TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH**

**HỌC KỲ I, NĂM HỌC 2025-2026**

**XÂY DỰNG WEBSITE CỬA HÀNG**

**BÁN ĐIỆN THOẠI**

**QUANGHƯNG MOBILE**

|  |  |
| --- | --- |
| *Giáo viên hướng dẫn*  ThS. Phạm Minh Đương | *Sinh viên thực hiện:*  Họ tên: Hồ Quang Vinh  MSSV: 110122201  Lớp: DA22TTC |

***Vĩnh Long, Tháng 12 Năm 2025***

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH**

**HỌC KỲ I, NĂM HỌC 2025-2026**

**XÂY DỰNG WEBSITE CỬA HÀNG**

**BÁN ĐIỆN THOẠI**

**QUANGHƯNG MOBILE**

|  |  |
| --- | --- |
| *Giáo viên hướng dẫn*  ThS. Phạm Minh Đương | *Sinh viên thực hiện:*  Họ tên: Hồ Quang Vinh  MSSV: 110122201  Lớp: DA22TTC |

***Vĩnh Long, Tháng 12 Năm 2025***

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

*Vĩnh Long, ngày … tháng … năm 2025*

**Giáo viên hướng dẫn**

*(Ký và ghi rõ họ tên)*

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

*Vĩnh Long, ngày … tháng … năm 2025*

**Thành viên hội đồng**

*(Ký và ghi rõ họ tên)*

LỜI CẢM ƠN

Trước hết, em xin gửi lời cảm ơn chân thành đến quý thầy cô, các bạn bè đã luôn quan tâm, giúp đỡ em trong suốt quá trình học tập và thực hiện đề tài này.

Đề tài này là một thử thách lớn đối với em, đòi hỏi em phải có kiến thức chuyên môn vững vàng và kỹ năng lập trình thành thạo. Tuy nhiên, nhờ sự hướng dẫn tận tình của thầy Phạm Minh Đương người đã tận tình giảng dạy và trực tiếp hướng dẫn em trong suốt quá trình thực hiện Đồ án của mình. Những lời nhận xét, đánh giá và nhất là những chia sẻ kinh nghiệm làm việc của thầy là những thông tin vô cùng hữu ích cho việc hoàn thành đề tài đúng tiến độ và đạt được kết quả tốt.

Em xin trân trọng cảm ơn thầy, chúc thầy và gia đình luôn luôn mạnh khỏe và đạt được mọi thành công trong cuộc sống. Tuy có nhiều cố gắng trong quá trình học tập, cũng như trong quá trình làm báo cáo tuy nhiên không thể tránh khỏi những thiếu sót, em rất mong được sự góp ý quý báu của tất cả các thầy cô giáo cũng như tất cả các bạn để kết quả của em được hoàn thiện hơn.

Cuối cùng em xin kính chúc các thầy cô mạnh khỏe, luôn luôn là những người lái đò ân cần dìu dắt, đưa những chuyến đò tiếp theo tới bến đỗ. Một lần nữa em xin chân thành cảm ơn!

Em xin chân thành cảm ơn!

Trân trọng,

MỤC LỤC

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN…………………………………………..i](#_Toc216700236)

[NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG ii](#_Toc216700237)

[LỜI CẢM ƠN iii](#_Toc216700238)

[MỤC LỤC iv](#_Toc216700239)

[DANH MỤC HÌNH ẢNH – BẢNG BIỂU v](#_Toc216700240)

[1. TỔNG QUAN NGHIÊN CỨU 1](#_Toc216700241)

[1.1. Giới thiệu đề tài 1](#_Toc216700242)

[1.2. Mục đích nghiên cứu 1](#_Toc216700243)

[1.3. Đối tượng nghiên cứu 2](#_Toc216700244)

[1.4. Phạm vi nghiên cứu 3](#_Toc216700245)

[1.5. Phương pháp nghiên cứu 3](#_Toc216700246)

[2. NGHIÊN CỨU LÝ THUYẾT 4](#_Toc216700247)

[2.1. Tổng quan về công nghệ phát triển web 4](#_Toc216700248)

[2.1.1 Khái niệm về website 4](#_Toc216700249)

[2.1.2 Công nghệ nền tảng 5](#_Toc216700250)

[2.1.3 Môi trường phát triển 5](#_Toc216700251)

[2.2. Ngôn ngữ lập trình 6](#_Toc216700252)

[2.2.1 HTML (Hyper Text Markup Language) 6](#_Toc216700253)

[2.3. Node.js 8](#_Toc216700254)

[2.3.1 Khái niệm về Node.js 8](#_Toc216700255)

[2.3.2 Các Thành phần của Node.js 8](#_Toc216700256)

[2.3.3 Ưu nhược điểm của Node.js 13](#_Toc216700257)

[2.4. Express.js 13](#_Toc216700258)

[2.4.1 Khái niệm về Express.js 13](#_Toc216700259)

[2.4.2 Các thành phần chính của Express.js 14](#_Toc216700260)

[2.5. Công nghệ giao diện 15](#_Toc216700261)

[2.5.1 Tailwind CSS 15](#_Toc216700262)

[2.6. MySQL 16](#_Toc216700263)

[2.6.1 MySQL là gì? 16](#_Toc216700264)

[2.6.2 Các đặc điểm chính của MySQL 17](#_Toc216700265)

[2.7. API 17](#_Toc216700266)

[2.7.1 Khái niệm về API 17](#_Toc216700267)

[2.8. Các nghiệp vụ liên quan đến đề tài 18](#_Toc216700268)

[2.8.1 Nghiệp vụ quản lý sản phẩm 18](#_Toc216700269)

[2.8.2 Nghiệp vụ bán hàng 18](#_Toc216700270)

[2.8.3 Nghiệp vụ giỏ hàng 18](#_Toc216700271)

[2.8.4 Nghiệp vụ đặt hàng 18](#_Toc216700272)

[2.8.5 Nghiệp vụ thanh toán 19](#_Toc216700273)

[2.8.6 Nghiệp vụ quản trị 19](#_Toc216700274)

[2.9. Các công trình nghiên cứu liên quan 19](#_Toc216700275)

[2.9.1 Nghiên cứu về hệ thống thương mại điện tử 19](#_Toc216700276)

[2.9.2 Các website thương mại điện tử phổ biến tại Việt Nam 19](#_Toc216700277)

[2.9.3 Các nghiên cứu về ứng dụng Node.js – Express.js 20](#_Toc216700278)

[2.9.4 Các nghiên cứu về bảo mật web 20](#_Toc216700279)

[2.10. Tổng quan các công cụ hỗ trợ phát triển và triển khai 21](#_Toc216700280)

[2.10.1 Visual Studio Code 21](#_Toc216700281)

[2.10.2 Git/GitHub 22](#_Toc216700282)

[2.11. Tổng kết chương 23](#_Toc216700283)

[3. HIỆN THỰC HÓA NGHIÊN CỨU 25](#_Toc216700284)

[3.1. Mô hình thực thể kết hợp ERD 25](#_Toc216700285)

[3.2. Mô tả hệ thống 25](#_Toc216700286)

[3.2.1 Kiến trúc hệ thống 26](#_Toc216700287)

[3.2.2 Kiến trúc dự án 27](#_Toc216700288)

[3.3. Xác định các yêu cầu chức năng của hệ thống 28](#_Toc216700289)

[3.3.1 Sơ đồ Usecase 28](#_Toc216700290)

[3.3.2 Chức năng dành cho khách hàng 28](#_Toc216700291)

[3.3.3 Chức năng dành cho quản trị viên (Admin) 29](#_Toc216700292)

[3.3.4 Chức năng dành cho Khách vãng lai: 30](#_Toc216700293)

[3.4. Phân tích thiết kế dữ liệu hệ thống 30](#_Toc216700294)

[3.4.1 Thiết kế API 30](#_Toc216700295)

[3.4.2 Sơ đồ tuần tự (Sequence Diagram) 32](#_Toc216700296)

[3.5. Mô hình hoạt động của hệ thống 35](#_Toc216700297)

[4. KẾT QUẢ NGHIÊN CỨU 37](#_Toc216700298)

[4.1. Tổng quan 37](#_Toc216700299)

[4.1.1 Công nghệ sử dụng 37](#_Toc216700300)

[4.2. Giao diện người dùng 37](#_Toc216700301)

[4.2.1 Giao diện trang chủ 37](#_Toc216700302)

[4.2.2 Giao diện trang giới thiệu 39](#_Toc216700303)

[4.2.3 Giao diện trang sản phẩm 39](#_Toc216700304)

[4.2.4 Giao diện giỏ hàng 40](#_Toc216700305)

[4.2.5 Giao diện thanh toán 41](#_Toc216700306)

[4.2.6 Giao diện trang tin tức 42](#_Toc216700307)

[4.2.7 Giao diện tài khoản người dùng 42](#_Toc216700308)

[4.3. Giao diện Admin 43](#_Toc216700309)

[4.3.1 Dashboard 43](#_Toc216700310)

[4.4. Kết luận 45](#_Toc216700311)

[5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 45](#_Toc216700312)

[5.1. Kết luận 45](#_Toc216700313)

[5.1.1 Kết quả đạt được 45](#_Toc216700314)

[5.1.2 Ý nghĩa thực tiễn 47](#_Toc216700315)

[5.2. Hạn chế 47](#_Toc216700316)

[5.3. Hướng phát triển 48](#_Toc216700317)

[5.4. Lời kết 48](#_Toc216700318)

[TÀI LIỆU THAM KHẢO 49](#_Toc216700319)

DANH MỤC BẢNG BIỂU

[Bảng 2. 1 Thành phần cấu trúc Node.js 9](#_Toc216700491)

[Bảng 2.2 Điều khiển Node.js 11](#_Toc216700492)

DANH MỤC HÌNH ẢNH

[Hình 1. Mô hình ERD 25](#_Toc216700504)

[Hình 2. Mô Hình kiến trúc hệ thống 26](#_Toc216700505)

[Hình 3. Kiến Trúc dự án 27](#_Toc216700506)

[Hình 4. Sơ đồ Usecase 28](#_Toc216700507)

[Hình 5.Quy trình Đăng ký 32](#_Toc216700508)

[Hình 6. Quy trình đăng nhập 33](#_Toc216700509)

[Hình 7. Quy trình đặt hàng và thanh toán 34](#_Toc216700510)

[Hình 8. Sơ đồ hoạt động đặt hàng trực tuyến 35](#_Toc216700511)

[Hình 9. Sơ đồ hoạt động quản lý đơn hàng 36](#_Toc216700512)

[Hình 10. Giao diện trang chủ 38](#_Toc216700513)

[Hình 11. Giao diện trang giới thiệu. 39](#_Toc216700514)

[Hình 12. Giao diện trang sản phẩm 40](#_Toc216700515)

[Hình 13. Giao diện trang giỏ hàng 40](#_Toc216700516)

[Hình 14. Giao diện trang thanh toán 41](#_Toc216700517)

[Hình 15. Giao diện trang tin tức 42](#_Toc216700518)

[Hình 16. Giao diện trang người dùng 42](#_Toc216700519)

[Hình 17. Giao diện trang Dashboard 43](#_Toc216700520)

[Hình 18. Các trang giao diện khác 45](#_Toc216700521)

# TỔNG QUAN NGHIÊN CỨU

## Giới thiệu đề tài

Trong thời đại công nghệ số phát triển mạnh mẽ, mua sắm trực tuyến đã trở thành một phần thiết yếu trong đời sống hàng ngày. Người tiêu dùng ngày càng ưu tiên tìm kiếm, so sánh và đặt mua sản phẩm qua các nền tảng online nhờ tính tiện lợi, nhanh chóng và khả năng truy cập mọi lúc, mọi nơi. Đặc biệt, đối với các sản phẩm công nghệ như điện thoại di động, nhu cầu mua sắm trực tuyến ngày càng tăng cao.

Tuy nhiên, nhiều cửa hàng hiện nay vẫn gặp phải những hạn chế như giao diện website chưa thân thiện, tốc độ tải trang chậm, quy trình mua hàng phức tạp, thiếu tính năng hỗ trợ khách hàng và khả năng quản lý dữ liệu hạn chế. Những vấn đề này ảnh hưởng trực tiếp đến trải nghiệm người dùng và hiệu quả kinh doanh của cửa hàng. Xuất phát từ thực tế trên, đề tài “Xây dựng website Quang Hưng Mobile” được lựa chọn nhằm tạo ra một nền tảng thương mại điện tử hiện đại, dễ sử dụng, thân thiện với khách hàng, đồng thời hỗ trợ quản lý sản phẩm và bán hàng hiệu quả.

Tóm lại, đề tài không chỉ góp phần hoàn thiện kỹ thuật xây dựng giao diện người dùng cho một nền tảng mua sắm trực tuyến hiện đại mà còn hướng đến việc nâng cao trải nghiệm, tăng động lực học tập và thúc đẩy hiệu quả thực tiễn của mô hình giáo dục số hóa hiện nay.

## Mục đích nghiên cứu

Mục đích nghiên cứu của đề tài là xây dựng và hoàn thiện một hệ thống website bán hàng trực tuyến cho Quang Hưng Mobile, đáp ứng nhu cầu quản lý và trải nghiệm mua sắm trong lĩnh vực thiết bị công nghệ. Đề tài hướng đến các mục tiêu sau thiết kế giao diện trực quan, thân thiện và dễ thao tác cho người dùng, hướng đến việc phát triển một nền tảng quản lý đầy đủ các nghiệp vụ như quản lý sản phẩm, khách hàng và đơn hàng, qua đó giảm thiểu thao tác thủ công và nâng cao độ chính xác trong quá trình xử lý dữ liệu. Bên cạnh đó còn chủ yếu tập trung cải thiện trải nghiệm người dùng thông qua giao diện hiện đại, dễ sử dụng và tương thích trên nhiều thiết bị, giúp khách hàng dễ dàng tìm kiếm, so sánh sản phẩm và thực hiện mua hàng nhanh chóng. Đề tài cũng đặt mục tiêu ứng dụng các công nghệ Web hiện đại như Node.js, Express.js, Tailwind CSS và MySQL để xây dựng hệ thống có hiệu năng cao, ổn định và dễ mở rộng. Ngoài ra, việc tích hợp các giải pháp nâng cao như thanh toán QR Code có giới hạn thời gian và Chatbot AI hỗ trợ tư vấn tự động cũng góp phần gia tăng tính tiện ích và tính cạnh tranh cho hệ thống, đáp ứng nhu cầu ngày càng cao của người dùng trong môi trường thương mại điện tử hiện nay.

## Đối tượng nghiên cứu

Các công nghệ frontend hiện đại: Tìm hiểu chuyên sâu về các framework ghiên cứu framework Tailwind CSS để thiết kế giao diện người dùng hiện đại và áp dụng đảm bảo khả năng hiển thị tương thích trên các thiết bị (PC, tablet, mobile).

Các công nghệ frontend: Node.js kết hợp framework Express.js được nghiên cứu và sử dụng để xây dựng hệ thống máy chủ của website. Đây là môi trường cho phép xử lý logic nghiệp vụ một cách linh hoạt, đồng thời cung cấp các API cần thiết để giao tiếp giữa giao diện người dùng và cơ sở dữ liệu.

Nghiên cứu Hệ quản trị cơ sở dữ liệu MySQL để thiết kế cấu trúc bảng, lưu trữ và truy vấn dữ liệu về sản phẩm, khách hàng, đơn hàng một cách hiệu quả. Nhu cầu và hành vi người dùng.

Các thành phần hỗ trợ trải nghiệm người dùng và nâng cao hiệu quả vận hành: Giải pháp thanh toán qua QR Code, Chatbot AI và các kỹ thuật đảm bảo bảo mật thông tin, chống gian lận khi thanh toán và bảo vệ dữ liệu người dùng.

Nghiên cứu hành vi người dùng khi mua sắm thiết bị công nghệ trên website, cụ thể: Thói quen tìm kiếm sản phẩm, nhu cầu xem và so sánh thông số kỹ thuật kỳ vọng về tốc độ phản hồi và trải nghiệm giao diện, mong muốn được tư vấn hỗ trợ nhanh thông qua chat hoặc chatbot.Những yếu tố này ảnh hưởng trực tiếp đến thiết kế giao diện, chức năng và chất lượng trải nghiệm người.

## Phạm vi nghiên cứu

Về phạm vi kỹ thuật: Đề tài tập trung vào quá trình xây dựng và hoàn thiện giao diện frontend cho website bán điện thoại Quang Hưng Mobile. Các hoạt động kỹ thuật bao gồm thiết kế giao diện người dùng, lập trình các chức năng tương tác cho hai vai trò chính: khách hàng và quản trị viên, cùng với việc kiểm thử và tập trung vào xây dựng phần giao diện và logic xử lý để tối ưu trãi nghiệm sử dụng.

Về mặt chuyên môn: Đề tài tập trung mô phỏng các nghiệp vụ cốt lõi của một website thương mại điện tử trong lĩnh vực điện thoại di động. Các chức năng chính bao gồm xem và quản lý sản phẩm, tìm kiếm theo thương hiệu hoặc mức giá, giỏ hàng, đặt hàng. Đối với bộ phận quản trị, hệ thống cung cấp các chức năng quản lý danh mục, sản phẩm, đơn hàng, tài khoản người dùng và tin tức, khuyến mãi. đồng thời phát triển giao diện trực quan hóa dữ liệu học tập bằng các biểu đồ, thống kê.

Đối tượng phục vụ: Khách hàng có nhu cầu tìm hiểu, so sánh và mua sắm điện thoại và quản trị viên phụ trách quản lý nội dung, sản phẩm và đơn hàng.

## Phương pháp nghiên cứu

Phương pháp nghiên cứu lý thuyết: Đề tài tiến hành khảo sát và tổng hợp các tài liệu chuyên ngành, giáo trình và các khóa học trực tuyến liên quan đến thương mại điện tử, trải nghiệm người dùng. Đồng thời, nghiên cứu chuyên sâu về các framework và thư viện frontend, cũng như các nguyên lý thiết kế UI/UX nhằm đảm bảo giao diện trực quan, hiện đại và thân thiện với người dùng.

Phương pháp nghiên cứu thực nghiệm: Phần thực nghiệm tập trung vào việc triển khai thực tế hệ thống. Các bước bao gồm cài đặt môi trường phát triển, lập trình các tính năng tương tác cho từng vai trò người dùng. Hệ thống được kiểm thử qua nhiều tình huống sử dụng thực tế, điều chỉnh, tối ưu nhằm hoàn thiện sản phẩm cuối cùng, đảm bảo đáp ứng cả về chức năng lẫn trải nghiệm người dùng.

# NGHIÊN CỨU LÝ THUYẾT

## Tổng quan về công nghệ phát triển web

### Khái niệm về website

Website là một tập hợp các trang thông tin bao gồm văn bản, hình ảnh, video, liên kết được truy cập và hiển thị thông qua trình duyệt Internet. Dựa trên cấu trúc và chức năng, website có thể được phân loại thành hai nhóm chính: website tĩnh và website động. Trong đó, website động là dạng phổ biến hiện nay, cho phép tương tác trực tiếp với người dùng, đồng thời cập nhật và thay đổi nội dung liên tục thông qua cơ sở dữ liệu hoặc hệ thống quản trị nội dung. Website động đóng vai trò quan trọng trong việc quản lý hoạt động, giới thiệu sản phẩm, hỗ trợ giao dịch trực tuyến và cung cấp dịch vụ trực tiếp, mang lại trải nghiệm linh hoạt và tiện lợi cho người dùng.

Kiến trúc Client – Server là nền tảng của hầu hết các hệ thống web hiện đại. Trong mô hình này, Client thường là trình duyệt của người dùng, chịu trách nhiệm gửi các yêu cầu tới Server để truy cập tài nguyên hoặc dịch vụ. Server nhận và xử lý các yêu cầu, tương tác với hệ quản trị cơ sở dữ liệu để truy xuất hoặc lưu trữ thông tin, sau đó trả về kết quả cho client.

Kiến trúc này thường được triển khai theo mô hình 3 lớp:

Lớp trình bày (Presentation Layer): đảm nhận giao diện người dùng và tương tác trực quan.

Lớp logic nghiệp vụ (Business Logic Layer): xử lý dữ liệu, quản lý nghiệp vụ và điều phối các hoạt động của hệ thống.

Lớp dữ liệu (Data Layer): thực hiện truy xuất, lưu trữ và quản lý dữ liệu trong cơ sở dữ liệu.

Hiện nay, HTTP và HTTPS là các giao thức trao đổi thông tin phổ biến nhất. Trong đó, HTTPS cung cấp khả năng truyền tải dữ liệu một cách bảo mật và ổn định, bảo vệ thông tin người dùng khỏi việc bị truy cập hoặc giả mạo trong quá trình truyền.

### Công nghệ nền tảng

Để xây dựng một ứng dụng web hiện đại, việc lựa chọn công nghệ nền tảng đóng vai trò tiên quyết. Khởi đầu từ những cơ bản là HTML, CSS và JavaScript, phát triển hệ thống lên một tầm cao mới bằng cách áp dụng các Framework mạnh mẽ như React, Vue.js hoặc Angular. Cách tiếp cận này cho phép chia nhỏ giao diện thành các thành phần (components) độc lập, giúp mã nguồn trở nên gọn gàng và dễ dàng nâng cấp.

Song song với logic, diện mạo của ứng dụng được trau chuốt nhờ các thư viện hỗ trợ như Tailwind CSS hay Material UI, đảm bảo trải nghiệm mượt mà trên mọi kích thước màn hình. Khi quy mô ứng dụng mở rộng, các giải pháp quản lý trạng thái như Redux hay TanStack Query trở thành trợ thủ đắc lực để kiểm soát luồng dữ liệu.

Không dừng lại ở đó, để ứng dụng thực sự "sống động", các công nghệ giao tiếp thời gian thực (Socket.IO) và thư viện trực quan hóa dữ liệu (Chart.js) cũng được tích hợp. Tất cả tạo nên một giải pháp công nghệ toàn diện: vừa mạnh mẽ về hiệu năng, vừa tối ưu về trải nghiệm, đồng thời tuân thủ nghiêm ngặt các tiêu chuẩn phát triển phần mềm hiện hành.

### Môi trường phát triển

Để tạo ra một sản phẩm web chất lượng, không chỉ cần kỹ năng lập trình mà còn cần một môi trường phát triển vững chất. Như Visual Studio Code hay WebStorm, nơi từng dòng mã được hỗ trợ bởi các tính năng gợi ý và kiểm tra lỗi tức thời.

Trong quá trình phát triển, việc quản lý mã nguồn được thực hiện thông qua Git và GitHub. Sự kết hợp này không chỉ giúp lưu trữ an toàn mà còn tạo điều kiện cho việc phối hợp nhóm nhịp nhàng và tự động hóa các khâu kiểm thử, triển khai (CI/CD). Để đảm bảo mã nguồn luôn sạch và hoạt động đúng như mong đợi, chúng tôi áp dụng các công cụ kiểm định khắt khe như Jest, Cypress cho việc test lỗi, và Lighthouse hay Postman để tối ưu hóa hiệu năng cũng như API.

Một yếu tố không thể thiếu là việc quản lý các thư viện bổ trợ thông qua npm hay yarn, giúp dự án luôn hoạt động trơn tru và tránh xung đột. Cuối cùng, đích đến của quy trình là đưa sản phẩm đến tay người dùng thông qua các nền tảng triển khai hiện đại như Vercel, Netlify hay Docker, đảm bảo ứng dụng luôn vận hành ổn định và tốc độ cao nhất.

## Ngôn ngữ lập trình

### HTML (Hyper Text Markup Language)

#### Giới thiệu HTML

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu siêu văn bản cơ bản nhất được sử dụng trong việc xây dựng các trang web. Đây là nền tảng để định nghĩa cấu trúc và nội dung của trang web, giúp trình duyệt hiểu và hiển thị đúng các thông tin mà nhà phát triển muốn truyền tải. Tuy nhiên, để có thể trình bày nội dung đẹp mắt hơn, ngoài HTML còn cần đến các công nghệ khác như CSS (dùng để định dạng và thiết kế giao diện) và JavaScript (dùng để tạo ra các tương tác và chức năng động).

#### Lịch sử phát triển

HTML bắt nguồn từ đề xuất của nhà vật lý Tim Berners-Lee tại CERN vào năm 1980, dẫn đến việc ông phát triển HTML và phần mềm trình duyệt, máy chủ đầu tiên vào cuối những năm 1990. Mô tả công khai đầu tiên về HTML, “HTML Tags”, được Berners-Lee công bố vào cuối năm 1991, mô tả 18 phần tử trong thiết kế ban đầu của HTML,. Mười một trong số đó vẫn tồn tại trong HTML4.

HTML được xác định là một ứng dụng của SGML (Standard Generalized Markup Language) và được chính thức hóa bởi Internet Engineering Task Force (IETF) vào giữa năm 1993. Phiên bản đầu tiên, “HTML 2.0”, được hoàn thiện bởi IETF vào năm 1995.

Sự phát triển tiếp theo của HTML đã diễn ra dưới sự bảo trợ của W3C, bắt đầu từ năm 1996. Phiên bản quan trọng như HTML 4.01 được xuất bản vào cuối năm 1999, và HTML5, một phiên bản đột phá, được W3C hoàn thiện vào ngày 28 tháng 10 năm 2014.

HTML5 đã đánh dấu một bước ngoặt trong lịch sử phát triển của HTML, bằng cách giới thiệu nhiều tính năng mới và cải thiện đáng kể so với các phiên bản trước, nhấn mạnh vào khả năng tương tác và đa phương tiện.

#### Cách hoạt động của HTML

HTML sử dụng các "markup" để chú thích nội dung như văn bản, hình ảnh và các loại nội dung khác để hiển thị trên trình duyệt web. Các markup này bao gồm các thành phần (elements) đặc biệt như:

Các thẻ cơ bản: <head>, <title>, <body>, <header>, <footer>, <article>, <section>.

Các thẻ định dạng: <p> (đoạn văn), <div> (phân chia khối), <span> (phân chia dòng).

Các thẻ đa phương tiện: <img> (hình ảnh), <audio> (âm thanh), <video> (video), <canvas> (vẽ đồ họa).

Các thẻ tương tác: <form> (biểu mẫu), <input> (trường nhập liệu), <button> (nút bấm).

Các thẻ điều hướng: <nav> (thanh điều hướng), <a> (liên kết).

#### Cấu trúc của một phần tử HTML

Mỗi phần tử HTML được xác định bởi các "thẻ" (tags) bao quanh tên phần tử đó, ví dụ <p>, <h1>, <div>, v.v. Tên của phần tử có thể được viết bằng chữ hoa, chữ thường hoặc kết hợp cả hai, ví dụ như <Title>, <TITLE>, nhưng chuẩn khuyến nghị hiện nay là nên viết bằng chữ thường để đảm bảo tính nhất quán và dễ đọc.

#### Kết luận

HTML là ngôn ngữ cơ bản nhưng vô cùng quan trọng trong lập trình web. Hiểu và thành thạo HTML là bước đầu tiên giúp người dùng xây dựng các trang web và ứng dụng trực tuyến hiệu quả.

## Node.js

### Khái niệm về Node.js

Node.js là một nền tảng mã nguồn mở được xây dựng trên V8 Javascript engine. Node.js được viết bằng c++ và [Javascript](https://vietnix.vn/javascript-la-gi/). Nền tảng này được phát triển Node.js vào năm 2009 bởi Ryan Lienhart Dahl.

Mục đích của Node.js là cung cấp một môi trường chạy ứng dụng web hiệu quả và không đồng bộ. Nền tảng này sử dụng mô hình I/O không chặn, cho phép xử lý nhiều yêu cầu cùng lúc mà không bị tắc nghẽn.

### Các Thành phần của Node.js

Module trong Node.js giống như những gói thư viện nhỏ, chứa các hàm, đối tượng và các lớp được viết sẵn để thực hiện các tác vụ cụ thể. Điều này giúp cho việc phát triển ứng dụng trở nên modun hóa và dễ dàng quản lý. Để sử dụng một module trong ứng dụng của bạn, bạn chỉ cần sử dụng hàm require().

Ở Node.JS có nhiều Module với nhiều chức năng khác nhau và đều cần thiết cho một ứng dụng web. Dưới đây là một số Module thường được sử dụng trong ứng dụng web:

|  |  |
| --- | --- |
| **Module chính** | **Mô tả** |
| **http** | Tạo và quản lý máy chủ HTTP dùng để xây dựng API và xử lý các yêu cầu từ client trong ứng dụng web. |
| **util** | Cung cấp các hàm tiện ích hỗ trợ lập trình như định dạng dữ liệu, kiểm tra kiểu dữ liệu, kế thừa đối tượng,… giúp giảm bớt công việc xử lý thủ công. |
| **fs** | Tương tác với hệ thống file: đọc, ghi, cập nhật và xóa file; thường dùng để lưu log server hoặc thao tác với tệp cấu hình. |
| **url** | Phân tích cú pháp URL để trích xuất các thành phần như protocol, host, pathname, query,… phục vụ xử lý request web. |
| **querystring** | Xử lý chuỗi truy vấn (query string) trong URL, cho phép chuyển đổi qua lại giữa dạng chuỗi và dạng đối tượng. |
| **stream** | Làm việc với dữ liệu dạng dòng (stream) giúp tối ưu hiệu suất khi xử lý tệp dung lượng lớn như hình ảnh sản phẩm. |
| **zlib** | Nén và giải nén dữ liệu để giảm kích thước truyền tải, cải thiện tốc độ phản hồi của server. |

Bảng 2. 1 Thành phần cấu trúc Node.js

**Bảng Điều khiển**

Đây là công cụ giao tiếp giữa lập trình viên và hệ thống. Nó giúp bạn in ra các thông tin cần thiết (như giá trị biến, kết quả tính toán) hoặc thông báo lỗi lên màn hình để dễ dàng kiểm tra.

Một trong những phương pháp phổ biến để sử dụng bảng điều khiển là sử dụng hàm console.log(). Hàm này cho phép in ra bất kỳ giá trị nào lên màn hình console.

**Cluster**

NodeJS, nổi tiếng với khả năng lập trình không đồng bộ, thường được xây dựng dựa trên mô hình đơn luồng. Tuy nhiên, để cải thiện hiệu năng và tận dụng tối đa tài nguyên CPU, Cluster là một giải pháp hoàn hảo.

Module Cluster trong NodeJS cho phép bạn tạo ra nhiều tiến trình con (worker processes), mỗi tiến trình sẽ xử lý các yêu cầu đến. Điều này giúp cân bằng tải và ngăn chặn việc một tiến trình bị quá tải. Các worker process này chia sẻ cùng một cổng máy chủ, tạo nên một cụm các tiến trình làm việc cùng nhau.

**Đối tượng cục bộ**

Trong Node.JS các đối tượng toàn cục đóng vai trò vô cùng quan trọng. Đây là những biến hoặc hàm được sẵn có trong mọi module của NodeJS, cho phép các lập trình viên truy cập và sử dụng chúng trực tiếp mà không cần khai báo lại. Một số đối tượng toàn cục phổ biến ở trong bảng dưới đây:

|  |  |
| --- | --- |
| **Đối tượng** | **Mô tả** |
| **global** | Đối tượng gốc chứa các biến và hàm toàn cục, có thể truy cập ở mọi module. |
| **\_\_dirname** | Đường dẫn thư mục chứa file hiện tại. |
| **\_\_filename** | Đường dẫn đầy đủ kèm tên file hiện tại. |
| **exports** | Dùng để xuất biến/hàm ra ngoài module. |
| **module** | Đại diện cho module hiện tại, quản lý thông tin xuất – nhập. |
| **require()** | Hàm dùng để import module khác vào file hiện tại. |
| **process** | Cung cấp thông tin về tiến trình Node.js, biến môi trường và sự kiện hệ thống. |
| **setTimeout/ setInterval** | Thực thi hàm sau delay hoặc lặp theo chu kỳ. |
| **clearTimeout/ clearInterval** | Hủy timeout hoặc interval đã tạo trước đó. |

Bảng 2.2 Điều khiển Node.js

**Streaming (Luồng)**

Luồng (Streaming) là một dòng dữ liệu liên tục được truyền từ nguồn đến đích. Trong lập trình, luồng được sử dụng để đọc hoặc ghi dữ liệu vào các thiết bị ngoại vi như file, mạng, hoặc các bộ nhớ đệm (buffer).

Có 4 **loại luồng** chính:

**Luồng đọc:** Chỉ cho phép đọc dữ liệu từ nguồn.

**Luồng ghi:** Chỉ cho phép ghi dữ liệu vào đích.

**Luồng duplex:** Cho phép vừa đọc vừa ghi dữ liệu.

**Luồng chuyển đổi:** Thay đổi định dạng hoặc mã hóa dữ liệu trong quá trình truyền.

**Buffer (Bộ nhớ đệm)**

Buffer là một cấu trúc dữ liệu được sử dụng để lưu trữ và thao tác hiệu quả với dữ liệu nhị phân. Buffer cung cấp một cách linh hoạt để làm việc với dữ liệu thô trong bộ nhớ.

**Tên miền (Domain)**

Module tên miền giúp ngăn chặn và khắc phục các lỗi tiềm ẩn trong website của bạn. Module này sử dụng hai cơ chế chính để đảm bảo hoạt động ổn định của hệ thống:

**Liên kết nội bộ:** Bộ phát lỗi sẽ được tích hợp trực tiếp vào mã nguồn của website, giúp xác định và xử lý lỗi một cách nhanh chóng và hiệu quả.

**Liên kết bên ngoài:** Bạn có thể thêm thủ công các bộ phát lỗi vào tên miền thông qua cấu hình hệ thống. Điều này đặc biệt hữu ích khi bạn muốn theo dõi và quản lý các lỗi từ các nguồn bên ngoài.

**DNS**

Module DNS dùng để tương tác với các máy chủ Hệ thống tên miền (DNS). Module DNS cho phép các nhà phát triển hiệu quả phân giải tên miền thành địa chỉ IP.

Có hai phương pháp chính để phân giải DNS:

dns.resolve(): Phương pháp này thiết lập kết nối mạng với máy chủ DNS để thực hiện quá trình phân giải. Phù hợp cho các ứng dụng yêu cầu tra cứu DNS thời gian thực hoặc cần xử lý các truy vấn DNS phức tạp.

dns.lookup(): Phương pháp này thực hiện phân giải DNS mà không cần kết nối mạng. Phù hợp cho các trường hợp kết nối mạng bị hạn chế hoặc không đáng tin cậy, chẳng hạn như ứng dụng ngoại tuyến hoặc cơ chế lưu trữ cache.

**Debugger (Trình gỡ lỗi)**

Node.JS có công cụ gỡ lỗi tích hợp vô cùng tiện lợi, giúp các nhà phát triển kiểm tra và khắc phục lỗi trong mã nguồn một cách hiệu quả. Mặc dù không sở hữu nhiều tính năng phức tạp như các trình gỡ lỗi chuyên dụng, nhưng công cụ này hoàn toàn đáp ứng nhu cầu kiểm tra code đơn giản trong quá trình phát triển ứng dụng.

### Ưu nhược điểm của Node.js

**Ưu điểm:**

IO hướng sự kiện không đồng bộ, cho phép xử lý nhiều yêu cầu đồng thời.

Sử dụng JavaScript – một ngôn ngữ lập trình dễ học.

Chia sẻ cùng code ở cả phía client và server.

NPM(Node Package Manager) và module Node đang ngày càng phát triển mạnh mẽ.

Cộng đồng hỗ trợ tích cực.

Cho phép stream các file có kích thước lớn.

**Nhược điểm:**

Không có khả năng mở rộng, vì vậy không thể tận dụng lợi thế mô hình đa lõi trong các phần cứng cấp server hiện nay.

Khó thao tác với cơ sử dữ liệu quan hệ.

Mỗi callback sẽ đi kèm với rất nhiều callback lồng nhau khác.

Cần có kiến thức tốt về JavaScript.

Không phù hợp với các tác vụ đòi hỏi nhiều CPU.

## Express.js

### Khái niệm về Express.js

Express.js là một trong những framework được sử dụng rộng rãi nhất để xây dựng ứng dụng web và API trên nền tảng Node.js. Framework này cung cấp cơ chế xử lý yêu cầu HTTP một cách rõ ràng, quản lý tuyến đường (routing), tích hợp middleware và nhiều tiện ích cần thiết khác, giúp việc phát triển ứng dụng web trở nên nhanh chóng và hiệu quả hơn.

Về mặt kiến thức nền tảng, Express.js hỗ trợ lập trình viên xây dựng cấu trúc ứng dụng theo hướng rõ ràng và tối ưu. Mọi luồng yêu cầu – phản hồi đều được xử lý thông qua hệ thống middleware linh hoạt, cho phép mở rộng tính năng bằng cách tích hợp các thư viện bên ngoài. Điều này giúp ứng dụng dễ dàng mở rộng, bảo trì và phù hợp với nhiều mô hình phát triển web hiện đại.

### Các thành phần chính của Express.js

**Router (Bộ định tuyến)**

Router là thành phần cốt lõi dùng để xác định và xử lý các tuyến đường (routes) của ứng dụng. Mỗi route sẽ gắn với một HTTP method (GET, POST, PUT, DELETE…) và một đường dẫn cụ thể. Giúp ứng dụng tổ chức luồng điều hướng rõ ràng và tách biệt logic xử lý theo từng chức năng.

**Middleware (Phần mềm trung gian)**

Middleware là các hàm được thực thi theo chuỗi khi có yêu cầu gửi đến server.  
Vai trò của middleware:

Xử lý dữ liệu request trước khi vào route

Xác thực người dùng

Kiểm tra token, session

Log hoạt động

Quản lý lỗi

Đây là điểm mạnh nhất của Express.js vì giúp mở rộng ứng dụng cực kỳ linh hoạt.

## Công nghệ giao diện

### Tailwind CSS

Tailwind CSS là một bộ công cụ CSS hiện đại, tập trung vào việc dùng các lớp nhỏ đơn giản để xây dựng giao diện web nhanh chóng và linh hoạt. Thay vì viết code CSS riêng cho từng phần, thì chỉ cần thêm các lớp sẵn có trực tiếp vào HTML để tạo thiết kế đẹp.

**Đặc điểm chính**

**Ưu tiên lớp tiện ích**: Mỗi lớp chỉ làm một việc nhỏ, như bg-blue-500 để đổi màu nền xanh, p-4 để thêm khoảng cách bên trong, font-bold để chữ in đậm. Kết hợp chúng để tạo thành phần giao diện hoàn chỉnh mà không cần file CSS riêng.

**Tùy chỉnh dễ dàng**: Dùng file tailwind.config.js để thay đổi màu sắc, kích thước chữ, khoảng cách, và các điểm ngắt cho màn hình khác nhau. Bạn có thể thêm lớp mới hoặc dùng phần mở rộng.

**Hỗ trợ màn hình đa dạng**: Có các lớp cho màn hình nhỏ, trung bình, lớn, và biến thể như hover hoặc focus để giao diện thay đổi phù hợp trên điện thoại, máy tính bảng hay máy tính.

**Tiết kiệm tài nguyên**: Tự động loại bỏ các lớp không dùng để file CSS nhỏ hơn, giúp trang web tải nhanh.

**Dễ kết hợp**: Làm việc tốt với React, Vue, Next.js. Có nhiều mẫu sẵn và cộng đồng hỗ trợ lớn.

**Cách sử dụng**

Cài đặt qua npm: Chạy lệnh npm install tailwindcss, rồi tạo file cấu hình bằng npx tailwindcss init.

Thêm lớp vào HTML:

<button class="bg-blue-600 text-white px-4 py-2 rounded shadow-lg hover:bg-blue-700">

Click Me

</button>

Chỉnh sửa màu, khoảng cách, hoặc lớp cho màn hình khác nhau trong file cấu hình theo nhu cầu dự án.

Dùng công cụ gợi ý lớp trong VS Code để viết code nhanh hơn.

Ví dụ cho giao diện thay đổi theo màn hình:

<div class="w-full md:w-1/2 lg:w-1/3 p-4">

Hộp thay đổi kích thước

</div>

**Kết luận**

Tailwind CSS giúp bạn tiết kiệm thời gian thiết kế, giữ giao diện nhất quán, tải trang nhanh và dễ thích ứng với mọi thiết bị. Đây là lựa chọn phổ biến cho lập trình viên làm phần giao diện web.

## MySQL

### MySQL là gì?

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS - Relational Database Management System) mã nguồn mở, được sử dụng phổ biến trong phát triển web. Với vai trò là nền tảng lưu trữ, quản lý và truy vấn dữ liệu, MySQL đóng góp quan trọng trong việc xây dựng các ứng dụng web động. Nó hỗ trợ các thao tác cơ bản như CRUD (Create, Read, Update, Delete) và hoạt động hiệu quả với nhiều ngôn ngữ lập trình như PHP, Python, và Java.

Bên cạnh đó, MySQL cung cấp các tính năng bảo mật mạnh mẽ, bao gồm phân quyền người dùng và mã hóa dữ liệu, giúp bảo vệ hệ thống khỏi các mối đe dọa tiềm ẩn. Đồng thời, cơ sở dữ liệu này hỗ trợ tối ưu hóa truy vấn thông qua chỉ mục và cung cấp các cơ chế như khóa chính, khóa ngoại để đảm bảo tính toàn vẹn dữ liệu.

Nhờ vào hiệu suất cao, dễ sử dụng và khả năng tích hợp tốt, MySQL trở thành lựa chọn lý tưởng cho nhiều loại ứng dụng web, từ các trang thương mại điện tử đến hệ thống quản lý nội dung.

### Các đặc điểm chính của MySQL

Mã nguồn mở: MySQL miễn phí sử dụng, nhưng cũng cung cấp các phiên bản thương mại với tính năng nâng cao.

Kiến trúc quan hệ: Dữ liệu được lưu trữ dưới dạng bảng (tables), với các hàng (rows) và cột (columns), hỗ trợ chuẩn hóa dữ liệu.

Hiệu suất cao: MySQL được thiết kế tối ưu hóa cho các ứng dụng có lượng truy vấn lớn và yêu cầu tốc độ xử lý nhanh.

Đa nền tảng: MySQL có thể chạy trên nhiều hệ điều hành như Windows, Linux, macOS, và Unix.

Hỗ trợ đa người dùng: Nhiều người dùng có thể kết nối và làm việc với cơ sở dữ liệu cùng lúc mà không gây xung đột.

Tương thích tốt: MySQL dễ dàng tích hợp với nhiều ngôn ngữ lập trình phổ biến như PHP, Python, Java, .NET, và các framework web như Laravel, Django, Spring.

Bảo mật cao: Cung cấp hệ thống quản lý người dùng, mật khẩu, mã hóa SSL và hỗ trợ các quyền truy cập tùy chỉnh.

## API

### Khái niệm về API

**API (Application Programming Interface)** là một tập hợp các quy tắc, giao thức và định dạng dữ liệu cho phép các phần mềm hoặc hệ thống khác nhau giao tiếp với nhau. API đóng vai trò như **cầu nối** giúp một ứng dụng có thể truy cập và sử dụng chức năng hoặc dữ liệu của một ứng dụng khác mà không cần biết cách thức hoạt động nội bộ của nó. Nói cách khác, API giống như một **“giao diện trung gian”** cung cấp các **phương thức (methods)** để một phần mềm gửi yêu cầu (request) và nhận lại phản hồi (response).

## Các nghiệp vụ liên quan đến đề tài

Trong quá trình xây dựng phát triển website bán điện thoại Quang Hưng Mobile, các nghiệp vụ chính cần được nghiên cứu và mô phỏng bao gồm:

### Nghiệp vụ quản lý sản phẩm

Thêm mới sản phẩm tức là người quản lý sẽ thêm sản phẩm vào hệ thống website để người sử dụng được đa dạng sự lựa chọn kèm theo đó là tăng độ nhận diện sản phẩm của cửa hàng.

Cập nhật thông tin sản phẩm (giá, số lượng, thương hiệu, hình ảnh) cho người dùng hiểu thêm về chi tiết sản phẩm.

Xóa hoặc vô hiệu hóa sản phẩm khi mà sản phẩm đó không đủ đáp ứng như cầu hiện nay.

Quản lý danh mục (hãng như Apple, Samsung, Xiaomi…)

### Nghiệp vụ bán hàng

Hiển thị sản phẩm

Lọc theo giá, thương hiệu

Xem chi tiết sản phẩm

Thêm vào giỏ hàng

### Nghiệp vụ giỏ hàng

Cập nhật số lượng

Xóa sản phẩm khỏi giỏ

Tính tổng tiền

### Nghiệp vụ đặt hàng

Người dùng nhập thông tin cá nhân

Xác nhận đơn hàng

Tạo hóa đơn

Lưu lịch sử đơn hàng

### Nghiệp vụ thanh toán

Thanh toán trực tiếp (COD)

Thanh toán qua mã QR (có thời gian hết hạn)

### Nghiệp vụ quản trị

Quản lý sản phẩm

Quản lý người dùng

Quản lý đơn hàng

Quản lý tin tức/khuyến mãi

Thống kê doanh thu

## Các công trình nghiên cứu liên quan

Nhiều công trình và sản phẩm thương mại điện tử hiện nay đã được phát triển nhanh chống với các tính năng nổi bật và mạnh mẽ. Một số nghiên cứu, ứng dụng liên quan đến đề tài này.

### Nghiên cứu về hệ thống thương mại điện tử

Các nghiên cứu mô tả kiến trúc hiện nay chủ yếu tập trung vào mô hình hoạt động của hệ thống thương mại điện tử bao gồm:

Mô hình bán hàng trực tuyến

Quản lý sản phẩm – khách hàng – đơn hàng

Xử lý thanh toán trực tuyến

Các nghiên cứu này cho thấy việc xây dựng website bán hàng cần chú tâm vào đảm bảo năng cao tính bảo mật, tốc độ phản hồi và trải nghiệm người dùng.

### Các website thương mại điện tử phổ biến tại Việt Nam

Thế Giới Di Động (thegioididong.com): Tập trung vào giao diện đơn giản, dễ tìm sản phẩm .

FPT Shop (fptshop.com.vn): Tập chung vàoTối ưu bộ lọc tìm kiếm, hỗ trợ trả góp xây dựng giao diện thu hút người mua ngoài ra còn có các tính năng hỗ trợ người dùng.

CellphoneS (cellphones.com.vn): Giao diện hiện đại thu hút người mua, hỗ trợ tư vấn trực tuyến đáp ứng nhu cầu hiện nay, chatbot.

Việc tham khảo các mô hình này giúp rút ra kinh nghiệm trong việc:

Tối ưu UI/UX xây dựng một gia diện trực quan đảm bảo đủ các yêu cầu đã đề ra.

Xây dựng cấu trúc menu – danh mục hợp lý đáp ứng cho sự ưu tiên người dùng.

Tích hợp chatbot hỗ trợ khách hàng.

Tối ưu tốc độ tải trang và hình ảnh đảm bảo trãi nghiệm người dùng.

### Các nghiên cứu về ứng dụng Node.js – Express.js

Nhiều công trình chỉ ra ưu điểm của Node.js trong:

Xử lý yêu cầu nhanh

Khả năng mở rộng

Tích hợp RESTful API

Các nghiên cứu này là nền tảng quan trọng cho việc xây dựng backend của hệ thống.

### Các nghiên cứu về bảo mật web

Tham khảo các tài liệu về bảo mật liên quan đến:

SQL Injection

XSS

CSRF

Bảo mật JWT

Những nghiên cứu này giúp đảm bảo hệ thống an toàn khi triển khai thực tế.

## Tổng quan các công cụ hỗ trợ phát triển và triển khai

### Visual Studio Code

#### Giới thiệu

Visual Studio Code chính là ứng dụng cho phép biên tập, soạn thảo các đoạn code để hỗ trợ trong quá trình thực hiện xây dựng, thiết kế website một cách nhanh chóng. Visual Studio Code hay còn được viết tắt là VS Code. Trình soạn thảo này vận hành mượt mà trên các nền tảng như Windows, macOS, Linux. Hơn thế nữa, VS Code còn cho khả năng tương thích với những thiết bị máy tính có cấu hình tầm trung vẫn có thể sử dụng dễ dàng.

Visual Studio Code hỗ trợ đa dạng các chức năng Debug, đi kèm với Git, có Syntax Highlighting. Đặc biệt là tự hoàn thành mã thông minh, Snippets, và khả năng cải tiến mã nguồn. Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép các lập trình viên thay đổi Theme, phím tắt, và đa dạng các tùy chọn khác. Mặc dù trình soạn thảo Code này tương đối nhẹ, nhưng lại bao gồm các tính năng mạnh mẽ.

Dù mới được phát hành nhưng VSCode là một trong những Code Editor mạnh mẽ và phổ biến nhất dành cho lập trình viên. Nhờ hỗ trợ nhiều ngôn ngữ lập trình phổ biến, tích hợp đầy đủ các tính năng và khả năng mở rộng, nên VSCode trở nên cực kì thân thuộc với bất kì lập trình viên nào.

#### Những ưu điểm nổi bật của Visual Studio Code

Visual Studio Code là gì được rất nhiều người tìm hiểu. Đây cũng là một trong các ứng dụng được dân IT “săn đón” và tải về và sử dụng rất nhiều. Visual Studio Code cũng luôn có những cải tiến và tạo ra đa dạng các tiện ích đi kèm từ đó giúp cho các lập trình viên sử dụng dễ dàng hơn. Trong đó có thể kể đến những ưu điểm sau:

Đa dạng ngôn ngữ lập trình giúp người dùng thỏa sức sáng tạo và sử dụng như HTML, CSS, JavaScript, C++,…

Ngôn ngữ, giao diện tối giản, thân thiện, giúp các lập trình viên dễ dàng định hình nội dung.

Các tiện ích mở rộng rất đa dạng và phong phú.

Tích hợp các tính năng quan trọng như tính năng bảo mật (Git), khả năng tăng tốc xử lý vòng lặp (Debug),…

Đơn giản hóa việc tìm quản lý hết tất cả các Code có trên hệ thống.

Visual Studio Code là một trong những trình biên tập Code rất phổ biến nhất hiện nay. Ứng dụng này cũng ngày càng chứng tỏ ưu thế vượt trội của mình khi so sánh với những phần mềm khác. Tuy bản miễn phí không có nhiều các tính năng nâng cao nhưng Visual Studio Code thực sự có thể đáp ứng được hầu hết nhu cầu cơ bản của lập trình viên.

### Git/GitHub

Git là hệ thống quản lý phiên bản mã nguồn phân tán, giúp theo dõi lịch sử thay đổi dự án từ nhỏ đến lớn. GitHub là nền tảng trực tuyến dựa trên Git, hỗ trợ lưu trữ, cộng tác, kiểm tra code và tự động hóa quy trình cho lập trình viên.​

* **Đặc điểm chính của Git**

Theo dõi toàn bộ lịch sử: Ghi lại mọi thay đổi file, người thực hiện, thời gian và lý do, dễ quay lại phiên bản cũ nếu lỗi.​

Làm việc phân tán: Mỗi người có bản sao đầy đủ trên máy cá nhân, không phụ thuộc máy chủ trung tâm, tăng an toàn và tốc độ.​

Nhánh và gộp mạnh mẽ: Tạo nhánh riêng cho tính năng mới, dễ gộp thay đổi và giải quyết xung đột.​

Quy trình linh hoạt: Hỗ trợ nhiều cách làm việc như Git Flow hoặc GitHub Flow.​

* **Các lệnh Git cơ bản**

git init: Tạo kho lưu trữ mới.​

git clone <url>: Tải kho từ xa về máy.​

git add <file>: Thêm file vào danh sách chuẩn bị lưu.​

git commit -m "msg": Lưu thay đổi với ghi chú.​

git status: Kiểm tra trạng thái file.​

git branch: Xem hoặc tạo nhánh.​

git merge <branch>: Gộp nhánh.​

git pull: Lấy và cập nhật code từ xa.​

git push: Đẩy code lên kho từ xa.​

* **Đặc điểm chính của GitHub**

Lưu trữ code: Quản lý kho riêng tư hoặc công khai, sao lưu và truy cập mọi nơi.​

Yêu cầu kéo và kiểm tra: Thảo luận thay đổi, duyệt code trước khi gộp vào nhánh chính.​

Quản lý nhóm và nhiệm vụ: Giao việc, theo dõi tiến độ qua issue và mốc thời gian.​

Tự động hóa: Kiểm tra, xây dựng và triển khai với GitHub Actions hoặc công cụ khác.​

Bảo mật: Hỗ trợ xác thực hai lớp và kiểm soát quyền truy cập.​

**Kết luận**

Git kết hợp GitHub là công cụ thiết yếu cho mọi dự án phần mềm, giúp quản lý code an toàn, cộng tác chuyên nghiệp và hiện đại cho lập trình viên.

## Tổng kết chương

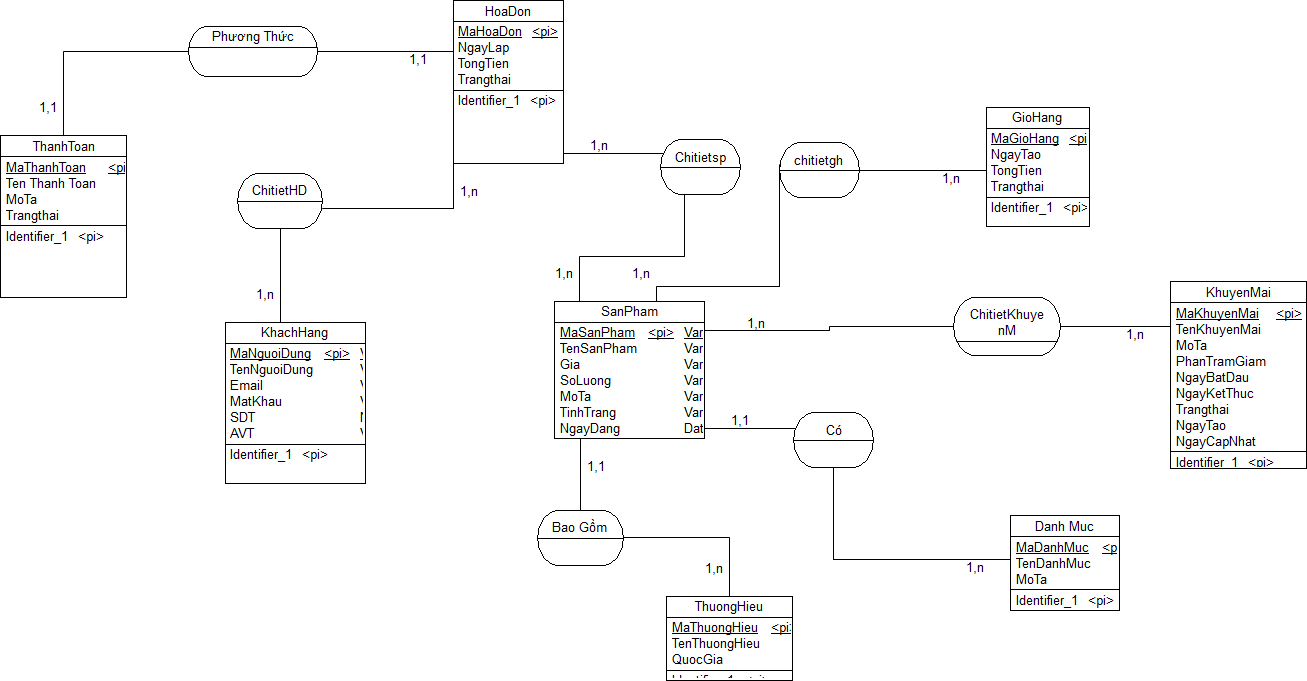
Chương này đã cung cấp cái nhìn toàn diện về các khía cạnh lý thuyết quan trọng trong phát triển ứng dụng web hiện đại. Nội dung tập trung vào các yếu tố nền tảng như thiết kế giao diện responsive, kiến trúc Client–Server, các ngôn ngữ và công nghệ cốt lõi (HTML5, CSS3, JavaScript, Node.js, Express.js, MySQL), cùng với các framework và thư viện hỗ trợ như Tailwind CSS, giúp xây dựng ứng dụng web hiệu quả, linh hoạt và trực quan.

Chương cũng phân tích các công cụ phát triển và triển khai thực tiễn, bao gồm Visual Studio Code, Git/GitHub, cùng các công cụ kiểm thử và tối ưu hiệu năng như Postman. Đồng thời, tối ưu hóa cơ sở dữ liệu cũng được đề cập, tạo nền tảng vững chắc cho việc phát triển các dự án web và tương tác.

Nhìn chung, chương này trang bị cho người đọc kiến thức lý thuyết và thực hành cần thiết để xây dựng các ứng dụng web an toàn, hiệu quả, dễ mở rộng và bảo trì, đồng thời chuẩn bị nền tảng cho việc nghiên cứu và ứng dụng các công nghệ hiện đại trong các chương tiếp theo.

# HIỆN THỰC HÓA NGHIÊN CỨU

## Mô hình thực thể kết hợp ERD

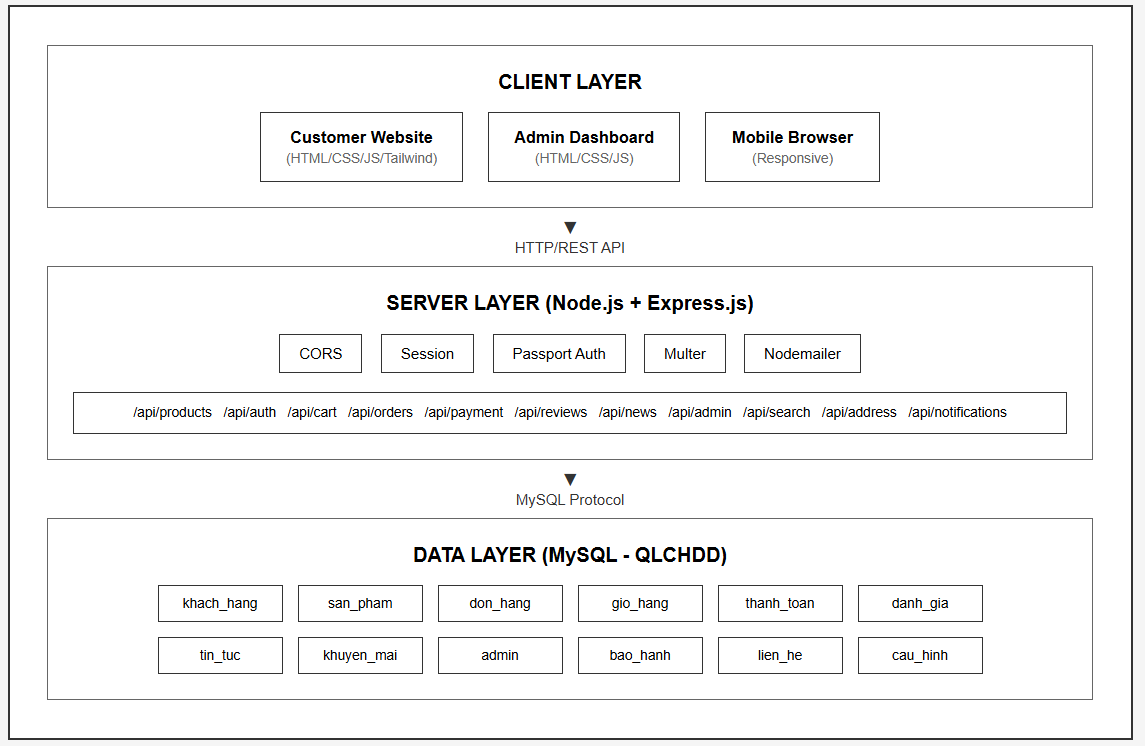


Hình 1. Mô hình ERD

## Mô tả hệ thống

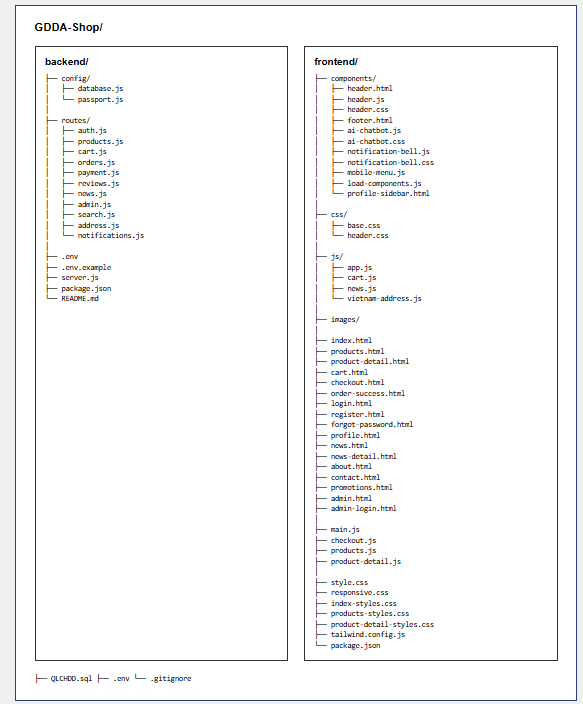
Hệ thống website **Quang Hưng Mobile** được xây dựng nhằm hỗ trợ khách hàng tra cứu thông tin sản phẩm, thực hiện mua sắm trực tuyến một cách tiện lợi nhanh chống và giúp quản trị viên quản lý toàn bộ hoạt động kinh doanh của cửa hàng. Hệ thống được phát triển theo mô hình Client – Server, trong đó giao diện người dùng được xây dựng bằng HTML, CSS, JavaScript và Tailwind CSS; backend được phát triển bằng Node.js và Express.js dữ liệu được quản lý bằng hệ quản trị cơ sở dữ liệu MySQL.

### Kiến trúc hệ thống



Hình 2. Mô Hình kiến trúc hệ thống

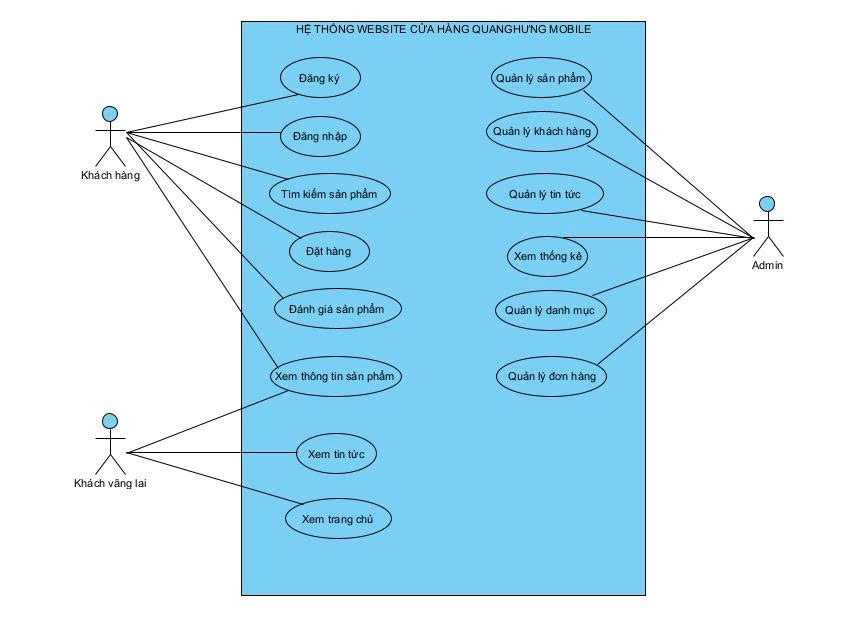
### Kiến trúc dự án



Hình 3. Kiến Trúc dự án

## Xác định các yêu cầu chức năng của hệ thống

### Sơ đồ Usecase



Hình 4. Sơ đồ Usecase

### Chức năng dành cho khách hàng

**Xem danh sách sản phẩm** là: Khách hàng có thể xem danh sách điện thoại theo thương hiệu, mức giá hoặc danh mục sản phẩm có trong trang web. Mỗi sản phẩm được hiển thị đầy đủ tên, hình ảnh, giá bán, khuyến mãi và số lượng còn lại của chúng.

**Tìm kiếm & lọc sản phẩm:** Hệ thống hỗ trợ tìm kiếm theo từ khóa và lọc nâng cao theo: Thương hiệu (Apple, Samsung, Xiaomi,…), khoảng giá, dung lượng bộ nhớ**,** tính năng nổi bậtqua đây giúp người dùng nhanh chóng tìm sản phẩm phù hợp.

**Xem chi tiết sản phẩm bao gồm:** Mô tả sản phẩm, hình ảnh chất lượng cao,cấu hình chi tiết, số lượng hàng, chính sách bảo hành qua đâyngười dùng có thể thêm sản phẩm vào giỏ hàng ngay tại trang này hoặc ấn mua trực tiếp mà không cần thêm vào giỏ.

**Giỏ hàng:** ở đây người dùng có thể thêm/xóa sản phẩm, cập nhật số lượng, tính tổng tiền tự động và áp dụng mã khuyến mãi (nếu có) cho từng sản phẩm.

**Đặt hàng:** Khách hàng sẽ nhập thông tin cá nhân gồm: Họ tên, số điện thoại, địa chỉ nhận hàng sau khi nhận được thông tin hệ thống tạo đơn hàng và ghi nhận vào cơ sở dữ liệu cho đối tựng người dùng.

**Thanh toán hỗ trợ hai hình thức**: COD (trả tiền khi nhận hàng) hoặc có thể Thanh toán QR Code (mã QR có thời gian hết hạn để tránh gian lận)

**Hỗ trợ tư vấn:** Website tích hợp chatbot AI để trả lời các câu hỏi như

Tư vấn chọn sản phẩm

Chính sách bảo hành

Tình trạng đơn hàng

### Chức năng dành cho quản trị viên (Admin)

**Quản lý sản phẩm Admin:** SẽThêm sản phẩm mới, sửa thông tin (giá, cấu hình, hình ảnh, tồn kho,…), xóa hoặc vô hiệu hóa sản phẩm, hình ảnh được tải lên máy chủ và liên kết tự động với sản phẩm.

**Quản lý danh mục & thương hiệu phân chia các nhóm điện thoại theo:** Hãng sản xuất và Loại sản phẩm Giúp người quản lý dễ truy vẫn và kiểm tra một cách nhanh chống và tiện lợi hơn

**Quản lý người dùng:** theo dõitài khoản khách hàng, lịch sử mua hàng và trạng thái hoạt động của tài khoản để quản trị viên đánh giá tiềm năng của khách hàng đó cùng với nhu cầu mua sắm để tổng hợp lại và phát triển hệ thống.

**Quản lý đơn hàng Admin:** có thể xác nhận đơn hàng, cập nhật trạng thái (đã xử lý, đang giao, hoàn thành, đã hủy), xem chi tiết đơn hàng theo thời gian, xuất báo cáo theo ngày,tháng

**Quản lý tin tức và khuyến mãi:** Cho phép thêm, sửa, xóa bài viết, quản lý chương trình giảm giá sau đó gửi thông báo cho khách hàng biết được những tin tức và khuyến mãi mới nhât.

**Thống kê và báo cáo:** Tổng hợp thống kê doanh thu theo tháng, sản phẩm bán chạy, số lượng đơn hàng theo trạng thái**,** dữ liệu được trực quan hóa bằng biểu đồ.

### Chức năng dành cho Khách vãng lai:

**Xem tin tức:** Đọc các bài viết, tin tức về công nghệ

**Xem trang chủ:** Truy cập và xem giao diện trang chủ website

## Phân tích thiết kế dữ liệu hệ thống

### Thiết kế API

**A. Xác thực (Authentication)**

POST /api/auth/send-otp dùng để gửi mã OTP đến email người dùng.  
POST /api/auth/verify-otp dùng để xác thực mã OTP.  
POST /api/auth/register dùng để đăng ký tài khoản mới.  
POST /api/auth/login cho phép khách hàng đăng nhập hệ thống.  
POST /api/auth/admin/login cho phép quản trị viên đăng nhập.  
PUT /api/auth/profile/:id dùng để cập nhật thông tin hồ sơ người dùng.  
POST /api/auth/forgot-password gửi OTP để đặt lại mật khẩu.  
POST /api/auth/verify-reset-otp xác thực OTP đặt lại mật khẩu.  
POST /api/auth/reset-password dùng để thiết lập mật khẩu mới.  
PUT /api/auth/change-password/:id cho phép thay đổi mật khẩu.  
GET /api/auth/google hỗ trợ đăng nhập bằng Google cho khách hàng.  
GET /api/auth/google/callback tiếp nhận phản hồi xác thực từ Google.  
GET /api/auth/admin/google hỗ trợ đăng nhập Google cho quản trị viên.

**B. Sản phẩm (Products)**

GET /api/products lấy danh sách sản phẩm, hỗ trợ tìm kiếm và sắp xếp.  
GET /api/products/:id lấy thông tin chi tiết sản phẩm.  
GET /api/products/brand/:brandId lấy sản phẩm theo hãng.  
GET /api/products/top-selling lấy danh sách sản phẩm bán chạy.  
GET /api/products/related/:id lấy danh sách sản phẩm liên quan.

**C. Giỏ hàng (Cart)**

GET /api/cart/:userId lấy giỏ hàng của người dùng.  
POST /api/cart thêm sản phẩm vào giỏ hàng.  
DELETE /api/cart/:userId/:productId xóa một sản phẩm khỏi giỏ hàng.  
DELETE /api/cart/:userId xóa toàn bộ giỏ hàng.

**D. Đơn hàng (Orders)**

POST /api/orders tạo đơn hàng mới.  
GET /api/orders/:orderId xem chi tiết đơn hàng.  
GET /api/orders/user/:userId lấy lịch sử đơn hàng của người dùng.  
PUT /api/orders/:orderId/cancel hủy đơn hàng.  
PUT /api/orders/:orderId/payment cập nhật trạng thái thanh toán.

**E. Đánh giá (Reviews)**

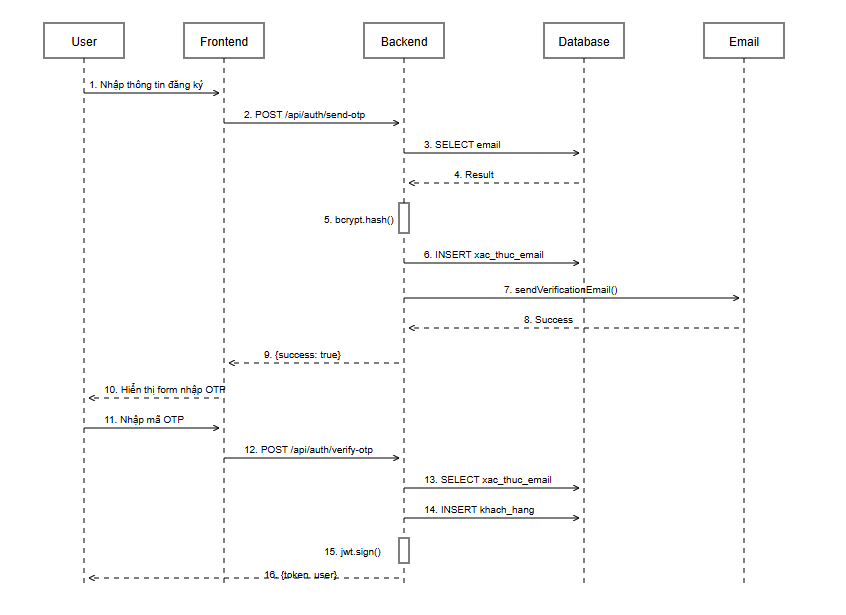
GET /api/reviews/product/:productId lấy danh sách đánh giá theo sản phẩm.  
GET /api/reviews/product/:productId/stats thống kê đánh giá sản phẩm.  
POST /api/reviews thêm đánh giá mới.  
PUT /api/reviews/:id cập nhật đánh giá.  
DELETE /api/reviews/:id xóa đánh giá.  
GET /api/reviews/can-review/:productId/:userId kiểm tra quyền đánh giá.  
GET /api/reviews/user/:userId lấy danh sách đánh giá của người dùng.

**F. Tin tức (News)**

GET /api/news lấy danh sách tin tức.  
GET /api/news/featured lấy tin tức nổi bật.  
GET /api/news/:id xem chi tiết tin tức.  
POST /api/news thêm tin tức mới (quản trị viên).  
PUT /api/news/:id cập nhật tin tức (quản trị viên).  
DELETE /api/news/:id xóa tin tức (quản trị viên).

### Sơ đồ tuần tự (Sequence Diagram)

1. **Quy trình đăng ký tài khoản**



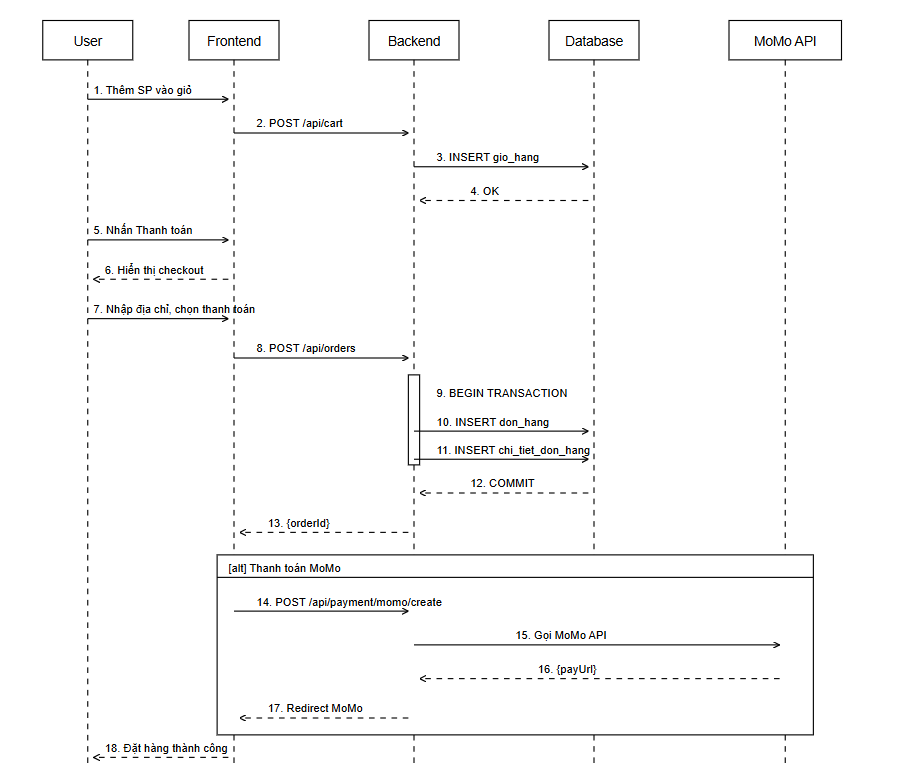
Hình 5.Quy trình Đăng ký

1. **Quy trình đăng nhập**



Hình 6. Quy trình đăng nhập

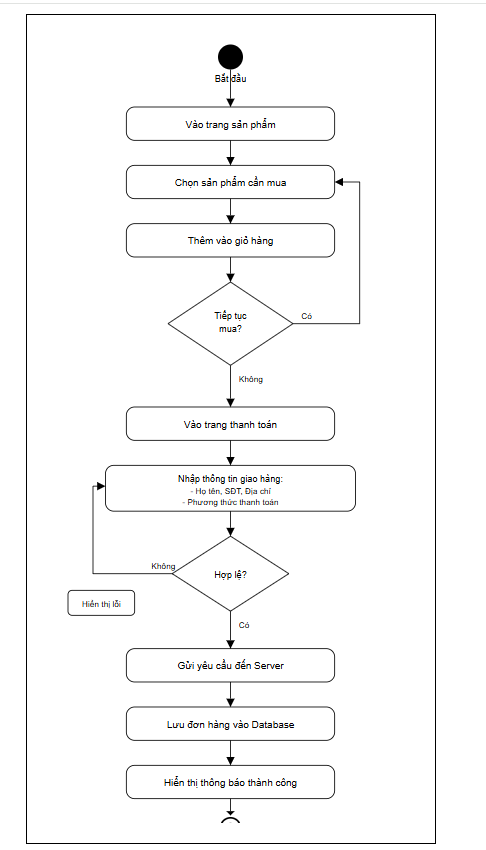
1. **Quy trình đặt hàng và thanh toán**

****

Hình 7. Quy trình đặt hàng và thanh toán

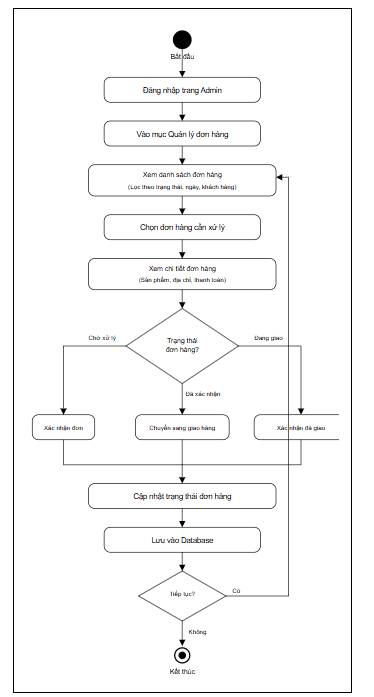
## Mô hình hoạt động của hệ thống

**Quy trình đặt hàng trực tuyến**



Hình 8. Sơ đồ hoạt động đặt hàng trực tuyến

**Quy Trình Quản lý đơn hàng (Admin)**

****

Hình 9. Sơ đồ hoạt động quản lý đơn hàng

**Kết luận**

Chương 3 đã trình bày tổng quan quá trình hiện thực hóa hệ thống website Cửa hàng điện thoại QuangHung Mobile. Nội dung bao gồm kiến trúc Client–Server, các yêu cầu chức năng của hệ thống, thiết kế dữ liệu và thiết kế xử lý. Bên cạnh đó, chương cũng mô tả các RESTful endpoint, cơ chế xác thực OTP, tích hợp thanh toán MoMo và đăng nhập Google OAuth. Cuối cùng, các sơ đồ tuần tự và sơ đồ hoạt động giúp làm rõ quy trình đăng ký, đặt hàng và quản lý đơn hàng, tạo nền tảng cho việc triển khai và đánh giá hệ thống ở các chương tiếp theo.

# KẾT QUẢ NGHIÊN CỨU

## Tổng quan

Sau quá trình phân tích, thiết kế và triển khai, hệ thống website Cửa hàng điện thoại QuangHung Mobile đã được hoàn thành với đầy đủ các chức năng theo yêu cầu đặt ra. Hệ thống gồm hai phần chính. Giao diện người dùng (Frontend) cho phép khách hàng xem và tìm kiếm sản phẩm, đăng ký – đăng nhập, quản lý giỏ hàng, đặt hàng, thanh toán và theo dõi đơn hàng. Giao diện quản trị (Admin) cho phép quản trị viên quản lý toàn bộ hoạt động của cửa hàng như sản phẩm, danh mục, khách hàng, đơn hàng, tin tức, khuyến mãi và thống kê hệ thống.

### Công nghệ sử dụng

## Giao diện người dùng

### Giao diện trang chủ

Mô tả: Trang chủ là điểm tiếp xúc đầu tiên với khách hàng, hiển thị banner quảng cáo các chương trình khuyến mãi, sản phẩm giá sốc và điện thoại chính hãng từ các thương hiệu lớn như iPhone, Samsung, OPPO, Xiaomi của cửa hàng.

Có các chức năng chính sau:

Banner slider tự động với 3 slide quảng cáo (OPPO, Samsung, khuyến mãi), có nút điều khiển và indicator dots.

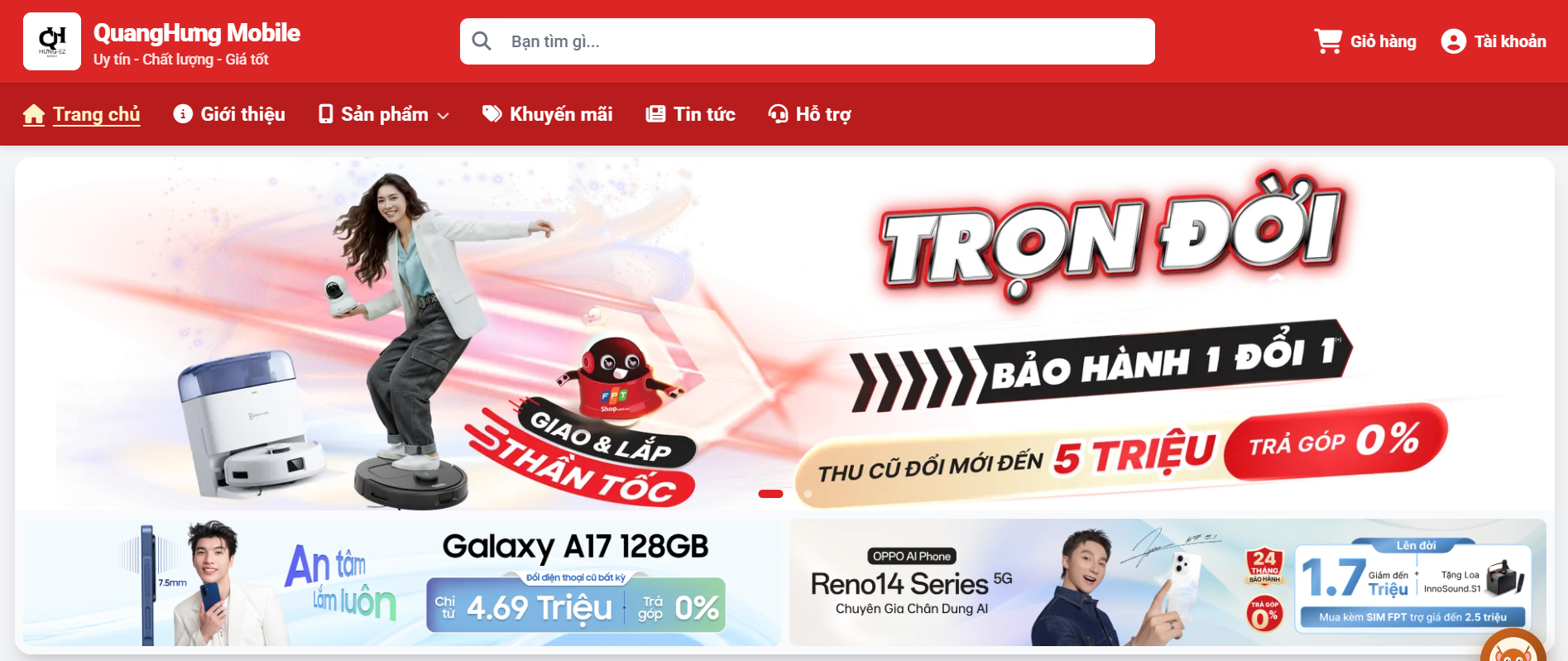
Section "Giá Sốc Mỗi Ngày" với đồng hồ đếm ngược và grid 5 sản phẩm deal hot.

Section "Sản Phẩm Chính Hãng" hiển thị các điện thoại có bảo hành toàn quốc.

Mỗi card sản phẩm hiển thị: hình ảnh, badge giảm giá, tên, giá khuyến mãi/giá gốc, đánh giá sao, nút "Mua ngay" và "Thêm giỏ hàng".

Thiết kế responsive, tối ưu SEO với meta tags và Open Graph.

Header component với logo, menu điều hướng, giỏ hàng, tài khoản.



Hình 10. Giao diện trang chủ

### Giao diện trang giới thiệu

Mô tả: Trang giới thiệu cung cấp thông tin chi tiết về QuangHưng Mobile - cửa hàng điện thoại uy tín với hơn 10 năm kinh nghiệm, giúp khách hàng hiểu rõ về thương hiệu, giá trị cốt lõi và đội ngũ lãnh đạo.



Hình 11. Giao diện trang giới thiệu.

### Giao diện trang sản phẩm

Mô tả: Trang danh sách sản phẩm cho phép khách hàng duyệt, tìm kiếm và lọc điện thoại theo nhiều tiêu chí khác nhau. Giao diện được thiết kế theo phong cách hiện đại phù hợp với người dùng.

Chức năng chính:

Banner slider quảng cáo với 3 slide (iPhone, Samsung, Khuyến mãi), có nút điều khiển và indicator dots

Thanh thương hiệu nổi bật: 8 hãng (Apple, Samsung, Xiaomi, OPPO, Vivo, HONOR, Realme, TECNO) - scroll ngang trên mobile, grid trên desktop

Sidebar bộ lọc tìm kiếm (sticky) với các tiêu chí cơ bản là:

Hãng sản xuất (checkbox với số lượng sản phẩm)

Loại phụ kiện (tai nghe, sạc, cáp, ốp lưng, cường lực...)

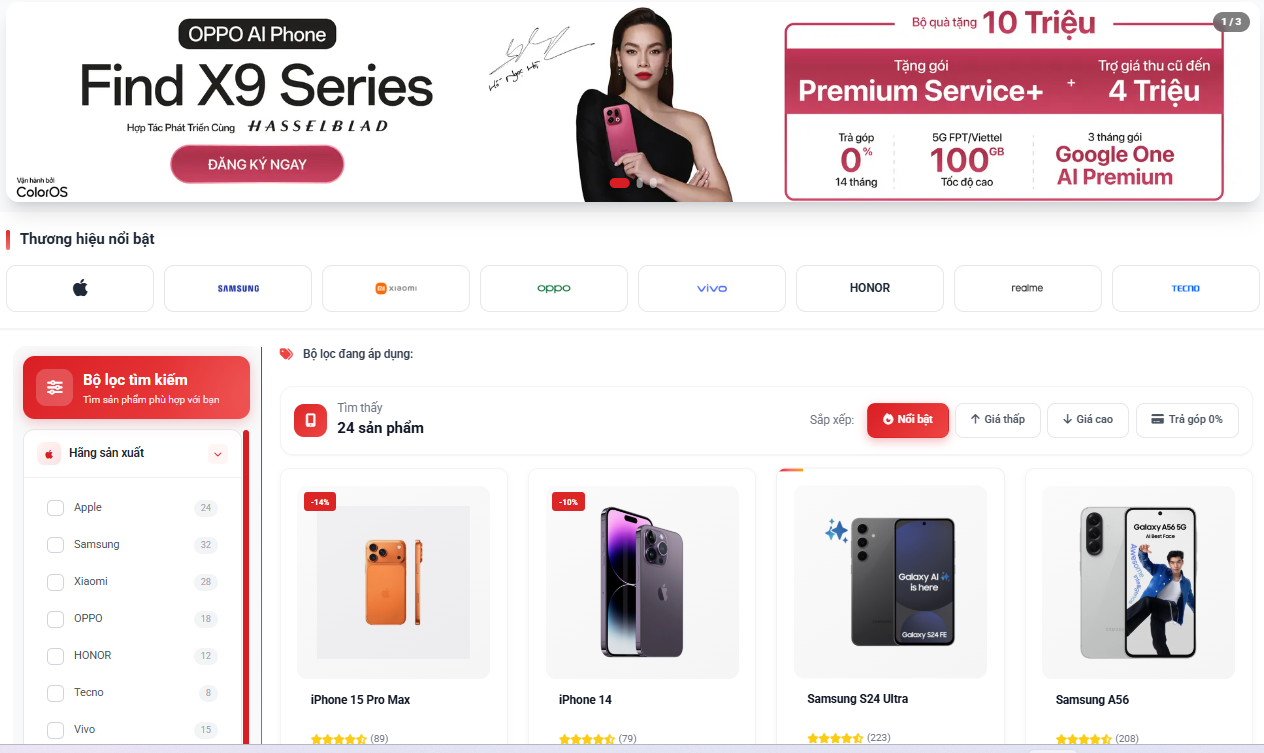
Mức giá (6 khoảng giá + nhập khoảng giá tùy chỉnh)

Hệ điều hành (iOS, Android)

Dung lượng ROM (128GB, 256GB, 512GB, 1TB)

Kết nối (NFC, Bluetooth, Hồng ngoại)

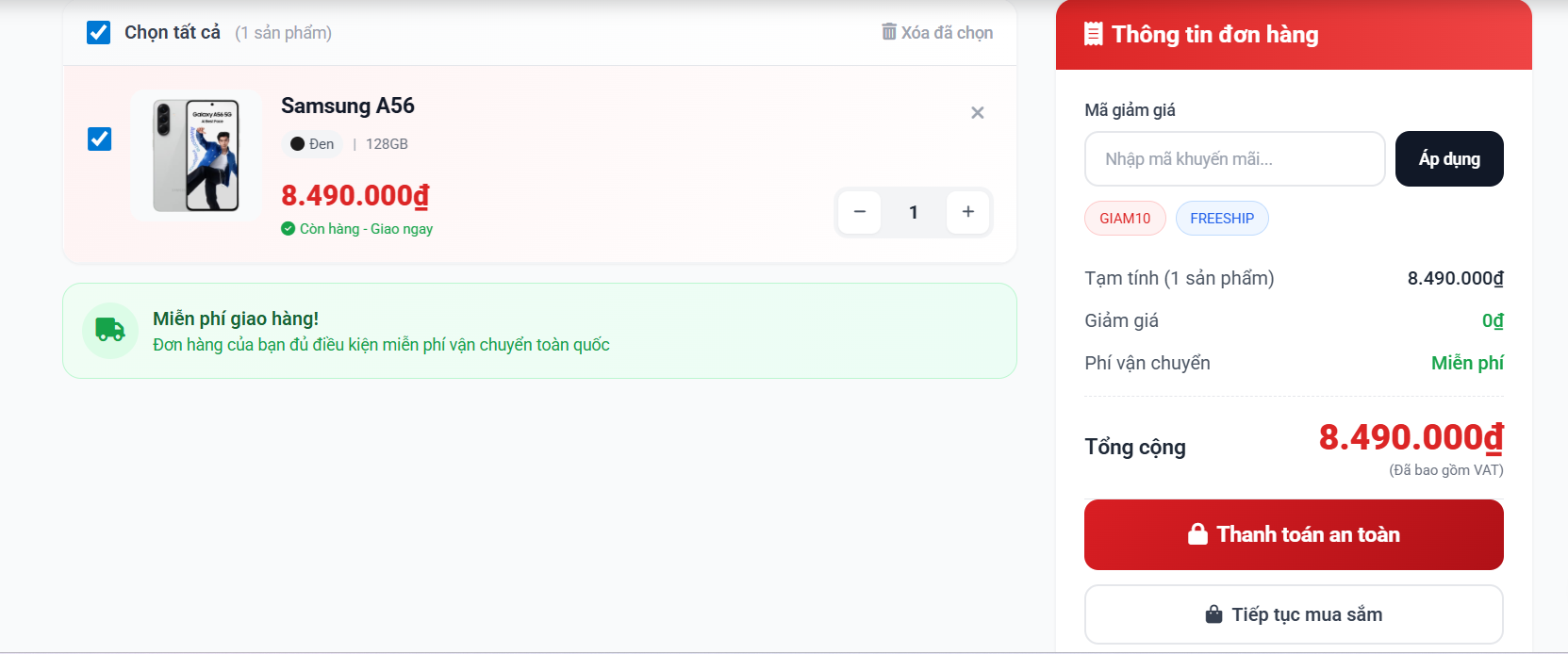
Hiệu năng và Pin (theo dung lượng mAh)



Hình 12. Giao diện trang sản phẩm

### Giao diện giỏ hàng

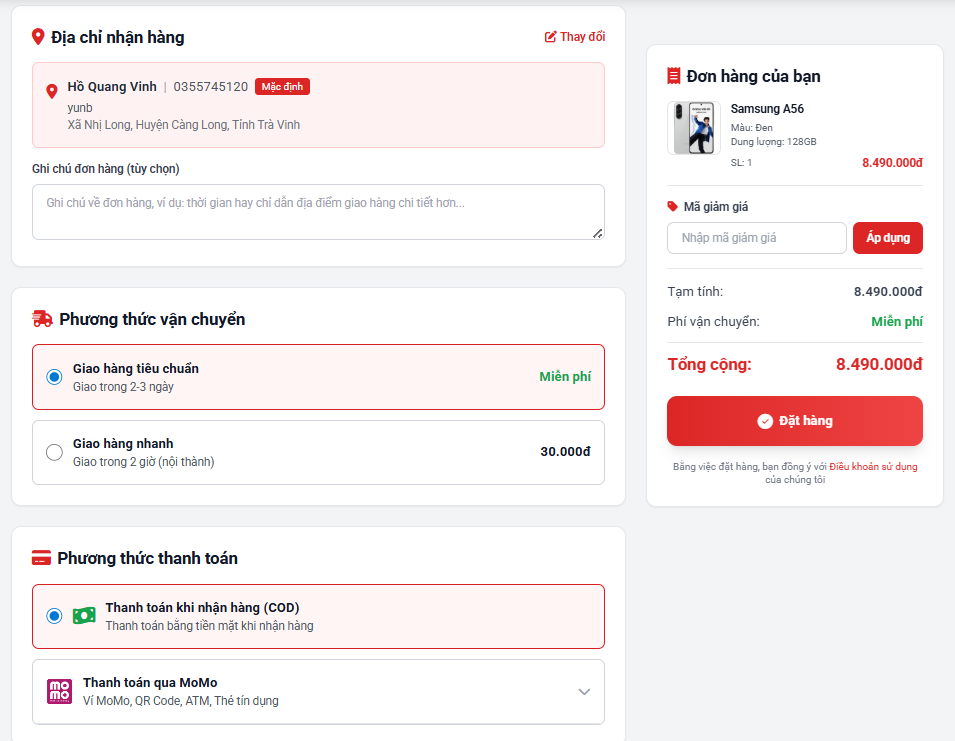
Mô tả: Trang giỏ hàng hiển thị danh sách sản phẩm đã thêm vào giỏ, cho phép khách hàng quản lý số lượng, áp dụng mã giảm giá và tiến hành thanh toán.



Hình 13. Giao diện trang giỏ hàng

### Giao diện thanh toán

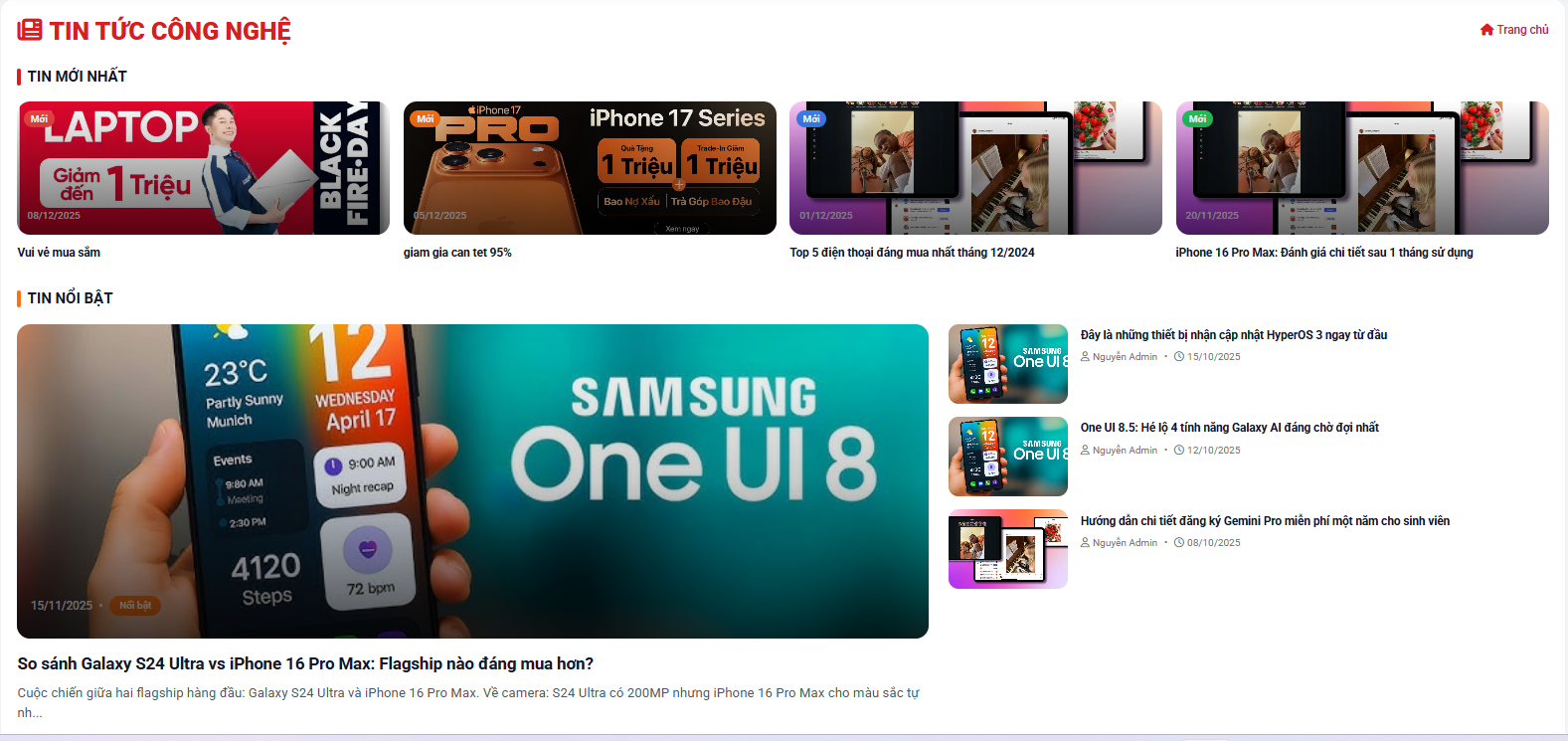
Mô tả: Trang thanh toán cho phép khách hàng hoàn tất đơn hàng với thông tin giao hàng, chọn phương thức vận chuyển và thanh toán.



Hình 14. Giao diện trang thanh toán

### Giao diện trang tin tức

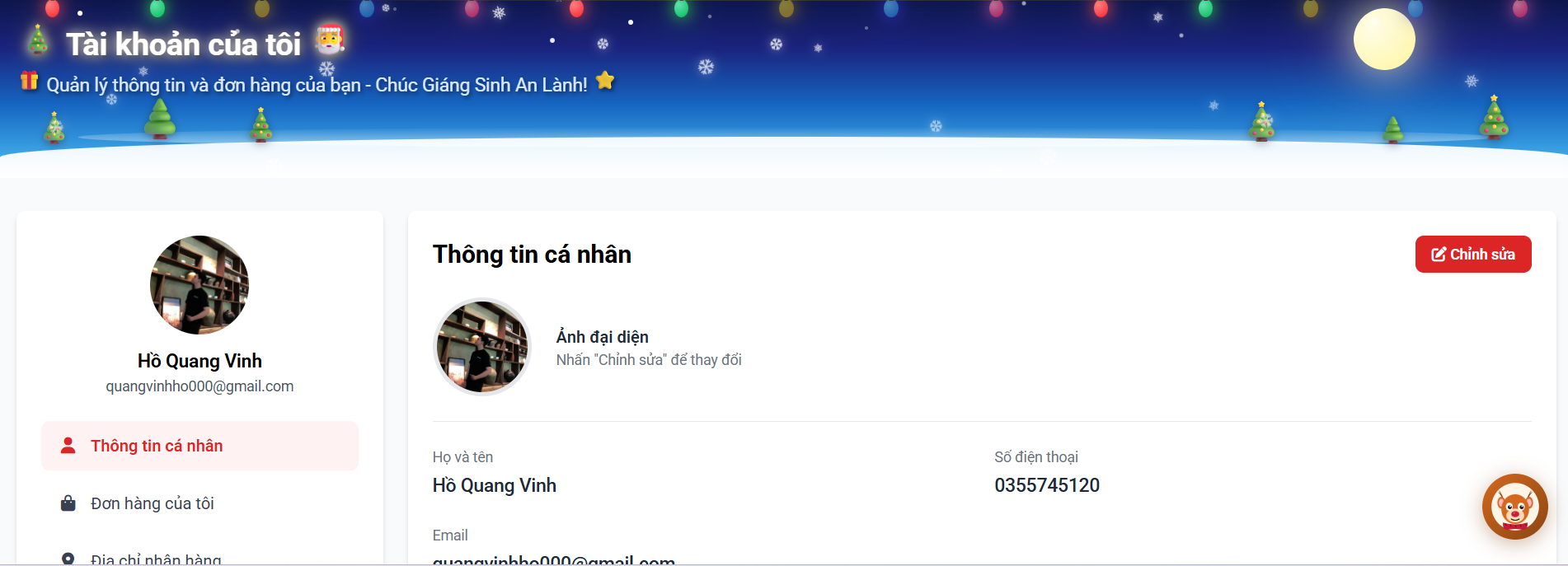
Mô tả: Trang tin tức hiển thị các bài viết về công nghệ, điện thoại, khuyến mãi giúp khách hàng cập nhật thông tin mới nhất từ QuangHưng Mobile.



Hình 15. Giao diện trang tin tức

### Giao diện tài khoản người dùng

Mô tả: Trang quản lý tài khoản cho phép người dùng xem và chỉnh sửa thông tin cá nhân, quản lý đơn hàng và địa chỉ nhận hàng.



Hình 16. Giao diện trang người dùng

## Giao diện Admin

### Dashboard

Mô tả: Trang quản trị dành cho Admin để theo dõi hoạt động kinh doanh của cửa hàng QuangHưng Mobile.

Đơn hàng

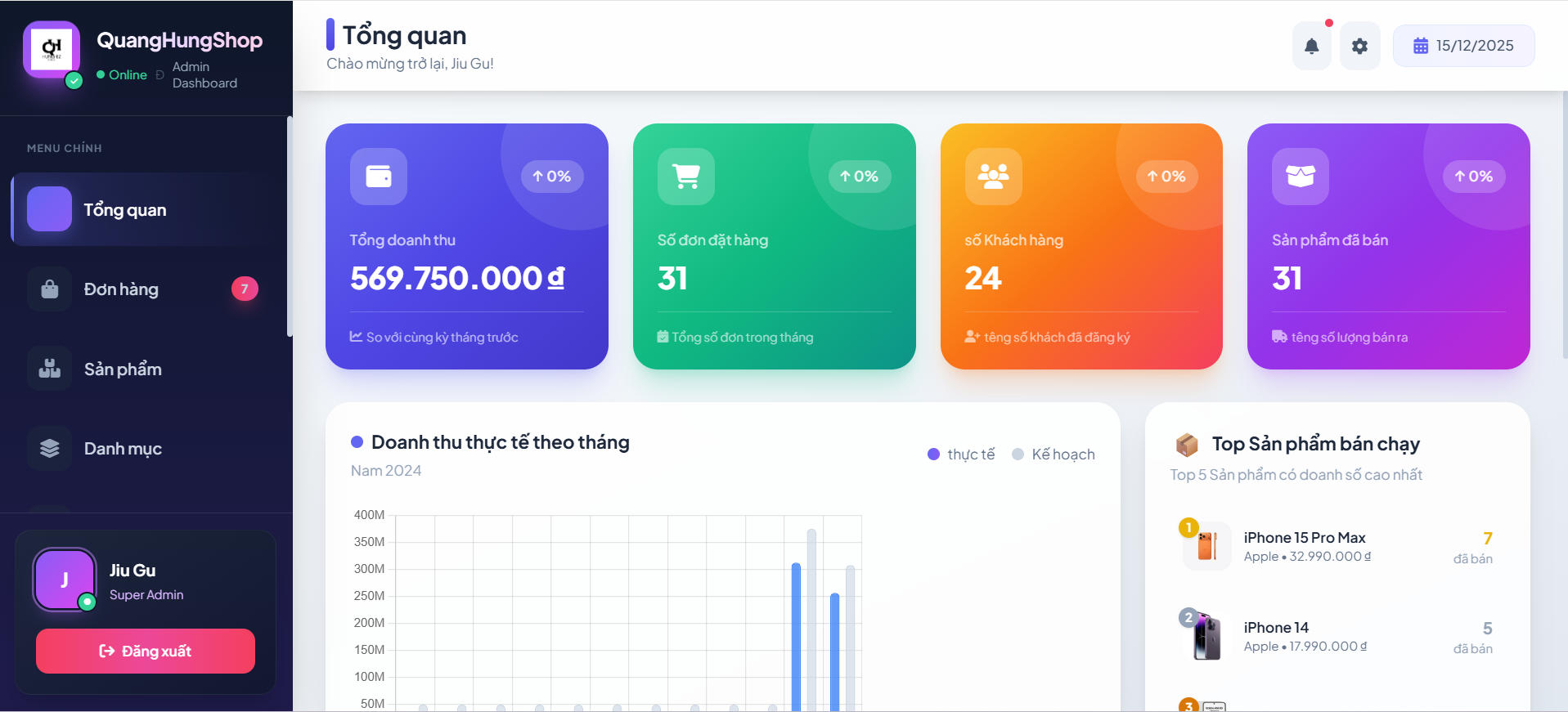
Sản phẩm

Danh mục

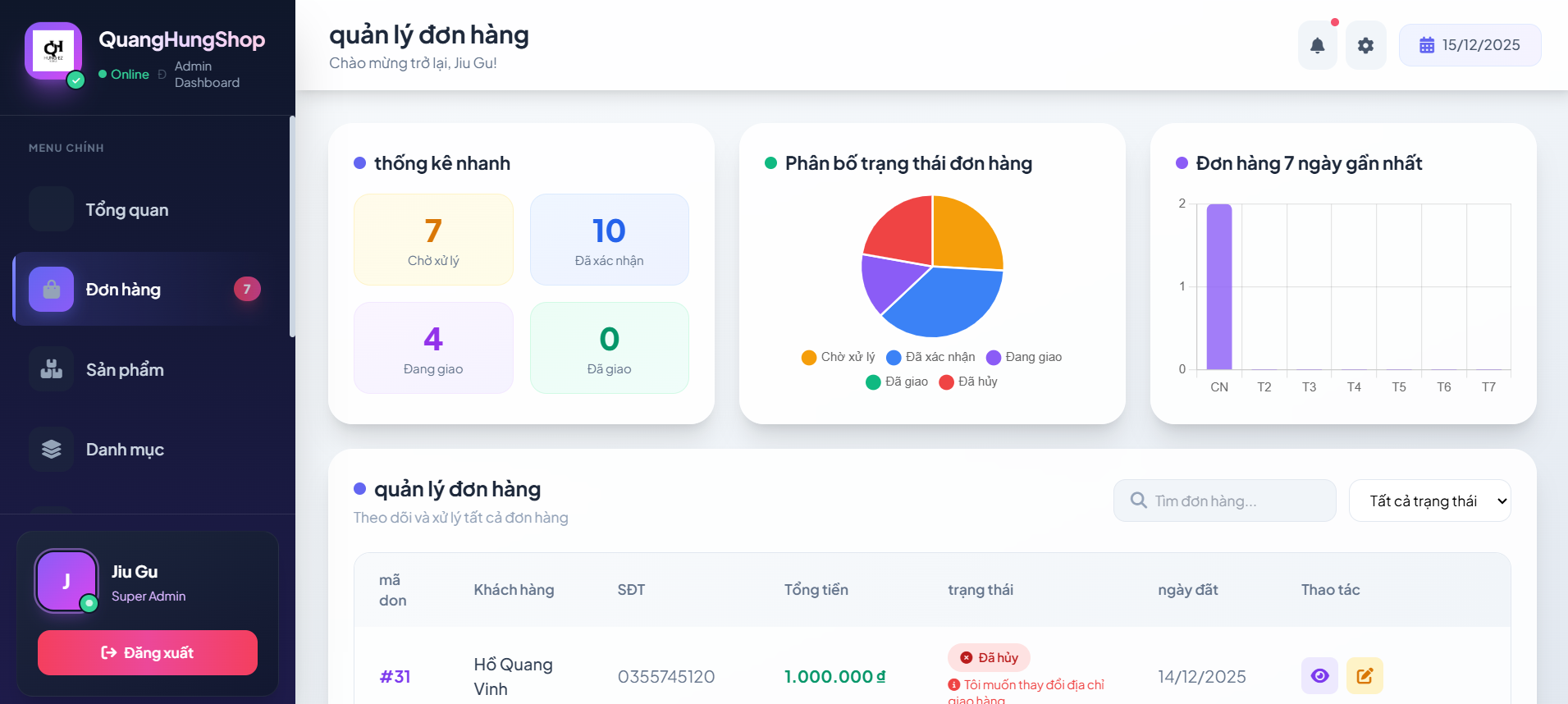
Khách hàng

Tin tức

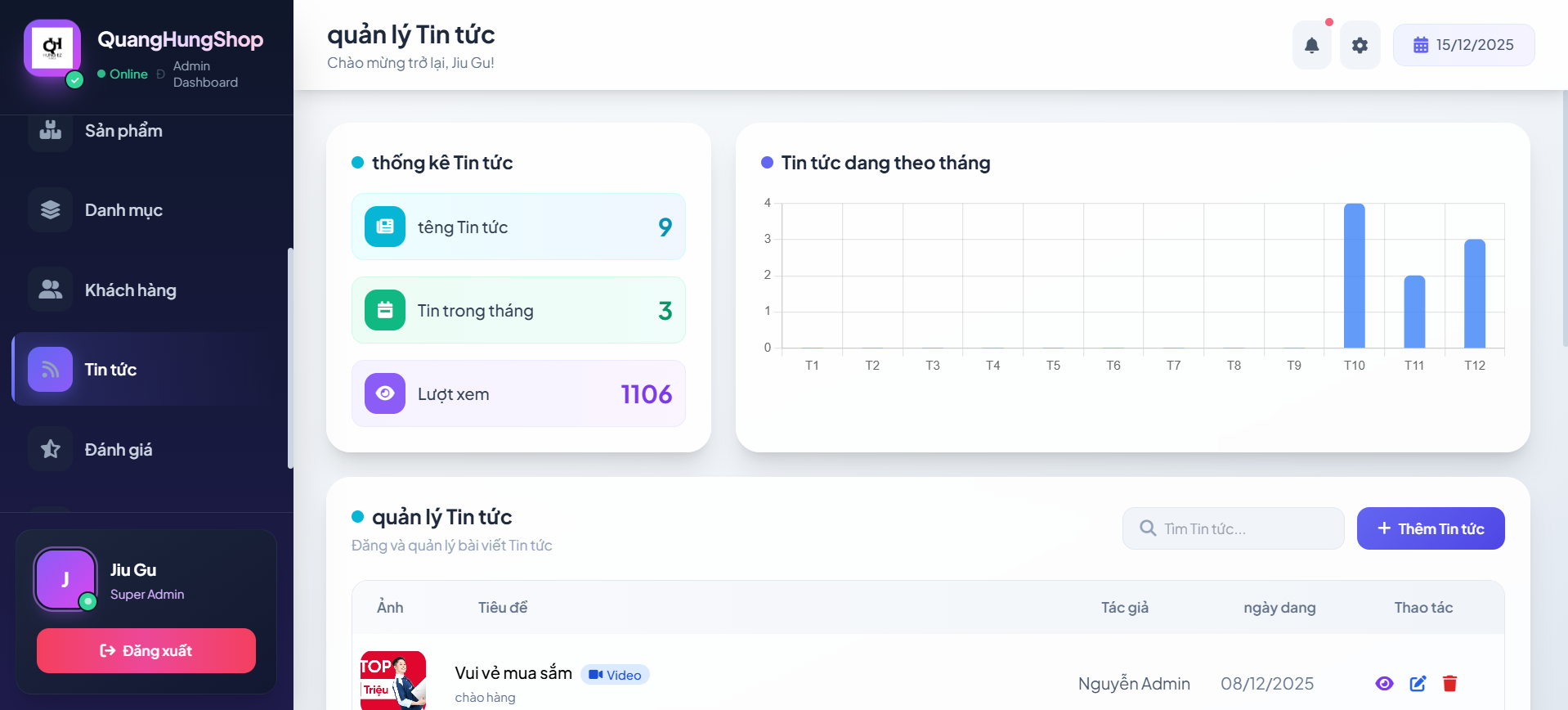
Đánh giá

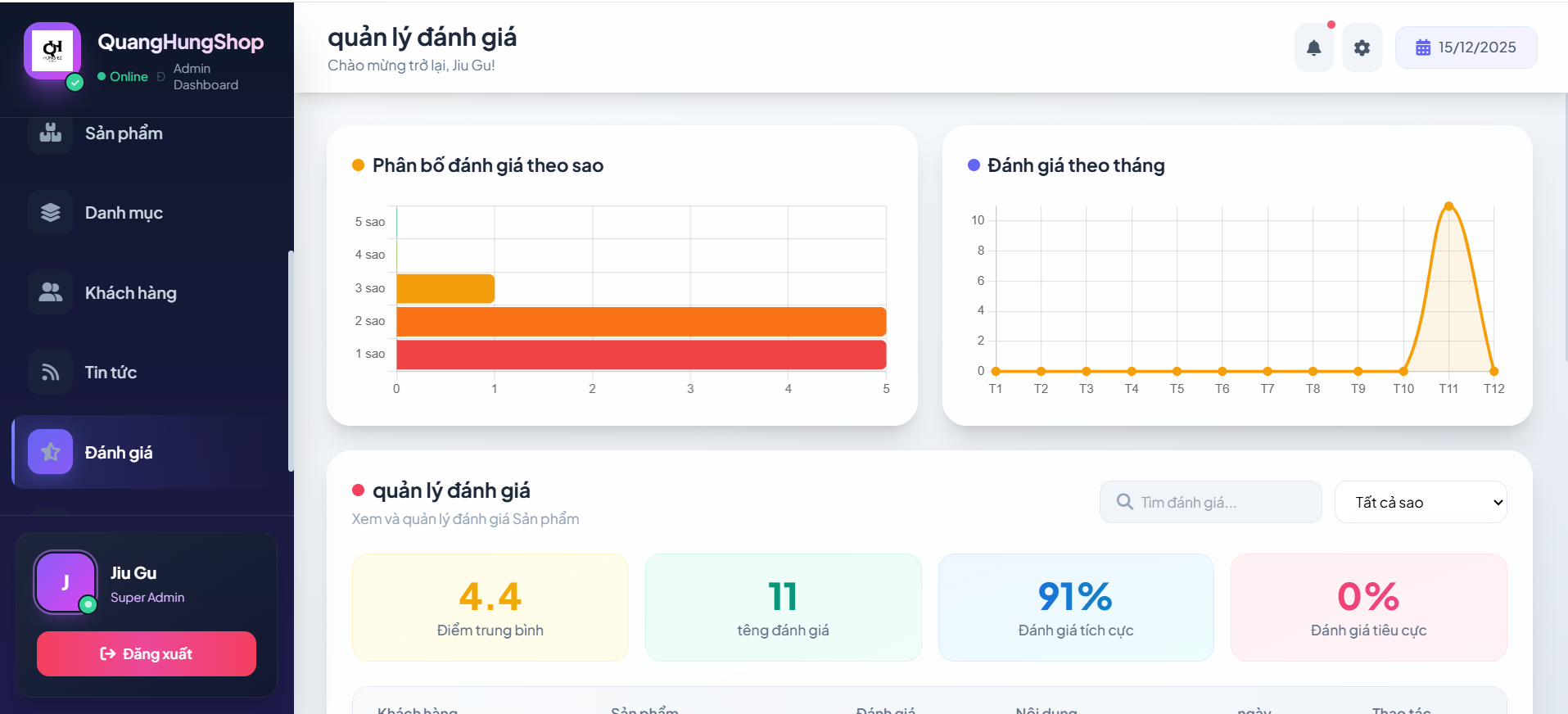


Hình 17. Giao diện trang Dashboard









Hình 18. Các trang giao diện khác

## Kết luận

Giao diện người dùng được thiết kế theo tiêu chuẩn hiện đại, thân thiện, dễ sử dụng. Khách hàng có thể dễ dàng tìm kiếm sản phẩm, mua hàng, thanh toán và quản lý tài khoản. Admin có thể theo dõi hoạt động kinh doanh, quản lý sản phẩm, đơn hàng một cách hiệu quả thông qua dashboard trực quan.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

### Kết quả đạt được

Sau quá trình nghiên cứu, phân tích, thiết kế và triển khai, đồ án đã hoàn thành việc xây dựng Website cửa hàng bán điện thoại Quang Hưng Mobile – một hệ thống thương mại điện tử hoàn chỉnh, hỗ trợ hiệu quả cho hoạt động kinh doanh và quản lý bán hàng trực tuyến, với các kết quả chính sau:

**Về mặt chức năng:**

Xây dựng thành công hệ thống website bán điện thoại với đầy đủ chức năng dành cho khách hàng như: xem danh sách sản phẩm, tìm kiếm và lọc theo nhiều tiêu chí, xem chi tiết sản phẩm, quản lý giỏ hàng, đặt hàng và thanh toán trực tuyến.

Phát triển hệ thống quản trị (Admin) với giao diện dashboard trực quan, cho phép quản lý toàn diện sản phẩm, danh mục, khách hàng, đơn hàng, tin tức và thống kê doanh thu.

Hỗ trợ đăng ký, đăng nhập người dùng bằng tài khoản thông thường và Google OAuth.

Tích hợp thanh toán QR Code có giới hạn thời gian nhằm đảm bảo an toàn và hạn chế gian lận.

Xây dựng hệ thống tin tức và khuyến mãi giúp cửa hàng truyền tải thông tin đến khách hàng một cách hiệu quả.

**Về mặt kỹ thuật:**

Áp dụng kiến trúc **Client – Server** kết hợp **RESTful API**.

Sử dụng các công nghệ hiện đại như **Node.js, Express.js, MySQL và Tailwind CSS**.

Thiết kế cơ sở dữ liệu quan hệ chuẩn hóa, đáp ứng tốt nhu cầu lưu trữ và truy xuất dữ liệu.

Xây dựng hệ thống API phục vụ các chức năng chính của website như xác thực, sản phẩm, giỏ hàng, đơn hàng và tin tức.

Áp dụng cơ chế xác thực **JWT** nhằm đảm bảo an toàn thông tin người dùng.

Thiết kế giao diện **responsive**, hoạt động ổn định trên Desktop, Tablet và Mobile.

**Về mặt giao diện:**

Giao diện website được thiết kế hiện đại, thân thiện với người dùng và dễ thao tác.

Sử dụng Tailwind CSS giúp giao diện đồng nhất, dễ mở rộng và bảo trì.

Bố cục rõ ràng, màu sắc hài hòa, phù hợp với lĩnh vực kinh doanh thiết bị công nghệ.

Hiệu năng website ổn định, thời gian tải trang nhanh, đáp ứng tốt trải nghiệm người dùng.

### Ý nghĩa thực tiễn

**Đối với cửa hàng:**

Hệ thống giúp số hóa quy trình kinh doanh, quản lý hiệu quả sản phẩm, đơn hàng và khách hàng; đồng thời mở rộng kênh bán hàng trực tuyến, góp phần nâng cao hiệu quả kinh doanh và tăng doanh thu.

**Đối với khách hàng:**

Website mang lại trải nghiệm mua sắm điện thoại tiện lợi, cho phép tìm kiếm và đặt mua sản phẩm mọi lúc, mọi nơi, với nhiều hình thức thanh toán linh hoạt và an toàn.

**Đối với lĩnh vực thương mại điện tử:**

Đồ án cung cấp một mô hình website bán lẻ thiết bị công nghệ có tính ứng dụng cao, có thể làm tài liệu tham khảo cho việc xây dựng và triển khai các hệ thống thương mại điện tử tương tự.

## Hạn chế

Bên cạnh những kết quả đạt được, đồ án vẫn còn tồn tại một số hạn chế như:

Chưa phát triển ứng dụng mobile native, hệ thống hiện chỉ hoạt động dưới dạng website responsive.

Chưa tích hợp nhiều cổng thanh toán điện tử phổ biến như VNPay, ZaloPay hoặc thẻ quốc tế.

Chưa có chương trình khách hàng thân thiết và hệ thống tích điểm.

Chưa tích hợp chatbot tư vấn tự động chuyên sâu cho lĩnh vực bán điện thoại.

Website chưa được tối ưu đầy đủ cho công cụ tìm kiếm (SEO).

Chưa tích hợp hệ thống phân tích hành vi người dùng (analytics).

## Hướng phát triển

Trong thời gian tới, hệ thống website Quang Hưng Mobile có thể được tiếp tục mở rộng và hoàn thiện theo các hướng sau:

Tích hợp thêm các cổng thanh toán trực tuyến phổ biến như ví điện tử và ngân hàng nhằm đa dạng hóa phương thức thanh toán cho người dùng.

Nâng cấp chatbot AI để hỗ trợ tư vấn thông minh hơn, cá nhân hóa nội dung dựa trên hành vi mua sắm của khách hàng.

Phát triển ứng dụng mobile (Android/iOS) nhằm mở rộng khả năng tiếp cận và nâng cao trải nghiệm người dùng.

Tăng cường các giải pháp bảo mật như xác thực đa yếu tố, mã hóa dữ liệu và kiểm soát truy cập nâng cao.

Tối ưu hiệu năng hệ thống, mở rộng khả năng chịu tải để đáp ứng số lượng người dùng lớn khi triển khai thực tế.

Những hướng phát triển trên sẽ giúp hệ thống hoàn thiện hơn, đáp ứng tốt hơn nhu cầu thực tế và nâng cao giá trị ứng dụng của đề tài trong tương lai.

## Lời kết

Đồ án “Xây dựng Website cửa hàng bán điện thoại Quang Hưng Mobile” đã hoàn thành các mục tiêu đề ra, xây dựng được một hệ thống thương mại điện tử tương đối hoàn chỉnh, đáp ứng tốt các yêu cầu về chức năng, kỹ thuật và giao diện. Kết quả đạt được không chỉ giúp sinh viên củng cố kiến thức chuyên môn mà còn tạo nền tảng cho việc nghiên cứu và phát triển các hệ thống web thương mại điện tử trong thực tế.

TÀI LIỆU THAM KHẢO

[1] Nguyễn Văn Minh, *Phân tích và thiết kế hệ thống thông tin*, NXB Khoa học và Kỹ thuật, Hà Nội, 2018.

[2] Trần Thị Thu Hà, *Giáo trình Phát triển ứng dụng Web*, NXB Đại học Quốc gia TP. Hồ Chí Minh, 2019.

[3] Lê Văn Hòa, *Cơ sở dữ liệu và hệ quản trị cơ sở dữ liệu*, NXB Thống Kê, Hà Nội, 2017.

[4] Nguyễn Thành Nam, *Lập trình Web với NodeJS và Express*, NXB Xây Dựng, Hà Nội, 2020.

[5] Bộ Thông tin và Truyền thông, “Thương mại điện tử Việt Nam – Xu hướng và giải pháp,” 2023. [Trực tuyến]. Có tại: <https://www.moit.gov.vn>. [Truy cập: Thg 12, 2025].

[6] Trường Đại học Công nghệ Thông tin, *Bài giảng Thương mại điện tử*, TP. Hồ Chí Minh, 2022.

[7] I. Sommerville, *Software Engineering*, 10th ed. Boston, MA, USA: Pearson Education, 2016.

[8] R. Fielding, “Architectural Styles and the Design of Network-based Software Architectures,” Ph.D. dissertation, University of California, Irvine, USA, 2000.

[9] Node.js Foundation, “Node.js Documentation,” 2024. [Online]. Available: <https://nodejs.org/en/docs>. [Accessed: Dec. 2025].