

[illegible]

Vĩnh Long, ngày tháng năm
Giảng viên hướng dẫn
(Ký tên và ghi rõ họ tên)

[illegible]

Thành viên hội đồng
(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Đặc biệt, em xin chân thành cảm ơn Thầy Thạc sĩ Lê Minh Tự đã luôn dành thời gian, tâm huyết để hướng dẫn, góp ý và giải đáp những thắc mắc của em trong suốt quá trình nghiên cứu và thực hiện đề tài. Những ý kiến và định hướng quý báu của Thầy Lê Minh Tự đã giúp em hiểu rõ hơn về nội dung đề tài, từ đó có thể hoàn thiện đồ án theo đúng mục tiêu và yêu cầu đề ra.

Em cũng xin gửi lời cảm ơn đến gia đình và bạn bè đã luôn động viên, khích lệ và hỗ trợ em trong quá trình học tập cũng như khi thực hiện đồ án. Sự quan tâm và giúp đỡ của mọi người là nguồn động lực to lớn giúp em vượt qua những khó khăn và hoàn thành đề tài đúng tiến độ.

Mặc dù đã rất cố gắng, nhưng do kiến thức và thời gian có hạn, đồ án chắc chắn không tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp quý báu từ quý thầy cô và các bạn để đề tài có thể được hoàn thiện hơn trong tương lai.

Cuối cùng, em xin chân thành cảm ơn và kính chúc quý thầy cô cùng các bạn luôn dồi dào sức khỏe, hạnh phúc và thành công trong công tác và học tập.

Trân trọng.

Vĩnh Long, ngày tháng năm

SINH VIÊN

Lê khánh duy

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN.....	14
1.1. Tổng quan về website bán linh kiện máy tính.....	14
1.2. Nhu cầu lắp ráp máy tính trong bối cảnh công nghệ số	14
1.3. Thực trạng mua sắm linh kiện máy tính hiện nay	15
1.4. Vấn đề cần giải quyết	15
1.5. Định hướng giải pháp của đề tài.....	16
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT	17
2.1. ReactJS (Xây dựng giao diện người dùng)	17
2.1.1. Tổng quan và vai trò	17
2.1.2. Kiến trúc hướng thành phần (Component-based)	18
2.2. Node.js (Môi trường thực thi phía Server)	19
2.2.1. Giới thiệu chung	19
2.2.2. Khả năng xử lý và hiệu năng	20
2.3. Axios (Giao tiếp HTTP Client)	20
2.3.1. Vai trò cầu nối dữ liệu	20
2.3.2. Tính năng xử lý phản hồi.....	21
2.4. Express (Framework xây dựng Backend API).....	21
2.4.1. Cấu trúc hóa hệ thống Backend	21
2.4.2. Middleware (Phần mềm trung gian)	22
2.5. Mongoose (Quản lý Cơ sở dữ liệu MongoDB).....	22
2.5.1. Mô hình hóa dữ liệu (Data Modeling)	22
2.5.2. Kiểm soát và xác thực dữ liệu	23
2.6. JsonWebToken - JWT (Bảo mật và Xác thực).....	23
2.6.1. Cơ chế xác thực hiện đại.....	23
2.6.2. Phân quyền người dùng (Authorization)	24

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU	25
3.1. Mô tả hệ thống.....	25
3.2. Phân tích yêu cầu hệ thống.....	26
3.2.1. Yêu cầu chức năng.....	26
3.2.1.1. Chức năng cho người dùng (Khách hàng)	26
3.2.1.2. Chức năng cho Nhân viên.....	27
3.2.1.3. Chức năng cho quản trị viên (Admin)	27
3.2.2. Yêu cầu phi chức năng.....	28
3.2.3. Sơ đồ Use Case	29
3.3. Thiết kế hệ thống.....	30
3.3.1. Thiết kế kiến trúc tổng thể hệ thống	30
3.3.2. Thiết kế cơ sở dữ liệu	31
3.3.3. Thiết kế giao diện người dùng	42
3.4. Thiết kế, kiến trúc ứng dụng.....	45
3.4.1. Xây dựng backend	45
3.4.2. Xây dựng frontend.....	47
3.4.3. Xây dựng chức năng giao tiếp thời gian thực (Socket.IO).....	48
3.5. Cài đặt và triển khai hệ thống.....	49
3.5.1. Môi trường và công cụ phát triển	49
3.5.2. Triển khai và chạy thử nghiệm	49
CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU	51
4.1. Kiểm thử API với Postman	51
4.2. Giao diện người dùng	60
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	68
5.1. Kết luận	68
5.2. Hướng phát triển.....	68

DANH MỤC TÀI LIỆU THAM KHẢO	70
Tài liệu và Giáo trình (Lý thuyết nền tảng).....	70
Nguồn Hình ảnh và Dữ liệu sản phẩm	70
Tài liệu Kỹ thuật và Cộng đồng lập trình (Code & Giải thuật).....	70

DANH MỤC HÌNH ẢNH

Hình 2.1. Logo ReactJS	17
Hình 2.2. Kiến trúc SPA so với MPA (Dùng cho phần ReactJS).....	18
Hình 2.3. Cấu trúc cây Component trong ứng dụng ReactJS	19
Hình 2.4. Logo Node.JS.....	19
Hình 2.5. Luồng dữ liệu với Axios (Dùng cho phần Axios).....	21
Hình 2.6. Kiến trúc Express & Routing (Dùng cho phần Express)	22
Hình 2.7. Mongoose Schema (Dùng cho phần Mongoose)	23
Hình 2.7. Quy trình xác thực JWT (Dùng cho phần JWT).....	24
Hình 3.1. Sơ đồ Use Case	29
Hình 3.2. Sơ đồ kiến trúc tổng thể hệ thống	31
Hình 3.3. Mô hình dữ liệu MongoDB của hệ thống	32
Hình 3.4. Phác thảo giao diện trang chủ	43
Hình 3.5. Phác thảo giao diện chi tiết sản phẩm.....	44
Hình 3.6. Phác thảo giao diện giỏ hàng và thanh toán.....	44
Hình 3.7. Phác thảo giao diện trang quản trị.....	45
Hình 3.8. Cấu trúc thư mục backend.....	47
Hình 3.9. Cấu trúc thư mục frontend	48
Hình 4.1. Kiểm thử api đăng ký.....	51
Hình 4.2. Kiểm thử api đăng nhập	52
Hình 4.3. Kiểm thử api đặt lại mật khẩu khi quên	52
Hình 4.4. Kiểm thử api lấy danh sách linh kiện (có phân trang)	53
Hình 4.5. Kiểm thử api tìm sản phẩm theo tiêu chí (Giá, Thương hiệu, Danh mục).....	54
Hình 4.6. Kiểm thử api xem chi tiết thông số kỹ thuật.....	54
Hình 4.7. Kiểm thử api thêm mới sản phẩm (quyền Admin).	55
Hình 4.8. Kiểm thử api thêm sản phẩm vào giỏ hàng.....	55

Hình 4.9. Kiểm thử api thêm mới sản phẩm vào giỏ hàng	56
Hình 4.10. Kiểm thử api tạo đơn hàng và thanh toán thành công.....	56
Hình 4.11. Kiểm thử api gửi đánh giá cho sản phẩm đã mua thành công	57
Hình 4.12. Kiểm thử api lấy danh sách đánh giá công khai	57
Hình 4.13. Kiểm thử api thống kê doanh thu của admin	58
Hình 4.14. Kiểm thử api quản lý danh sách người dùng	58
Hình 4.15. Kiểm thử api cập nhật trạng thái đơn hàng (Duyệt/Giao hàng).....	59
Hình 4.16. Kiểm thử api quản lý mã giảm giá	59
Hình 4.17. Giao diện đăng nhập.....	60
Hình 4.18. Giao diện đăng ký	60
Hình 4.22. Giao diện hiển thị hình ảnh rõ nét, giá bán, tình trạng kho và bảng thông số kỹ thuật chi tiết	63
Hình 4.23. Giao diện giỏ hàng	63
Hình 4.24. Giao diện thanh toán	64
Hình 4.25. Giao diện theo dõi đơn hàng	64
Hình 4.26. Giao diện thông tin người dùng	65
Hình 4.27. Giao diện trang Quản trị của Admin.....	65
Hình 4.28. Giao diện Quản lý đơn hàng	66
Hình 4.29. Giao diện Quản lý sản phẩm	66
Hình 4.30. Giao diện chat tư vấn khách hàng	67
Hình 4.31. Giao diện phản hồi chat của Admin.....	67

DANH MỤC BẢNG BIỂU

Bảng 3.1. Chi tiết các thực thể User.....	33
Bảng 3.2. Chi tiết các thực thể Products	34
Bảng 3.3. Chi tiết các thực thể Categories	35
Bảng 3.4. Chi tiết các thực thể Carts.....	36
Bảng 3.5. Chi tiết các thực thể Cart_Items	36
Bảng 3.6. Chi tiết các thực thể Orders	37
Bảng 3.7. Chi tiết thực thể Order_Items	38
Bảng 3.8. Chi tiết các thực thể Coupons.....	39
Bảng 3.9. Chi tiết các thực thể Reviews	40
Bảng 3.10. Chi tiết các thực thể Chat_Messages	41
Bảng 3.11. Chi tiết các thực thể Message_Details.....	41
Bảng 3.12. Chi tiết các thực thể Filters.....	42

TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

Vấn đề nghiên cứu

Trong kỷ nguyên thương mại điện tử, việc kinh doanh linh kiện máy tính đối mặt với những thách thức đặc thù:

- Dữ liệu phức tạp: Linh kiện máy tính (CPU, GPU, RAM, Mainboard...) có thông số kỹ thuật không đồng nhất, gây khó khăn cho việc quản lý bằng cơ sở dữ liệu quan hệ truyền thống.
- Trải nghiệm người dùng: Khách hàng cần một giao diện tốc độ cao, khả năng lọc sản phẩm thông minh và quy trình thanh toán tinh gọn.
- Bài toán quản lý: Nhu cầu về một hệ thống quản trị (Admin Dashboard) đồng bộ để theo dõi tồn kho và biến động giá theo thời gian thực.

Các hướng tiếp cận

Để giải quyết bài toán trên, đồ án lựa chọn các hướng tiếp cận sau:

- Tiếp cận kiến trúc Micro-services (đơn giản hóa): Tách biệt hoàn toàn Frontend và Backend thông qua chuẩn giao tiếp RESTful API.
- Tiếp cận dữ liệu phi cấu trúc (NoSQL): Sử dụng MongoDB để giải quyết sự đa dạng trong thuộc tính sản phẩm linh kiện máy tính, cho phép mở rộng trường dữ liệu mà không cần thay đổi cấu trúc bảng như SQL truyền thống.
- Tiếp cận Component-based: Sử dụng React.js để tái sử dụng các thành phần giao diện, tăng tốc độ phát triển và tính nhất quán của website.

Cách giải quyết vấn đề

Hệ thống được xây dựng dựa trên nền tảng **MERN Stack** với quy trình cụ thể:

- Xây dựng Cơ sở dữ liệu: Thiết kế các Schemas trong MongoDB để lưu trữ thông tin Người dùng, Sản phẩm và Đơn hàng.
- Phát triển Backend (Server-side): Sử dụng Node.js và Express để xây dựng các API endpoints, thực hiện các nghiệp vụ: xác thực người dùng (Authentication) bằng JWT, mã hóa mật khẩu bằng Bcrypt, và xử lý logic giỏ hàng/thanh toán.

- Phát triển Frontend (Client-side): Xây dựng giao diện Single Page Application (SPA) bằng React.js, gọi dữ liệu từ Backend thông qua Axios và quản lý trạng thái ứng dụng (State management).

Một số kết quả đạt được

- Hoàn thiện mô hình 3 vai trò: Phân quyền thành công cho Admin, Nhân viên và User với các không gian làm việc và quyền hạn riêng biệt, không chồng chéo.

- Quản lý dữ liệu linh hoạt: Xây dựng thành công cơ sở dữ liệu linh kiện có khả năng mở rộng thông tin chi tiết cho hàng nghìn mã sản phẩm khác nhau.

- Chu trình nghiệp vụ khép kín và tự động hóa: Hệ thống đã xử lý trọn vẹn luồng công việc thực tế:

- + Khách hàng: Thực hiện tìm kiếm, chọn lọc và khởi tạo đơn hàng thành công.

- + Nhân viên: Tiếp nhận thông báo, kiểm tra và chuyển trạng thái đơn hàng (Xác nhận/Đang giao/Hoàn tất).

- + Hệ thống: Tự động cập nhật số lượng tồn kho theo thời gian thực và lưu trữ lịch sử giao dịch vào cơ sở dữ liệu để phục vụ việc đối soát.

- Giao diện phản hồi tốt: Website đạt tốc độ tải trang nhanh và hiển thị tương thích (Responsive) trên các thiết bị, tạo nền tảng vững chắc để triển khai lên các dịch vụ Cloud trong giai đoạn tiếp theo.

MỞ ĐẦU

Lý do chọn đề tài

Trong kỷ nguyên số hóa, thương mại điện tử đã trở thành một phần không thể thiếu của nền kinh tế. Đặc biệt, nhu cầu xây dựng cấu hình máy tính (PC Building) và nâng cấp linh kiện công nghệ đang tăng trưởng mạnh mẽ. Tuy nhiên, linh kiện máy tính là nhóm hàng đặc thù với hệ thống thông số kỹ thuật đa dạng và phức tạp (Socket, Bus RAM, Chipset...), gây khó khăn cho người dùng trong việc tìm kiếm và so sánh nếu không có một nền tảng hỗ trợ chuyên biệt.

Về mặt kỹ thuật, các website truyền thống thường gặp hạn chế về tốc độ tải trang và khả năng mở rộng khi dữ liệu lớn. Để giải quyết vấn đề này, việc ứng dụng các công nghệ mới như Node.js, React và MongoDB là giải pháp tối ưu. Sự kết hợp này cho phép xây dựng ứng dụng đơn trang (Single Page Application) với trải nghiệm người dùng mượt mà, đồng thời tận dụng sức mạnh của cơ sở dữ liệu phi quan hệ (NoSQL) để lưu trữ linh hoạt các thuộc tính sản phẩm không đồng nhất.

Mục đích nghiên cứu

Đề tài tập trung vào hai mục đích chính:

Mục đích lý thuyết: Nghiên cứu và làm chủ quy trình phát triển ứng dụng web hiện đại; hiểu sâu về cơ chế hoạt động của RESTful API, xác thực bảo mật và thiết kế cơ sở dữ liệu với MongoDB.

Mục đích thực tiễn: Xây dựng hoàn thiện một hệ thống thương mại điện tử với quy trình nghiệp vụ khép kín, bao gồm:

- Cung cấp giao diện mua sắm trực quan, tốc độ cao cho khách hàng.
- Xây dựng công cụ quản lý đơn hàng và kho vận hiệu quả cho nhân viên.
- Cung cấp hệ thống quản trị, báo cáo thống kê doanh thu cho người quản lý.

Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

- Công nghệ: Các kỹ thuật lập trình JavaScript, thư viện React.js để xây dựng giao diện, môi trường Node.js cho phía máy chủ và hệ quản trị cơ sở dữ liệu MongoDB.

- Hệ thống: Các quy trình nghiệp vụ thương mại điện tử bao gồm: Quản lý danh mục sản phẩm, quy trình xử lý đơn hàng, quản lý phân quyền người dùng và bảo mật thông tin.

Phạm vi nghiên cứu:

Phạm vi chức năng: Hệ thống được thiết kế với 3 phân hệ người dùng rõ rệt:

- Khách hàng (User): Thực hiện tìm kiếm, lọc sản phẩm theo thông số kỹ thuật, quản lý giỏ hàng và đặt hàng trực tuyến.

- Nhân viên (Staff): Tiếp nhận đơn hàng từ hệ thống, quản lý và cập nhật trạng thái đơn hàng theo quy trình (*Xác nhận* → *Đang giao* → *Hoàn tất*), đồng thời cập nhật số lượng tồn kho.

- Quản trị viên (Admin): Quản lý toàn bộ hệ thống, quản lý tài khoản nhân viên/người dùng và xem báo cáo thống kê.

Phạm vi triển khai:

- Hệ thống được xây dựng, vận hành và kiểm thử hoàn thiện trên môi trường phát triển cục bộ (Localhost).

- Mô phỏng đầy đủ quy trình từ lúc đặt hàng đến khi hoàn tất đơn hàng và lưu trữ lịch sử giao dịch vào cơ sở dữ liệu.

CHƯƠNG 1: TỔNG QUAN

1.1. Tổng quan về website bán linh kiện máy tính

Website bán linh kiện máy tính là một hệ thống thương mại điện tử trực tuyến cho phép người dùng tìm kiếm, xem thông số kỹ thuật và thực hiện giao dịch mua sắm các thiết bị phần cứng. Nội dung trên website thường được tổ chức theo cấu trúc phân cấp gồm các danh mục sản phẩm (category) như CPU, RAM, VGA..., chi tiết sản phẩm và các bài viết tư vấn công nghệ. Cấu trúc này giúp người dùng dễ dàng tiếp cận thông tin, so sánh cấu hình, đồng thời hỗ trợ việc quản lý kho hàng, đơn hàng và lưu trữ dữ liệu kinh doanh một cách khoa học.

Khác với các website bán hàng tổng hợp, website bán linh kiện máy tính tập trung sâu vào các thông số kỹ thuật và tính tương thích giữa các thiết bị. Nó không chỉ là nơi mua bán mà còn hướng đến việc xây dựng một kho dữ liệu phần cứng, nơi người dùng có thể tra cứu thông tin chính xác để xây dựng cấu hình máy tính phù hợp.

Trong những năm gần đây, cùng với sự phát triển của công nghệ web, các website bán linh kiện ngày càng được cải tiến về giao diện, tốc độ tải trang và khả năng xử lý dữ liệu lớn, đáp ứng tốt hơn nhu cầu ngày càng khắt khe của cộng đồng yêu công nghệ.

1.2. Nhu cầu lắp ráp máy tính trong bối cảnh công nghệ số

Chuyển đổi số đang diễn ra mạnh mẽ trong mọi lĩnh vực, kéo theo nhu cầu sử dụng các thiết bị máy tính hiệu năng cao để làm việc, học tập và giải trí. Người dùng hiện đại không còn phụ thuộc hoàn toàn vào các bộ máy tính đồng bộ lắp sẵn mà ngày càng chủ động trong việc tự chọn lựa linh kiện để "build" (lắp ráp) máy tính theo nhu cầu cá nhân.

Trong bối cảnh đó, website bán linh kiện máy tính đóng vai trò là một kênh cung ứng quan trọng, hỗ trợ hiệu quả cho quá trình tìm hiểu và mua sắm. Website cho phép người dùng:

- Tra cứu và so sánh giá, hiệu năng của các linh kiện mới nhất trên thị trường.
- Đặt mua hàng trực tuyến và nhận tư vấn cấu hình mà không cần đến cửa hàng vật lý.

- Cập nhật xu hướng công nghệ phần cứng, các dòng sản phẩm mới ra mắt.
- Theo dõi lịch sử đơn hàng và trạng thái bảo hành thiết bị.

Thông qua các hoạt động này, website góp phần thúc đẩy thị trường công nghệ, giúp người dùng dễ dàng tiếp cận sản phẩm chính hãng và tối ưu hóa chi phí đầu tư thiết bị.

1.3. Thực trạng mua sắm linh kiện máy tính hiện nay

Hiện nay, một bộ phận lớn người dùng vẫn sử dụng các nền tảng mạng xã hội, các nhóm mua bán tự phát hoặc các trang thương mại điện tử tổng hợp để tìm mua linh kiện. Các nền tảng này có ưu điểm là lượng người dùng đông đảo, dễ tiếp cận. Tuy nhiên, khi áp dụng vào lĩnh vực đặc thù như phần cứng máy tính, các kênh này bộc lộ nhiều hạn chế.

Thứ nhất, thông tin sản phẩm trên các trang tổng hợp thường không được chuẩn hóa về mặt thông số kỹ thuật, dẫn đến khó khăn trong việc lọc sản phẩm (ví dụ: khó lọc Mainboard theo đúng loại Socket hay Chipset). Thứ hai, việc mua bán qua các nhóm chat hoặc mạng xã hội tiềm ẩn nhiều rủi ro về chất lượng hàng hóa, chế độ bảo hành không rõ ràng và thiếu sự đảm bảo từ hệ thống. Thứ ba, việc quản lý đơn hàng và tra cứu lịch sử mua sắm trên các nền tảng này thường rời rạc, thiếu tính hệ thống và khó kiểm soát đối với người bán.

Những hạn chế trên gây khó khăn cho người dùng trong việc tìm kiếm sản phẩm tương thích và an tâm khi mua sắm. Do đó, cần có một nền tảng bán hàng chuyên biệt, được thiết kế phù hợp với đặc thù kỹ thuật của linh kiện máy tính.

1.4. Vấn đề cần giải quyết

Từ thực trạng nêu trên, có thể nhận thấy vấn đề cốt lõi cần giải quyết là thiếu một hệ thống kinh doanh trực tuyến chuyên biệt, có tổ chức và tối ưu cho việc tra cứu linh kiện. Một website bán linh kiện máy tính cần đáp ứng các yêu cầu sau:

- Sản phẩm được phân loại rõ ràng theo danh mục và thông số chi tiết.
- Hỗ trợ tìm kiếm, lọc sản phẩm thông minh.
- Dễ dàng quản lý giỏ hàng, đặt hàng và theo dõi trạng thái vận chuyển.
- Có cơ chế quản lý kho hàng, đơn hàng và phân quyền người dùng (Khách hàng, Nhân viên, Quản trị viên) phù hợp.

Việc giải quyết những vấn đề này sẽ góp phần nâng cao trải nghiệm mua sắm, đồng thời giúp quy trình quản lý kinh doanh trở nên chuyên nghiệp và hiệu quả hơn.

1.5. Định hướng giải pháp của đề tài

Xuất phát từ các vấn đề đã phân tích, đề tài “Xây dựng Website bán linh kiện máy tính” đề xuất giải pháp xây dựng một hệ thống website thương mại điện tử với kiến trúc hiện đại, giao diện trực quan và chức năng chuyên sâu.

Hệ thống được xây dựng theo mô hình client–server, sử dụng các công nghệ Node.js, React và MongoDB, trong đó:

- Phía Client (React): Đảm nhiệm việc hiển thị giao diện người dùng tương tác cao (SPA), giúp trải nghiệm mượt mà khi tìm kiếm và xem sản phẩm.
- Phía Server (Node.js & Express): Xử lý nghiệp vụ bán hàng, xác thực người dùng, tính toán giỏ hàng và cung cấp API.
- Cơ sở dữ liệu (MongoDB): Dùng để lưu trữ thông tin linh hoạt cho các sản phẩm có cấu trúc thông số khác nhau, cũng như lưu trữ đơn hàng và người dùng.

Giải pháp này nhằm đảm bảo hệ thống hoạt động ổn định, tốc độ phản hồi nhanh, dễ bảo trì và có khả năng mở rộng dữ liệu linh kiện trong tương lai.

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1. ReactJS (Xây dựng giao diện người dùng)

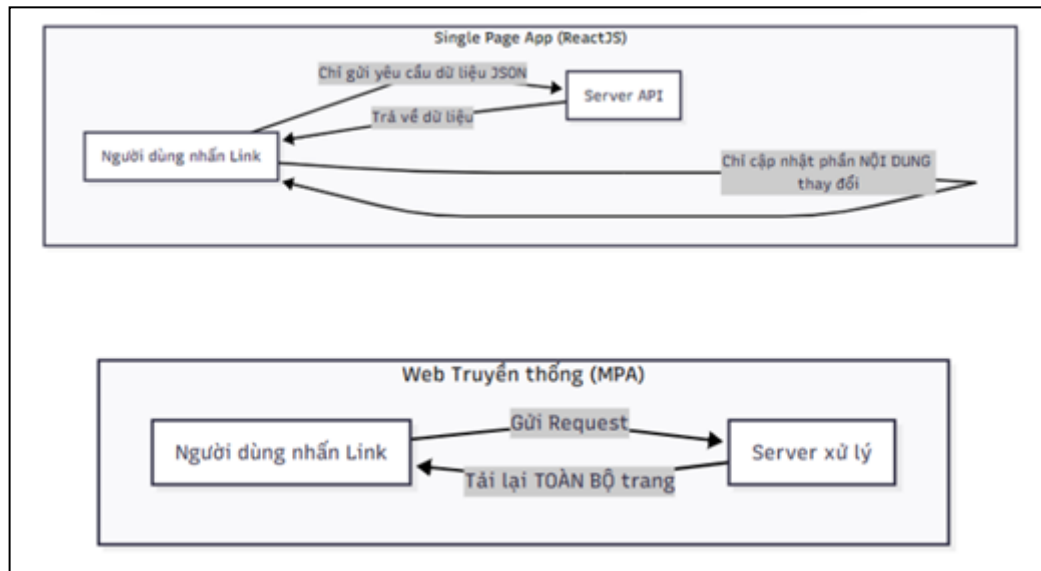
2.1.1. Tổng quan và vai trò

ReactJS là một thư viện JavaScript mã nguồn mở do Facebook (Meta) phát triển, đóng vai trò nền tảng trong việc xây dựng giao diện người dùng (User Interface - UI). Trong kiến trúc của hệ thống website bán linh kiện máy tính, ReactJS đảm nhận toàn bộ trách nhiệm phía Client (Frontend), nơi tương tác trực tiếp với khách hàng.



Hình 2.1. Logo ReactJS

Khác với các công nghệ web truyền thống thường tải lại toàn bộ trang khi có thay đổi, ReactJS cho phép xây dựng ứng dụng dạng **Single Page Application (SPA)**. Điều này có nghĩa là khi khách hàng chuyển từ trang "Danh sách Card màn hình" sang trang "Chi tiết sản phẩm", trình duyệt chỉ cần cập nhật lại phần nội dung cần thiết mà không cần tải lại toàn bộ mã nguồn của trang web. Điều này giúp giảm thiểu băng thông mạng và mang lại trải nghiệm mượt mà, tốc độ cao tương tự như các ứng dụng trên máy tính.



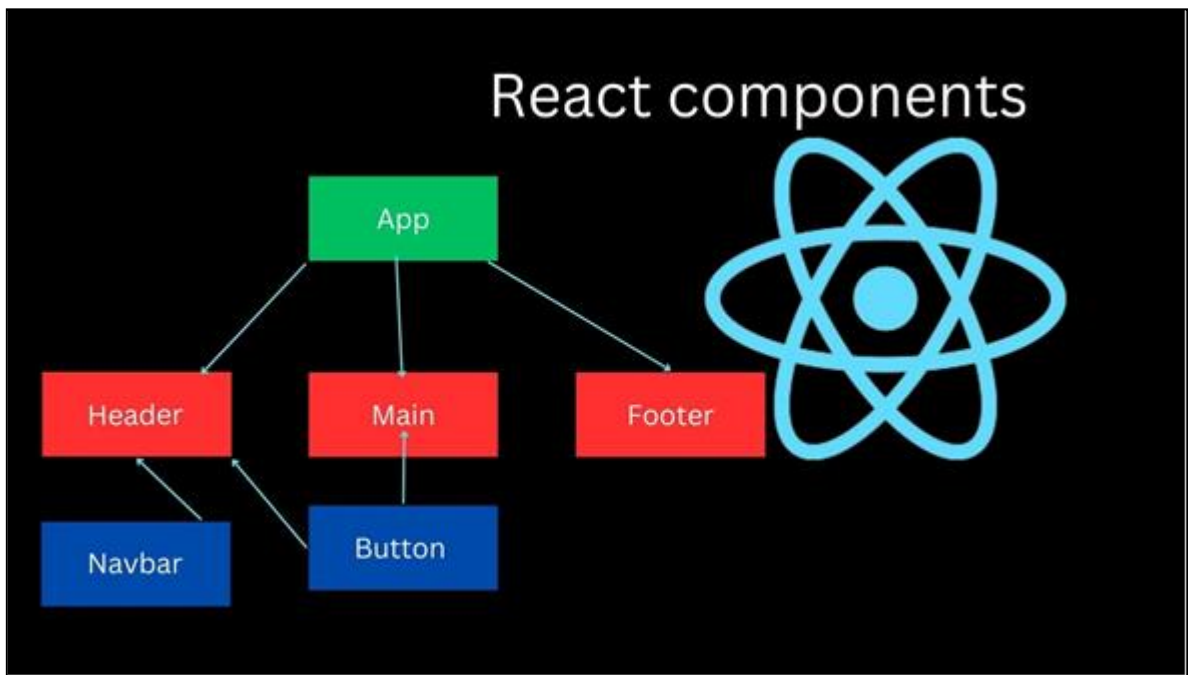
Hình 2.2. Kiến trúc SPA so với MPA (Dùng cho phần ReactJS)

2.1.2. Kiến trúc hướng thành phần (Component-based)

Một trong những lý do chính để lựa chọn ReactJS cho đề tài này là khả năng chia nhỏ giao diện phức tạp thành các thành phần độc lập gọi là Component.

- Tính tái sử dụng: Ví dụ, một thành phần hiển thị sản phẩm (Product Card) bao gồm hình ảnh, tên linh kiện, giá tiền và nút mua hàng. Component này chỉ cần được lập trình một lần và có thể tái sử dụng hàng trăm lần để hiển thị danh sách các loại linh kiện khác nhau như RAM, CPU, Mainboard.

- Dễ dàng bảo trì: Các thành phần như Thanh điều hướng (Header), Chân trang (Footer), Giỏ hàng (Cart) được tách biệt hoàn toàn. Khi cần sửa lỗi hoặc nâng cấp giao diện của một phần, lập trình viên không cần lo lắng làm ảnh hưởng đến các phần còn lại của hệ thống.



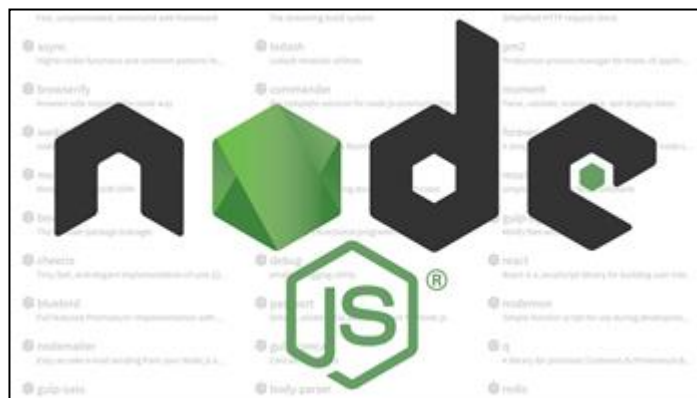
Hình 2.3. Cấu trúc cây Component trong ứng dụng ReactJS

2.2. Node.js (Môi trường thực thi phía Server)

2.2.1. Giới thiệu chung

Node.js là một môi trường thực thi mã JavaScript đa nền tảng, được xây dựng dựa trên V8 JavaScript Engine của Google Chrome. Đây là công nghệ cốt lõi cho phép sử dụng ngôn ngữ JavaScript để lập trình phía máy chủ (Server-side), thay vì chỉ giới hạn ở trình duyệt như trước đây.

Việc lựa chọn Node.js cho dự án tạo nên sự đồng bộ về ngôn ngữ (Fullstack JavaScript). Lập trình viên có thể sử dụng cùng một cú pháp, cùng một kiểu dữ liệu JSON cho cả phía Frontend (ReactJS) và Backend, giúp quá trình phát triển nhanh chóng hơn và giảm thiểu các lỗi chuyển đổi dữ liệu.



Hình 2.4. Logo Node.JS

2.2.2. Khả năng xử lý và hiệu năng

Trong bối cảnh website thương mại điện tử bán linh kiện máy tính, hệ thống thường xuyên phải xử lý hàng loạt các yêu cầu đồng thời (concurrent requests) như: hàng trăm người cùng xem sản phẩm, tìm kiếm từ khóa, lọc theo giá tiền. Node.js giải quyết vấn đề này bằng cơ chế xử lý bất đồng bộ (Non-blocking I/O). Nó có thể tiếp nhận và xử lý hàng nghìn kết nối cùng lúc mà không làm treo hệ thống, đảm bảo tính ổn định và tốc độ phản hồi nhanh ngay cả trong những thời điểm có lượng truy cập cao.

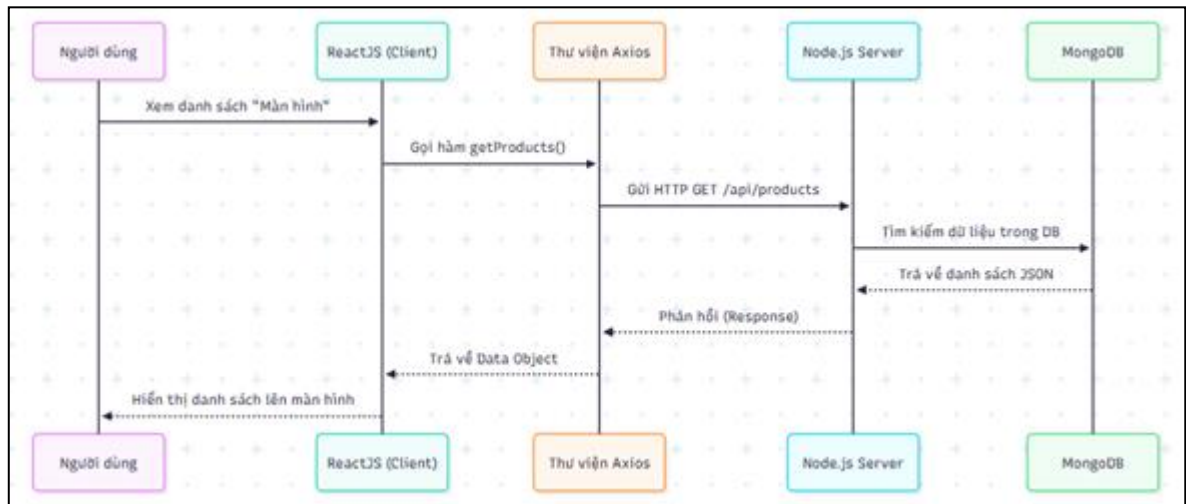
2.3. Axios (Giao tiếp HTTP Client)

2.3.1. Vai trò cầu nối dữ liệu

Axios là một thư viện HTTP Client hoạt động dựa trên Promise, đóng vai trò là "người vận chuyển" dữ liệu giữa giao diện ReactJS và máy chủ Node.js. ReactJS là nơi hiển thị, còn Node.js là nơi chứa dữ liệu; Axios chính là công cụ kết nối hai đầu này lại với nhau.

Thông qua Axios, hệ thống thực hiện các thao tác nghiệp vụ chuẩn (CRUD) thông qua các phương thức HTTP:

- GET: Dùng để lấy danh sách linh kiện, lấy chi tiết đơn hàng.
- POST: Dùng để gửi dữ liệu đăng ký, đăng nhập, hoặc tạo đơn hàng mới.
- PUT/PATCH: Dùng để cập nhật thông tin cá nhân hoặc thay đổi trạng thái đơn hàng.
- DELETE: Dùng để xóa sản phẩm trong giỏ hàng hoặc xóa dữ liệu quản trị.



Hình 2.5. Luồng dữ liệu với Axios (Dùng cho phần Axios)

2.3.2. Tính năng xử lý phản hồi

Axios cung cấp khả năng tự động chuyển đổi dữ liệu JSON. Khi máy chủ gửi về thông tin danh sách sản phẩm dưới dạng chuỗi văn bản JSON, Axios sẽ tự động phân tích và chuyển nó thành đối tượng JavaScript để ReactJS có thể sử dụng ngay lập tức. Ngoài ra, Axios còn hỗ trợ bắt lỗi (Error Handling) hiệu quả, giúp hiển thị các thông báo lỗi rõ ràng cho người dùng khi có sự cố mạng hoặc lỗi từ máy chủ.

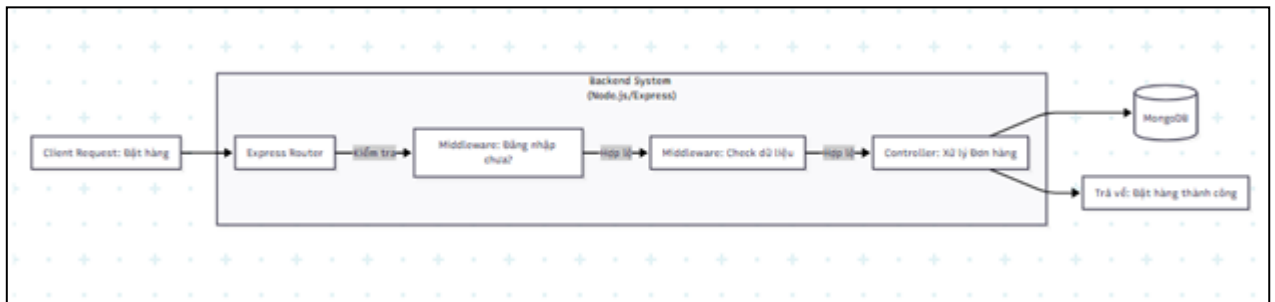
2.4. Express (Framework xây dựng Backend API)

2.4.1. Cấu trúc hóa hệ thống Backend

Nếu Node.js là động cơ, thì Express chính là bộ khung sườn của chiếc xe. Express là một framework tối giản nhưng mạnh mẽ, giúp tổ chức cấu trúc của ứng dụng phía Server một cách khoa học. Nó cung cấp các công cụ để xây dựng các API (Application Programming Interface) chuẩn RESTful.

Trong dự án, Express giúp phân chia mã nguồn thành các module rõ ràng thông qua cơ chế Định tuyến (Routing). Ví dụ:

- Nhóm đường dẫn /api/products: Chuyên xử lý các logic liên quan đến hiển thị, tìm kiếm linh kiện.
- Nhóm đường dẫn /api/auth: Xử lý đăng ký, đăng nhập, cấp lại mật khẩu.
- Nhóm đường dẫn /api/orders: Xử lý quy trình thanh toán và vận chuyển.



Hình 2.6. Kiến trúc Express & Routing (Dùng cho phần Express)

2.4.2. Middleware (Phần mềm trung gian)

Express sở hữu tính năng Middleware mạnh mẽ, cho phép chèn các bước xử lý trung gian trước khi một yêu cầu thực sự được thực hiện.

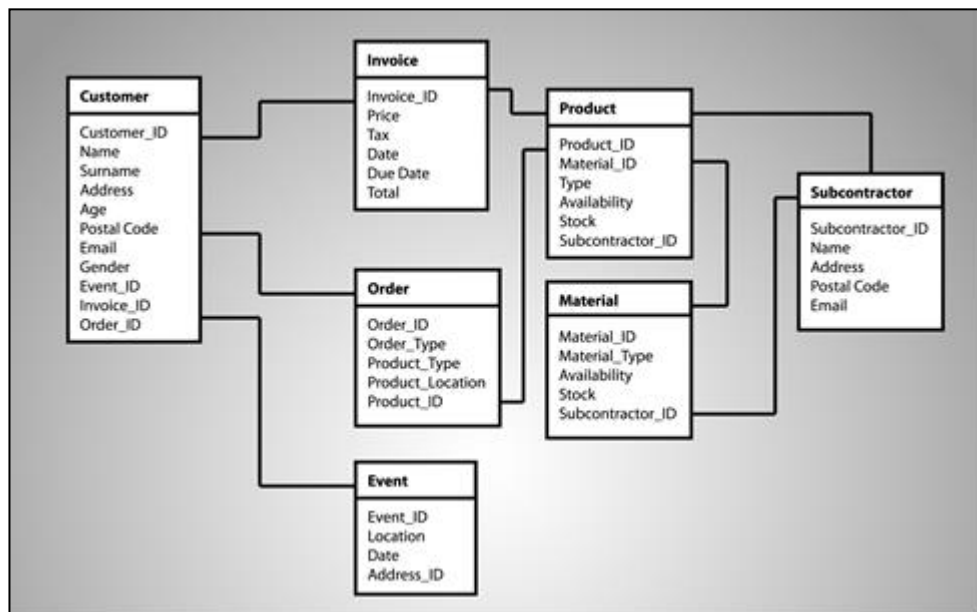
Ví dụ thực tế: Khi người dùng nhấn nút "Đặt hàng", một Middleware xác thực sẽ chạy trước để kiểm tra xem người dùng đã đăng nhập chưa. Sau đó, một Middleware khác kiểm tra xem dữ liệu gửi lên (số lượng, địa chỉ) có hợp lệ không. Nếu tất cả đều đúng, Express mới chuyển yêu cầu đến bộ xử lý lưu đơn hàng.

2.5. Mongoose (Quản lý Cơ sở dữ liệu MongoDB)

2.5.1. Mô hình hóa dữ liệu (Data Modeling)

Mongoose là thư viện cầu nối giúp ứng dụng Node.js làm việc với cơ sở dữ liệu MongoDB một cách dễ dàng và chặt chẽ hơn. Dù MongoDB là cơ sở dữ liệu phi cấu trúc (NoSQL), nhưng trong kinh doanh, dữ liệu cần sự nhất quán. Mongoose giải quyết vấn đề này bằng cách cung cấp Schema (Lược đồ).

Schema đóng vai trò như một bản thiết kế, quy định rõ ràng cấu trúc của một đối tượng dữ liệu. Đối với đề tài bán linh kiện máy tính, Mongoose đảm bảo rằng một sản phẩm khi được lưu vào kho dữ liệu bắt buộc phải có đủ các trường thông tin: Tên sản phẩm (String), Giá bán (Number), Số lượng tồn kho (Number), và Hình ảnh (String).



Hình 2.7. Mongoose Schema (Dùng cho phần Mongoose)

2.5.2. Kiểm soát và xác thực dữ liệu

Mongoose tích hợp sẵn các bộ kiểm tra (Validators) mạnh mẽ. Nó ngăn chặn việc lưu trữ các dữ liệu rác hoặc không hợp lệ vào hệ thống.

- Ví dụ: Nếu người quản trị nhập giá sản phẩm là số âm, hoặc nhập sai định dạng email khách hàng, Mongoose sẽ tự động báo lỗi và từ chối lưu bản ghi đó. Điều này giúp cơ sở dữ liệu luôn sạch sẽ, chính xác và đáng tin cậy.

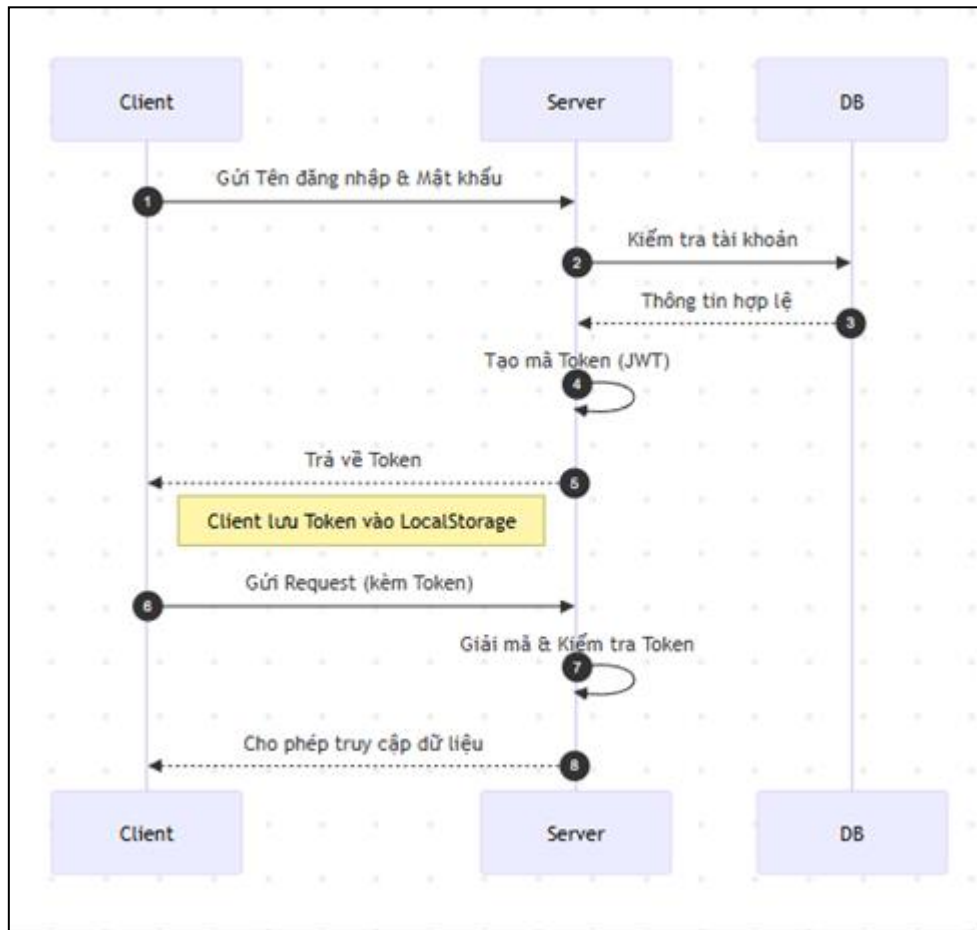
2.6. JsonWebToken - JWT (Bảo mật và Xác thực)

2.6.1. Cơ chế xác thực hiện đại

JsonWebToken (JWT) là tiêu chuẩn mở để truyền tải thông tin an toàn giữa các bên dưới dạng đối tượng JSON. Đây là công nghệ then chốt để xây dựng chức năng Đăng ký - Đăng nhập và bảo mật phiên làm việc của người dùng.

Quy trình hoạt động của JWT trong hệ thống:

1. Người dùng nhập tên đăng nhập và mật khẩu.
2. Server kiểm tra, nếu đúng sẽ tạo ra một chuỗi mã hóa (Token) chứa thông tin định danh (ID người dùng, Quyền hạn) và gửi về cho Client.
3. Client lưu trữ Token này (thường ở LocalStorage).
4. Ở các lần truy cập sau, Client gửi kèm Token này trong mỗi yêu cầu gửi lên Server để chứng minh danh tính.



Hình 2.7. Quy trình xác thực JWT (Dùng cho phần JWT)

2.6.2. Phân quyền người dùng (Authorization)

Ngoài việc xác thực danh tính, JWT còn đóng vai trò quan trọng trong việc phân quyền. Thông tin bên trong Token sẽ cho hệ thống biết người dùng đó là "Khách hàng thường" hay "Quản trị viên" (Admin). Dựa vào Token này, Server sẽ quyết định cho phép truy cập hoặc chặn các hành động nhạy cảm. Ví dụ: Chỉ những Token có quyền Admin mới được phép gọi API để thêm sản phẩm mới hoặc xóa đơn hàng, trong khi Khách hàng chỉ được phép xem và mua hàng. Điều này đảm bảo an toàn tuyệt đối cho dữ liệu hệ thống.

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

3.1. Mô tả hệ thống

Website bán linh kiện máy tính là một hệ thống thương mại điện tử chuyên dụng (E-commerce System), được xây dựng và phát triển nhằm giải quyết bài toán mua sắm các thiết bị công nghệ phức tạp trong thời đại chuyển đổi số. Khác với các mặt hàng tiêu dùng thông thường, linh kiện máy tính (như Mainboard, CPU, VGA, RAM, PSU) đòi hỏi sự chính xác tuyệt đối về thông số kỹ thuật, khả năng tương thích phần cứng và nguồn gốc xuất xứ. Do đó, hệ thống này không chỉ đơn thuần là nơi thực hiện giao dịch mua bán, mà còn đóng vai trò là một nền tảng cung cấp thông tin kỹ thuật chuyên sâu, hỗ trợ người dùng so sánh, đánh giá và ra quyết định mua hàng chính xác nhất.

Hệ thống cung cấp trọn vẹn các chức năng của một quy trình thương mại điện tử khép kín (End-to-End E-commerce): bắt đầu từ việc người dùng (Sinh viên, Khách hàng) đăng ký tài khoản, tìm kiếm sản phẩm thông qua bộ lọc chuyên sâu, xem chi tiết cấu hình, quản lý giỏ hàng, cho đến các bước thanh toán bảo mật, áp dụng mã giảm giá và theo dõi lộ trình vận đơn. Song song đó, hệ thống tích hợp các công cụ tương tác thời gian thực (Real-time) như Chat hỗ trợ trực tuyến và Hệ thống thông báo tự động, giúp thu hẹp khoảng cách giữa người bán và người mua, nâng cao trải nghiệm khách hàng.

Về mặt kỹ thuật, Website bán linh kiện máy tính được xây dựng theo mô hình Client–Server hiện đại, tách biệt hoàn toàn giữa Frontend và Backend (Decoupled Architecture). Trong đó:

- Frontend (Client): Chịu trách nhiệm hiển thị giao diện trực quan, xử lý các tương tác người dùng mượt mà theo cơ chế Single Page Application (SPA), giảm thiểu việc tải lại trang.

- Backend (Server): Đảm nhiệm xử lý các logic nghiệp vụ phức tạp như tính toán giá thành, kiểm tra tồn kho theo thời gian thực (Real-time Inventory), xác thực bảo mật và quản lý luồng dữ liệu.

- Database: Sử dụng cơ sở dữ liệu NoSQL (MongoDB) để lưu trữ dữ liệu sản phẩm với cấu trúc linh động (Schema-less), phù hợp với sự đa dạng và không đồng nhất của các thông số kỹ thuật phần cứng máy tính.

3.2. Phân tích yêu cầu hệ thống

3.2.1. Yêu cầu chức năng

3.2.1.1. Chức năng cho người dùng (Khách hàng)

Người dùng là đối tượng chính sử dụng hệ thống để thực hiện hành vi mua sắm. Các nhóm chức năng chi tiết bao gồm:

Đăng ký và Đăng nhập: Người dùng có thể tạo tài khoản mới bằng email và mật khẩu, hoặc đăng nhập nhanh thông qua Google OAuth. Hệ thống sử dụng cơ chế JWT (JSON Web Token) để duy trì phiên làm việc an toàn và bảo mật.

Quản lý thông tin cá nhân: Cho phép người dùng cập nhật thông tin giao hàng (địa chỉ, số điện thoại), thay đổi mật khẩu bảo vệ và quản lý ảnh đại diện cá nhân.

Tìm kiếm và Lọc sản phẩm: Hệ thống cung cấp công cụ tìm kiếm thông minh theo từ khóa (Full-text search) và bộ lọc nâng cao đa tiêu chí (Lọc theo khoảng giá, Thương hiệu, Dung lượng RAM, Loại Socket CPU...) giúp người dùng nhanh chóng tìm được linh kiện phù hợp với nhu cầu build PC.

Xem chi tiết sản phẩm: Hiển thị đầy đủ hình ảnh độ nét cao, giá bán niêm yết, giá khuyến mãi, tình trạng kho và bảng thông số kỹ thuật chi tiết của từng linh kiện.

Quản lý Giỏ hàng: Người dùng có thể thêm sản phẩm vào giỏ hàng (Shopping Cart), điều chỉnh số lượng mua hoặc xóa sản phẩm. Hệ thống tự động tính toán tổng tiền tạm tính và lưu trạng thái giỏ hàng.

Đặt hàng và Thanh toán: Hỗ trợ quy trình đặt hàng 3 bước (Thông tin giao hàng -> Phương thức thanh toán -> Xác nhận). Tích hợp cổng thanh toán ZaloPay (QR Code, thẻ ATM) và phương thức COD (Thanh toán khi nhận hàng).

Đánh giá và Bình luận: Cho phép người dùng đã mua hàng để lại bình luận và chấm điểm sao (Rating), giúp tăng độ uy tín cho sản phẩm và hỗ trợ cộng đồng.

Chat hỗ trợ trực tuyến: Tích hợp cửa sổ chat realtime, cho phép khách hàng gửi thắc mắc và nhận phản hồi tức thì từ nhân viên kỹ thuật mà không cần chuyển sang ứng dụng khác.

3.2.1.2. Chức năng cho Nhân viên

Nhân viên là người hỗ trợ vận hành hệ thống, có quyền hạn hạn chế hơn Quản trị viên (Admin). Họ tập trung vào việc xử lý đơn hàng và chăm sóc khách hàng. Các chức năng bao gồm:

Xử lý Đơn hàng: Tiếp nhận đơn hàng mới, kiểm tra kho, xác nhận đơn hàng và cập nhật trạng thái vận chuyển (Chờ lấy hàng -> Đang giao). Nhân viên không có quyền xóa đơn hàng.

Hỗ trợ Khách hàng (Chat): Tiếp nhận các tin nhắn từ khách hàng thông qua hệ thống Chat Real-time, tư vấn kỹ thuật và giải đáp thắc mắc về bảo hành.

Quản lý Kho hàng (Cơ bản): Xem danh sách sản phẩm để kiểm tra số lượng tồn kho. Có quyền cập nhật số lượng tồn kho sau khi kiểm kê thực tế, nhưng không được phép thêm mới, xóa sản phẩm hoặc sửa giá bán (quyền này thuộc về Admin).

Quản lý Bình luận: Duyệt các bình luận/đánh giá của khách hàng trước khi hiển thị lên website, ẩn các bình luận spam hoặc thiếu văn hóa.

3.2.1.3. Chức năng cho quản trị viên (Admin)

Quản trị viên là người vận hành, chịu trách nhiệm quản lý toàn bộ dữ liệu và quy trình kinh doanh:

Quản lý Sản phẩm (Inventory Management): Thêm mới linh kiện với đầy đủ thông số kỹ thuật, cập nhật giá bán theo biến động thị trường, quản lý số lượng tồn kho (Stock) và tải lên hình ảnh sản phẩm lên Cloud.

Quản lý Đơn hàng (Order Processing): Theo dõi danh sách đơn hàng mới theo thời gian thực, cập nhật trạng thái vận chuyển (Chờ xác nhận -> Đang giao -> Hoàn thành -> Đã hủy) và in phiếu giao hàng.

Quản lý Người dùng (User Management): Xem danh sách khách hàng đăng ký, thực hiện khóa/mở khóa tài khoản đối với các trường hợp vi phạm chính sách hoặc spam.

Quản lý Danh mục (Categories): Tạo mới, chỉnh sửa hoặc xóa các nhóm linh kiện (Ví dụ: tách nhóm VGA AMD và VGA NVIDIA, nhóm Ổ cứng SSD và HDD).

Quản lý Mã giảm giá (Coupons): Tạo các mã khuyến mãi, thiết lập mức giảm giá (%), số tiền), số lượng mã và thời gian hết hạn.

Thống kê và Báo cáo (Dashboard): Xem Dashboard tổng quan về doanh thu theo ngày/tháng/năm, biểu đồ tăng trưởng, danh sách top sản phẩm bán chạy và cảnh báo các sản phẩm sắp hết hàng.

3.2.2. Yêu cầu phi chức năng

Tính bảo mật (Security): Dữ liệu mật khẩu phải được mã hóa một chiều (Hashing) bằng thuật toán Bcrypt. Các API nhạy cảm phải được bảo vệ bằng Token xác thực. Ngăn chặn các lỗi bảo mật phổ biến như XSS (Cross-site Scripting) và NoSQL Injection.

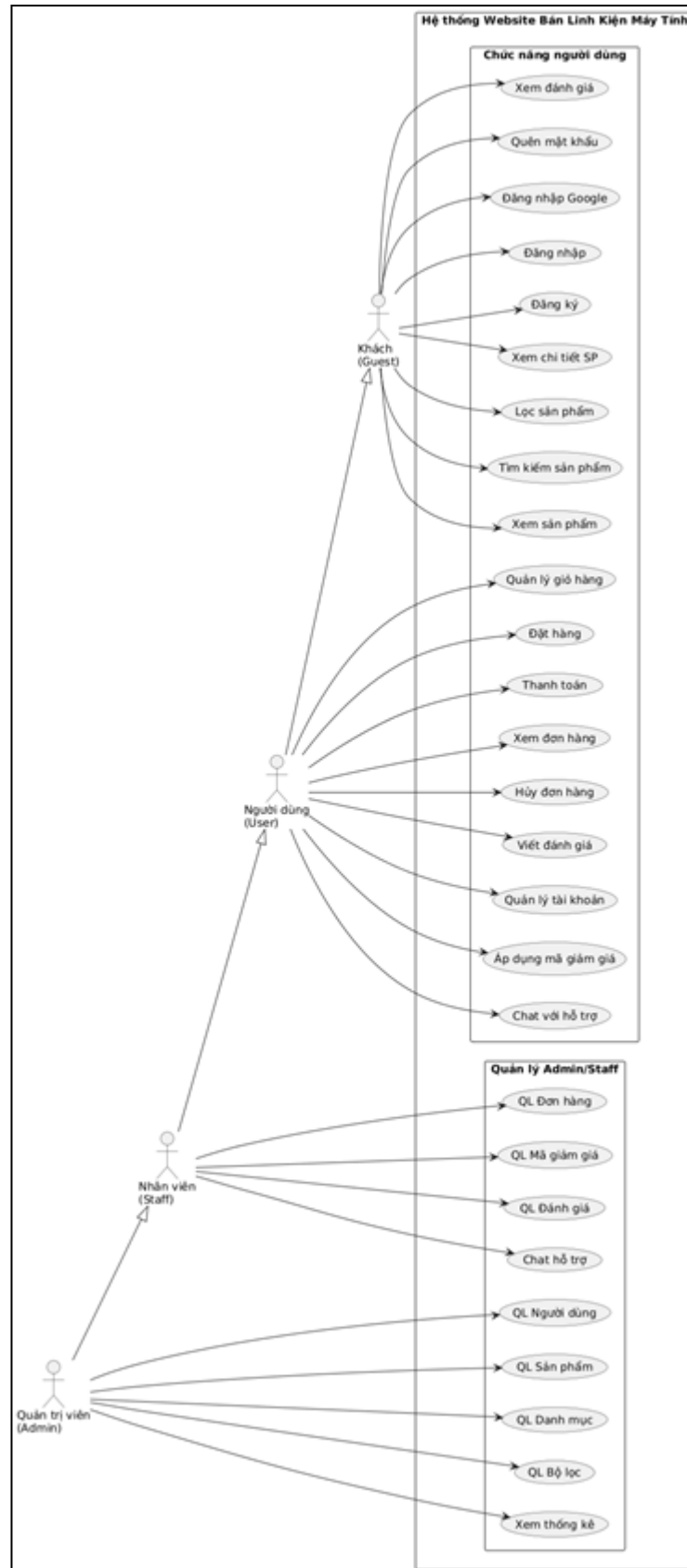
Hiệu năng (Performance): Thời gian tải trang chủ và danh sách sản phẩm phải dưới 2 giây. Hệ thống chịu tải được nhiều người dùng truy cập đồng thời mà không bị treo hoặc phản hồi chậm.

Tính ổn định (Reliability): Hệ thống hoạt động liên tục 24/7. Dữ liệu đơn hàng và kho hàng phải đảm bảo tính nhất quán (Consistency), không xảy ra tình trạng bán quá số lượng tồn kho.

Khả năng tương thích (Compatibility): Giao diện hiển thị tốt trên mọi kích thước màn hình (Responsive Design) từ Desktop, Laptop đến Tablet, Mobile; tương thích với các trình duyệt phổ biến (Chrome, Firefox, Safari, Edge).

Khả năng mở rộng (Scalability): Kiến trúc mã nguồn phải rõ ràng, dễ dàng thêm các module chức năng mới (như PC Builder, Trả góp) trong tương lai.

3.2.3. Sơ đồ Use Case



Hình 3.1. Sơ đồ Use Case

3.3. Thiết kế hệ thống

3.3.1. Thiết kế kiến trúc tổng thể hệ thống

Hệ thống website bán linh kiện máy tính được thiết kế theo mô hình Client - Server phân tán, đảm bảo sự tách biệt rõ ràng giữa các tầng logic, giúp dễ dàng bảo trì và nâng cấp.

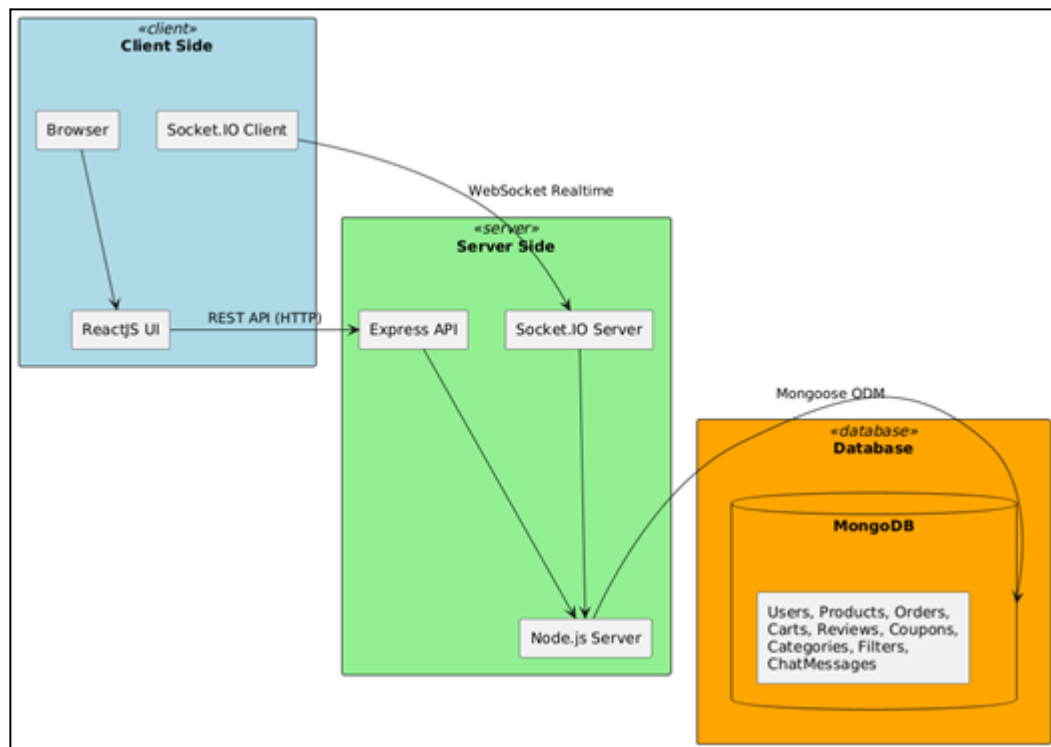
Tầng Presentation (Client): Sử dụng thư viện ReactJS. Client đảm nhiệm việc hiển thị dữ liệu (View), bắt sự kiện từ người dùng và gửi yêu cầu (Request) về Server. Việc sử dụng React giúp giảm tải cho Server vì chỉ cần gửi dữ liệu JSON thay vì toàn bộ mã HTML.

Tầng Business Logic (Server): Sử dụng Node.js và Express.js. Đây là trung tâm xử lý của hệ thống, thực hiện các công việc: Kiểm tra xác thực (Auth Middleware), Validate dữ liệu đầu vào, Xử lý logic nghiệp vụ (Tính toán tổng tiền, Trừ kho) và Gọi xuống Database.

Tầng Data Access (Database): Sử dụng MongoDB. Lưu trữ toàn bộ dữ liệu hệ thống dưới dạng Document (JSON). Mô hình NoSQL giúp việc lưu trữ các thuộc tính động của linh kiện máy tính trở nên dễ dàng và linh hoạt.

Tầng Real-time Communication: Sử dụng Socket.IO để thiết lập kênh giao tiếp hai chiều (Bi-directional), phục vụ cho tính năng Chat hỗ trợ và Thông báo đơn hàng mới.

Tầng Media Storage: Sử dụng dịch vụ Cloudinary để lưu trữ hình ảnh sản phẩm, tối ưu hóa tốc độ tải ảnh (CDN).

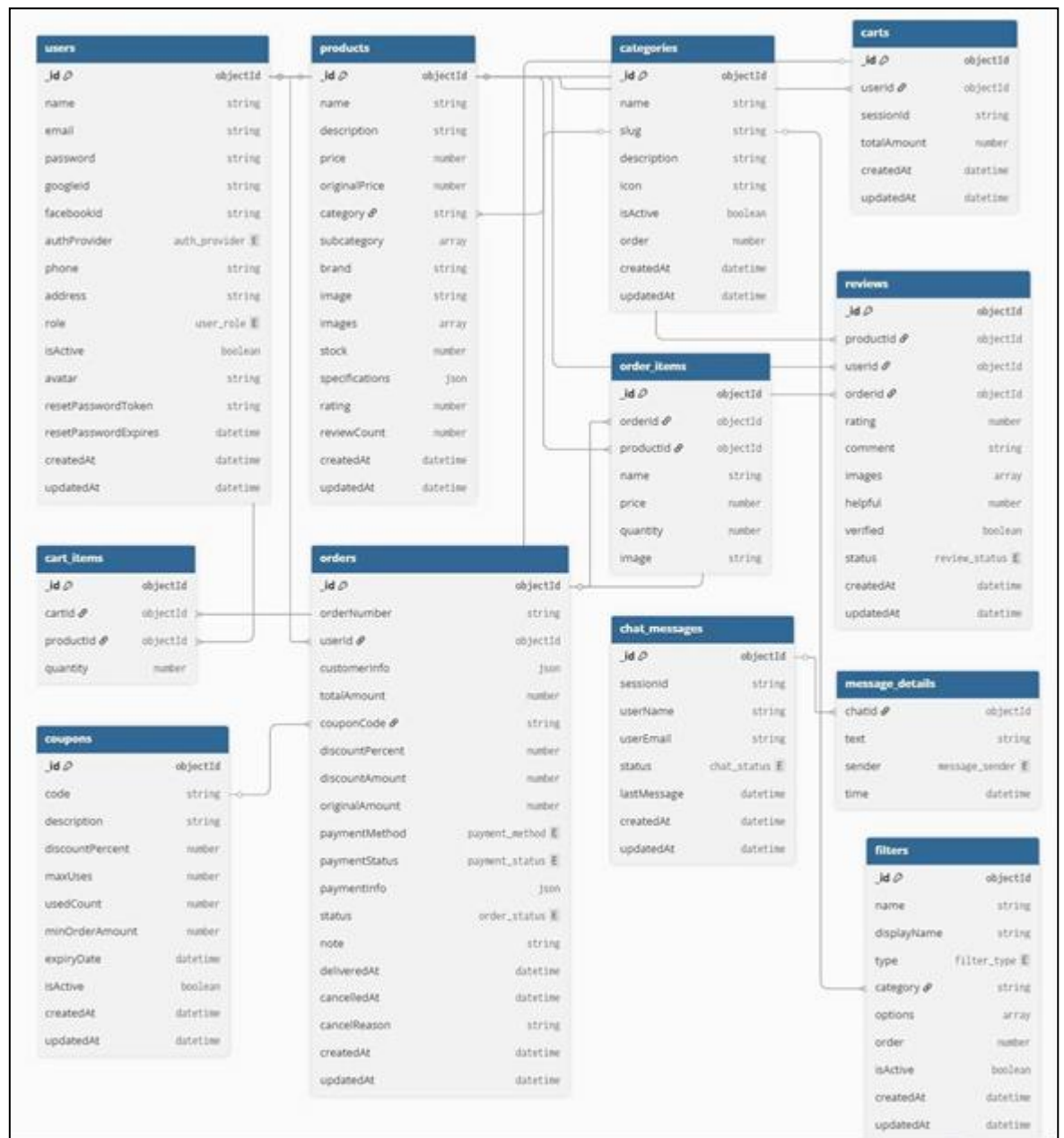


Hình 3.2. Sơ đồ kiến trúc tổng thể hệ thống

3.3.2. Thiết kế cơ sở dữ liệu

Hệ thống website bán linh kiện máy tính sử dụng MongoDB làm hệ quản trị cơ sở dữ liệu chính. MongoDB là một cơ sở dữ liệu NoSQL, lưu trữ dữ liệu dưới dạng Document (JSON/BSON). Cấu trúc này đặc biệt phù hợp với các hệ thống thương mại điện tử kinh doanh thiết bị công nghệ, nơi mà thông tin sản phẩm có cấu trúc phức tạp, đa dạng về thông số kỹ thuật (ví dụ: thông số của RAM khác hoàn toàn với thông số của Mainboard) và thường xuyên cần cập nhật.

Mặc dù là cơ sở dữ liệu phi quan hệ, mối quan hệ giữa các Collection vẫn được thiết lập chặt chẽ thông qua các khóa tham chiếu (ObjectId). Cơ chế này đảm bảo khả năng liên kết dữ liệu (ví dụ: liên kết người dùng với đơn hàng họ đã đặt) giúp đảm bảo tính toàn vẹn của dữ liệu trong khi vẫn giữ được sự linh hoạt và hiệu năng truy xuất cao đặc trưng của mô hình NoSQL.



Hình 3.3. Mô hình dữ liệu MongoDB của hệ thống

Chi tiết các thực thể

Một số bảng chính trong dự án bao gồm:

Bảng 3.1. Chi tiết các thực thể User

STT	Thuộc tính	Kiểu	Mô tả
1	_id	ObjectId	Khóa chính(Primary Key)
2	name	String	Họ và tên người dùng
3	email	String	Địa chỉ email (duy nhất)
4	password	String	Mật khẩu đã mã hóa
5	googleId	String	ID tài khoản Google (nếu có)
6	facebookId	String	ID tài khoản Facebook (nếu có)
7	authProvider	Enum	Nhà cung cấp xác thực (Local/Google/Facebook)
8	phone	String	Số điện thoại liên lạc
9	address	String	Địa chỉ giao hàng mặc định
10	role	Enum	Vai trò (User/Admin/Staff)
11	isActive	Boolean	Trạng thái hoạt động của tài khoản
12	avatar	String	Đường dẫn ảnh đại diện

STT	Thuộc tính	Kiểu	Mô tả
13	resetPasswordToken	String	Token để khôi phục mật khẩu
14	resetPasswordExpires	DateTime	Thời gian hết hạn của token
15	createdAt	DateTime	Ngày tạo tài khoản
16	updatedAt	DateTime	Ngày cập nhật gần nhất

Bảng 3.2. Chi tiết các thực thể Products

STT	Thuộc tính	Kiểu	Mô tả
1	_id	ObjectId	Khóa chính
2	name	String	Tên sản phẩm
3	description	String	Mô tả chi tiết
4	price	Number	Giá bán hiện tại
5	originalPrice	Number	Giá gốc (trước khi giảm)
6	category	String	ID danh mục sản phẩm
7	subcategory	Array	Danh sách danh mục con
8	brand	String	Thương hiệu sản phẩm
9	image	String	Ảnh đại diện chính
10	images	Array	Danh sách các ảnh phụ
11	stock	Number	Số lượng tồn kho

STT	Thuộc tính	Kiểu	Mô tả
12	specifications	JSON	Thông số kỹ thuật chi tiết
13	rating	Number	Điểm đánh giá trung bình
14	reviewCount	Number	Tổng số lượt đánh giá
15	createdAt	DateTime	Ngày tạo sản phẩm
16	updatedAt	DateTime	Ngày cập nhật sản phẩm

Bảng 3.3. Chi tiết các thực thể Categories

STT	Thuộc tính	Kiểu	Mô tả
1	_id	ObjectId	Khóa chính
2	name	String	Tên danh mục
3	slug	String	Đường dẫn tĩnh (SEO)
4	description	String	Mô tả danh mục
5	icon	String	Biểu tượng danh mục
6	isActive	Boolean	Trạng thái kích hoạt
7	order	Number	Thứ tự hiển thị
8	createdAt	DateTime	Ngày tạo
9	updatedAt	DateTime	Ngày cập nhật

Bảng 3.4. Chi tiết các thực thể Carts

STT	Thuộc tính	Kiểu	Mô tả
1	_id	ObjectId	Khóa chính
2	userId	ObjectId	ID người dùng (nếu đã đăng nhập)
3	sessionId	String	ID phiên (cho khách vãng lai)
4	totalAmount	Number	Tổng tiền tạm tính
5	createdAt	DateTime	Ngày tạo giỏ hàng
6	updatedAt	DateTime	Ngày cập nhật giỏ hàng

Bảng 3.5. Chi tiết các thực thể Cart_Items

STT	Thuộc tính	Kiểu	Mô tả
1	_id	ObjectId	Khóa chính
2	cartId	ObjectId	ID giỏ hàng liên kết
3	productId	ObjectId	ID sản phẩm
4	quantity	Number	Số lượng mua

Bảng 3.6. Chi tiết các thực thể Orders

STT	Thuộc tính	Kiểu	Mô tả
1	_id	ObjectId	Khóa chính
2	orderNumber	String	Mã đơn hàng (dùng để tra cứu)
3	userId	ObjectId	ID người đặt hàng
4	customerInfo	JSON	Thông tin khách hàng (Tên, SĐT, Đ/C)
5	totalAmount	Number	Tổng giá trị đơn hàng
6	couponCode	String	Mã giảm giá đã áp dụng
7	discountPercent	Number	Phần trăm giảm giá
8	discountAmount	Number	Số tiền được giảm
9	originalAmount	Number	Giá gốc trước giảm
10	paymentMethod	Enum	Phương thức thanh toán
11	paymentStatus	Enum	Trạng thái thanh toán
12	paymentInfo	JSON	Chi tiết giao dịch thanh toán
13	status	Enum	Trạng thái đơn hàng

STT	Thuộc tính	Kiểu	Mô tả
14	note	String	Ghi chú của khách hàng
15	deliveredAt	DateTime	Thời gian giao hàng thành công
16	cancelledAt	DateTime	Thời gian hủy đơn (nếu có)
17	cancelReason	String	Lý do hủy đơn
18	createdAt	DateTime	Ngày đặt hàng
19	updatedAt	DateTime	Ngày cập nhật đơn hàng

Bảng 3.7. Chi tiết thực thể Order_Items

STT	Thuộc tính	Kiểu	Mô tả
1	_id	ObjectId	Khóa chính
2	orderId	ObjectId	ID đơn hàng liên kết
3	productId	ObjectId	ID sản phẩm
4	name	String	Tên sản phẩm tại thời điểm mua
5	price	Number	Giá bán tại thời điểm mua
6	quantity	Number	Số lượng mua

STT	Thuộc tính	Kiểu	Mô tả
7	image	String	Ảnh sản phẩm

Bảng 3.8. Chi tiết các thực thể Coupons

STT	Thuộc tính	Kiểu	Mô tả
1	_id	ObjectId	Khóa chính
2	code	String	Mã code giảm giá
3	description	String	Mô tả chương trình khuyến mãi
4	discountPercent	Number	Phần trăm giảm giá
5	maxUses	Number	Số lần sử dụng tối đa
6	usedCount	Number	Số lần đã sử dụng
7	minOrderAmount	Number	Giá trị đơn hàng tối thiểu
8	expiryDate	DateTime	Ngày hết hạn
9	isActive	Boolean	Trạng thái kích hoạt
10	createdAt	DateTime	Ngày tạo
11	updatedAt	DateTime	Ngày cập nhật

Bảng 3.9. Chi tiết các thực thể Reviews

STT	Thuộc tính	Kiểu	Mô tả
1	_id	ObjectId	Khóa chính
2	productId	ObjectId	ID sản phẩm được đánh giá
3	userId	ObjectId	ID người đánh giá
4	orderId	ObjectId	ID đơn hàng liên quan
5	rating	Number	Số sao đánh giá (1-5)
6	comment	String	Nội dung bình luận
7	images	Array	Ảnh đính kèm trong đánh giá
8	helpful	Number	Số lượt thấy hữu ích
9	verified	Boolean	Xác thực đã mua hàng
10	status	Enum	Trạng thái duyệt (Pending/Approved)
11	createdAt	DateTime	Ngày tạo đánh giá
12	updatedAt	DateTime	Ngày cập nhật đánh giá

Bảng 3.10. Chi tiết các thực thể Chat_Messages

STT	Thuộc tính	Kiểu	Mô tả
1	_id	ObjectId	Khóa chính
2	sessionId	String	ID phiên chat
3	userName	String	Tên người chat
4	userEmail	String	Email người chat
5	status	Enum	Trạng thái phiên (Active/Closed)
6	lastMessage	DateTime	Thời gian tin nhắn cuối
7	createdAt	DateTime	Thời gian bắt đầu
8	updatedAt	DateTime	Thời gian cập nhật

Bảng 3.11. Chi tiết các thực thể Message_Details

STT	Thuộc tính	Kiểu	Mô tả
c	chatId	ObjectId	ID phiên chat liên kết
2	text	String	Nội dung tin nhắn
3	sender	Enum	Người gửi (User/Admin)
4	time	DateTime	Thời gian gửi

Bảng 3.12. Chi tiết các thực thể Filters

STT	Thuộc tính	Kiểu	Mô tả
1	_id	ObjectId	Khóa chính
2	name	String	Tên kỹ thuật của bộ lọc
3	displayName	String	Tên hiển thị (VD: Dung lượng)
4	type	Enum	Loại lọc (Checkbox/Range/Radio)
5	category	String	ID danh mục áp dụng
6	options	Array	Các lựa chọn lọc (VD: 8GB, 16GB)
7	order	Number	Thứ tự hiển thị
8	isActive	Boolean	Trạng thái kích hoạt
9	createdAt	DateTime	Ngày tạo
10	updatedAt	DateTime	Ngày cập nhật

3.3.3. Thiết kế giao diện người dùng

Giao diện người dùng (UI) của hệ thống website bán linh kiện máy tính được thiết kế theo phong cách hiện đại (Material Design), tập trung vào sự tối giản và trực quan để làm nổi bật hình ảnh và thông số kỹ thuật của sản phẩm.

Các bản mẫu giao diện (Prototypes) được xây dựng trước khi lập trình để thống nhất luồng trải nghiệm người dùng (UX Flow).

- Giao diện Trang chủ (Home Page) Trang chủ là nơi gây ấn tượng đầu tiên với khách hàng. Bố cục được chia thành các khối rõ ràng:

+ Thanh điều hướng (Navbar): Chứa logo, thanh tìm kiếm thông minh (Live Search), giỏ hàng và menu tài khoản. Thanh này được ghim cố định (Sticky) để người dùng dễ dàng truy cập.

+ Banner quảng cáo: Sử dụng Carousel (Slide ảnh) để hiển thị các chương trình khuyến mãi hoặc sản phẩm mới ra mắt.

+ Danh mục sản phẩm: Hiển thị các icon đại diện cho từng loại linh kiện (VGA, CPU, RAM...) giúp điều hướng nhanh.

+ Khối sản phẩm nổi bật: Hiển thị lưới sản phẩm (Product Grid) với hình ảnh, tên, giá bán và nút "Thêm vào giỏ" nhanh.



Hình 3.4. Phác thảo giao diện trang chủ

- Giao diện Chi tiết sản phẩm (Product Detail Page) Đây là trang quan trọng nhất hỗ trợ quyết định mua hàng. Giao diện được chia làm hai cột chính:

+ Cột trái: Hiển thị thư viện ảnh sản phẩm với tính năng Zoom chi tiết.

+ Cột phải: Hiển thị tên sản phẩm, giá bán, tình trạng kho hàng (Còn hàng/Hết hàng), các tùy chọn cấu hình và nút kêu gọi hành động (Call To Action).

+ Phần dưới: Chứa các tab thông tin: "Mô tả chi tiết", "Thông số kỹ thuật" (dạng bảng so sánh) và "Đánh giá từ khách hàng".

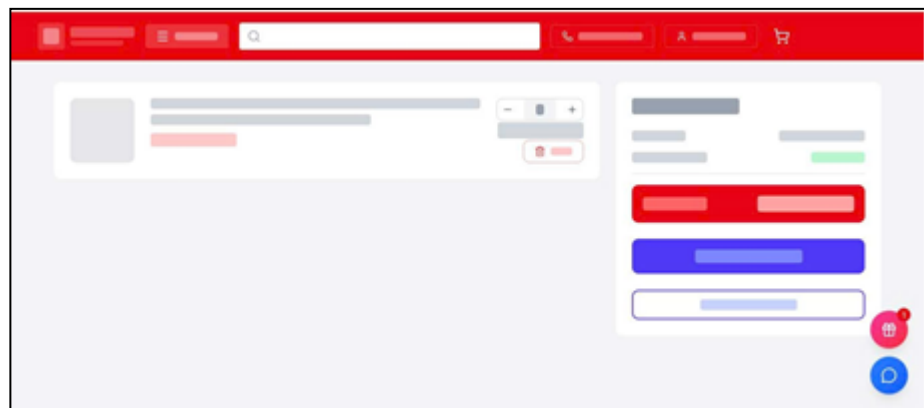


Hình 3.5. Phác thảo giao diện chi tiết sản phẩm

- Giao diện Giỏ hàng và Thanh toán:

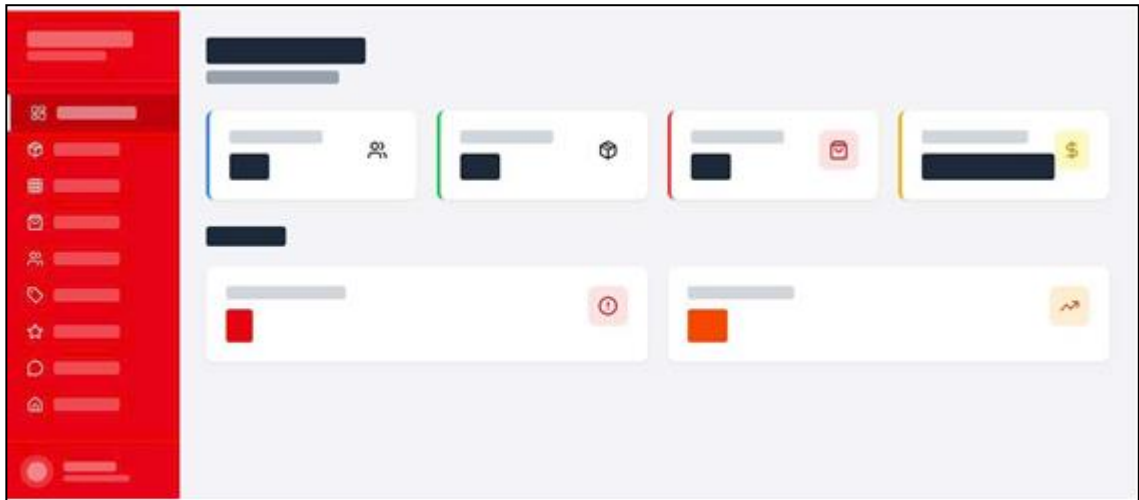
+ Giỏ hàng: Thiết kế dạng bảng liệt kê sản phẩm, cho phép người dùng điều chỉnh số lượng hoặc xóa sản phẩm. Cột bên phải hiển thị tổng tiền tạm tính.

+ Thanh toán: Giao diện tối giản, chia làm 3 bước rõ ràng: Thông tin giao hàng -> Chọn phương thức thanh toán -> Xác nhận đơn hàng.



Hình 3.6. Phác thảo giao diện giỏ hàng và thanh toán

- Giao diện Quản trị (Admin Dashboard) Giao diện dành cho quản trị viên được thiết kế dạng Dashboard với thanh Sidebar bên trái chứa các menu quản lý. Khu vực trung tâm hiển thị các biểu đồ thống kê (Chart.js) và các bảng dữ liệu (Data Tables) hỗ trợ phân trang, sắp xếp và lọc dữ liệu.



Hình 3.7. Phác thảo giao diện trang quản trị

3.4. Thiết kế, kiến trúc ứng dụng

3.4.1. Xây dựng backend

Backend của hệ thống được xây dựng trên nền tảng NodeJS kết hợp với framework Express.js, đóng vai trò là trung tâm xử lý logic nghiệp vụ và cung cấp API RESTful cho Frontend.

Cấu trúc thư mục Backend được tổ chức theo mô hình phân lớp (Layered Architecture) để đảm bảo tính rõ ràng và dễ bảo trì:

Entry Point (server.js) Đây là điểm khởi đầu của ứng dụng. Tại đây, hệ thống thực hiện:

- Kết nối đến cơ sở dữ liệu MongoDB Atlas.
- Cấu hình các Global Middleware (CORS, Body Parser).
- Khởi tạo Passport.js để xử lý đăng nhập Google.
- Tích hợp Socket.IO để lắng nghe sự kiện Chat và Thông báo thời gian thực.
- Định tuyến các request đến thư mục routes/.

Quản lý Cơ sở dữ liệu (models/) Hệ thống sử dụng thư viện Mongoose để tương tác với MongoDB. Các Schema được định nghĩa chặt chẽ để đảm bảo tính toàn vẹn dữ liệu:

- Nhóm Bán hàng: Product.js, Category.js, Coupon.js quản lý kho hàng và khuyến mãi.
- Nhóm Giao dịch: Order.js, Cart.js lưu trữ thông tin mua sắm và thanh toán.

- Nhóm Người dùng: User.js lưu trữ thông tin tài khoản, tích hợp các trường cho Google ID.

- Nhóm Tương tác: Review.js, ChatMessage.js phục vụ tính năng đánh giá và chat hỗ trợ.

- Tính năng nâng cao: Filter.js hỗ trợ cấu hình bộ lọc động cho sản phẩm.

Xác thực và Bảo mật (config/ & middleware/) Hệ thống áp dụng cơ chế bảo mật đa lớp:

- Authentication: Sử dụng config/passport.js để xử lý xác thực qua Google OAuth, giúp người dùng đăng nhập nhanh chóng mà không cần nhớ mật khẩu.

- Authorization: Middleware middleware/auth.js kiểm tra JWT (JSON Web Token) trong Header của mỗi request để xác định quyền hạn (Guest, Customer, Staff, Admin).

- File Upload: Middleware middleware/upload.js (sử dụng Multer) kiểm tra định dạng và kích thước file ảnh trước khi lưu vào thư mục uploads/.

Hệ thống Định tuyến (routes/) Các API chia nhỏ theo chức năng nghiệp vụ:

- Thanh toán: routes/zalopay.js chuyên biệt để xử lý việc tạo đơn hàng ZaloPay và nhận Callback xác nhận thanh toán từ cổng thanh toán.

- Quản trị: routes/admin.js cung cấp các API thống kê doanh thu và quản lý hệ thống dành riêng cho Admin.

- Tương tác: routes/chat.js quản lý lịch sử tin nhắn và routes/reviews.js xử lý đánh giá sản phẩm.

- Nghiệp vụ cơ bản: Các route products.js, orders.js, auth.js xử lý các tác vụ CRUD thông thường.

```
backend/
├── config/           # Cấu hình hệ thống (Passport cho Google OAuth)
├── data/             # Dữ liệu tĩnh (Fallback data)
├── middleware/       # Các hàm trung gian xử lý request (Auth, Upload)
├── models/           # Định nghĩa Schema Mongoose (10 bảng dữ liệu)
├── routes/           # Định nghĩa các API Endpoint (RESTful)
├── uploads/          # Thư mục lưu trữ tạm thời file upload
├── server.js         # Entry point (Khởi tạo server, kết nối DB, Socket.IO)
└── .env              # Biến môi trường (Bảo mật)
```

Hình 3.8. Cấu trúc thư mục backend

3.4.2. Xây dựng frontend

Frontend của hệ thống được xây dựng dựa trên thư viện ReactJS phiên bản 18, áp dụng kiến trúc Single Page Application (SPA) để tối ưu hóa trải nghiệm người dùng, giúp chuyển trang mượt mà không cần tải lại trình duyệt. Mã nguồn được tổ chức theo hướng Component-based (dựa trên thành phần), chia nhỏ giao diện thành các khối độc lập để dễ dàng tái sử dụng và bảo trì.

Thành phần giao diện tái sử dụng (components/) Thư mục này chứa 13 component được sử dụng lặp lại ở nhiều nơi, giúp đồng bộ giao diện toàn hệ thống:

- Nhóm điều hướng: Header.js, Footer.js, MegaMenu.js (Menu đa cấp giúp hiển thị danh mục sản phẩm rõ ràng), ScrollToTop.js.
- Nhóm sản phẩm: ProductCard.js (Hiển thị tóm tắt sản phẩm), ProductFilter.js & FilterBar.js (Bộ lọc tìm kiếm), ProductReviews.js (Hiển thị đánh giá).

Quản lý Trang màn hình (pages/) Hệ thống phân tách rõ ràng giữa giao diện Khách hàng (Client) và Quản trị viên (Admin):

- Phân hệ khách hàng:
 - + Luồng mua sắm: home.js (trang chủ), productdetail.js, cart.js (giỏ hàng), checkout.js (thanh toán), ordersuccess.js (thông báo kết quả).
 - + Luồng xác thực: login.js, register.js, forgotpassword.js, resetpassword.js. đặc biệt authcallback.js dùng để xử lý redirect khi đăng nhập thành công qua google.
 - + Luồng cá nhân: profile.js, myorders.js.
 - + Thanh toán: zalopaycallback.js xử lý kết quả trả về từ cổng thanh toán zalopay.
- Phân hệ Quản trị (pages/Admin/):
 - + Sử dụng AdminLayout.js làm khung sườn chung (Sidebar, Header).
 - + Dashboard.js: Hiển thị biểu đồ thống kê.

+ Các trang quản lý nghiệp vụ CRUD: AdminProducts.js, AdminOrders.js, AdminUsers.js, AdminCoupons.js.

+ AdminChat.js: Giao diện trung tâm để nhân viên phản hồi tin nhắn khách hàng.

Quản lý trạng thái toàn cục (context/) thay vì truyền dữ liệu qua nhiều cấp (props drilling), hệ thống sử dụng react context api:

- AuthContext.js: Lưu trữ thông tin người dùng đăng nhập, Token xác thực và trạng thái phân quyền (Admin/User).

- CartContext.js: Quản lý danh sách sản phẩm trong giỏ hàng, tự động tính tổng tiền và đồng bộ với LocalStorage.

Lớp dịch vụ api (services/) file api.js đóng vai trò là cầu nối giữa frontend và backend. hệ thống sử dụng thư viện axios được cấu hình sẵn (axios instance):

- Interceptors: Tự động đính kèm JWT Token vào Header của mọi request gửi.

- Error Handling: Tự động đăng xuất nếu Token hết hạn hoặc server trả về lỗi xác thực (401).

Giao diện và Responsive (css/) Bên cạnh App.css chứa các style chung, hệ thống tách biệt file mobile-responsive.css để xử lý các Media Query, đảm bảo giao diện hiển thị tốt trên điện thoại di động và máy tính bảng (ẩn hiện menu, thay đổi kích thước lưới sản phẩm).

```
frontend/src/  
├── components/      # Các thành phần giao diện tái sử dụng (UI Blocks)  
├── pages/           # Các trang màn hình chính (Screens)  
│   └── Admin/       # Phân hệ quản trị viên  
├── context/         # Quản lý trạng thái toàn cục (Global State)  
├── services/        # Cấu hình gọi API (Axios)  
└── mobile-responsive.css # Cấu hình hiển thị đa thiết bị
```

Hình 3.9. Cấu trúc thư mục frontend

3.4.3. Xây dựng chức năng giao tiếp thời gian thực (Socket.IO)

Nhằm nâng cao tính tương tác, hệ thống tích hợp thư viện Socket.IO để thiết lập kênh giao tiếp hai chiều (Bi-directional) giữa Client và Server.

Cơ chế hoạt động:

- Kết nối: Khi người dùng truy cập website, Client sẽ khởi tạo một kết nối WebSocket bền vững đến Server.

- Lắng nghe sự kiện (Listening): Server lắng nghe các sự kiện cụ thể như `send_message` (gửi tin nhắn chat) hoặc `order_created` (đặt hàng thành công).

- Phát sự kiện (Emitting):

- + Chat: khi server nhận tin nhắn từ khách hàng, nó sẽ phát sự kiện `receive_message` tới đúng admin đang online để hiển thị tin nhắn ngay lập tức.

- + Thông báo: Khi có đơn hàng mới, Server phát sự kiện `notification` tới Dashboard của Admin, kích hoạt thông báo popup hoặc âm thanh cảnh báo.

3.5. Cài đặt và triển khai hệ thống

3.5.1. Môi trường và công cụ phát triển

Hệ thống được phát triển và kiểm thử trên môi trường tiêu chuẩn với các công cụ sau:

- Hệ điều hành: Windows 10.
- Ngôn ngữ lập trình: JavaScript (ES6+).
- Frontend: ReactJS v18.
- Backend: Node.js v16+, Express.js.
- Cơ sở dữ liệu: MongoDB Atlas (Cloud Database).
- Công cụ soạn thảo mã: Visual Studio Code.
- Postman: kiểm thử API
- Quản lý phiên bản: Git & GitHub.
- Quản lý Database: MongoDB Compass.
- Trình duyệt kiểm thử: Google Chrome.

3.5.2. Triển khai và chạy thử nghiệm

Sau khi hoàn tất triển khai cả hai phân hệ lên môi trường Internet, tiến hành chạy thử nghiệm tổng thể (End-to-End Testing):

- Kiểm thử luồng Mua hàng: Đăng ký tài khoản -> Tìm kiếm sản phẩm -> Thêm vào giỏ -> Đặt hàng và thanh toán (ZaloPay) -> Kiểm tra email xác nhận.

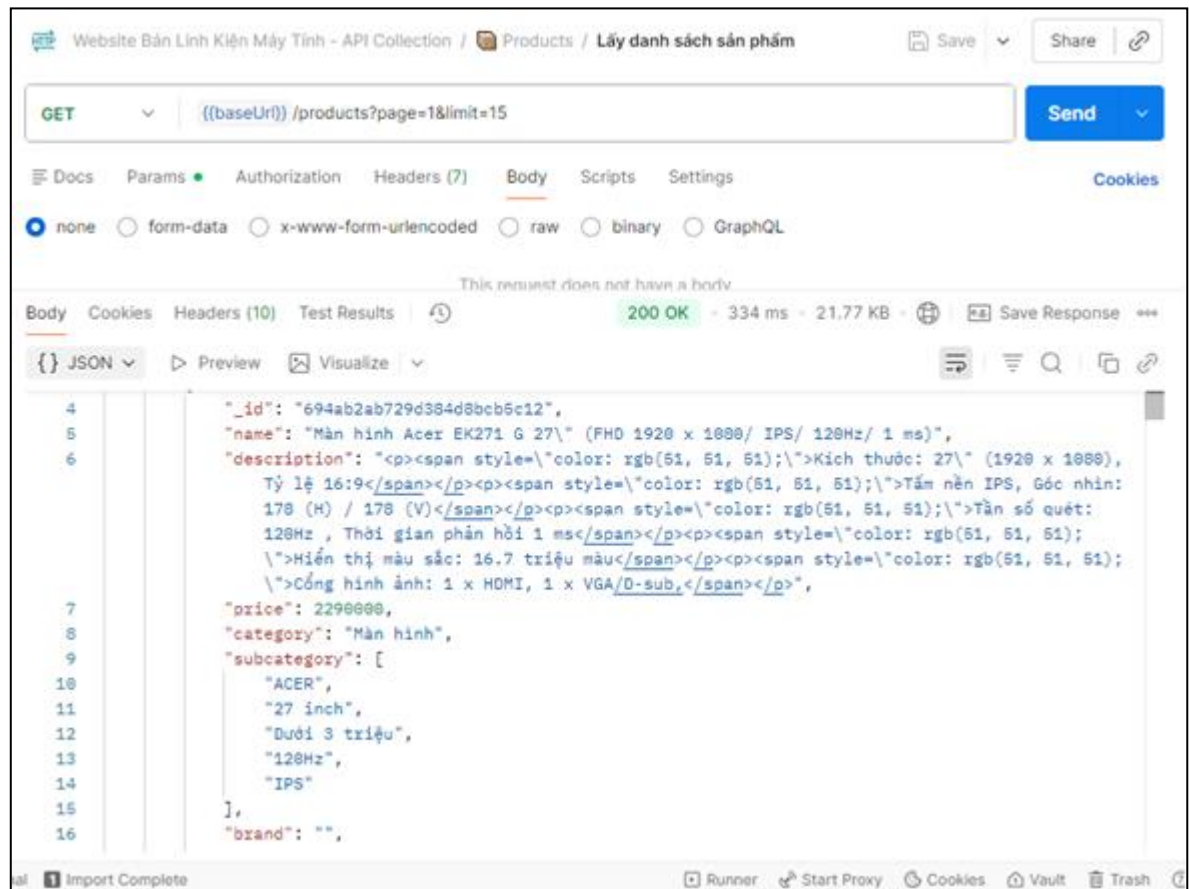
- Kiểm thử luồng Quản trị: Đăng nhập Admin -> Thêm sản phẩm mới kèm ảnh -> Kiểm tra sản phẩm xuất hiện trên trang chủ.

- Kiểm thử Real-time: Mở hai trình duyệt (Khách và Admin) để kiểm tra tốc độ phản hồi của tin nhắn Chat và thông báo đơn hàng mới.

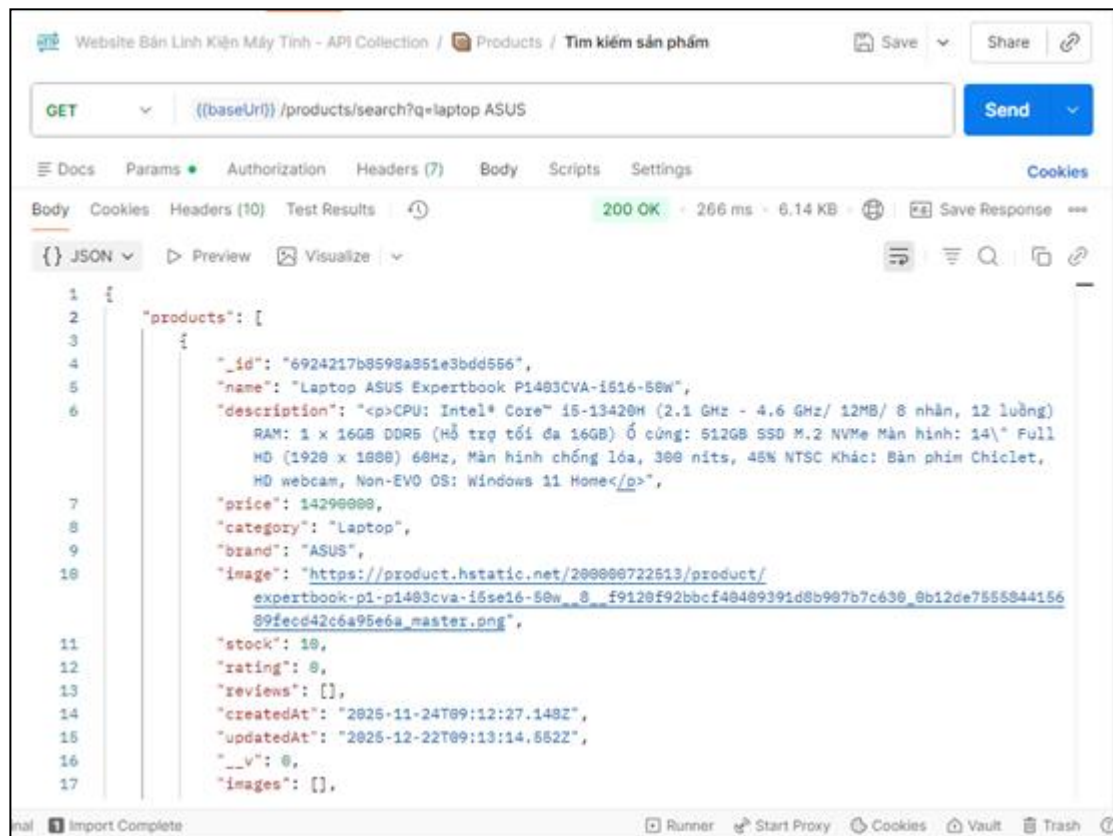
Kết quả thử nghiệm cho thấy hệ thống hoạt động ổn định, không phát sinh lỗi nghiêm trọng và đáp ứng tốt các yêu cầu chức năng cũng như phi chức năng đã đề ra.

100

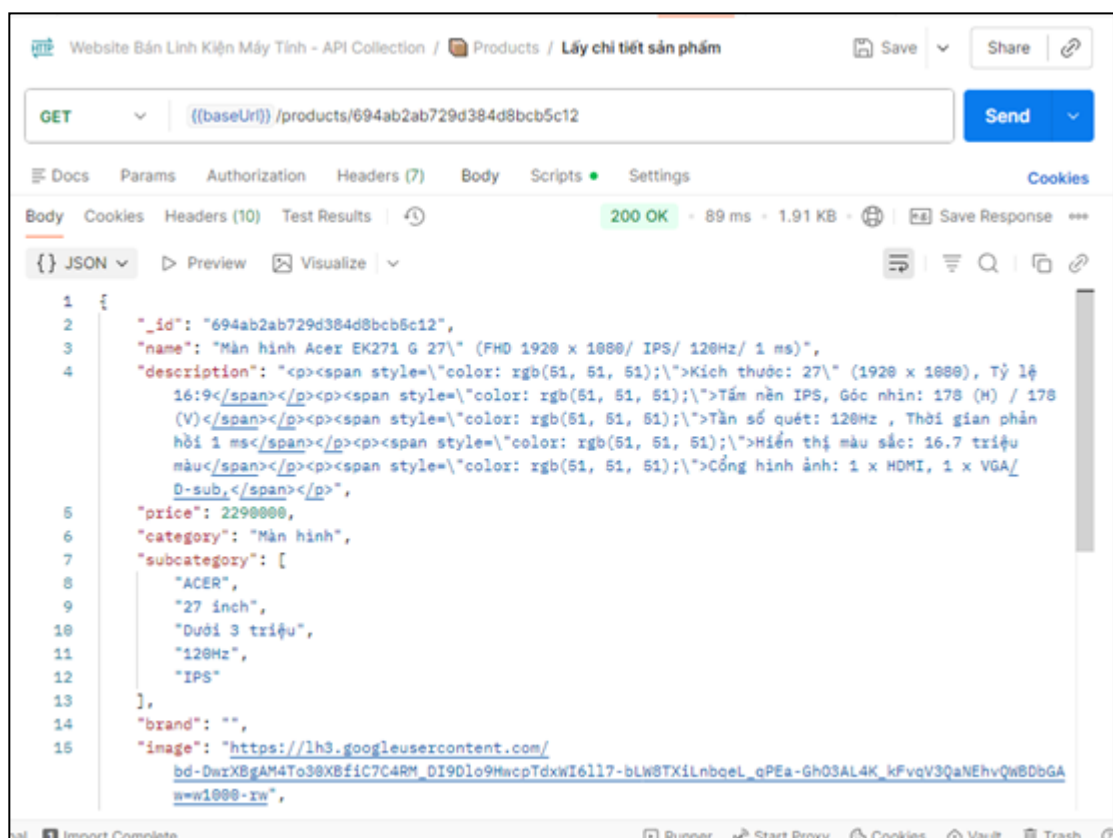
API quản lý sản phẩm: Lấy danh sách linh kiện (có phân trang), tìm sản phẩm theo tiêu chí (Giá, Thương hiệu, Danh mục), xem chi tiết thông số kỹ thuật và thêm mới sản phẩm (quyền Admin).



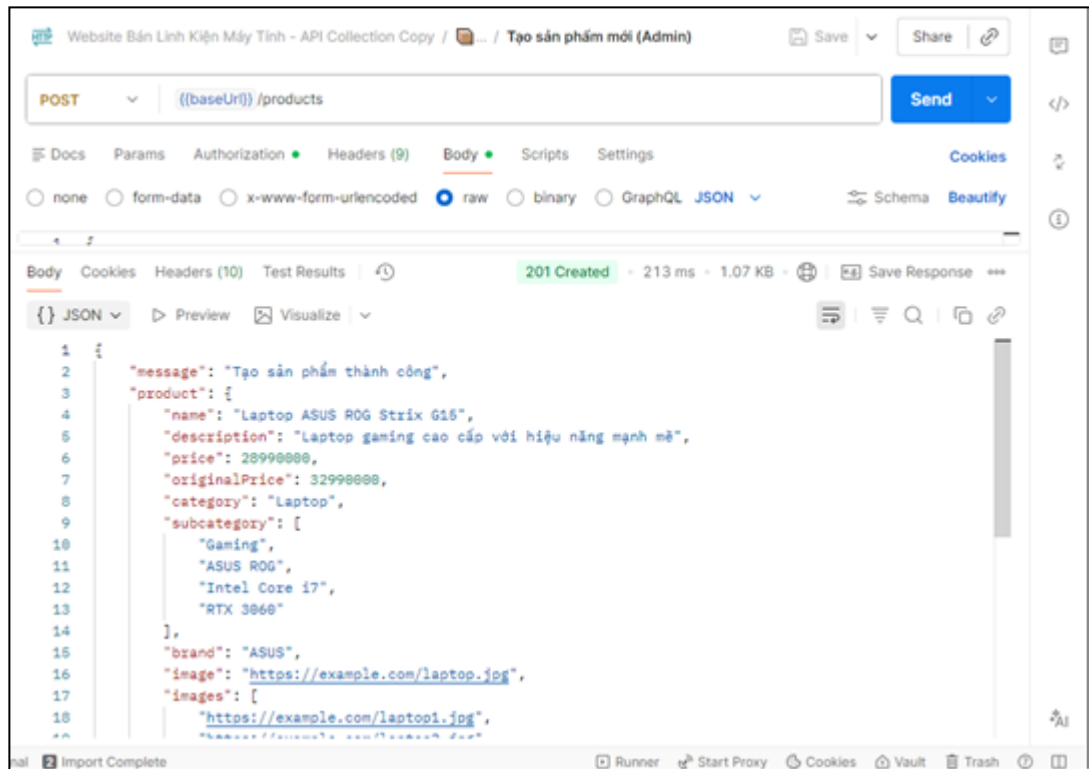
Hình 4.4. Kiểm thử api lấy danh sách linh kiện (có phân trang)



Hình 4.5. Kiểm thử api tìm sản phẩm theo tiêu chí (Giá, Thương hiệu, Danh mục)

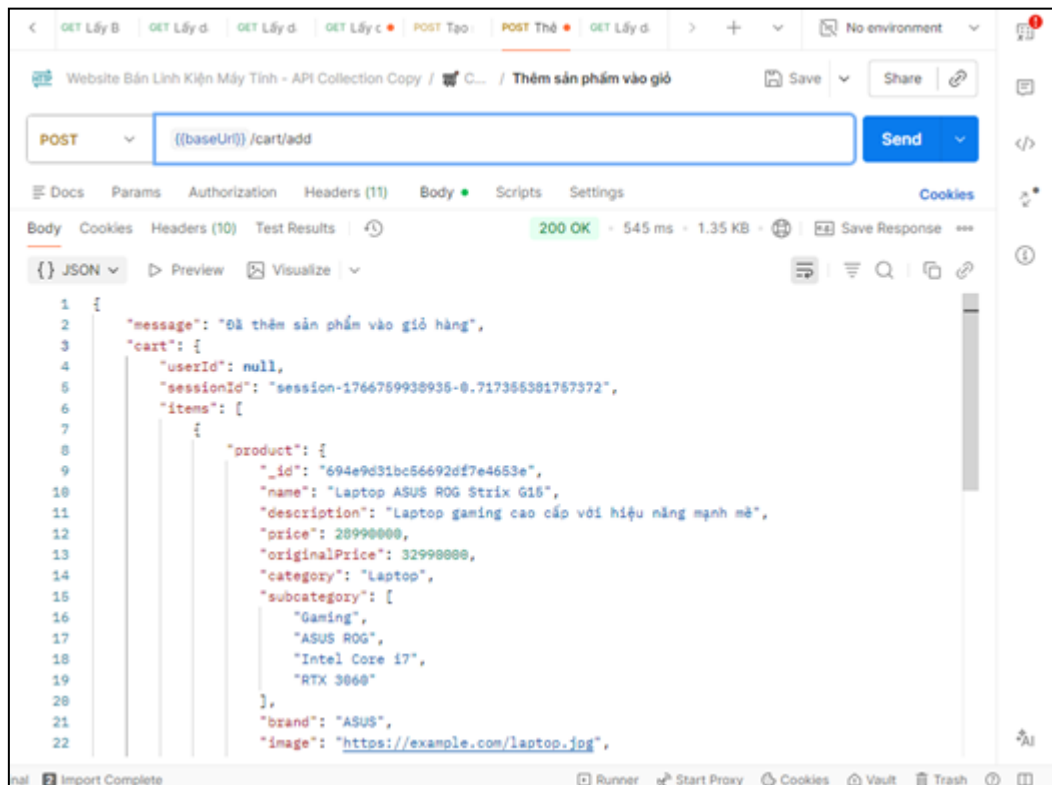


Hình 4.6. Kiểm thử api xem chi tiết thông số kỹ thuật

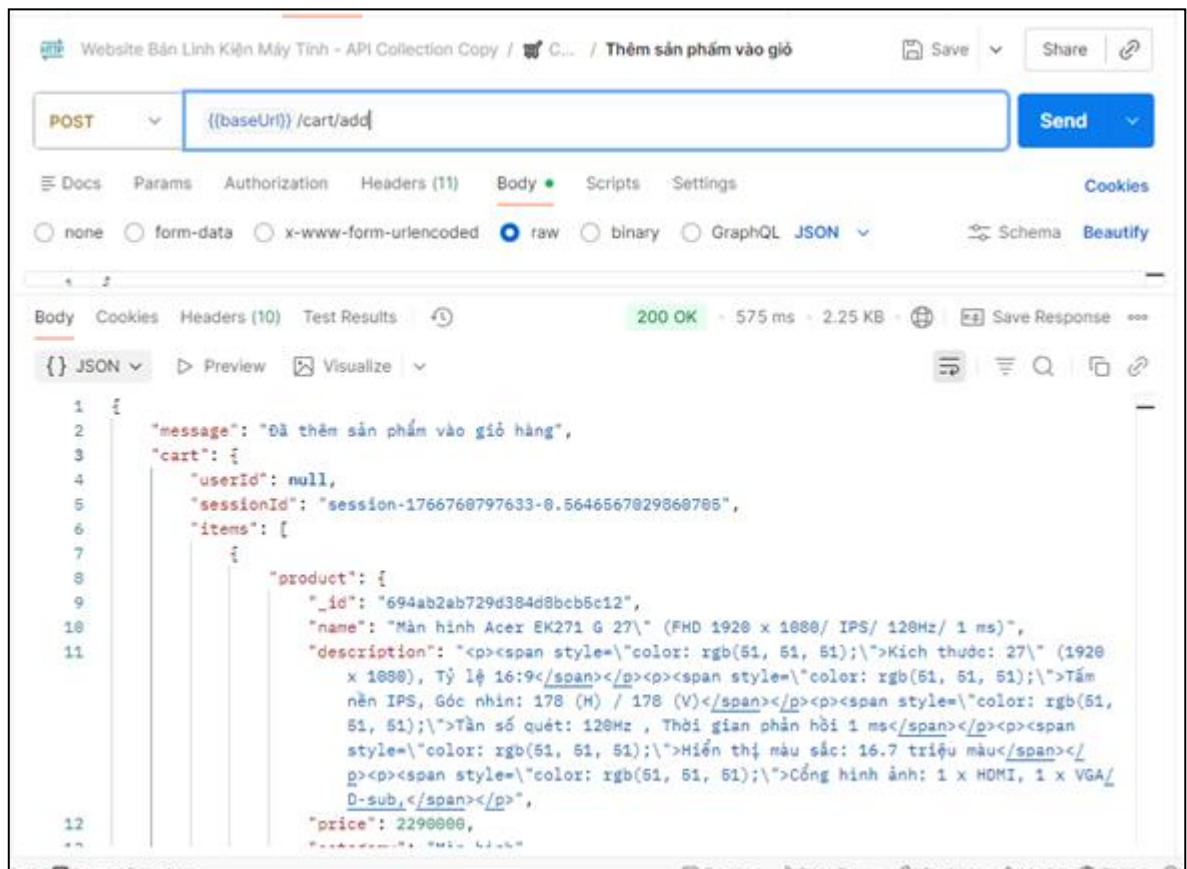


Hình 4.7. Kiểm thử api thêm mới sản phẩm (quyền Admin).

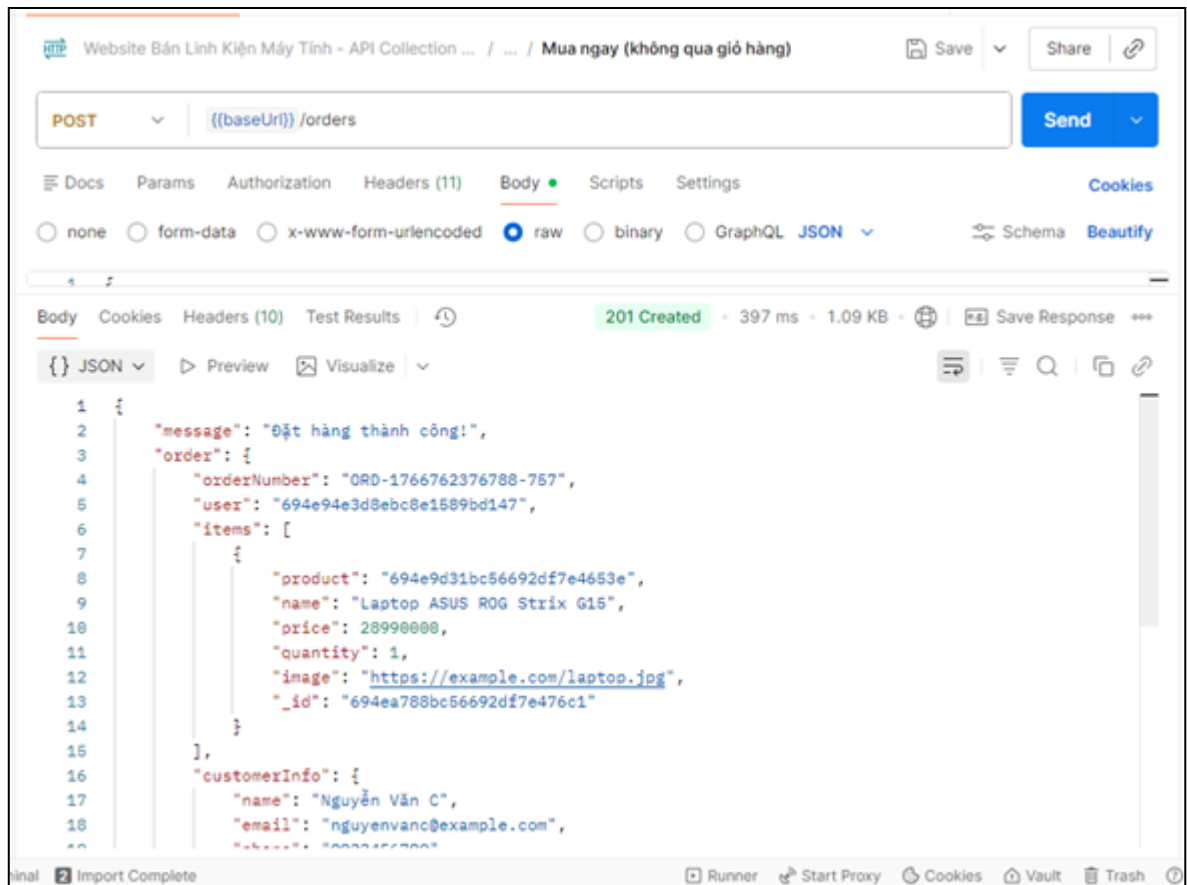
API Giỏ hàng và Đơn hàng: Thêm sản phẩm vào giỏ hàng, tạo đơn hàng mới, tạo đơn hàng và thanh toán thành công.



Hình 4.8. Kiểm thử api thêm sản phẩm vào giỏ hàng

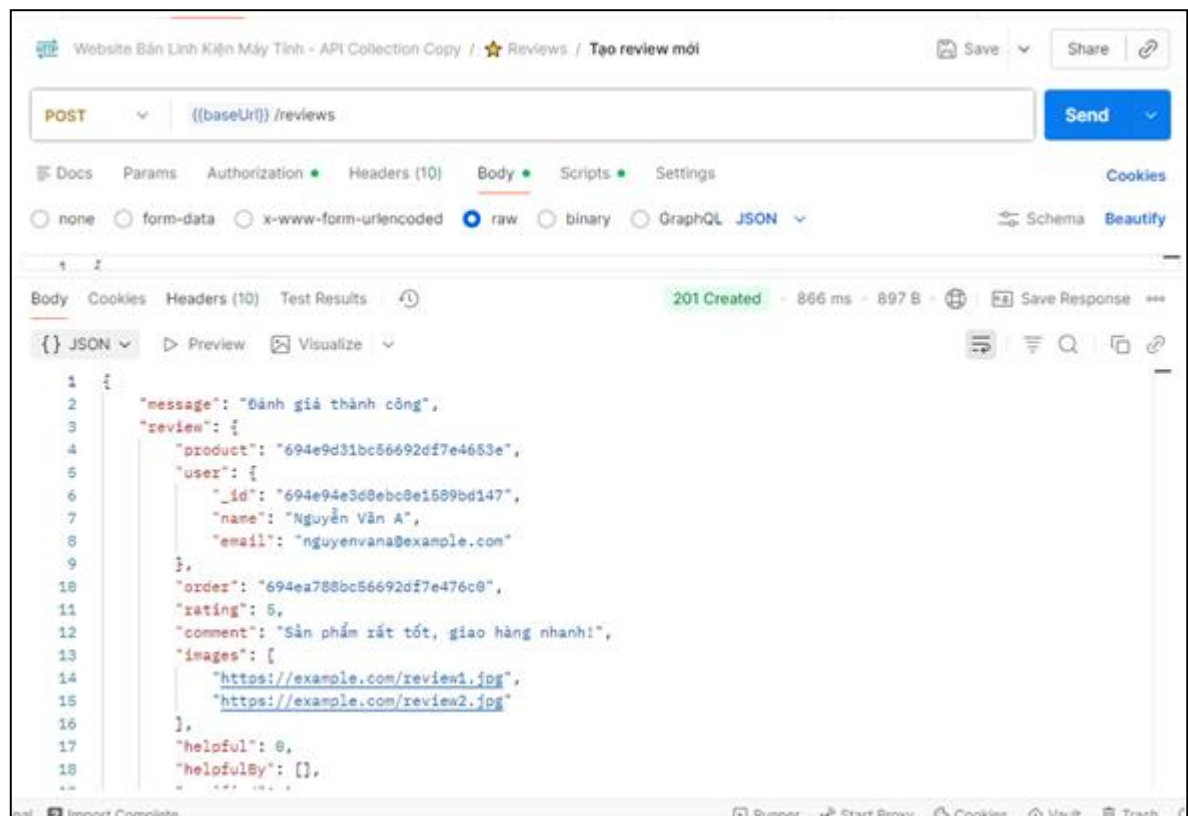


Hình 4.9. Kiểm thử api thêm mới sản phẩm vào giỏ hàng

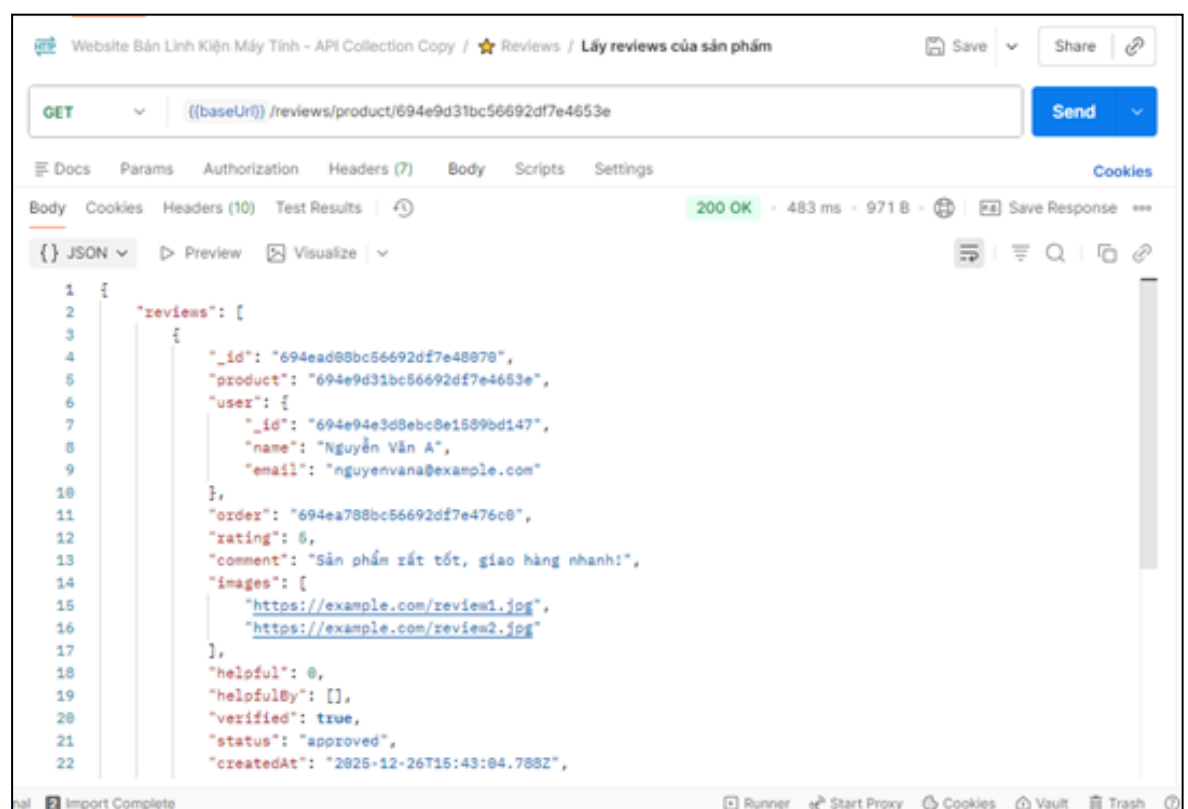


Hình 4.10. Kiểm thử api tạo đơn hàng và thanh toán thành công

API Đánh giá sản phẩm: Gửi đánh giá (Rating & Comment) cho sản phẩm đã mua và lấy danh sách đánh giá công khai.

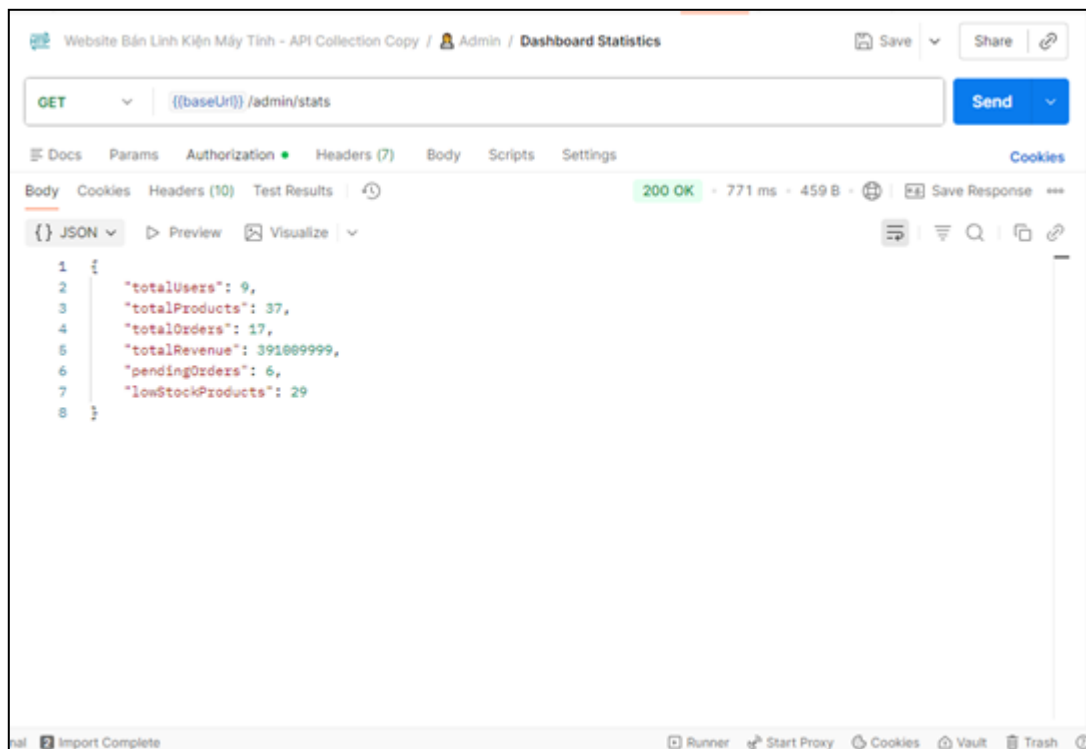


Hình 4.11. Kiểm thử api gửi đánh giá cho sản phẩm đã mua thành công

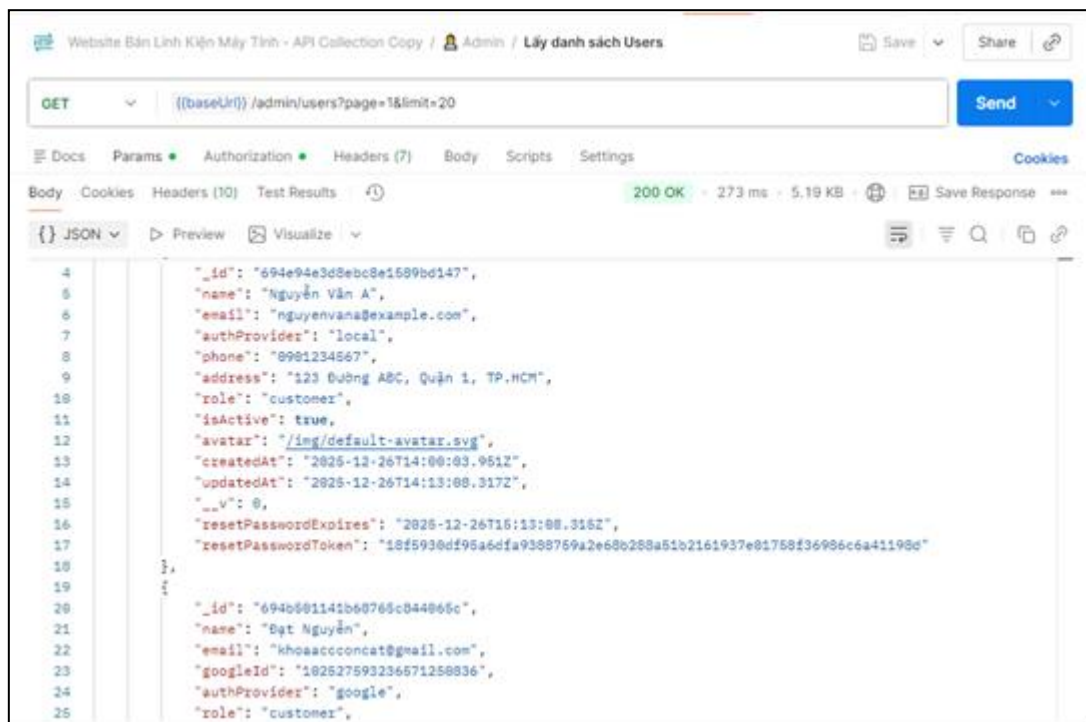


Hình 4.12. Kiểm thử api lấy danh sách đánh giá công khai

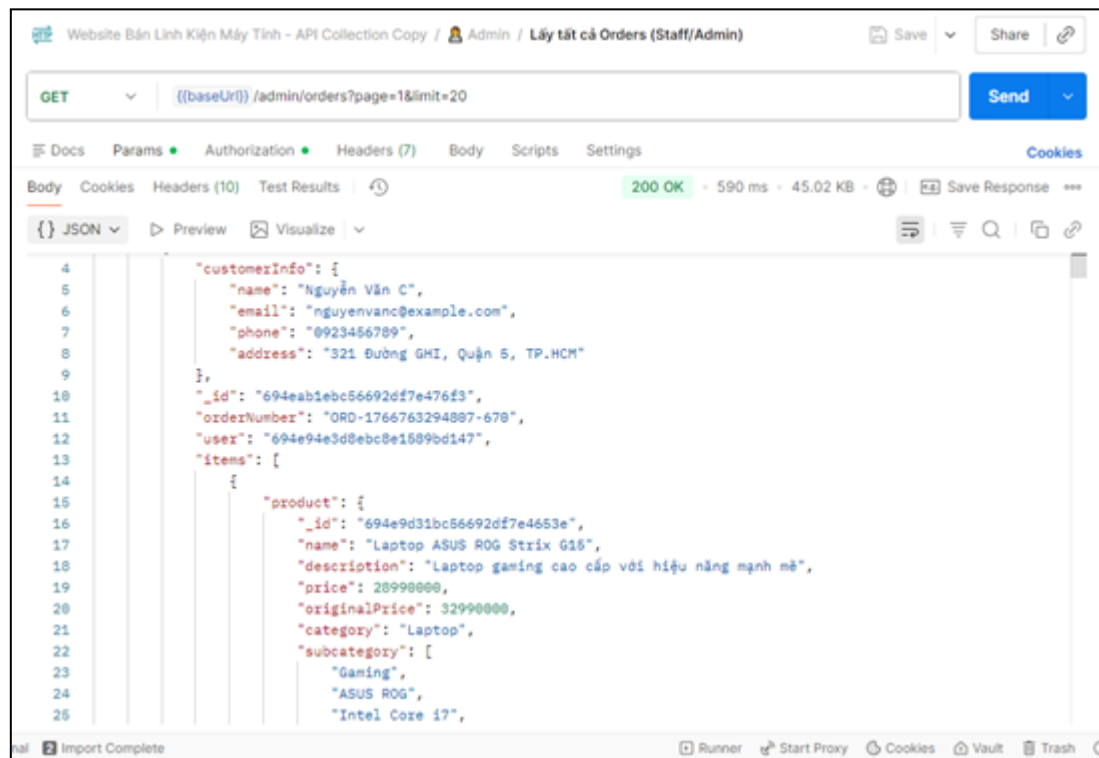
API Quản trị (Admin): Thống kê doanh thu, quản lý danh sách người dùng, cập nhật trạng thái đơn hàng (Duyệt/Giao hàng) và quản lý mã giảm giá.



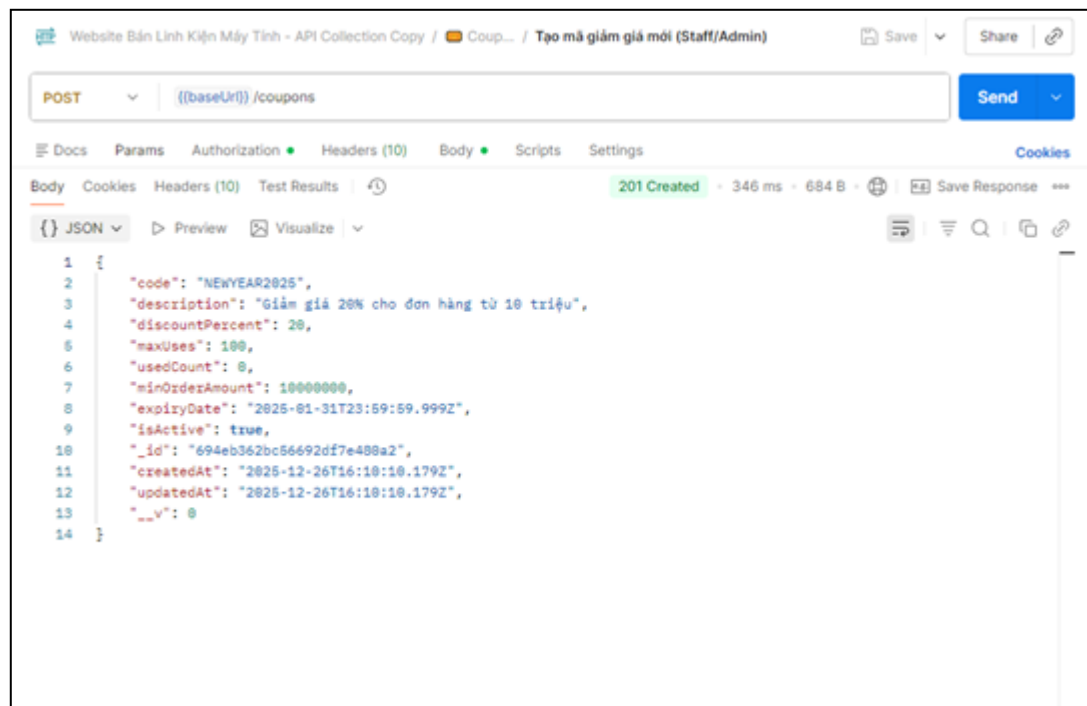
Hình 4.13. Kiểm thử api thống kê doanh thu của admin



Hình 4.14. Kiểm thử api quản lý danh sách người dùng



Hình 4.15. Kiểm thử api cập nhật trạng thái đơn hàng (Duyệt/Giao hàng)

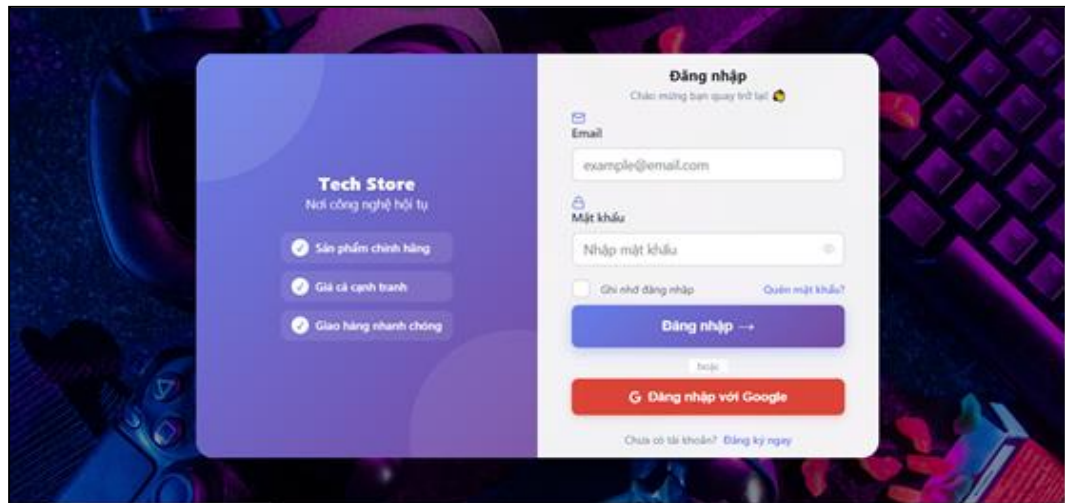


Hình 4.16. Kiểm thử api quản lý mã giảm giá

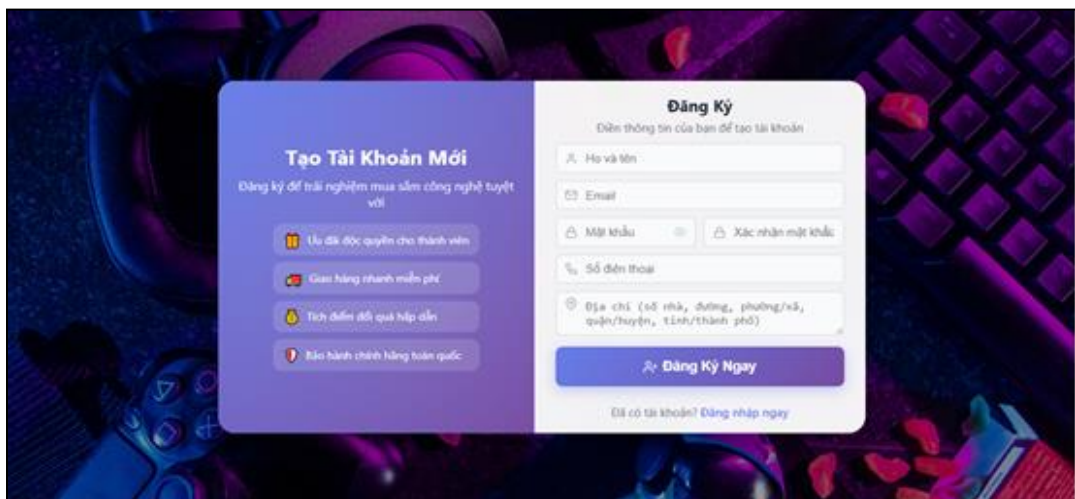
4.2. Giao diện người dùng

Giao diện người dùng của website bán linh kiện máy tính được xây dựng bằng ReactJS, hướng đến trải nghiệm mua sắm hiện đại, trực quan và tối ưu hóa cho việc tìm kiếm thông số kỹ thuật.

Giao diện Đăng ký và Đăng nhập: Thiết kế tối giản, hỗ trợ đăng nhập nhanh bằng Google và xác thực bảo mật.



Hình 4.17. Giao diện đăng nhập

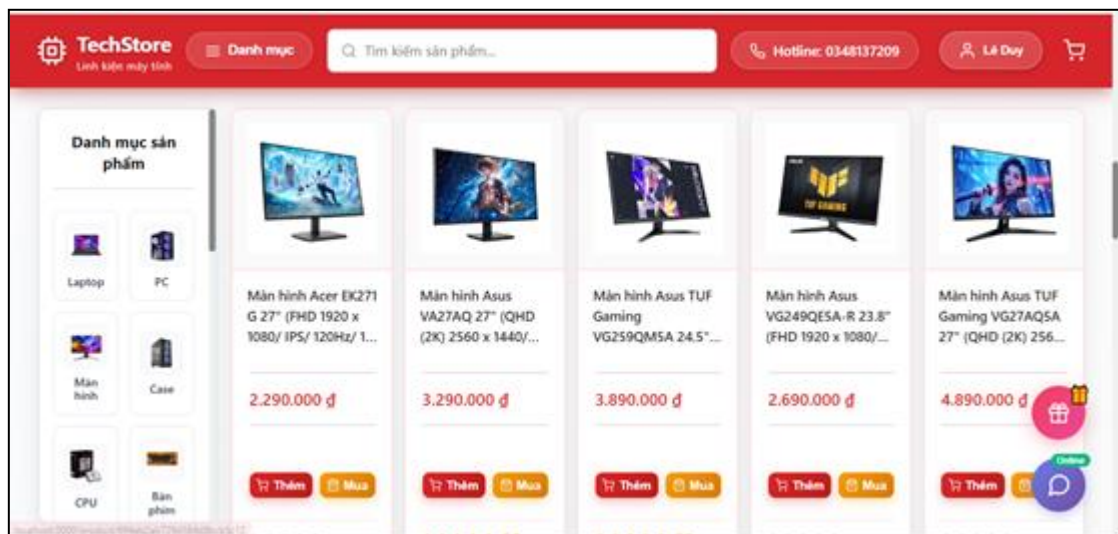


Hình 4.18. Giao diện đăng ký

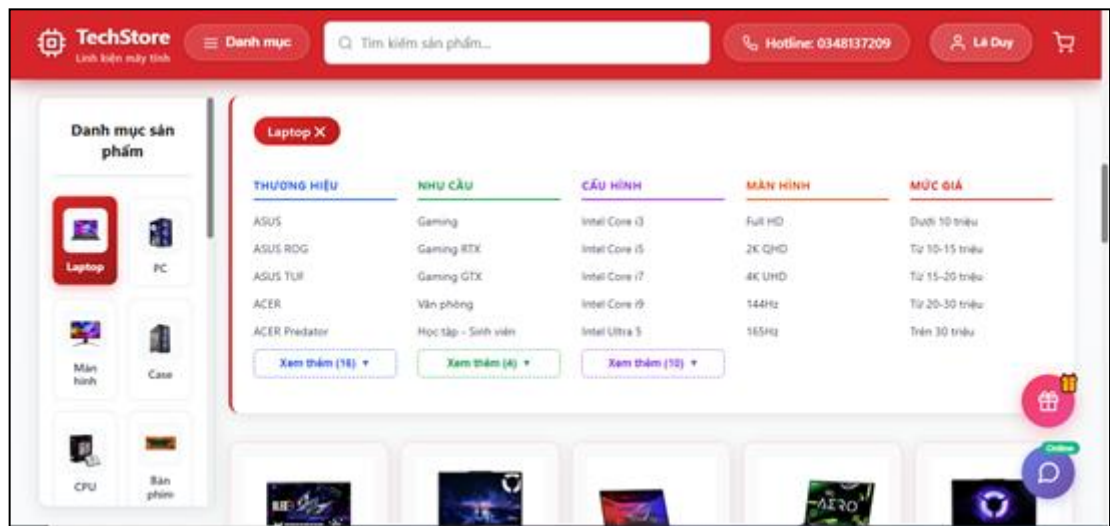
Trang chủ và Danh sách sản phẩm: Hiển thị Banner khuyến mãi, các danh mục linh kiện (VGA, CPU, RAM...) và danh sách sản phẩm nổi bật với bộ lọc tìm kiếm chi tiết bên thanh sidebar.



Hình 4.19. Giao diện trang chủ , banner khuyến mãi

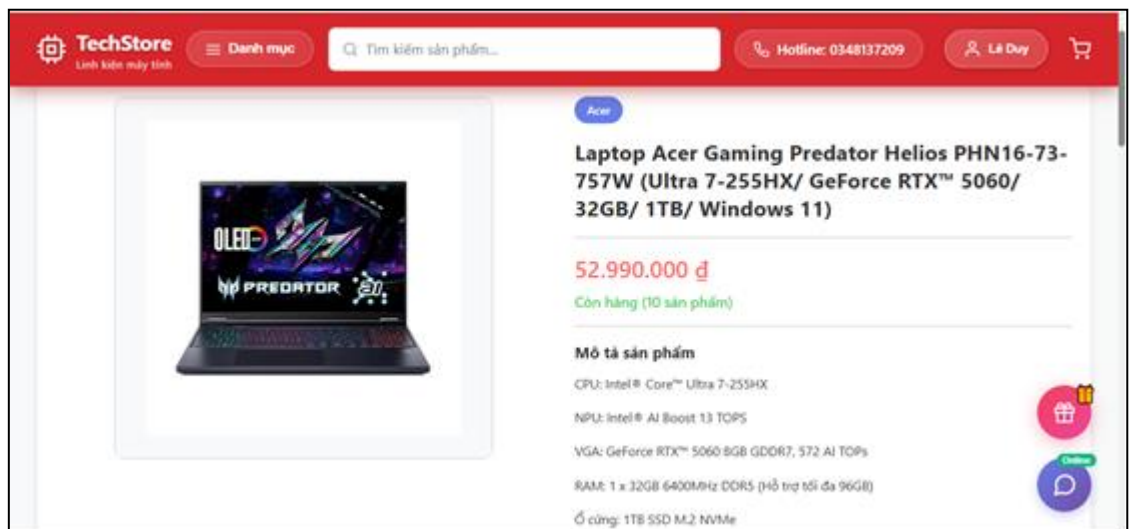


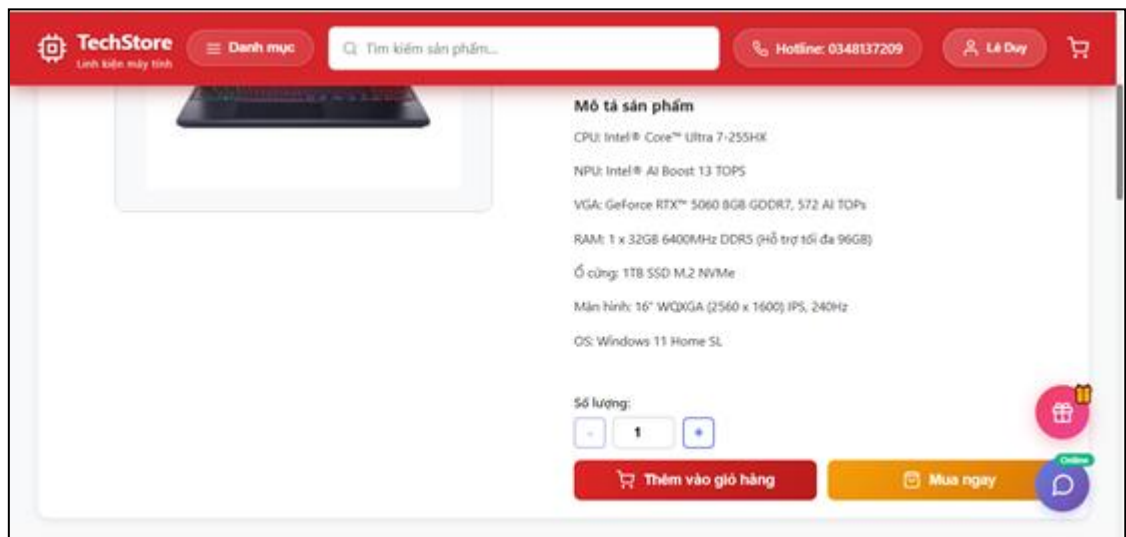
Hình 4.20. Giao diện các danh mục linh kiện (VGA, CPU, RAM...)



Hình 4.21. Giao diện danh sách sản phẩm nổi bật với bộ lọc tìm kiếm chi tiết bên thanh sidebar.

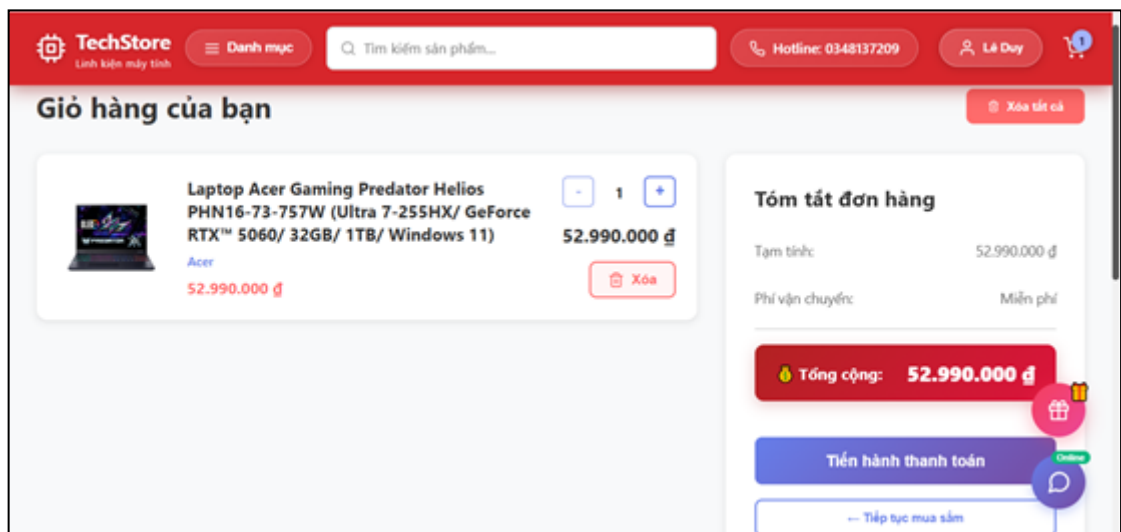
Trang Chi tiết sản phẩm: Hiện thị hình ảnh rõ nét, giá bán, tình trạng kho và bảng thông số kỹ thuật chi tiết – yếu tố quan trọng nhất khi mua linh kiện máy tính.





Hình 4.22. Giao diện hiển thị hình ảnh rõ nét, giá bán, tình trạng kho và bảng thông số kỹ thuật chi tiết

Giao diện Giỏ hàng và Thanh toán: Cho phép người dùng xem lại các sản phẩm đã chọn, nhập mã giảm giá, điền thông tin giao hàng và chọn phương thức thanh toán (COD hoặc ZaloPay).



Hình 4.23. Giao diện giỏ hàng

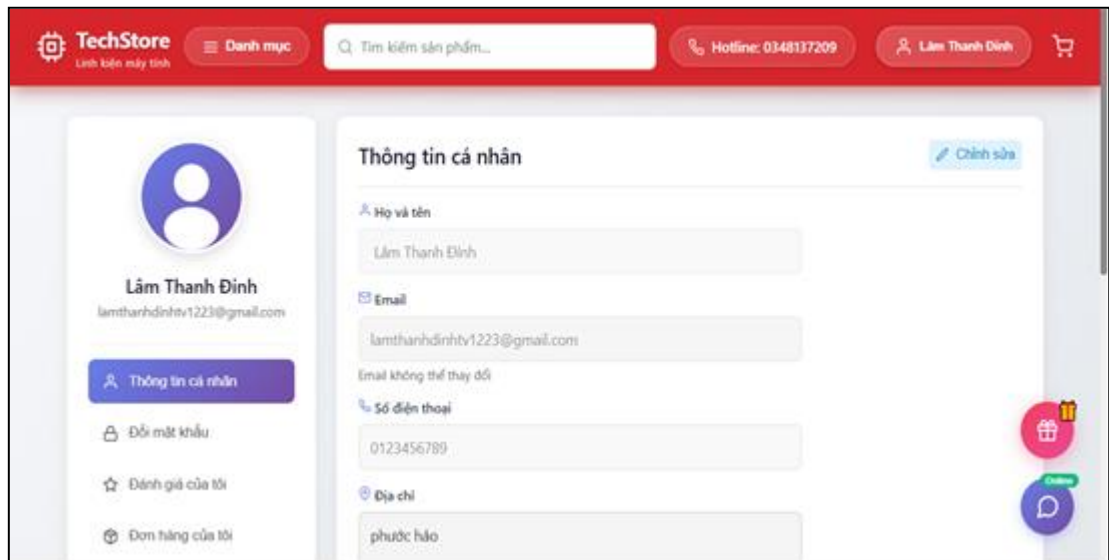
The screenshot displays the 'Thanh toán' (Checkout) page of the TechStore website. The page is divided into two main sections. On the left, there is a form titled 'Thông tin giao hàng' (Shipping Information) with fields for 'Họ và tên' (Full Name), 'Email', 'Số điện thoại' (Phone Number), 'Địa chỉ giao hàng' (Shipping Address) including 'Tỉnh/Thành phố' (Province/City), 'Quận/Huyện' (District/City), 'Phường/Xã' (Ward/Village), and 'Số nhà, tên đường' (House Number, Street Name). On the right, there is a section titled 'Đơn hàng của bạn' (Your Order) showing a 'Laptop Acer Gaming Predator Helios PHN16-73-757W (Ultra 7-255HX/ GeForce RTX™ 5060/ 32GB/ 1TB/ Windows 11)' for 'Số lượng: 1' (Quantity: 1) at a price of '52.990.000 đ'. Below this, there is a 'Mã giảm giá' (Discount Code) field with a placeholder 'NHẬP MÃ GIAM GIÁ' and an 'Áp dụng' (Apply) button. At the bottom right, there is a 'Tạm tính' (Subtotal) of '52.990.000 đ' and a 'Phí vận chuyển' (Shipping Fee) of 'Miễn phí' (Free). The top navigation bar includes the TechStore logo, a menu icon, a search bar, a hotline number '0348137209', and a user profile icon labeled 'Lê Duy'.

Hình 4.24. Giao diện thanh toán

Giao diện Quản lý tài khoản: Nơi người dùng theo dõi lịch sử đơn hàng, trạng thái vận chuyển và cập nhật thông tin cá nhân.

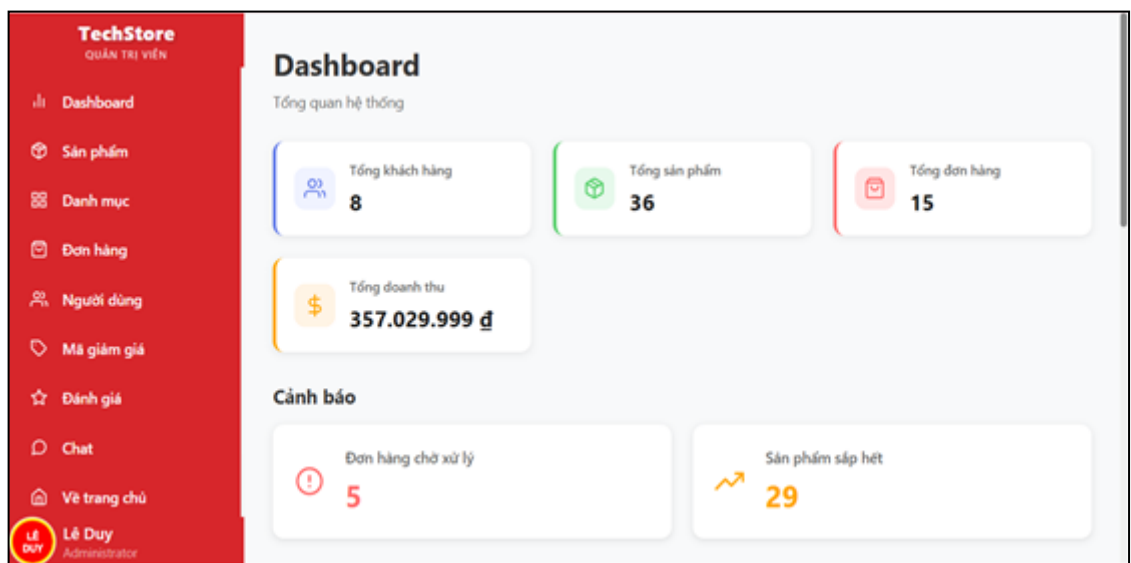
The screenshot displays the 'Theo dõi đơn hàng' (Track Order) page of the TechStore website. The page features a top navigation bar with the TechStore logo, a menu icon, a search bar, a hotline number '0348137209', and a user profile icon labeled 'Lâm Thanh Đình'. Below the navigation bar, there is a section titled 'Đơn hàng #ORD-1766762290777-929' with a timestamp 'lúc 22:18:26 tháng 12, 2025'. The order status is 'Chờ xác nhận' (Waiting for confirmation). Below this, there is a section titled 'Màn hình Acer EK271 G 27" (FHD 1920 x 1080/ IPS/ 120Hz/ 1 ms)' for 'Số lượng: 1' (Quantity: 1) at a price of '2.290.000 đ'. On the right side, there are icons for a gift, a chat bubble, and a speech bubble. The bottom navigation bar includes buttons for 'Tất cả (3)' (All), 'Chờ xác nhận (2)' (Waiting for confirmation), 'Đang xử lý (0)' (Processing), 'Đang giao (0)' (Shipping), and 'Đã giao (1)' (Delivered).

Hình 4.25. Giao diện theo dõi đơn hàng

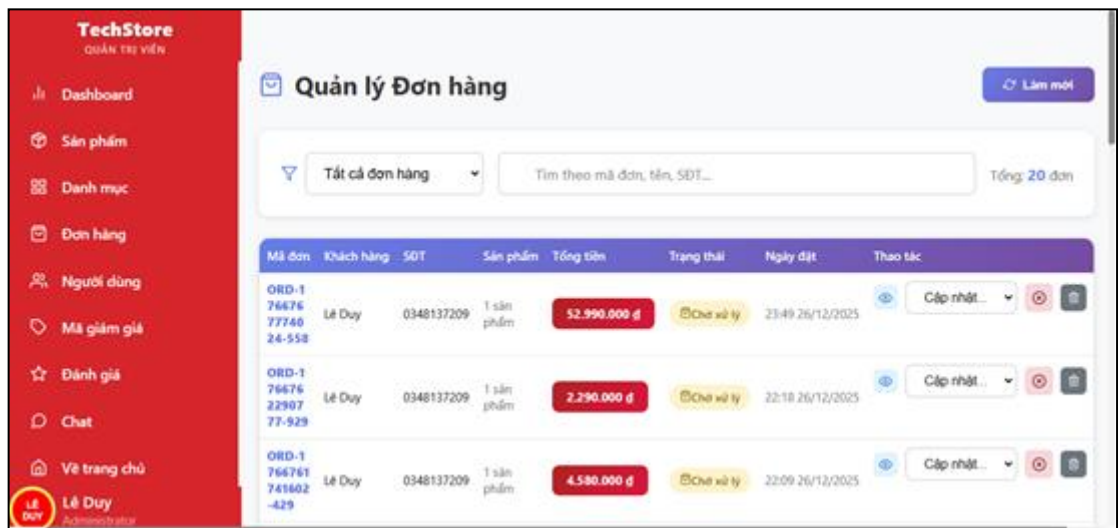


Hình 4.26. Giao diện thông tin người dùng

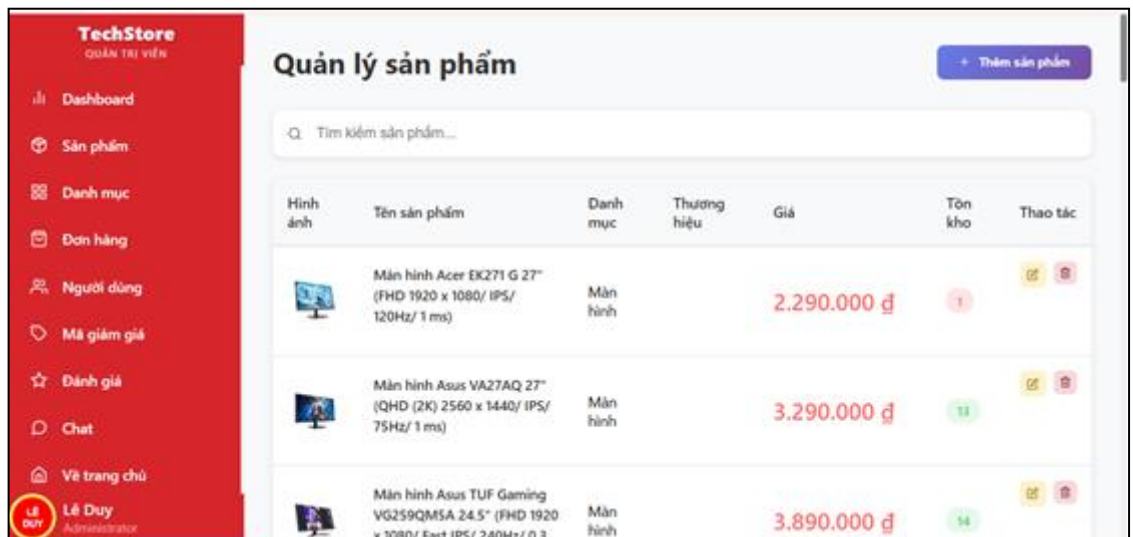
Giao diện Quản trị (Admin Dashboard): Cung cấp cái nhìn tổng quan cho người quản lý với các biểu đồ doanh thu, danh sách đơn hàng cần xử lý và công cụ quản lý kho hàng.



Hình 4.27. Giao diện trang Quản trị của Admin

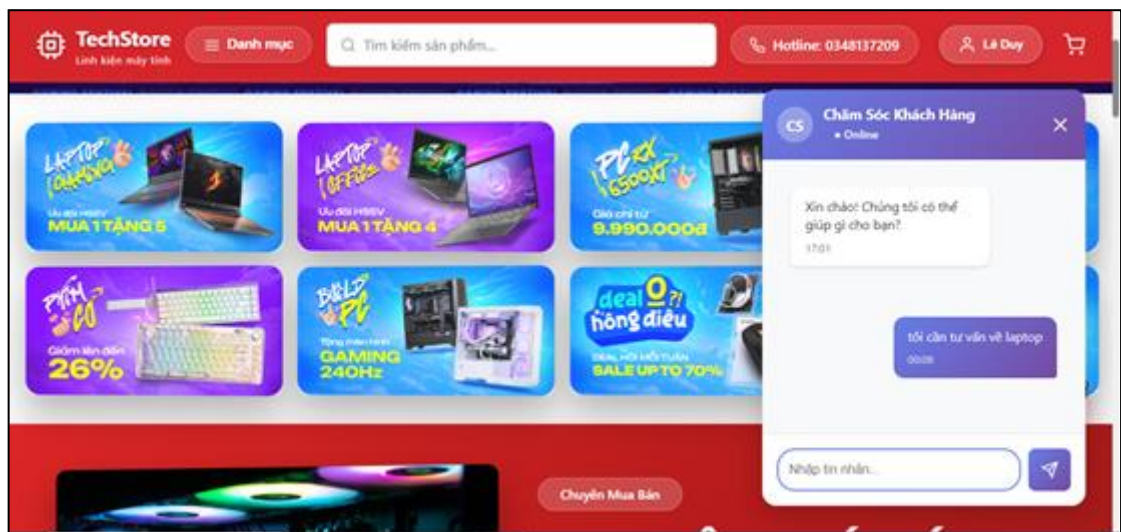


Hình 4.28. Giao diện Quản lý đơn hàng

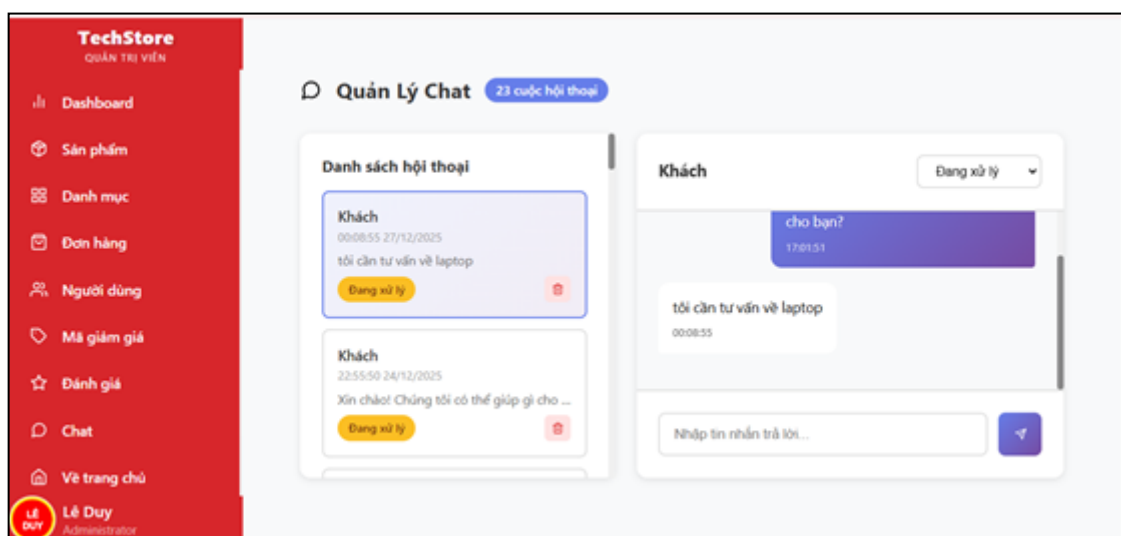


Hình 4.29. Giao diện Quản lý sản phẩm

Giao diện Chat hỗ trợ: Widget chat thời gian thực giúp kết nối khách hàng và nhân viên tư vấn ngay lập tức.



Hình 4.30. Giao diện chat tư vấn khách hàng



Hình 4.31. Giao diện phản hồi chat của Admin

Kết luận: Qua quá trình chạy thử nghiệm, giao diện hoạt động mượt mà, tốc độ tải trang nhanh nhờ công nghệ ReactJS. Các luồng nghiệp vụ quan trọng như "Tìm kiếm -> Thêm giỏ hàng -> Thanh toán" diễn ra trơn tru, logic. Chức năng Chat Real-time hoạt động ổn định, giúp nâng cao chất lượng dịch vụ khách hàng.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Đồ án “Xây dựng hệ thống Website bán linh kiện máy tính” được thực hiện với mục tiêu xây dựng một hệ thống thương mại điện tử phục vụ nhu cầu tìm kiếm, so sánh và mua sắm các thiết bị công nghệ trong thời đại số. Thông qua quá trình nghiên cứu, phân tích yêu cầu, thiết kế và hiện thực hệ thống, đồ án đã hoàn thành các mục tiêu đặt ra ban đầu.

Kết quả của đồ án là xây dựng được một website bán hàng hoạt động ổn định, đáp ứng đầy đủ các chức năng cơ bản như quản lý tài khoản người dùng, tìm kiếm và lọc sản phẩm, quản lý giỏ hàng, thanh toán trực tuyến, thông báo đơn hàng và hỗ trợ trực tuyến. Hệ thống được thiết kế theo mô hình client–server với frontend sử dụng ReactJS, backend sử dụng NodeJS kết hợp Express.js và cơ sở dữ liệu MongoDB, đảm bảo tính linh hoạt và khả năng mở rộng.

Đồ án đã vận dụng hiệu quả các kiến thức đã học về lập trình web full-stack, thiết kế kiến trúc hệ thống, cơ sở dữ liệu NoSQL và các công nghệ tích hợp (thanh toán, giao tiếp thời gian thực) vào việc xây dựng một sản phẩm phần mềm hoàn chỉnh. Kết quả đạt được cho thấy hệ thống đáp ứng tốt các yêu cầu chức năng và phi chức năng đã đề ra, có thể ứng dụng trong thực tế với vai trò là một nền tảng kinh doanh linh kiện điện tử hiệu quả.

5.2. Hướng phát triển

Mặc dù hệ thống đã hoàn thành các chức năng cơ bản của một trang thương mại điện tử, nhưng do giới hạn về thời gian và phạm vi đồ án, hệ thống vẫn còn nhiều tiềm năng để tiếp tục phát triển và hoàn thiện trong tương lai. Một số hướng phát triển có thể được đề xuất như sau:

- Mở rộng thêm các chức năng nâng cao như công cụ tự động xây dựng cấu hình PC (PC Builder), gợi ý sản phẩm dựa trên lịch sử mua hàng hoặc so sánh chi tiết thông số kỹ thuật.

- Cải thiện hiệu năng hệ thống khi số lượng sản phẩm và đơn hàng tăng cao, ứng dụng các công nghệ lưu trữ đệm (Caching) như Redis để tối ưu tốc độ truy vấn.

- Tăng cường các giải pháp bảo mật thanh toán, tích hợp thêm đa dạng các cổng thanh toán điện tử và cơ chế xác thực đa yếu tố cho tài khoản quản trị.

- Phát triển phiên bản ứng dụng trên thiết bị di động (Mobile App) nhằm nâng cao khả năng tiếp cận và sự tiện lợi cho khách hàng khi mua sắm.

- Bổ sung các công cụ phân tích dữ liệu kinh doanh và báo cáo thống kê chuyên sâu nhằm giúp quản trị viên dễ dàng theo dõi doanh thu và xu hướng mua sắm.

Các hướng phát triển trên sẽ góp phần hoàn thiện hệ thống website bán linh kiện máy tính, nâng cao trải nghiệm người dùng và mở rộng khả năng cạnh tranh trong môi trường thương mại điện tử.

DANH MỤC TÀI LIỆU THAM KHẢO

Tài liệu và Giáo trình (Lý thuyết nền tảng)

[1] PGS.TS. Nguyễn Ái Việt (2014), *Giáo trình Nhập môn Công nghệ phần mềm*, Nhà xuất bản Đại học Quốc gia Hà Nội.

[2] Vasan Subramanian (2019), *Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node*, Nhà xuất bản Apress.

[3] Tài liệu hướng dẫn đồ án tốt nghiệp ngành Công nghệ thông tin, [Tên trường của bạn].

Nguồn Hình ảnh và Dữ liệu sản phẩm

[4] Hệ thống bán lẻ CellphoneS. *Danh mục linh kiện máy tính và phụ kiện*. [Trực tuyến]. Địa chỉ: <https://cellphones.com.vn/linh-kien.html>.

[5] Công ty Cổ phần Thương mại Dịch vụ Phong Vũ. *Xây dựng cấu hình PC và Linh kiện máy tính*. [Trực tuyến]. Địa chỉ: <https://phongvu.vn/linh-kien-may-tinh>.

Tài liệu Kỹ thuật và Cộng đồng lập trình (Code & Giải thuật)

[6] React Documentation (Trang chủ). Tài liệu chính thức về React Hooks và Component Lifecycle. [Trực tuyến]. Địa chỉ: <https://react.dev/reference/react>.

[7] Cộng đồng Stack Overflow. Các thảo luận về xử lý giỏ hàng và xác thực JWT trong Node.js. [Trực tuyến]. Địa chỉ: <https://stackoverflow.com/>.

[8] Cộng đồng GitHub Open Source. *Mã nguồn tham khảo các dự án MERN Stack E-commerce*. [Trực tuyến]. Địa chỉ: <https://github.com/>.

[9] Diễn đàn Viblo/F8. Các bài viết chia sẻ về tích hợp cổng thanh toán ZaloPay và Socket.IO. [Trực tuyến]. Địa chỉ: <https://viblo.asia/>.

[10] MongoDB Documentation. Tài liệu thiết kế Schema cho hệ thống thương mại điện tử. [Trực tuyến]. Địa chỉ: <https://www.mongodb.com/docs/manual/>.