên quan sản phẩm, productId sẽ tham chiếu đến sản phẩm

Bảng 3.13. Bảng quan hệ collection user_voucher

<i>Collection chính</i>	<i>Quan hệ</i>	<i>Collection liên kết</i>
user_voucher	1 - n	users
user_voucher	1 - 1	vouchers
user_voucher	1 - 1	orders

- Users: Một user có thể sở hữu nhiều voucher.
- Vouchers: Mỗi bản ghi user_voucher liên kết với 1 voucher gốc (template).
- Orders: Nếu user dùng voucher để thanh toán, nó sẽ liên kết đến 1 order (qua orderId — nếu bạn có).

Bảng 3.14. Bảng quan hệ collection users.

<i>Collection chính</i>	<i>Quan hệ</i>	<i>Collection liên kết</i>
users	1 - n	products
users	1 - n	cart
users	1 - n	notifications
users	1 - n	user_voucher
users	1 - n	orders
users	1 - n	reviews
users	1 - n	wishlists

- Products: Một user có thể đăng nhiều sản phẩm (qua uid).
- Cart: Một user có nhiều sản phẩm trong giỏ (qua userId).
- Notifications: Một user có nhiều thông báo gửi đến.
- User_voucher: Một user có thể sở hữu nhiều voucher
- Orders: Một user có thể tạo nhiều đơn hàng.
- Reviews: Một user có thể viết nhiều đánh giá sản phẩm.
- Wishlists: Một user có thể lưu nhiều sản phẩm yêu thích.

Bảng 3.15 Bảng quan hệ collection vouchers.

<i>Collection chính</i>	<i>Quan hệ</i>	<i>Collection liên kết</i>
vouchers	1 - n	user_voucher
vouchers	1 - n	notifications

- User_voucher: Một voucher gốc có thể được cấp phát cho nhiều user.
- Notifications: Các thông báo liên quan voucher (hết hạn, dùng được...) được liên kết qua voucherId.

Bảng 3.16 Bảng quan hệ collection orders.

<i>Collection chính</i>	<i>Quan hệ</i>	<i>Collection liên kết</i>
orders	1 - n	order_items
orders	n - 1	users
orders	1 - 1	user_voucher
orders	1 - n	notifications

- order_items: Một đơn hàng có nhiều sản phẩm chi tiết.
- users: Một user có thể có nhiều đơn hàng.
- user_voucher: Mỗi đơn hàng có thể gắn 1 voucher đã claim.
- notifications: Mỗi đơn hàng có thể sinh nhiều thông báo (trạng thái, giao hàng, hủy...).

<i>Collection chính</i>	<i>Quan hệ</i>	<i>Collection liên kết</i>
User_activity	n - 1	users
User_activity	n - 1	products

- users: Một user có nhiều hành động.
- products: Một sản phẩm có nhiều lượt xem, lượt click.

3.4.2. Cấu trúc các collection thực thể trong Website

Collection User:

```

createdAt: October 18, 2025 at 4:39:38 PM UTC+7
email: "hungnguyenhai50@gmail.com"
emailVerified: true
name: "Hung Nguyễn hải"
uid: "6d2Ylq35UUZ3Vs8WBQjEI7uUjdF2"

```

Hình 3.17. Minh họa collection User

Bảng 3.17. Mô tả collection AdressUser.

<i>Thuộc tính</i>	<i>Kiểu dữ liệu</i>	<i>Mô tả</i>
uid	string	Mã định danh duy nhất của người dùng trong Firebase Authentication. Không trùng lặp.
createdAt	timestamp	Thời điểm tài khoản được tạo.
email	string	Địa chỉ email người dùng dùng để đăng ký / đăng nhập.
emailVerified	boolean	Trạng thái email đã được xác minh chưa: true = đã xác minh, false = chưa xác minh
name	string	Tên hiển thị của người dùng.

□ **Collection Products:**

```

    > catalogs: ["Điện thoại"]
    > colors: ["Trắng"]
    createdAt: October 5, 2025 at 8:43:47 PM UTC+7
    description: "Máy mới 100% Chính hãng Apple Việt Nam"
    > images: ["data:image/webp;base64,U..."]
    name: "Iphone 16 Pro Max"
    price: 30590000
    quantity: 2
    > sizes: ["356Gb"]
    status: "approved"
    > tags: [] (array) +
    uid: "qVvzvUfbqCMfv7cCo49jaeQG6V32"
  
```

Hình 3.18. Minh họa collection Products

Bảng 3.18. Mô tả collection Products.

<i>Thuộc tính</i>	<i>Kiểu dữ liệu</i>	<i>Mô tả</i>
catalogs	array(string)	Danh sách các danh mục sản phẩm thuộc về.
colors	array(string)	Các màu có sẵn của sản phẩm
createdAt	timestamp	Thời điểm tạo sản phẩm.
thumbnail	string	URL
description	string	Mô tả chi tiết sản phẩm.
images	array(string)	Base64 hình ảnh sản phẩm.
name	string	Tên sản phẩm.
price	number	Giá bán sản phẩm.
quantity	number	Số lượng có trong kho.
sizes	array(string)	Các phiên bản dung lượng.
status	string	Trạng thái duyệt sản phẩm trong hệ thống.

□ **Collection Catalogs:**

```
createdAt: September 26, 2025 at 11:45:19 PM UTC+7
name: "Phụ kiện"
```

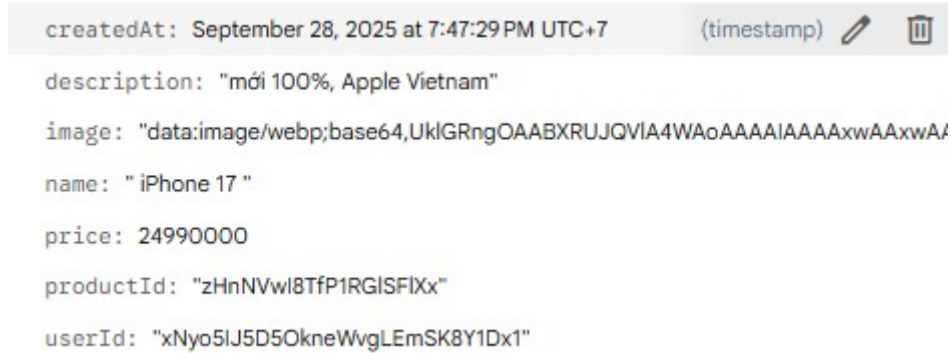
Hình 3.19. Minh họa collection catalogs.

Bảng 3.19. Mô tả collection catalogs.

<i>Thuộc tính</i>	<i>Kiểu dữ liệu</i>	<i>Mô tả</i>
name	string	Tên danh mục.

createdAt	timestamp	Thời điểm tạo danh mục.
-----------	-----------	-------------------------

□ **Collection Cart:**



Hình 3.20. Minh họa collection Cart.

Bảng 3.20. Mô tả collection Cart

Thuộc tính	Kiểu dữ liệu	Mô tả
createdAt	timestamp	Thời điểm tạo sản phẩm
description	string	Mô tả sản phẩm.
image	String (base64)	Ảnh sản phẩm dạng Base64.
name	string	Tên sản phẩm.
price	number	Giá sản phẩm.
productId	string	ID sản phẩm.
userId	string	ID người đăng sản phẩm.

□ **Collection Vouchers:**

```

autoDistribute: false
claimedCount: 2
code: "SINHVIEN"
createdAt: "2025-11-22T14:44:46.381Z"
discount: 10
discountType: "percentage"
endDate: "2025-11-30"
isActive: true
occasion: "Sinh viên "
startDate: "2025-11-22"
targetGender: ""
totalLimit: 2
    
```

Hình 3.21. minh họa collection Voucher.

Bảng 3.21. Mô tả collection Voucher.

<i>Thuộc tính</i>	<i>Kiểu dữ liệu</i>	<i>Mô tả</i>
autoDistribute	boolean	Tự phát cho người dùng.
claimedCount	number	Uniquesố lượng voucher mà người dùng đã lấy.
code	string	Mã giảm giá.
createdAt	string	Thời gian được tạo.
discount	number	Giá trị giảm.
discountType	string	Loại giảm giá, theo phần trăm , theo số tiền.
endDate	string	Ngày kết thúc.
isActive	boolean	Trạng thái mã còn hoạt động không.
occasion	string	Dịp áp dụng.

startDate	string	Ngày bắt đầu
targetGender	string	Muốn phát theo giới tính.
totalLimit	number	Số lượng sử dụng cho phép.
validHourt	number	Thời gian hiệu lực tính từ lúc nhận mã.

□ **Collection UserVouchers:**

```

claimedAt: "2025-11-22T14:45:53.803Z"
code: "SINHVIEN"
discount: 10
discountType: "percentage"
expiresAt: "2025-11-23T14:45:53.803Z"
isExpired: false
isUsed: false
occasion: "Sinh viên "
userId: "rGJMXA7OCIWP5WOaZrMUMc1PVBG3"
voucherId: "7hpqlkw5pW5XgfwvXTUL"

```

Hình 3.22. minh họa collection User Voucher.

Bảng 3.22. Mô tả collection UserVoucher.

<i>Thuộc tính</i>	<i>Kiểu dữ liệu</i>	<i>Mô tả</i>
claimedAt	string	Thời điểm người dùng nhận voucher.
code	string	Mã voucher.
discount	number	Loại giảm.
discountType	string	Thời điểm hết hạn voucher đối với người dùng.

expiresAt	string	Voucher của người dùng đã hết hạn hay chưa.
isExpired	boolean	Voucher đã được sử dụng hay chưa.
isUsed	boolean	Dịp chương trình.
occasion	string	ID người nhận voucher.
userId	string	ID voucher.
voucherId	string	default false

□ **Collection Notifications:**

```

additionalInfo: "🎁 Xem thêm voucher mới tại trang "Voucher của tôi"."
category: "voucher"
createdAt: November 25, 2025 at 6:22:45 PM UTC+7
deleted: false
isRead: false
message: "Voucher WOMEN83 đã hết hạn sử dụng."
orderId: null
productId: null
title: "🔴 Voucher đã hết hạn"
type: "voucher_expired"
userId: "wT7k0eydhDeZC3DiRhncUIFTPv1"
voucherCode: "WOMEN83"

```

Hình 3.23. Minh họa collection Notifications

Bảng 3.23. Mô tả collection Notifications.

Thuộc tính	Kiểu dữ liệu	Mô tả
additionalInfo	string	Thông tin phụ cho thông báo.
category	string	Loại thông báo.
createdAt	timestamp	Thời điểm tạo thông báo.

deleted	boolean	Người dùng đã xóa thông báo chưa.
isRead	boolean	Đã đọc hay chưa.
message	string	Nội dung thông báo.
orderId	string/null	Dùng khi thông báo liên quan đến đơn hàng (order).
productId	string/null	Dùng khi thông báo liên quan đến sản phẩm.
title	string	Tiêu đề thông báo.
type	string	Loại sự kiện.
userId	string	Người dùng nhận thông báo.
voucherCode	string	Mã voucher liên quan.
voucherId	string	ID của voucher gốc.

□ **Collection Order:**

```

_id: ObjectId('691c12eab9181b541825d074')
orderId: "DH1763447529526"
userId: "wT7k0eydhDeZC3DiRhncUIFTPvi1"
userEmail: "hungggg@gmail.com"
▶ customerInfo: Object
▶ items: Array (1)
totalPrice: 16690000
paymentMethod: "ZALOPAY"
status: "pending_payment"
zpTransToken: "ACFI9eB5aUhDBLR9zSEGYKSQ"
appTransId: "251118_245158"
paymentUrl: "https://qcgateway.zalopay.vn/openinapp?order=eyJ6cHRyYW5zdG9rZW4iOiJBQ..."

```

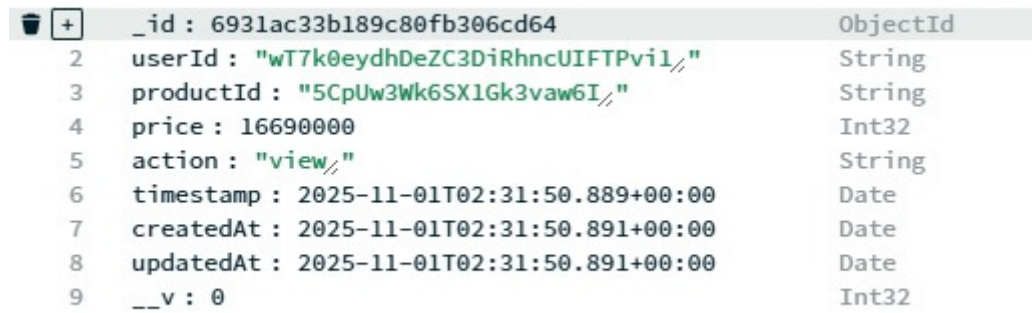
Hình 3.24. Minh họa collection Order

Bảng 3.24. Mô tả collection Order.

<i>Thuộc tính</i>	<i>Kiểu dữ liệu</i>	<i>Mô tả</i>
_id	ObjectId	Khóa chính

orderId	string	Mã đơn hàng hiển thị cho người dùng.
userId	string	ID người đặt hàng.
userEmail	string	Email khách hàng.
customerInfo	object	Thông tin người nhận.
items	array	Danh sách sản phẩm trong đơn.
totalPrice	int32	Tổng tiền đơn hàng.
paymentMethod	string	Phương thức thanh toán.
status	string	Trạng thái đơn hàng.
zpTransToken	string	Token giao dịch từ ZaloPay.
appTransId	string	Mã giao dịch app-side với ZaloPay.
paymentUrl	string	Link mở thanh toán ZaloPay.
createdAt	date	Ngày tạo đơn hàng.
updatedAt	date	Ngày cập nhật đơn hàng.
__v	int32	Version document.

□ **Collection Userbehaviors.**



1	_id : 6931ac33b189c80fb306cd64	ObjectId
2	userId : "wT7k0eydhDeZC3DiRhncUIFTPvi1/"	String
3	productId : "5CpUw3Wk6SX1Gk3vaw6I/"	String
4	price : 16690000	Int32
5	action : "view/"	String
6	timestamp : 2025-11-01T02:31:50.889+00:00	Date
7	createdAt : 2025-11-01T02:31:50.891+00:00	Date
8	updatedAt : 2025-11-01T02:31:50.891+00:00	Date
9	__v : 0	Int32

Hình 3.25. Minh họa collection Userbehaviors.

Bảng 3.25. Mô tả collection Userbehaviors.

Thuộc tính	Kiểu dữ liệu	Mô tả
_id	ObjectId	Khóa chính
userId	string	Khóa phụ liên kết người dùng.
productId	string	Khóa phụ liên kết sản phẩm.
price	Int32	Giá sản phẩm
action	string	"view" – hành động của user.
timestamp	date	Thời điểm xảy ra hành động.
createdAt	date	Thời gian tạo.
updatedAt	date	Thời gian cập nhật.
__v	Int32	Phiên bản.

CHƯƠNG 4. CẤU HÌNH, TRIỂN KHAI HỆ THỐNG VÀ KẾT QUẢ

4.1. Cấu hình hệ thống.

Hệ điều hành: Windows 10.

IDE: Visual Studio Code (VS Code).

Web server: Node.js (Runtime Environment).

Cơ sở dữ liệu: MongoDB và Firebase.

Ngôn ngữ: JavaScript (chính), HTML, CSS.

Thư viện thêm: Bootstrap.

4.2. Triển khai hệ thống.

4.2.1. Cài đặt môi trường.

Để phát triển và hoạt động hệ thống **website bán hàng công nghệ**, cần chuẩn bị đầy đủ các môi trường phần mềm được hoạt động một cách trơn chu và hiệu quả thì:

Hệ điều hành

Hệ điều hành sử dụng: windows 10.

Hỗ trợ cài đặt và chạy máy chủ website, cơ sở dữ liệu và công cụ phát triển.

Máy chủ và ngôn ngữ lập trình

JavaScript: ngôn ngữ lập trình chính bên phía server, kết hợp với Reactjs để phát triển hệ thống.

NodeJs: chạy server và quản lí các gói JavaScript.

Cơ sở dữ liệu

MongoDb: lưu cơ sở dữ liệu về lịch sử đơn hàng, chatbot, dữ liệu người dùng cũng như trạng thái người dùng.

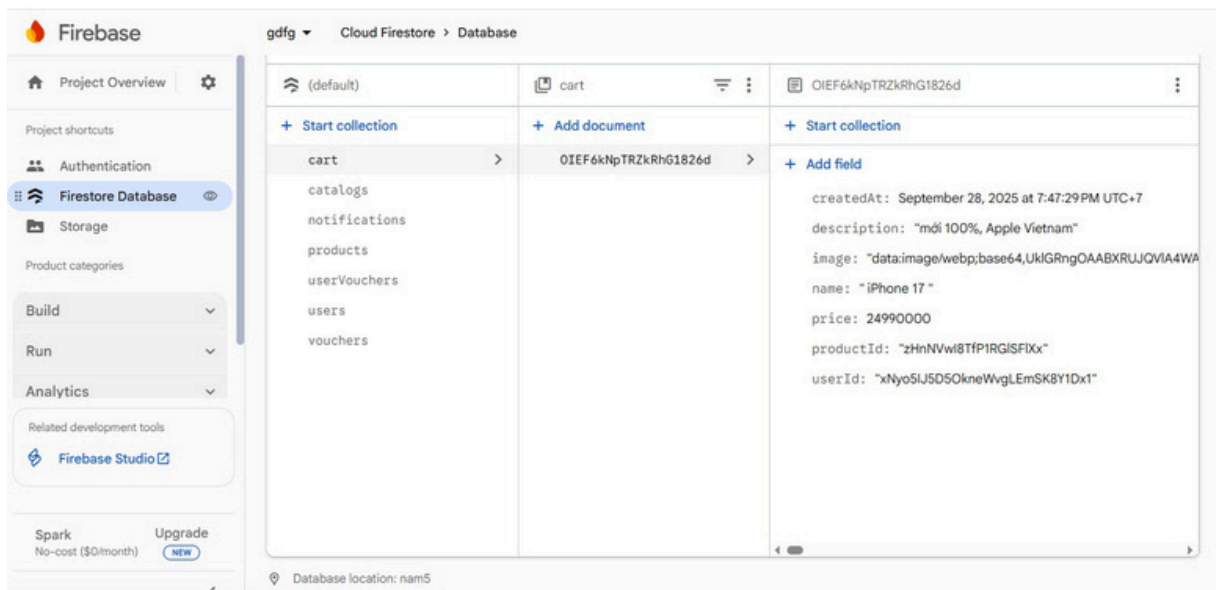
FireBase: lưu thông tin đơn hàng, người dùng, thông báo, cũng như voucher bên phía người dùng.

Công cụ hỗ trợ

Visual Studio Code: môi trường phát triển và tích hợp IDE.

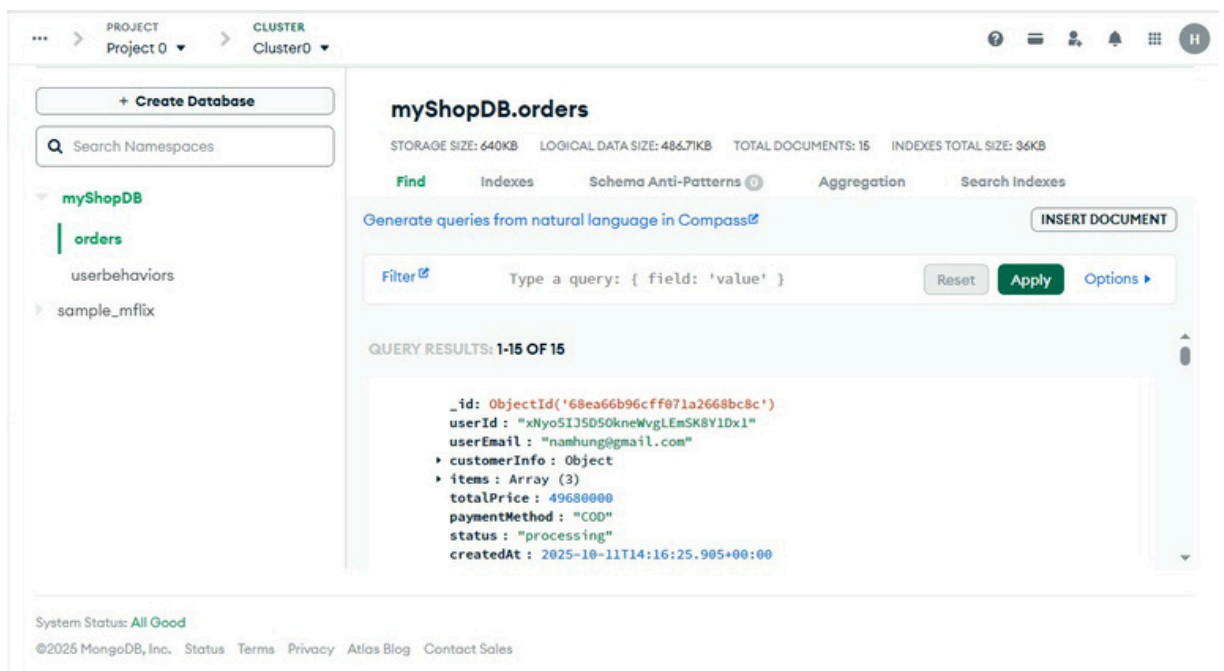
Git & GitHub: quản lí và các phiên bản mã nguồn.

4.2.2. Cài đặt CSDL.



Hình 4.1 Các collection được thêm vào Firebase trong Admin.

Back-end sẵn có (BaaS - Backend as a Service): Firebase cung cấp các dịch vụ Backend hoàn chỉnh (Auth, Storage, Database) dưới dạng API, cho phép lập trình viên tập trung 100% vào Front-end (ReactJS) và logic nghiệp vụ chính trên NodeJS



Hình 4.2 Các collection được thêm vào Mongoddb trong Admin.

Mongoddb lưu trữ dữ liệu dưới dạng Document (tương tự như JSON/BSON). Mỗi tài liệu trong cùng một Collection (tương đương Table) có thể có cấu trúc, số lượng trường, và kiểu dữ liệu khác nhau. Công dụng: Giúp phát triển nhanh chóng, dễ dàng thay đổi

cấu trúc dữ liệu mà không cần phải thực hiện các thao tác thay đổi Schema phức tạp như trong SQL

4.2.3. Khởi động dự án

Bước 1: khởi chạy dự án

```
PS E:\unicca> npm start

> unicca@0.1.0 start
> react-scripts start

Browserslist: browsers data (caniuse-lite) is 13 months old. Please run:
Compiled successfully!

You can now view unicca in the browser.

Local:            http://localhost:3000
On Your Network:  http://192.168.50.1:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
█
```

Hình 4.3 Khởi tạo và chạy server.

Bước 2: kết nối cơ sở dữ liệu

Tại bước này cũng nên kiểm tra kỹ lại rằng là cơ sở dữ liệu đã kết nối được hết chưa, và nếu chưa kết nối hết thì nó sẽ báo lỗi và phải cập nhập hết tất cả các api cũng như cơ sở dữ liệu để có thể hoạt động website 1 cách mượt mà nhất.

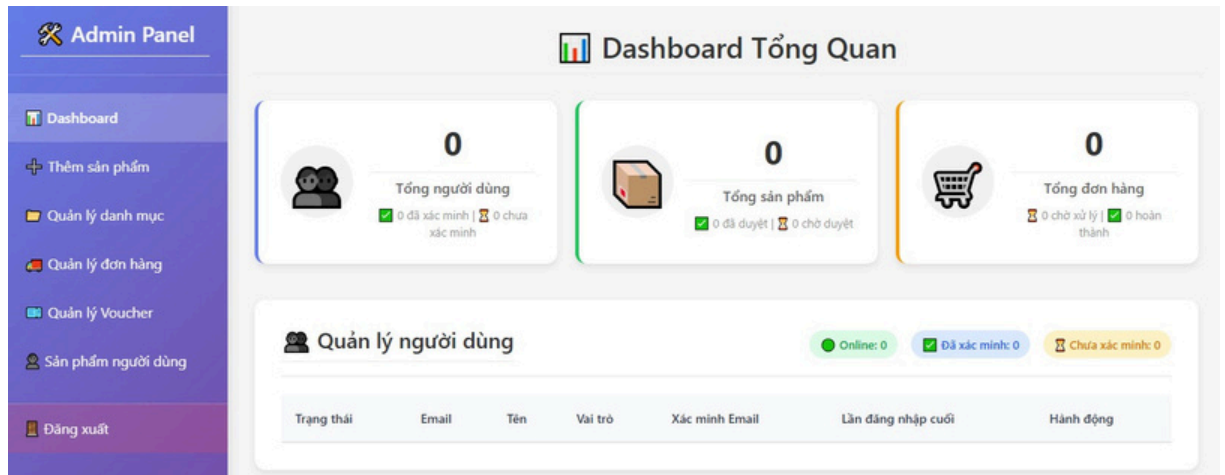
```
PS E:\unicca> node server.js
[dotenv@17.2.2] injecting env (2) from .env -- tip: ⚙ enable debug logging with { debug: true }

🚀 Server đang chạy tại http://localhost:5000
📋 API Endpoints:
✅ POST /api/orders
✅ GET /api/orders/:userId
✅ PATCH /api/orders/:id/cancel
📦 POST /api/payment/zalopay/create
📦 GET /api/payment/zalopay/verify
💬 POST /chat
🔴 POST /api/behavior/track
🔴 POST /api/recommendations/product
🔴 POST /api/recommendations/collaborative
🔴 GET /api/recommendations/user/:userId
📊 GET /api/analytics/user-behavior/:userId

✅ Đã kết nối thành công tới MongoDB!
█
```

Hình 4.4 Khởi tạo và kết nối với các api và cơ sở dữ liệu.

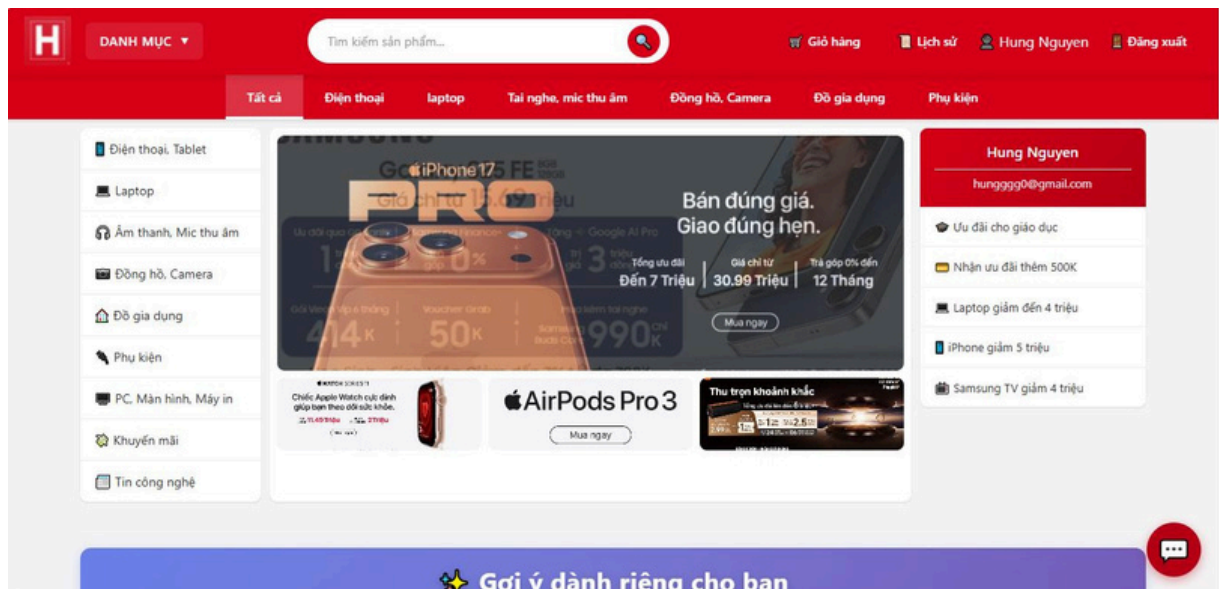
Bước 3: Khởi chạy Admin tại localhost:3001/admin



Khi khởi tạo thành công các bước sau là website có thể hoạt động một cách trơn tru.

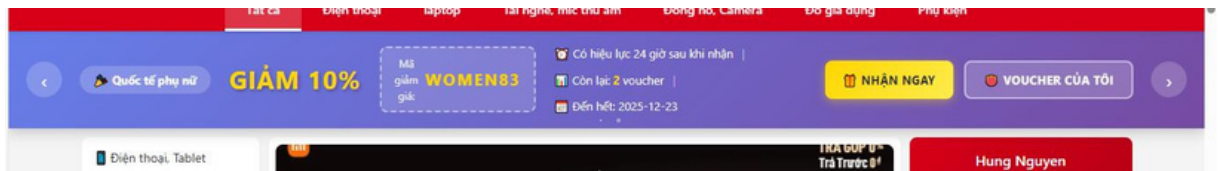
4.3. Kết quả đạt được.

4.3.1. Giao diện người dùng.



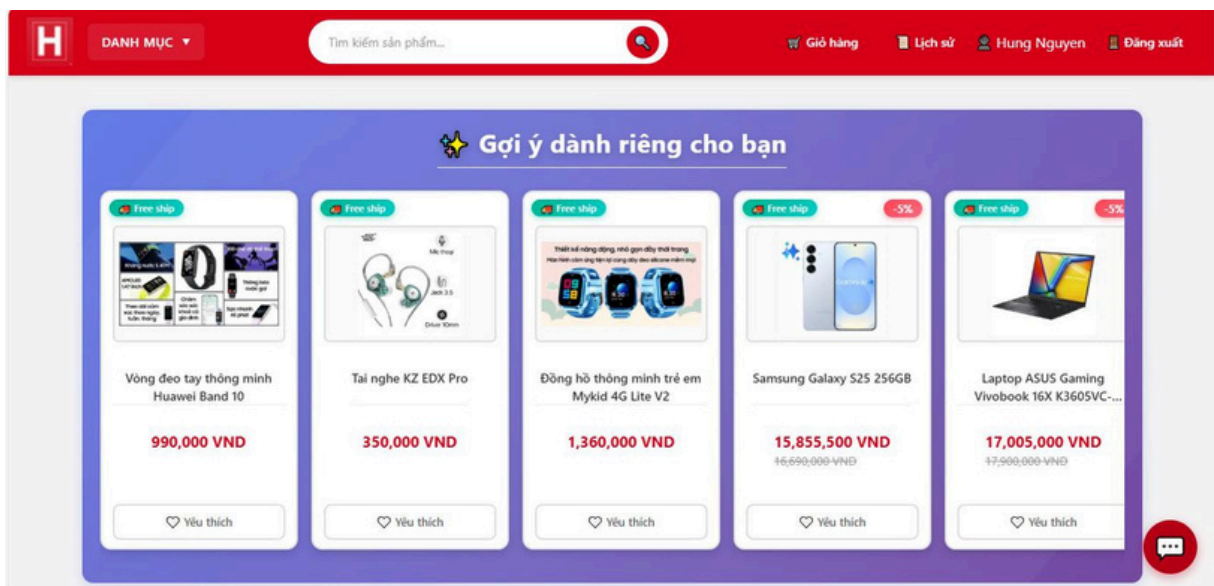
Hình 4.5. Trang giao diện Website.

Đây là giao diện chính khi người dùng vào website, thì người dùng có thể thấy được các banner quảng cáo sản phẩm, thấy các chức năng cơ bản như tìm kiếm sản phẩm, các thông tin khuyến mãi, nếu như đã đăng nhập thành công thì người dùng có thể thấy các thông tin cá nhân của mình ở phía góc trên màn hình. Kéo thêm xuống 1 chút thì sẽ thấy toàn bộ sản phẩm của cửa hàng, được chia theo các catalogs để dễ tìm kiếm cũng như dễ quản lý hơn so với người quản trị, người dùng cũng dễ dàng tìm kiếm cũng như tham khảo sản phẩm hơn.



Hình 4.6. Giao diện voucher.

Khi đăng nhập thì người dùng có thể thấy được các voucher đang được quảng cáo theo nhu cầu của người quản trị cũng như cửa hàng, để người dùng có thể dễ dàng thấy và nhận được voucher, thì voucher cũng có số lượng cũng như thời gian để sử dụng voucher đó nếu như hết hạn thì coi như bỏ.



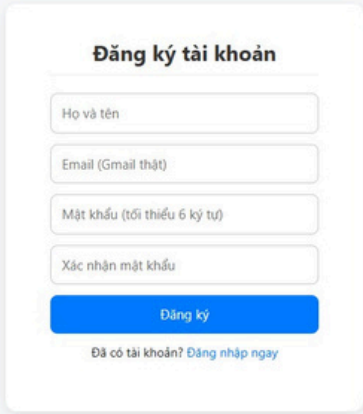
Hình 4.7. Giao diện AI recommendation.

AI recommendation dựa trên trạng thái người dùng, nó được sắp xếp theo lịch sử mua hàng hay cũng như hành vi của người dùng khi sử dụng website, dựa theo lượt click sản phẩm, lượt yêu thích sản phẩm, mua hàng, thêm vào giỏ hàng để có thể đề xuất cho người dùng mà những ứng dụng dựa trên hành vi cũng như lịch sử mua hàng của người dùng

Dựa trên các hành vi đó rồi được tích điểm sản phẩm như sau:

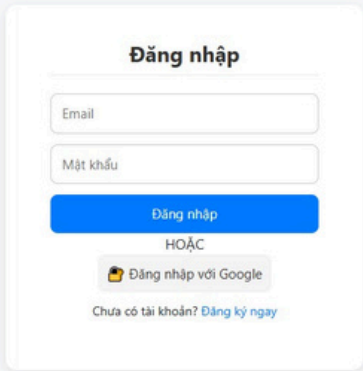
- Mua sản phẩm thành công là: 4đ
- Thêm vào giỏ hàng: 3đ
- Yêu thích sản phẩm: 2đ
- Xem sản phẩm: 1đ

Dựa vào đó để đưa sản phẩm phù hợp nhất đến với người dùng.



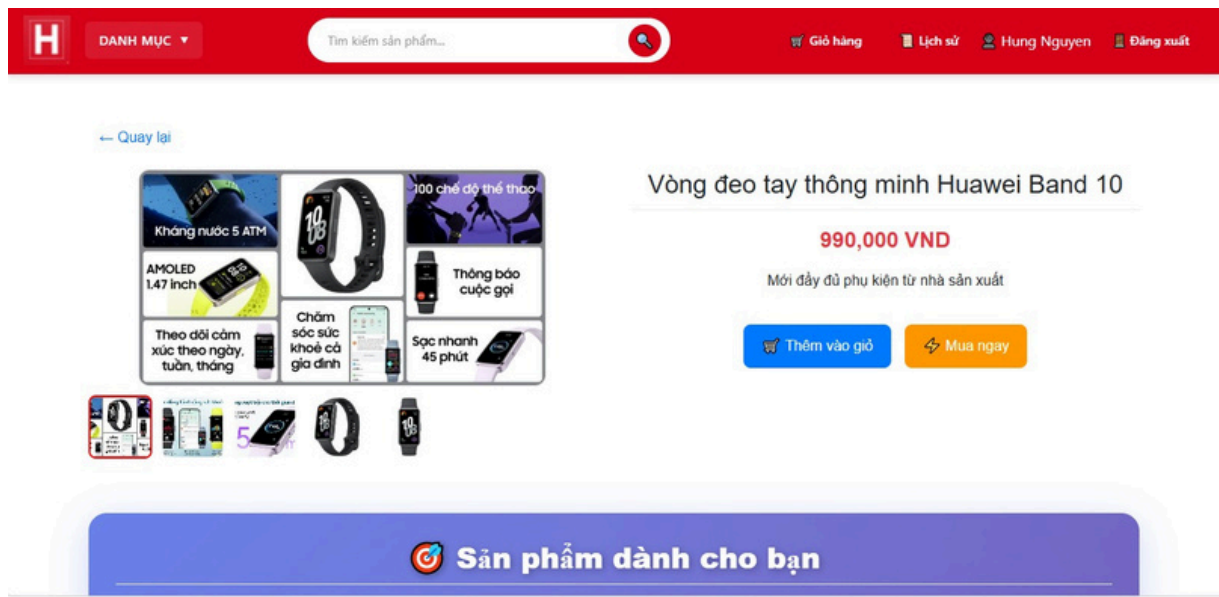
Hình 4.8. Trang đăng kí User.

Đăng kí để có thể sử dụng được hết các chức năng trong web



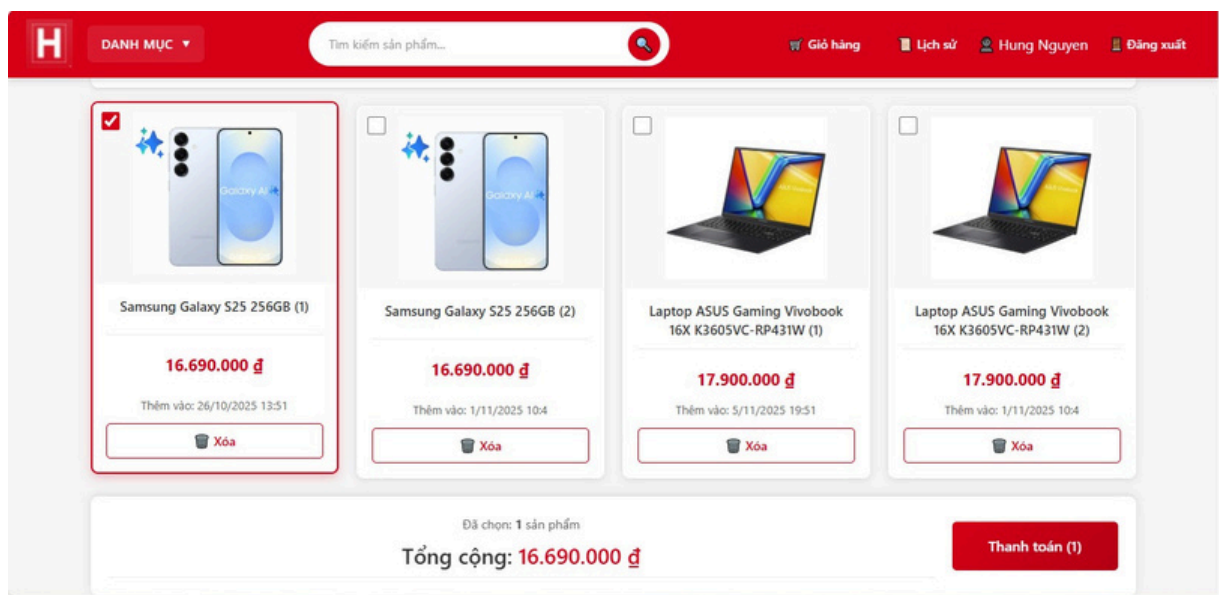
Hình 4.9. Trang đăng nhập User.

Đăng nhập để có thể mua hàng cũng như sử dụng web, có thể đăng nhập bằng google, hoặc cũng như đăng kí để đăng nhập, để lấy thông tin cũng như dữ liệu của người dùng



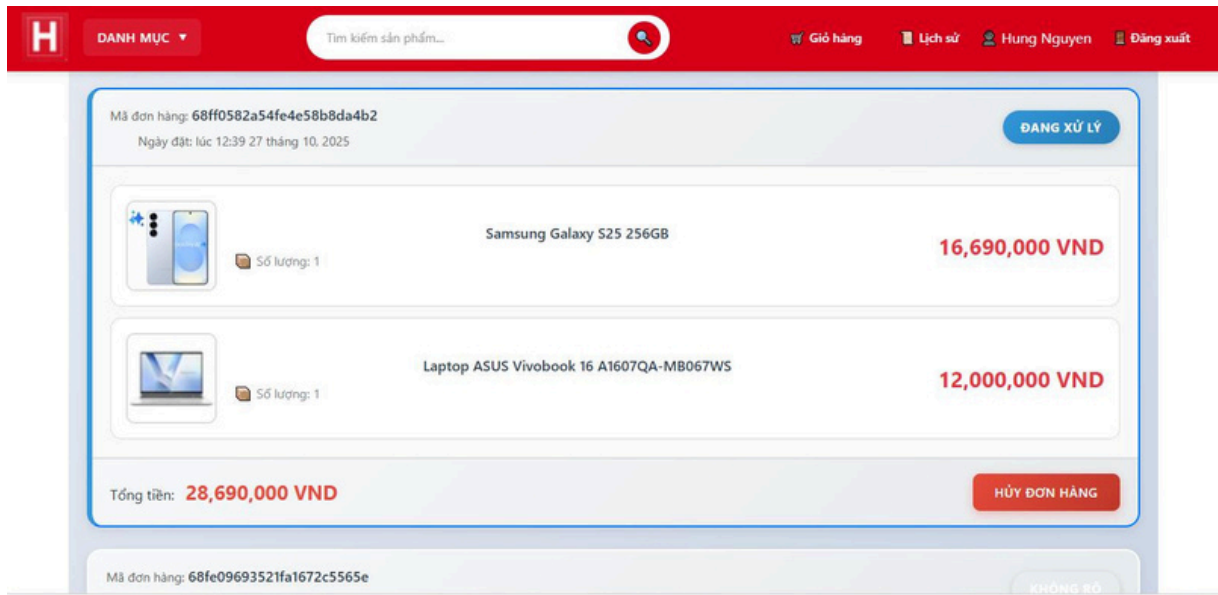
Hình 4.10. Trang chi tiết sản phẩm.

Giới thiệu chi tiết sản phẩm tới cho người dùng về thông tin cũng như: tổng tin hình ảnh giá cả về sản phẩm để cho người dùng dễ dàng để lựa chọn sản phẩm mà người dùng muốn mua.



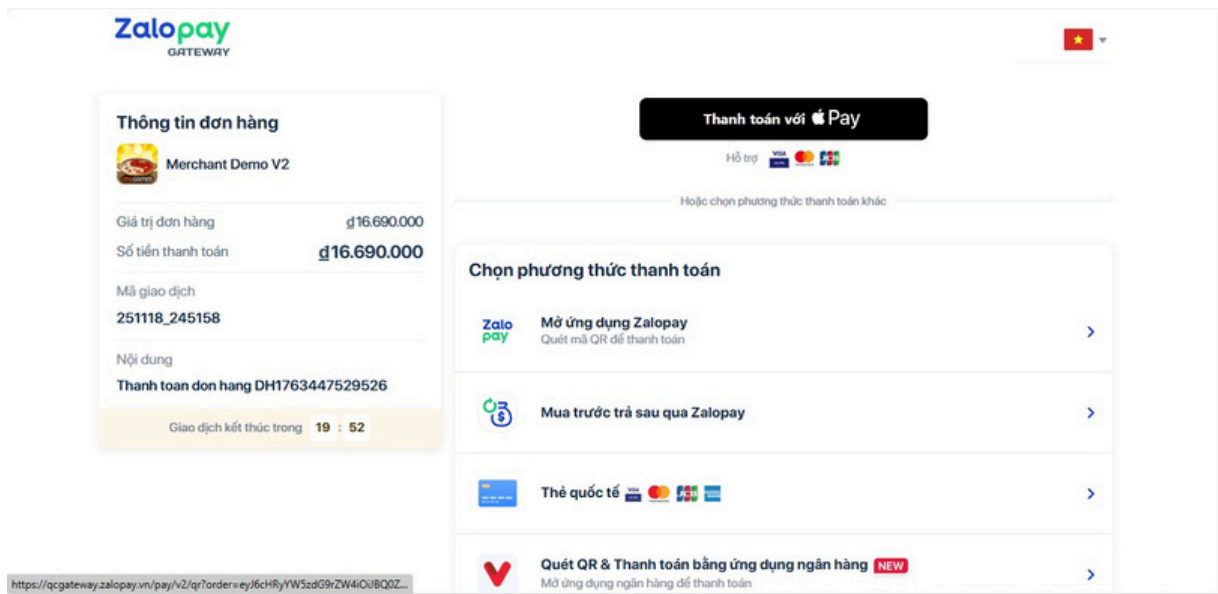
Hình 4.11. Trang giỏ hàng.

Khi người dùng đã lựa được sản phẩm ưng ý và muốn mua thì sẽ thêm sản phẩm vào giỏ hàng thì web sẽ chuyển qua trang giỏ hàng này, giỏ hàng cũng để chứa các sản phẩm mà người dùng để ý cũng như muốn mua sản phẩm, có thể xóa bớt sản phẩm.



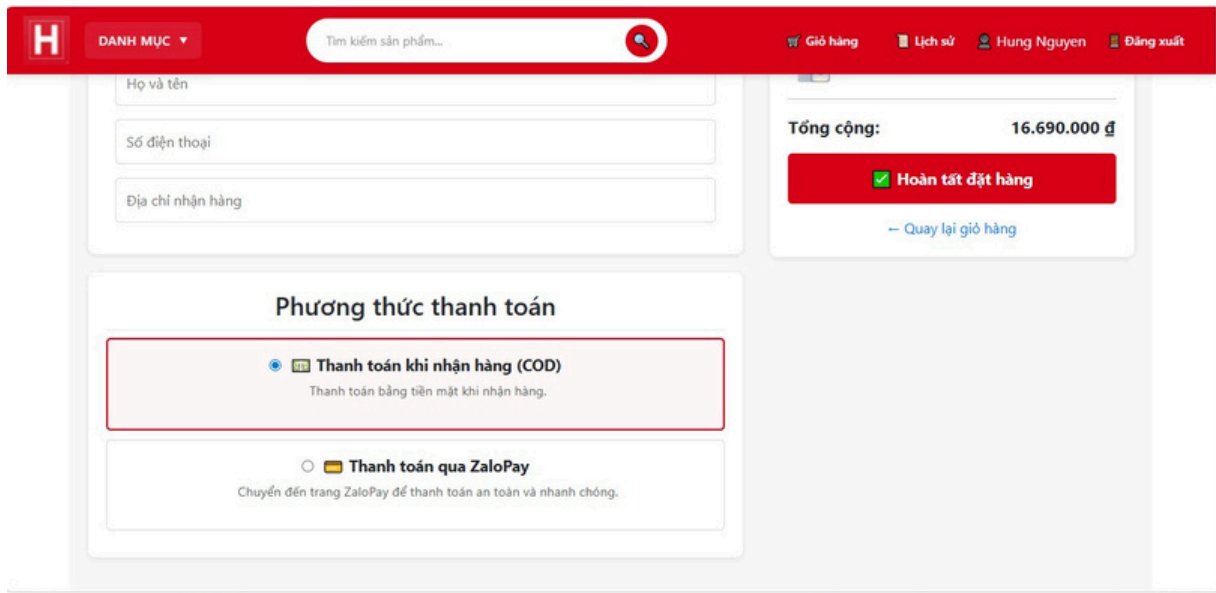
Hình 4.12. Trang lịch sử mua hàng.

Ở trang lịch sử mua hàng thì người dùng có thể thấy được mã đơn mà người dùng đã đặt hoặc mua và cũng có thể thấy được tình trạng đơn hàng cũng như có thể hủy đơn hàng nếu muốn (có lý do chính đáng) để có thể hủy đơn.



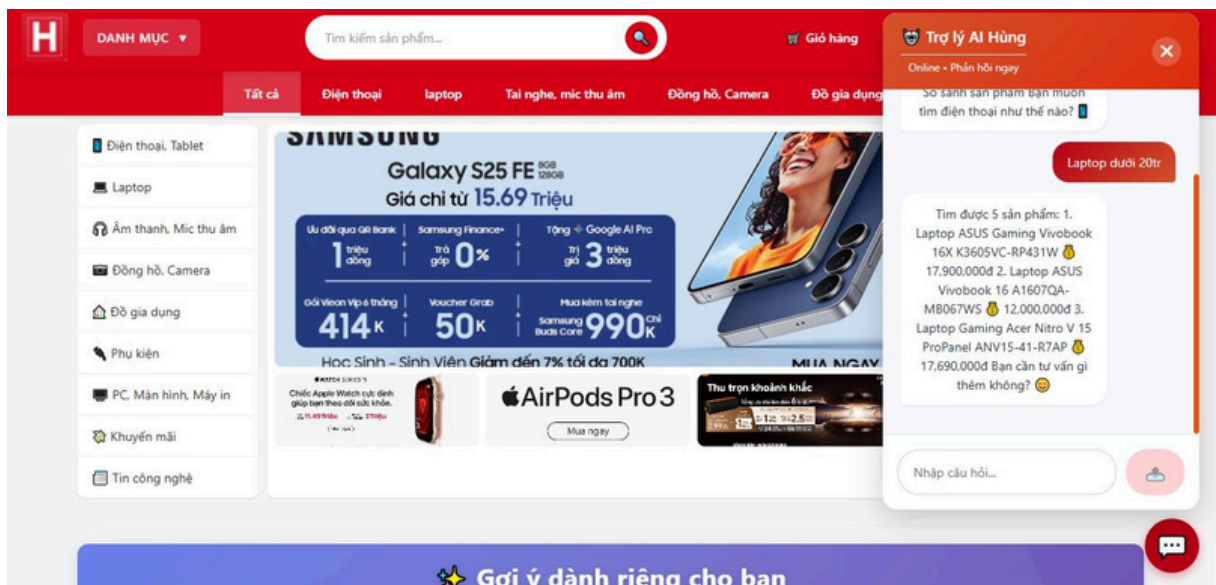
Hình 4.13. Trang thanh toán trực tuyến Zalopay.

Khi thực hiện thanh toán online thông qua Zalopay thì sẽ chuyển qua trang Zalopay và người dùng sẽ thực hiện các phương pháp thanh toán theo mô hình của zalo, khi thanh toán thành công thì sẽ chuyển trạng thái về trang chủ của web để xem chi tiết đơn hàng đang mua.



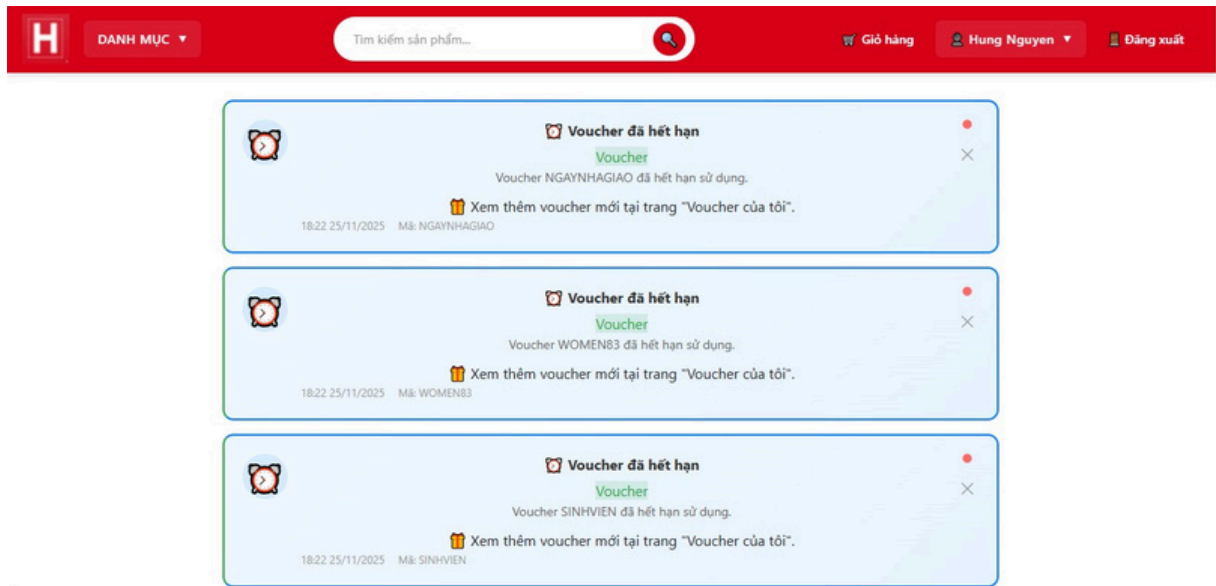
Hình 4.14. Trang phương thức thanh toán.

Ở trang thanh toán thì người dùng sẽ được thấy thông tin của người dùng cũng như có thể thêm sửa xóa địa chỉ trước khi đơn hàng được gửi đi, thấy được giá cả và có thể áp thêm voucher cũng như mã giảm giá để có thể giảm đi giá tiền khi mua sản phẩm.



Hình 4.15. Trang Chatbot hỗ trợ khách hàng.

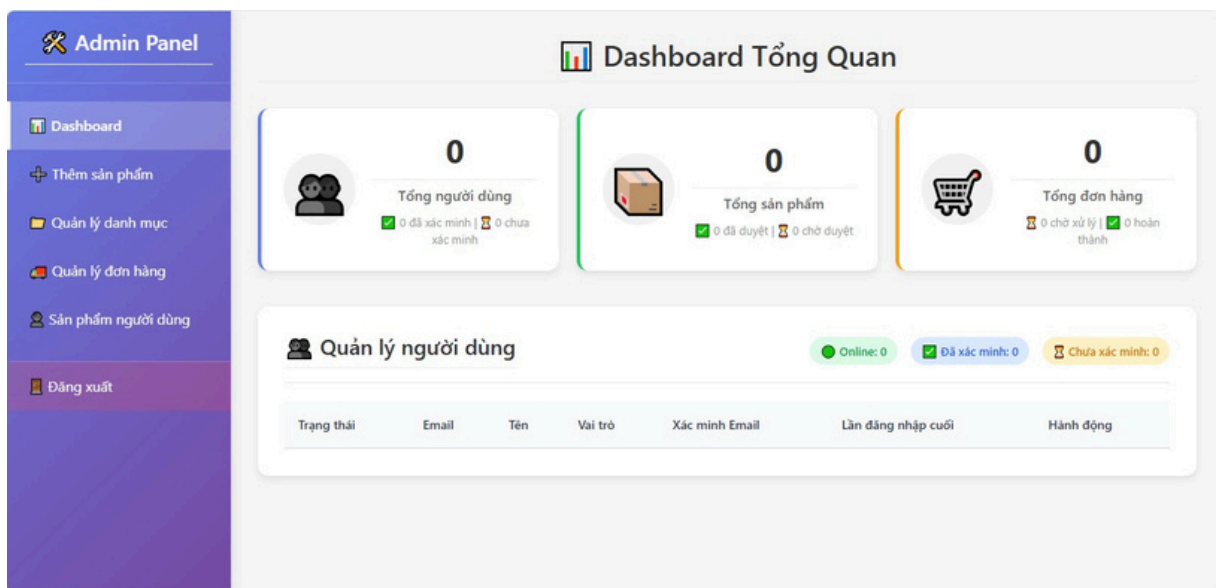
Hỗ trợ khách hàng có thể tư vấn cho khách hàng thay cho nhân viên trực web, tư vấn viên cho khách hàng về giá cả cũng như chất lượng sản phẩm tới tay người dùng.



Hình 4.16 Trang thông báo cho người dùng.

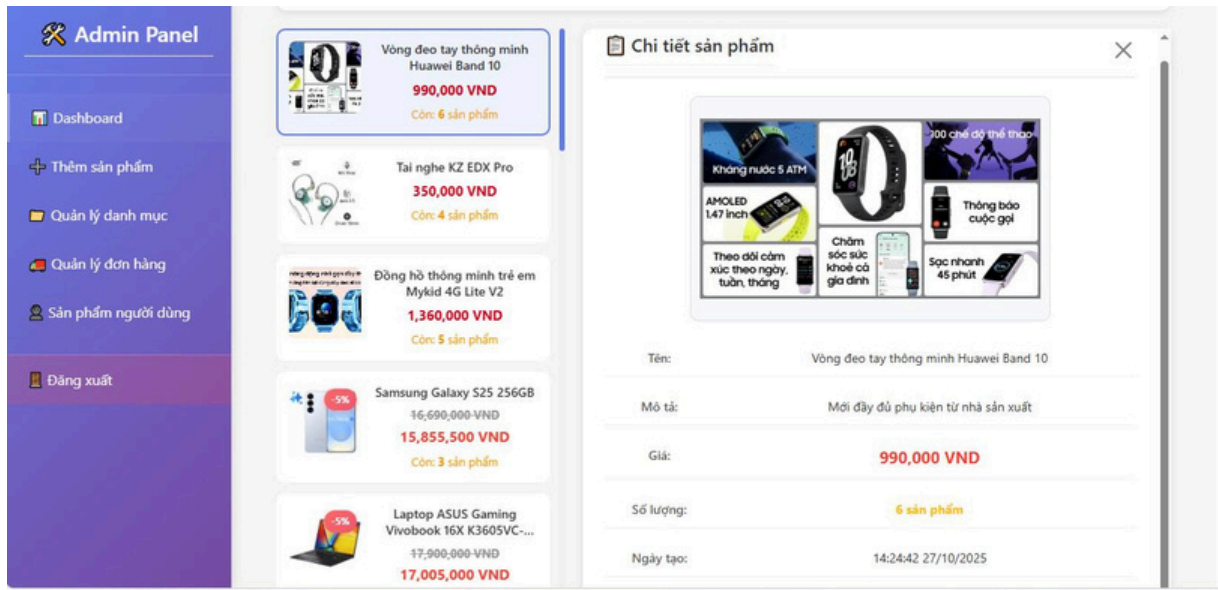
Trang thông báo cho người dùng về các hoạt động hay các trạng thái của người dùng ví dụ khi sản phẩm đã tới, thông báo voucher sắp hết hạn hoặc đã hết hạn, thông báo cho người dùng được biết.

4.3.2. Giao diện người quản trị.



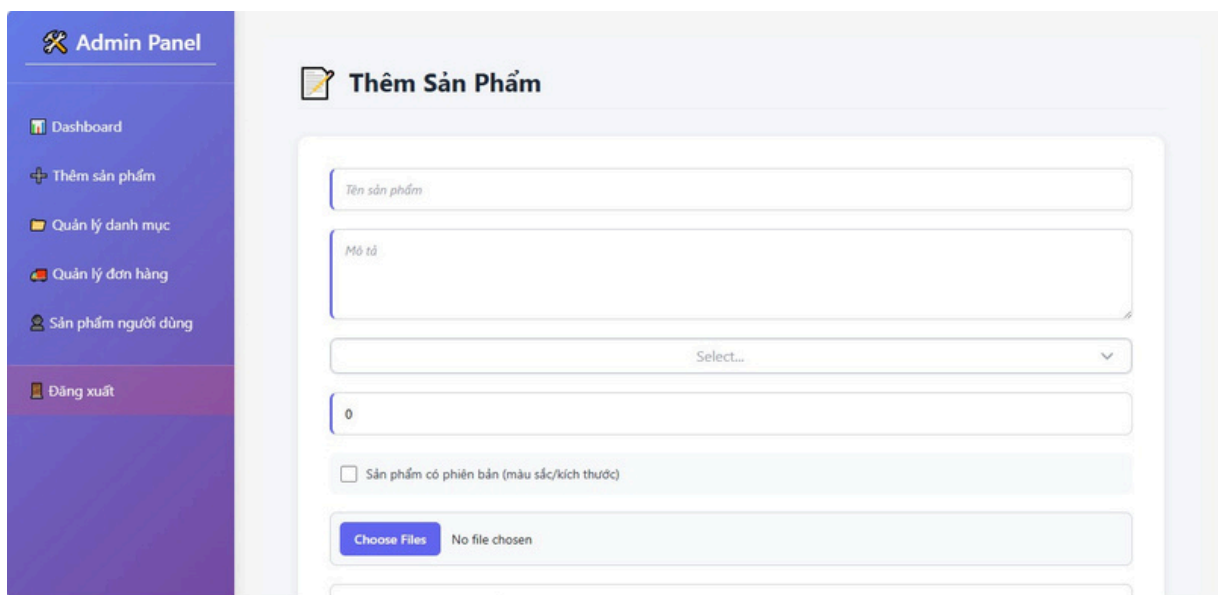
Hình 4.17. Trang quản trị Admin.

Dashboar dành cho người quản trị dễ dàng thấy được tổng doanh thu cũng như tổng đơn hàng đã được gửi đi cũng như thông tin của các người dùng đã hoạt động trong website.



Hình 4.18. Trang quản lý sản phẩm.

Giao diện cũng như dễ dàng quản lý sản phẩm cho người quản trị, có thể sửa xóa, cập nhập sản phẩm để dễ dàng hơn có thể giảm giá sản phẩm.



Hình 4.19. Trang thêm sản phẩm.

Thêm sản phẩm mới, người quản trị sẽ phải có thông tin cơ bản của sản phẩm cũng như hình ảnh của sản phẩm để có thể thêm sản phẩm, sản phẩm đã được chia theo catalogs vì đó người quản trị cũng dễ dàng thêm sản phẩm hơn.

Order ID	Customer Name	Phone Number	Payment Method	Time	Amount	Status	Actions
DH1763447529526	haihung123456789		ZaloPay	13:32:09 18/11/2025	16,690,000 VND	Chờ thanh toán	Đã nhận tiền, Chờ thanh toán
DH1761714867630	Hưng muốn uống sữa123456789		ZaloPay	12:14:27 29/10/2025	16,690,000 VND	Chờ thanh toán	Đã nhận tiền, Chờ thanh toán
DH1761479016883	haihung0332655565		ZaloPay	18:43:36 26/10/2025	33,380,000 VND	Chờ thanh toán	Đã nhận tiền, Chờ thanh toán
DH1761966331791	hung123456789		ZaloPay	10:05:31 1/11/2025	17,900,000 VND	Đã thanh toán	Đã thanh toán
DH1761965562697	hung hai123456789		ZaloPay	09:52:42 1/11/2025	16,690,000 VND	Đã thanh toán	Đã thanh toán

Hình 4.20. Trang quản lý đơn hàng.

Quản lý đơn hàng người quản trị sẽ thấy được rằng là trạng thái đơn hàng đã đi chưa khi đã nhận được tiền thì đơn hàng sẽ được chuyển đi.

Mã	Dịp lễ	Giảm giá	Thời gian phát	Hiệu lực đã phát/Tổng	Đối tượng	Trạng thái	Hành động
SINHVIENT	Sinh viên	10%	2025-11-22 – 2025-11-30	24h 2 / 2	Tất cả	Đã hết hạn	
WOMEN83	Quốc tế phụ nữ	10%	2025-11-22 – 2025-12-23	24h 1 / 3	Tất cả	Đã hết hạn	
NGAYNHAGIAO	Nhà giáo Việt Nam	20%	2025-11-20 – 2025-12-21	24h 2 / 20	Tất cả	Đã hết hạn	

Hình 4.21. Trang quản lý Voucher.

Thêm sửa xóa các voucher để phát ra cho người dùng, người quản trị có thể quản lý, số lượng voucher cũng như các mã voucher để đẩy ra trang chủ của web và tới tay người dùng.

KẾT LUẬN

Đề tài: “Xây dựng website bán hàng công nghệ” đã hoàn thành và đạt những kết quả khả quan, đã đạt được những mục tiêu mà cá nhân đã đề ra.

1. Thành tựu đạt được

Về công nghệ: qua quá trình thực hiện đồ án làm cho bản thân em hiểu thêm về các công nghệ ReactJS, NodeJS, MongoDB, Firebase, cách thực hiện cũng như cách hoạt động của các công nghệ, các cấu trúc cũng như là các api các modul cũng như luồng xử lý để có thể hoạt động 1 website cơ bản nhỏ lẻ

Về chương trình: Hoàn thiện chức năng cơ bản, đã xây dựng thành công một website bán hàng công nghệ với đầy đủ các tính năng cốt lõi như:

- Quản lý sản phẩm: Thêm, sửa, xóa, hiển thị sản phẩm, phân loại.
- Giỏ hàng và đặt hàng: Chức năng thêm sản phẩm vào giỏ hàng, cập nhật số lượng và tiến hành đặt hàng cơ bản.
- Quản lý người dùng: Đăng ký, đăng nhập, phân quyền cơ bản (người dùng/quản trị viên).
- Học hỏi quy trình: Đã tự tay triển khai toàn bộ dự án từ khâu thiết kế cơ sở dữ liệu, xây dựng Back-end API, phát triển Front-end UI/UX, đến khâu kiểm thử và triển khai ứng dụng cơ bản.

2. Hạn chế của đề tài

Mặc dù đã hoàn thành được dự án nhưng em cảm thấy dự án của chính bản thân em cũng thiếu sót rất nhiều, và những hạn chế mà em đã nhận ra là:

- Hiệu suất: cần tối ưu hóa hệ thống khi có số lượng lớn người dùng đăng nhập 1 lúc vào website, để ứng dụng có thể sử dụng và hoạt động 1 cách mượt mà hơn.
- Bảo mật: Tăng cường lớp bảo mật cũng như là thông tin khách hàng để khách hàng có thể yên tâm sử dụng và không cần lo nghĩ về chuyện lộ thông tin khách hàng.
- Mở rộng tính năng: cần theo dõi và phát triển thêm các nhu cầu của khách hàng và có thể ứng dụng thêm vào dự án, mở rộng quy mô website lớn hơn để có thể đáp ứng được nhiều nhu cầu của khách hàng hơn.

3. Hướng phát triển

Dựa vào những hạn chế đã có trong dự án thì em xin có các hướng phát triển dành cho dự án sâu hơn và nâng cao chất lượng của dự án như sau:

- Tối ưu hóa hiệu suất: Nghiên cứu và áp dụng các kỹ thuật tối ưu hóa hiệu suất mã hóa và nguồn cơ sở dữ liệu để cải thiện hiệu suất dự án.
- Bảo mật: Tăng cường lớp bảo mật bằng cách thực hiện các biện pháp như mã hóa dữ liệu, xác thực hai yếu tố, và kiểm soát truy cập.
- Mở rộng tính năng: Nghiên cứu và thêm các tính năng mới như tích hợp đánh giá, khách hàng có thể vote (1,2,3,..) để đánh giá chất lượng và cũng như thái độ lẫn dịch vụ của website và kết nối với các dịch vụ khác để cung cấp trải nghiệm đa dịch vụ

4. Kết luận

Đề tài này không chỉ mang lại một sản phẩm hoàn thiện của cá nhân em, mà nó đã đáp ứng được nhu cầu mua hàng cơ bản của người dùng trong thời đại mã số hóa, đề tài thể hiện được quá trình nghiên cứu, nỗ lực của bản thân em trong việc học hỏi và tiếp thu kiến thức để vận dụng vào đề tài. Sản phẩm đã chứng minh điều đó. Em hi vọng sản phẩm của mình sẽ có thể thực hiện và hoạt động trong một môi trường thực tế để có thể thấy được những thiếu sót trong dự án. Và lời cuối em xin cảm ơn thầy cô đã hướng dẫn và chỉ ra những hướng đi cho dự án của em.

TÀI LIỆU THAM KHẢO

- [1] ZaloPay, "Thanh toán trực tuyến," *ZaloPay Documentation*. Available: <https://docs.zalopay.vn/docs/developer-tools/test-instructions/test-wallets/>. Accessed: Nov. 2025.
- [2] StackOverflow, "ReactJS and NodeJS JSON parse unexpected character," *StackOverflow*. Available: <https://stackoverflow.com/questions/57863669/reactjs-and-node-js-json-parse-unexpected-character-at-line-1-column-1-of-the>. Accessed: Nov. 2025.
- [3] Google, "FirebaseDocumentation," *Firebase*. Available: <https://firebase.google.com/>. Accessed: Nov. 2025.
- [4] MongoDBInc., "MongoDBOfficialWebsite," *MongoDB*. Available: <https://www.mongodb.com/>. Accessed: Nov. 2025.
- [5] TaggoAI, "TaggoAI Chatbot Platform," *TaggoAI*. Available: <https://taggoai.com/>. Accessed: Nov. 2025.
- [6] Meta, "React – A JavaScript library for building user interfaces," *React Official Documentation*. Available: <https://react.dev/>. Accessed: Nov. 2025.
- [7] W3Schools, "ReactTutorial," *W3Schools*. Available: <https://www.w3schools.com/react/>. Accessed: Nov. 2025.
- [8] Node.jsFoundation, "Node.jsDocumentation," *Node.js*. Available: <https://nodejs.org/en/docs>. Accessed: Nov. 2025.