

LỜI CAM ĐOAN

Tôi xin cam đoan rằng toàn bộ nội dung của báo cáo này là kết quả của quá trình học tập, nghiên cứu và thực hiện đồ án một cách nghiêm túc dưới sự hướng dẫn của giảng viên Võ Thành C. Các tài liệu, dữ liệu và hình ảnh sử dụng trong báo cáo đều được trích dẫn rõ ràng nguồn gốc.

Tôi hoàn toàn chịu trách nhiệm về tính trung thực và nguyên bản của báo cáo. Nếu phát hiện có sự sao chép, gian lận hay vi phạm bản quyền, tôi xin hoàn toàn chịu trách nhiệm trước hội đồng và quy định của Nhà trường.

Vĩnh Long, ngày ... tháng ... năm 2025

Sinh viên thực hiện

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Trước tiên, em xin gửi lời cảm ơn chân thành và sâu sắc đến Thầy Võ Thành C – giảng viên hướng dẫn đã tận tình chỉ bảo, định hướng và hỗ trợ em trong suốt quá trình thực hiện đề tài tốt nghiệp. Những ý kiến đóng góp quý báu của thầy không chỉ giúp em hoàn thiện đề án mà còn giúp em nâng cao kiến thức và kỹ năng thực tế trong lĩnh vực lập trình web.

Em cũng xin chân thành cảm ơn quý thầy cô khoa Công nghệ Thông tin, Trường Đại học Trà Vinh, đã truyền đạt cho em những kiến thức nền tảng và chuyên sâu trong suốt quá trình học tập tại trường. Đây chính là hành trang quý báu giúp em tự tin triển khai và hoàn thiện đề tài của mình.

Bên cạnh đó, em xin gửi lời cảm ơn đến gia đình và bạn bè đã luôn động viên, hỗ trợ tinh thần cũng như góp ý trong quá trình thực hiện đề án.

Mặc dù đã cố gắng hết sức, nhưng do thời gian thực hiện có hạn và trình độ còn nhiều hạn chế, bài báo cáo không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý của quý thầy cô để em có thể hoàn thiện hơn trong tương lai.

Em xin chân thành cảm ơn!

MỤC LỤC

i.	LỜI CAM ĐOAN	i
ii.	LỜI CẢM ƠN.....	ii
iii.	MỤC LỤC	iii
iv.	LỜI MỞ ĐẦU	v
v.	DANH MỤC HÌNH ẢNH.....	x
vi.	DANH MỤC BẢNG BIỂU.....	xi
vii.	KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT.....	xii
	CHƯƠNG 1: TỔNG QUAN	1
1.1.	Tổng quan vấn đề nghiên cứu	1
1.2.	Hướng tiếp cận giải quyết vấn đề	1
1.3.	Nghiệp vụ.....	1
	CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT	3
2.1.	Công nghệ sử dụng	3
2.3.1.	Giới thiệu về Node.js	3
2.3.2.	Giới thiệu về React	4
2.2.	Các công cụ được sử dụng	6
	CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU.....	11
3.1.	Phân tích và đặc tả yêu cầu hệ thống	11
3.2.	Thiết kế mô hình dữ liệu.....	13
3.2.1.	Mô hình thực thể kết hợp.....	13
3.2.2.	Mô hình vật lý.....	14
3.2.3.	Sơ đồ use case	16
3.3.	Mô tả các bảng trong cơ sở dữ liệu.....	16
3.4.	Hướng dẫn cài đặt và chạy dự án.....	24
3.4.1.	Cài đặt môi trường	24
3.4.2.	Cấu hình cơ sở dữ liệu	24
3.4.3.	Cài đặt và chạy Backend (NodeJS Express).....	25
3.4.4.	Cài đặt và chạy Frontend (ReactJS).....	25
	CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU	27
4.1.	Các chức năng chính của người dùng	27
4.2.	Các chức năng chính của admin	30
	CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	36
5.1.	Kết luận	36

5.2. Hạn chế	36
5.3. Hướng phát triển	36
DANH MỤC TÀI LIỆU THAM KHẢO	38

LỜI MỞ ĐẦU

Lý do chọn đề tài

Trong thời đại công nghệ số hiện nay, nhu cầu mua sắm trực tuyến ngày càng phát triển mạnh mẽ, đặc biệt là trong lĩnh vực thiết bị công nghệ như laptop, điện thoại, gaming gear và các phụ kiện liên quan. Người tiêu dùng ngày càng ưa chuộng việc tìm kiếm và đặt mua sản phẩm qua các nền tảng thương mại điện tử do tính tiện lợi, nhanh chóng và khả năng so sánh giá cả. Trước thực trạng đó, việc xây dựng một website bán đồ công nghệ không chỉ mang tính ứng dụng thực tiễn cao mà còn giúp sinh viên tiếp cận và rèn luyện kỹ năng phát triển các hệ thống web hiện đại. Chính vì vậy, em đã chọn đề tài “Xây dựng website bán đồ công nghệ” làm đề án tốt nghiệp.

Mục đích nghiên cứu

Mục đích của đề tài là xây dựng một website thương mại điện tử chuyên bán các sản phẩm công nghệ, tích hợp đầy đủ các chức năng cần thiết như: xem và tìm kiếm sản phẩm, đặt hàng, quản lý giỏ hàng, thanh toán và quản trị hệ thống. Qua đó, đề tài giúp nâng cao kiến thức và kỹ năng lập trình web, ứng dụng các công nghệ phổ biến hiện nay như NodeJS và ReactJS vào thực tiễn.

Phương pháp nghiên cứu

Đề tài được thực hiện dựa trên phương pháp nghiên cứu tài liệu và phương pháp thực nghiệm. Sinh viên tìm hiểu, tham khảo các tài liệu chuyên ngành, các mô hình website thương mại điện tử phổ biến, đồng thời tiến hành phân tích, thiết kế và xây dựng hệ thống trên môi trường thực tế bằng cách lập trình và triển khai website.

Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài là hệ thống website bán hàng trực tuyến, tập trung vào các sản phẩm công nghệ. Bên cạnh đó, hệ thống cũng hướng tới người dùng cuối như khách hàng mua sắm và người quản trị hệ thống.

Phạm vi nghiên cứu

Đề tài tập trung xây dựng một website bán hàng đơn lẻ với quy mô vừa và nhỏ, chủ yếu phục vụ mục đích học tập và trình diễn. Các chức năng chính bao gồm: hiển thị và tìm kiếm sản phẩm, giỏ hàng, đặt hàng, thanh toán đơn giản và trang quản trị cơ bản. Website không đi sâu vào các chức năng nâng cao như phân tích hành vi khách hàng, tích hợp thanh toán quốc tế hay hệ thống vận chuyển phức tạp.

4. Điểm mới đề tài:

.....

.....

.....

.....

.....

5. Giá trị thực trên đề tài:

.....

.....

.....

.....

.....

.....

.....

7. Đề nghị sửa chữa bổ sung:

.....

.....

.....

.....

.....

.....

.....

8. Đánh giá:

.....

.....

.....

Trà Vinh, ngày tháng năm 20...

Giảng viên hướng dẫn

(Ký & ghi rõ họ tên)

BẢN NHẬN XÉT ĐỒ ÁN, KHÓA LUẬN TỐT NGHIỆP
(Của cán bộ chấm đồ án, khóa luận)

Họ và tên người nhận xét:

Chức danh: Học vị:

Chuyên ngành:

Cơ quan công tác:

Tên sinh viên:

Tên đề tài đồ án, khóa luận tốt nghiệp:

.....
.....

I. Ý KIẾN NHẬN XÉT

1. Nội dung:

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

2. Điểm mới các kết quả của đồ án, khóa luận:

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

3. Ứng dụng thực tế:

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

.....

.....

II. CÁC VẤN ĐỀ CẦN LÀM RÕ
(Các câu hỏi của giáo viên phản biện)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

III. KẾT LUẬN
(Ghi rõ đồng ý hay không đồng ý cho bảo vệ đề án khóa luận tốt nghiệp)

.....

.....

.....

.....

....., ngày tháng năm 20...

Người nhận xét
(Ký & ghi rõ họ tên)

DANH MỤC HÌNH ẢNH

Hình 3. 1 ERD (sơ đồ thực thể liên kết).....	13
Hình 3. 2 Mô hình vật lý	14
Hình 3. 3 Sơ đồ use case Khách hàng	16
Hình 3. 4 Sơ đồ use case Admin.....	16
Hình 4. 1 Giao diện trang chủ (slider).....	27
Hình 4. 2 Giao diện trang chủ Product Carousel.....	27
Hình 4. 3 Giao diện footer.....	28
Hình 4. 4 Giao diện đăng nhập.....	29
Hình 4. 5 Giao diện đăng nhập qua google	29
Hình 4. 6 Giao diện đăng ký.....	30
Hình 4. 7 Giao diện thống kê.....	31
Hình 4. 8 Giao diện thống kê Heatmap	31
Hình 4. 9 Giao diện thống kê Biểu đồ tròn.....	32
Hình 4. 10 Giao diện quản lý người dùng	32
Hình 4. 11 Giao diện quản lý sản phẩm	33
Hình 4. 12 Giao diện quản lý loại sản phẩm	33
Hình 4. 13 Giao diện quản lý đơn hàng.....	34
Hình 4. 14 Giao diện lịch sử đơn hàng.....	35

DANH MỤC BẢNG BIỂU

Bảng 3. 1 Bảng brand	17
Bảng 3. 2 Bảng cart	17
Bảng 3. 3 Bảng cart_item	18
Bảng 3. 4 Bảng category	18
Bảng 3. 5 Bảng comment	19
Bảng 3. 6 Bảng orders	20
Bảng 3. 7 Bảng order_item.....	20
Bảng 3. 8 Bảng product.....	21
Bảng 3. 9 Bảng product_details.....	22
Bảng 3. 10 Bảng promotion.....	23
Bảng 3. 11 Bảng role	23
Bảng 3. 12 Bảng user.....	24

KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT

STT	Từ viết tắt	Ý nghĩa đầy đủ (Full Form)	Giải thích / Ghi chú
1	API	Application Programming Interface	Giao diện lập trình ứng dụng – thường dùng để giao tiếp giữa front-end và back-end
2	CRUD	Create, Read, Update, Delete	Các thao tác cơ bản với dữ liệu
3	DB	Database	Cơ sở dữ liệu
4	UI	User Interface	Giao diện người dùng
5	UX	User Experience	Trải nghiệm người dùng
6	JWT	JSON Web Token	Chuỗi mã hóa dùng cho xác thực và phân quyền
7	MVC	Model – View – Controller	Mô hình kiến trúc phần mềm
8	RESTful API	Representational State Transfer	Giao thức truyền thông thường dùng trong Web API
9	JSON	JavaScript Object Notation	Định dạng trao đổi dữ liệu
10	HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
11	CSS	Cascading Style Sheets	Ngôn ngữ định kiểu cho HTML
12	JS	JavaScript	Ngôn ngữ lập trình front-end (cũng dùng cả ở back-end với Node.js)
15	SQL	Structured Query Language	Ngôn ngữ truy vấn dữ liệu trong DB
16	XAMPP	Cross-platform, Apache, MySQL, PHP, Perl	Bộ phần mềm giả lập máy chủ web cục bộ
17	ORM	Object Relational Mapping	Công cụ ánh xạ giữa CSDL và đối tượng trong mã (VD: Sequelize, Prisma)
20	CSR	Client-Side Rendering	Kết xuất giao diện phía trình duyệt

21	SSR	Server-Side Rendering	Kết xuất giao diện phía máy chủ
23	NPM	Node Package Manager	Trình quản lý gói của Node.js
24	IDE	Integrated Development Environment	Môi trường phát triển tích hợp (VD: VS Code)
25	UX/UI	User Experience / User Interface	Giao diện và trải nghiệm người dùng

CHƯƠNG 1: TỔNG QUAN

1.1. Tổng quan vấn đề nghiên cứu

Trong thời đại công nghệ phát triển mạnh mẽ, việc kinh doanh và mua sắm qua mạng ngày càng trở nên phổ biến. Các website thương mại điện tử đã trở thành kênh phân phối quan trọng, giúp doanh nghiệp tiếp cận khách hàng một cách nhanh chóng, hiệu quả và tiết kiệm chi phí. Đặc biệt, trong lĩnh vực thiết bị công nghệ như laptop, điện thoại, gaming gear và phụ kiện, người tiêu dùng có xu hướng tìm kiếm và mua hàng trực tuyến thay vì đến cửa hàng vật lý.

Tuy nhiên, để xây dựng được một website thương mại điện tử hiệu quả, cần phải đảm bảo nhiều yếu tố như giao diện thân thiện, dễ sử dụng, tốc độ truy cập nhanh, bảo mật cao và đặc biệt là đầy đủ chức năng hỗ trợ cả người mua lẫn người quản trị. Việc áp dụng các công nghệ hiện đại như ReactJS ở frontend và NodeJS (Express) ở backend đang là xu hướng phát triển phổ biến vì khả năng tương tác tốt và hiệu năng cao.

Do đó, việc nghiên cứu và xây dựng một website bán đồ công nghệ có tính ứng dụng cao, phục vụ nhu cầu thực tiễn, đồng thời giúp sinh viên rèn luyện kỹ năng lập trình là rất cần thiết.

1.2. Hướng tiếp cận giải quyết vấn đề

Để giải quyết bài toán xây dựng một website bán đồ công nghệ, đề tài áp dụng phương pháp tiếp cận theo quy trình phát triển phần mềm cơ bản gồm các bước:

- Thu thập yêu cầu: Xác định rõ các chức năng cần thiết cho cả người dùng và người quản trị hệ thống.
- Phân tích và thiết kế hệ thống: Thiết kế cơ sở dữ liệu, sơ đồ ERD và kiến trúc tổng thể.
- Lập trình: Xây dựng giao diện người dùng bằng ReactJS, phát triển API phía server bằng NodeJS với Express.
- Kiểm thử: Thực hiện kiểm thử chức năng và giao diện để đảm bảo hệ thống hoạt động đúng và ổn định.
- Triển khai và đánh giá: Chạy thử hệ thống, đánh giá hiệu năng và trải nghiệm người dùng.

1.3. Nghiệp vụ

a) Nghiệp vụ người dùng (khách hàng):

- Đăng ký và đăng nhập tài khoản.

- Xem danh sách sản phẩm theo danh mục: laptop, điện thoại, gaming gear, phụ kiện,...

- Tìm kiếm sản phẩm theo tên hoặc từ khóa.
- Xem chi tiết sản phẩm, hình ảnh, mô tả, giá bán,...
- Thêm sản phẩm vào giỏ hàng.
- Cập nhật và xóa sản phẩm khỏi giỏ hàng.
- Đặt hàng và thanh toán.
- Theo dõi trạng thái đơn hàng đã đặt.

b) Nghiệp vụ quản trị viên:

- Đăng nhập hệ thống quản trị.
- Quản lý danh sách sản phẩm: thêm, sửa, xóa sản phẩm.
- Quản lý đơn hàng: xem danh sách đơn hàng, xác nhận trạng thái giao hàng.
- Quản lý người dùng: xem thông tin khách hàng, khóa/mở tài khoản.
- Thống kê đơn hàng, doanh thu theo ngày, tháng, năm.

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1. Công nghệ sử dụng

2.3.1. Giới thiệu về Node.js

Node.js là một môi trường runtime mã nguồn mở, chạy trên nền tảng V8 JavaScript Engine của Google. Được thiết kế để xây dựng các ứng dụng web hiệu suất cao, Node.js sử dụng kiến trúc hướng sự kiện và mô hình xử lý không đồng bộ (asynchronous), rất phù hợp với các hệ thống yêu cầu tốc độ và khả năng mở rộng cao.

Lý do sử dụng Node.js trong đề tài:

- Xử lý không đồng bộ (Asynchronous): Giúp ứng dụng xử lý nhiều yêu cầu đồng thời mà không bị chặn, rất phù hợp với các hệ thống quản lý nhiều người dùng.
- Hiệu năng cao: Nhờ kiến trúc event-driven, Node.js hoạt động hiệu quả khi xử lý khối lượng công việc lớn.
- Ecosystem mạnh mẽ: NPM (Node Package Manager) cung cấp hàng ngàn thư viện hỗ trợ phát triển, trong đó có các thư viện chính được sử dụng trong dự án.
- Phát triển dễ dàng: Node.js sử dụng JavaScript – một ngôn ngữ quen thuộc với cả frontend và backend, giúp dễ dàng phát triển toàn diện ứng dụng (full-stack).

Các thành phần Node.js sử dụng trong dự án

Framework Express.js

Express là một framework nhẹ và linh hoạt cho Node.js, cung cấp các công cụ cần thiết để xây dựng API RESTful.

Vai trò trong dự án:

- Xử lý các yêu cầu HTTP (GET, POST, PUT, DELETE) từ frontend.
- Định tuyến các chức năng của hệ thống như đăng nhập, đăng ký giờ giảng, và xem báo cáo.
- Cung cấp middleware để kiểm tra xác thực, xử lý lỗi và tối ưu hóa dữ liệu.

MySQL2

Thư viện hỗ trợ Node.js kết nối và truy vấn cơ sở dữ liệu MySQL.

Vai trò trong dự án:

- Quản lý dữ liệu liên quan đến giảng viên, phân công giảng dạy, và giờ nhiệm vụ.
- Xử lý các truy vấn cơ sở dữ liệu phức tạp, đảm bảo tính hiệu quả và chính xác.

JWT (JSON Web Token)

Sử dụng để bảo mật ứng dụng bằng cách xác thực và phân quyền người dùng.

Vai trò trong dự án:

- Mã hóa thông tin người dùng khi đăng nhập và lưu trữ token trên frontend.
- Xác minh các yêu cầu từ phía frontend trước khi truy cập các tài nguyên nhạy cảm.

CORS (Cross-Origin Resource Sharing)

Thư viện cho phép frontend (React) giao tiếp với backend (Node.js) khi chạy trên các domain khác nhau. Đảm bảo rằng các yêu cầu API từ React được backend chấp nhận, tránh lỗi chặn cross-origin.

bcrypt

Thư viện mã hóa mật khẩu người dùng để lưu trữ an toàn.

Vai trò trong dự án:

- Bảo mật thông tin người dùng bằng cách mã hóa mật khẩu trước khi lưu vào cơ sở dữ liệu.
- Hỗ trợ so khớp mật khẩu trong quá trình đăng nhập.

dotenv

Thư viện dùng để quản lý các biến môi trường (environment variables). Lưu trữ thông tin nhạy cảm như URL cơ sở dữ liệu, khóa bí mật JWT, và cổng chạy server.

nodemon

Công cụ tự động khởi động lại server khi phát hiện thay đổi trong mã nguồn.

Vai trò trong dự án: Tăng tốc quá trình phát triển bằng cách loại bỏ việc khởi động lại server thủ công.

2.3.2. Giới thiệu về React

React là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, được thiết kế để xây dựng giao diện người dùng (User Interface - UI) một cách linh hoạt và hiệu quả. Với cơ chế quản lý DOM ảo (Virtual DOM) và cách tiếp cận hướng thành phần (Component-based), React trở thành lựa chọn phổ biến trong việc phát triển ứng dụng web hiện đại.

Lý do sử dụng React trong đề tài

- Hiệu suất cao: DOM ảo giúp React chỉ cập nhật các thành phần thực sự thay đổi, tăng tốc độ phản hồi của giao diện.
- Hướng thành phần: Ứng dụng được chia nhỏ thành các thành phần (components) tái sử dụng, dễ bảo trì và mở rộng.

- Dữ liệu động: Quản lý trạng thái với React State và Context API cho phép giao diện phản hồi nhanh chóng với dữ liệu thay đổi.

- Hệ sinh thái phong phú: React dễ dàng tích hợp với các thư viện UI như Bootstrap và Material-UI (MUI), cùng với các công cụ quản lý trạng thái như Redux hoặc Zustand.

Các thành phần React được sử dụng trong dự án

Component-based Architecture (Kiến trúc hướng thành phần)

Vai trò trong dự án:

Từng tính năng của hệ thống (như form đăng nhập, danh sách phân công giảng dạy, thống kê) được phát triển dưới dạng các thành phần độc lập.

Cách tiếp cận này giúp tối ưu hóa việc tái sử dụng mã nguồn và tăng tính nhất quán trong giao diện.

React Router

Thư viện quản lý điều hướng và định tuyến trong ứng dụng React.

Vai trò trong dự án:

- Xây dựng cấu trúc ứng dụng với các trang như: đăng nhập, quản lý phân công giảng dạy, thống kê.

- Hỗ trợ chuyển đổi giữa các trang mà không cần tải lại toàn bộ ứng dụng.

State Management (Quản lý trạng thái)

- React State: Quản lý dữ liệu tạm thời cho từng component, đảm bảo sự tương tác động trong giao diện.

- Context API: Cung cấp dữ liệu dùng chung giữa các components mà không cần truyền qua nhiều cấp.

Vai trò trong dự án: Quản lý thông tin đăng nhập của người dùng, dữ liệu phân công giảng dạy và các trạng thái giao diện khác.

Bootstrap và Material-UI (MUI)

Bootstrap: Framework CSS giúp xây dựng giao diện nhanh chóng với các thành phần giao diện cơ bản.

Material-UI (MUI): Thư viện React cung cấp các component theo phong cách Material Design của Google.

Vai trò trong dự án:

- Bootstrap được dùng để tạo bố cục (layout) đơn giản và đảm bảo giao diện phản hồi (responsive).

- MUI được sử dụng để xây dựng các thành phần giao diện nâng cao, đảm bảo tính thẩm mỹ và chuyên nghiệp cho ứng dụng.

Axios

Thư viện gửi yêu cầu HTTP từ frontend đến backend.

Vai trò trong dự án:

- Kết nối React với backend (Node.js) qua các API RESTful để lấy và gửi dữ liệu.
- Xử lý các thao tác như đăng nhập, đăng ký giờ giảng, và truy vấn thống kê.

Kiến trúc frontend sử dụng React

Single Page Application (SPA): Ứng dụng không tải lại toàn bộ trang khi chuyển đổi giữa các chức năng, cải thiện tốc độ và trải nghiệm người dùng.

Quản lý trạng thái hiệu quả: Các trạng thái như thông tin người dùng, trạng thái đăng nhập, và dữ liệu phân công giảng dạy được quản lý tập trung để đảm bảo tính đồng bộ và nhất quán. Phần mềm được sử dụng

2.2. Các công cụ được sử dụng

2.4.1. Visual Studio Code

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mở, miễn phí do Microsoft phát triển. Đây là một công cụ phổ biến trong giới lập trình viên nhờ giao diện thân thiện, khả năng tùy biến cao, và hỗ trợ phong phú từ các tiện ích mở rộng (extensions).

Lý do sử dụng Visual Studio Code trong đề tài

- Gọn nhẹ nhưng mạnh mẽ: VS Code có dung lượng nhỏ gọn, khởi động nhanh nhưng vẫn cung cấp đầy đủ các tính năng cần thiết cho việc lập trình.
- Hỗ trợ đa nền tảng: Có thể chạy trên các hệ điều hành phổ biến như Windows, macOS và Linux.
- Tích hợp Git: VS Code cho phép quản lý mã nguồn dễ dàng, giúp theo dõi thay đổi và quản lý phiên bản dự án.
- Hệ sinh thái mở rộng: Cung cấp hàng ngàn tiện ích mở rộng hỗ trợ cho các ngôn ngữ lập trình và công nghệ như Node.js, React, Bootstrap, và MySQL.
- Tích hợp Debugger: Hỗ trợ kiểm tra và sửa lỗi trực tiếp trong quá trình phát triển ứng dụng, giúp tiết kiệm thời gian.

Các tính năng chính sử dụng trong dự án

Code Editor

Cung cấp trình chỉnh sửa mã với tính năng tô màu cú pháp (syntax highlighting), tự động hoàn thành mã (IntelliSense), và gợi ý lệnh.

Vai trò trong dự án:

- Viết mã nguồn backend bằng Node.js và Express.
- Phát triển giao diện frontend bằng React.

Terminal tích hợp

VS Code tích hợp sẵn Terminal giúp chạy các lệnh như khởi động server Node.js, cài đặt thư viện qua NPM, hoặc kiểm tra MySQL.

Vai trò trong dự án: Dễ dàng kiểm tra và quản lý các tác vụ lập trình mà không cần chuyển đổi giữa các công cụ khác nhau.

Debugging Tools (Công cụ gỡ lỗi)

Tích hợp debugger cho Node.js và Chrome giúp dễ dàng phát hiện và sửa lỗi trong cả backend và frontend.

Vai trò trong dự án: Kiểm tra các chức năng như xác thực, xử lý dữ liệu phân công giảng dạy và tương tác giữa frontend và backend.

2.4.2. Postman

Postman là một công cụ mạnh mẽ và phổ biến để kiểm tra và phát triển API, được sử dụng bởi các nhà phát triển backend, frontend, và cả QA. Công cụ này cung cấp giao diện trực quan để gửi yêu cầu (request) HTTP đến máy chủ và nhận phản hồi (response), giúp kiểm tra và gỡ lỗi API nhanh chóng và hiệu quả.

Lý do sử dụng Postman trong đề tài

- Giao diện thân thiện: Postman không yêu cầu kỹ năng lập trình nâng cao, giúp dễ dàng gửi các yêu cầu API và kiểm tra phản hồi.
- Hỗ trợ nhiều loại yêu cầu: Hỗ trợ các loại HTTP như GET, POST, PUT, DELETE, PATCH, HEAD, giúp kiểm tra đầy đủ các chức năng CRUD của API.
- Xử lý dữ liệu phức tạp: Postman cho phép gửi dữ liệu trong các định dạng như JSON, XML, form-data, hoặc raw text.
- Kiểm thử tự động: Hỗ trợ viết script để kiểm tra phản hồi, đảm bảo rằng API hoạt động đúng như mong đợi.
- Tích hợp quy trình làm việc: Có khả năng lưu và tổ chức các bộ sưu tập (collection) các API, chia sẻ giữa các thành viên nhóm dễ dàng.

Các tính năng chính sử dụng trong dự án

- Gửi yêu cầu tới các API của backend Node.js (Express) để kiểm tra chức năng xử lý dữ liệu phân công giảng dạy.
- Xác minh tính chính xác và hiệu suất của các API như đăng nhập, đăng ký giờ giảng, và xem thống kê.
- Xử lý và kiểm tra dữ liệu JSON: Gửi và nhận dữ liệu JSON giữa frontend (React) và backend, đảm bảo dữ liệu được truyền tải chính xác.
- Kiểm thử bảo mật: Thử nghiệm các API với dữ liệu không hợp lệ hoặc thiếu xác thực (authorization token) để đảm bảo hệ thống không bị lỗi hoặc xâm nhập.

Lưu và tái sử dụng bộ sưu tập API: Postman cho phép lưu các yêu cầu API dưới dạng bộ sưu tập, dễ dàng tổ chức và kiểm tra lại sau này. Tạo bộ sưu tập API cho các chức năng chính như đăng nhập, phân công giảng dạy, và thống kê, giúp quản lý và kiểm tra tiện lợi hơn.

Kiểm tra tự động và script: Postman cung cấp khả năng viết script bằng JavaScript để kiểm tra tự động phản hồi từ API. Viết các script kiểm tra các điều kiện như mã trạng thái HTTP (200, 400, 401), cấu trúc phản hồi JSON, và nội dung dữ liệu.

Tích hợp với môi trường (Environment): Postman hỗ trợ quản lý các biến môi trường (environment variables) như URL máy chủ, token xác thực, giúp tiết kiệm thời gian khi làm việc với các môi trường khác nhau (development, staging, production).

2.4.3. XAMPP

XAMPP là một phần mềm mã nguồn mở cung cấp một bộ công cụ gồm các thành phần cần thiết để chạy các ứng dụng web động, bao gồm Apache HTTP Server, MySQL (hoặc MariaDB), PHP và Perl. XAMPP thường được sử dụng trong các môi trường phát triển để dễ dàng cài đặt và chạy các ứng dụng web mà không cần cấu hình phức tạp.

Lý do sử dụng XAMPP trong đề tài

- Dễ dàng cài đặt: XAMPP cung cấp một gói cài đặt đơn giản, cho phép người dùng nhanh chóng thiết lập môi trường phát triển với các công cụ như Apache, MySQL, và PHP.
- Quản lý cơ sở dữ liệu MySQL: XAMPP đi kèm với phpMyAdmin, một công cụ web để quản lý cơ sở dữ liệu MySQL dễ dàng qua giao diện người dùng. Điều này rất hữu ích trong việc tạo, quản lý và kiểm tra dữ liệu trong cơ sở dữ liệu.

- Môi trường phát triển tích hợp: Với các thành phần như Apache (Web Server) và MySQL (Cơ sở dữ liệu), XAMPP cung cấp một môi trường phát triển hoàn chỉnh, thuận tiện cho việc kiểm thử và phát triển ứng dụng web.

- Được sử dụng rộng rãi: XAMPP là công cụ phổ biến trong cộng đồng phát triển, dễ dàng tìm kiếm tài liệu hỗ trợ và giải pháp khi gặp vấn đề.

Các tính năng chính sử dụng trong dự án

Apache Web Server

- XAMPP cung cấp một máy chủ Apache tích hợp sẵn giúp dễ dàng kiểm tra ứng dụng web mà không cần cài đặt thêm phần mềm khác.

- Mặc dù dự án sử dụng Node.js và React, Apache vẫn có thể được sử dụng trong môi trường phát triển để chạy các ứng dụng phụ hoặc thử nghiệm API.

MySQL Database

- XAMPP cung cấp MySQL (hoặc MariaDB) để quản lý cơ sở dữ liệu cho ứng dụng. Mặc dù trong dự án này sử dụng MySQL2 với Node.js, XAMPP vẫn có thể hữu ích cho việc thiết lập cơ sở dữ liệu và kiểm tra ban đầu.

- Có thể sử dụng phpMyAdmin để quản lý và truy vấn cơ sở dữ liệu một cách dễ dàng trong quá trình phát triển.

phpMyAdmin

- Là công cụ quản lý cơ sở dữ liệu MySQL qua giao diện web. Giúp dễ dàng tạo bảng, xem dữ liệu, và kiểm tra các truy vấn SQL.

- Cung cấp giao diện đồ họa để sử dụng thay vì phải nhập lệnh SQL thủ công.

Hỗ trợ các ngôn ngữ lập trình: XAMPP hỗ trợ PHP và Perl, nhưng trong dự án này, các thành phần chính như Apache và MySQL vẫn có thể hỗ trợ kiểm thử các API hoặc ứng dụng có liên quan đến PHP hoặc các ứng dụng khác trong quá trình phát triển.

Quản lý đa môi trường: XAMPP hỗ trợ nhiều dự án cùng lúc, giúp người phát triển có thể chạy và thử nghiệm nhiều ứng dụng khác nhau trên cùng một máy tính mà không cần phải cấu hình quá nhiều môi trường.

2.4.4. Git/GitHub

Giới thiệu về Git/GitHub

Git là một hệ thống quản lý phiên bản phân tán (distributed version control system) phổ biến, được phát triển bởi Linus Torvalds vào năm 2005. Git cho phép các lập trình viên theo dõi sự thay đổi của mã nguồn qua từng phiên bản, giúp làm việc nhóm hiệu

quả, đồng thời hỗ trợ phục hồi lại các phiên bản trước của mã nguồn nếu cần thiết. Git có thể hoạt động cả trong môi trường trực tuyến và ngoại tuyến, cho phép các lập trình viên làm việc độc lập và sau đó hợp nhất công việc của mình với nhóm.

GitHub là một dịch vụ lưu trữ mã nguồn dựa trên Git, cung cấp không gian lưu trữ trực tuyến cho các dự án Git. GitHub hỗ trợ các tính năng như quản lý kho mã nguồn, theo dõi sự thay đổi của mã, và làm việc nhóm. Ngoài ra, GitHub cũng hỗ trợ các tính năng như Pull Request, Issues, Wiki, giúp các nhóm phát triển cộng tác dễ dàng hơn.

Lý do sử dụng Git/GitHub trong đề tài

Quản lý mã nguồn hiệu quả: Git giúp theo dõi mọi thay đổi trong mã nguồn, cho phép lập trình viên quản lý các phiên bản của dự án, đảm bảo rằng không mất mát dữ liệu trong quá trình phát triển. GitHub cung cấp không gian để lưu trữ mã nguồn và làm việc cộng tác.

Làm việc nhóm và cộng tác: GitHub hỗ trợ tính năng "Branch" và "Merge" giúp các lập trình viên có thể làm việc trên các nhánh riêng biệt mà không ảnh hưởng đến mã chính. Sau đó, những thay đổi có thể được hợp nhất lại mà không gây ra xung đột. Đây là một tính năng cực kỳ quan trọng khi làm việc nhóm.

Lưu trữ và chia sẻ mã nguồn: GitHub cho phép lưu trữ mã nguồn công khai hoặc riêng tư. Điều này rất hữu ích khi cần chia sẻ dự án với các thành viên trong nhóm, hoặc cung cấp mã nguồn cho cộng đồng.

Theo dõi và quản lý các vấn đề (Issues): GitHub cung cấp tính năng quản lý "Issues" giúp theo dõi các lỗi, tính năng cần triển khai hoặc các công việc phải thực hiện. Điều này giúp tăng cường sự tổ chức trong dự án và đảm bảo rằng các nhiệm vụ quan trọng không bị bỏ sót.

Tích hợp CI/CD (Continuous Integration/Continuous Deployment): GitHub hỗ trợ tích hợp với các công cụ CI/CD giúp tự động kiểm tra và triển khai mã nguồn sau mỗi lần commit. Điều này giúp tăng tốc quá trình phát triển và đảm bảo chất lượng phần mềm

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

3.1. Phân tích và đặc tả yêu cầu hệ thống

3.1.1. Yêu cầu chức năng

a) Chức năng người dùng (khách hàng)

- Đăng ký tài khoản: Người dùng cung cấp thông tin cá nhân (họ tên, email, mật khẩu, số điện thoại, địa chỉ) để đăng ký.
- Đăng nhập/Đăng xuất: Người dùng đăng nhập bằng email và mật khẩu.
- Quản lý thông tin cá nhân: Cho phép người dùng cập nhật thông tin cá nhân và mật khẩu.
- Xem sản phẩm:
 - Xem danh sách sản phẩm theo danh mục (laptop, điện thoại, gaming gear, phụ kiện...).
 - Xem chi tiết sản phẩm (hình ảnh, mô tả, giá).
- Tìm kiếm và lọc sản phẩm: Người dùng có thể tìm theo tên, từ khóa hoặc lọc theo danh mục/giá.
- Quản lý giỏ hàng:
 - Thêm sản phẩm vào giỏ hàng.
 - Cập nhật số lượng hoặc xóa sản phẩm trong giỏ hàng.
- Đặt hàng và thanh toán: Người dùng xác nhận đơn hàng, lựa chọn phương thức thanh toán.
- Theo dõi trạng thái đơn hàng: Xem lịch sử đơn hàng, tình trạng giao hàng.

b) Chức năng quản trị viên (admin)

- Đăng nhập quản trị: Quản trị viên truy cập bằng tài khoản admin.
- Quản lý sản phẩm: Thêm mới, chỉnh sửa, xóa sản phẩm và danh mục.
- Quản lý đơn hàng:
 - Xem danh sách đơn hàng.
 - Xác nhận/trạng thái giao hàng.
- Quản lý người dùng:
 - Xem thông tin khách hàng.
 - Khóa/mở tài khoản người dùng.

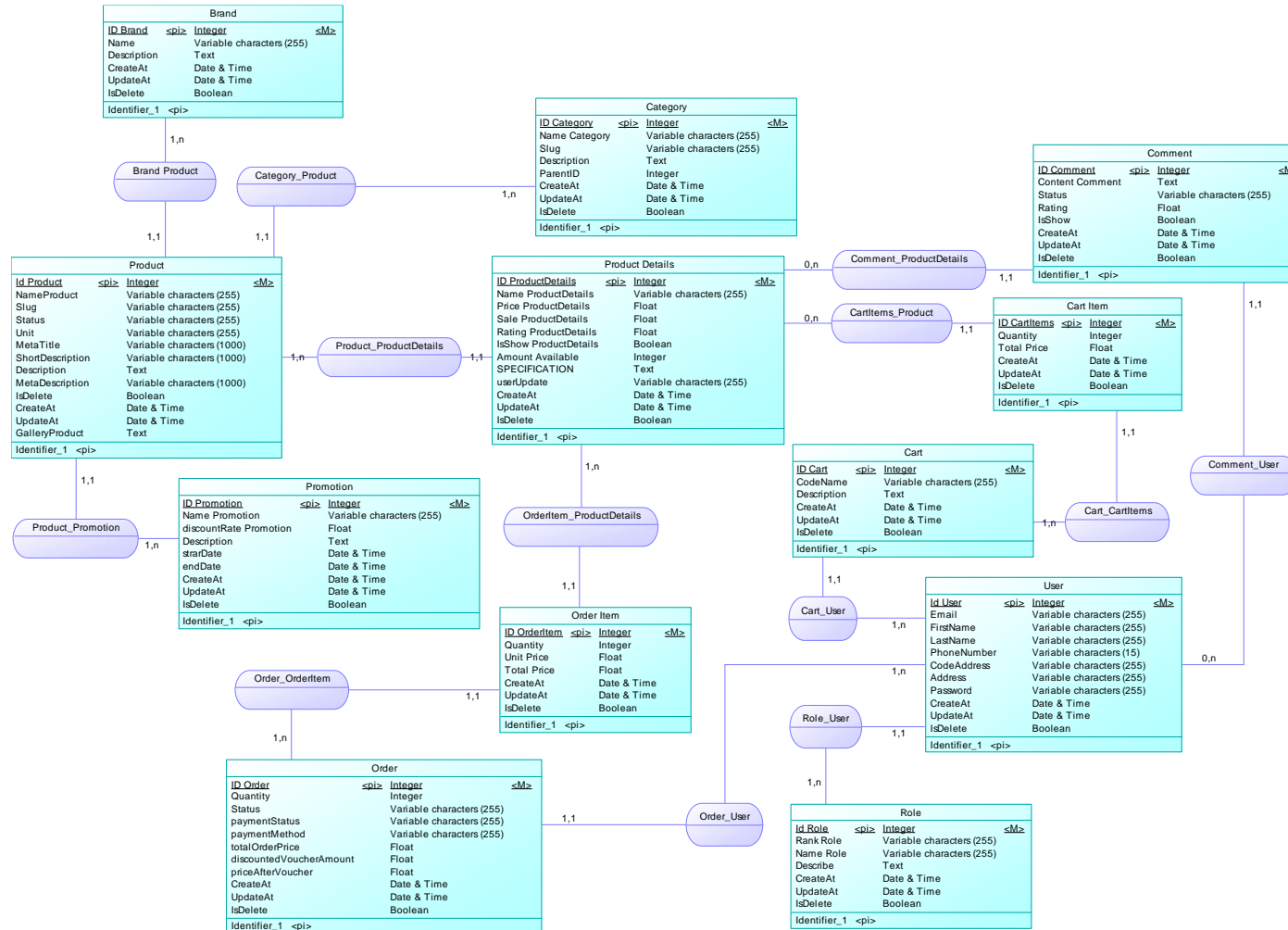
- Xem báo cáo doanh thu, số lượng đơn hàng theo ngày/tháng/năm.

3.1.2. Yêu cầu phi chức năng

- Hệ thống đáp ứng nhanh ($<3s$) khi tải sản phẩm và giỏ hàng.
- Bảo mật:
 - Mật khẩu được mã hóa.
 - Phân quyền rõ ràng giữa người dùng và admin.
- Cho phép bổ sung thêm danh mục sản phẩm, phương thức thanh toán trong tương lai.
- Giao diện thân thiện, dễ sử dụng, hỗ trợ responsive trên thiết bị di động.
- Hỗ trợ triển khai trên server web phổ biến, dễ dàng bảo trì.

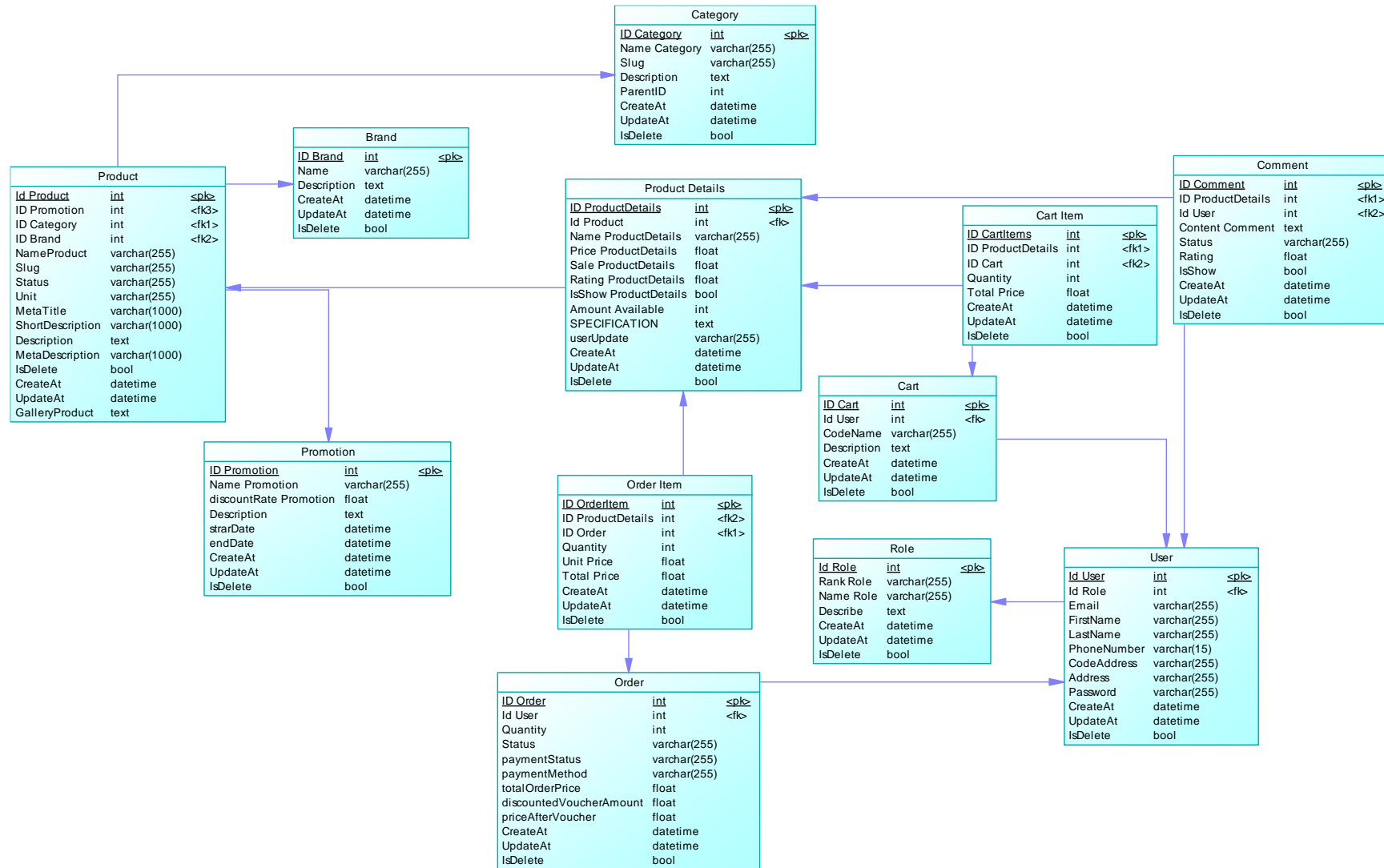
3.2. Thiết kế mô hình dữ liệu

3.2.1. Mô hình thực thể kết hợp



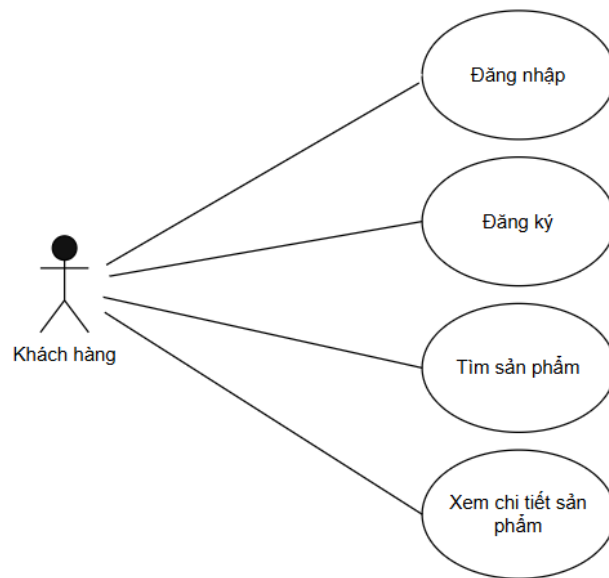
Hình 3. 1 ERD (sơ đồ thực thể liên kết)

3.2.2. Mô hình vật lý

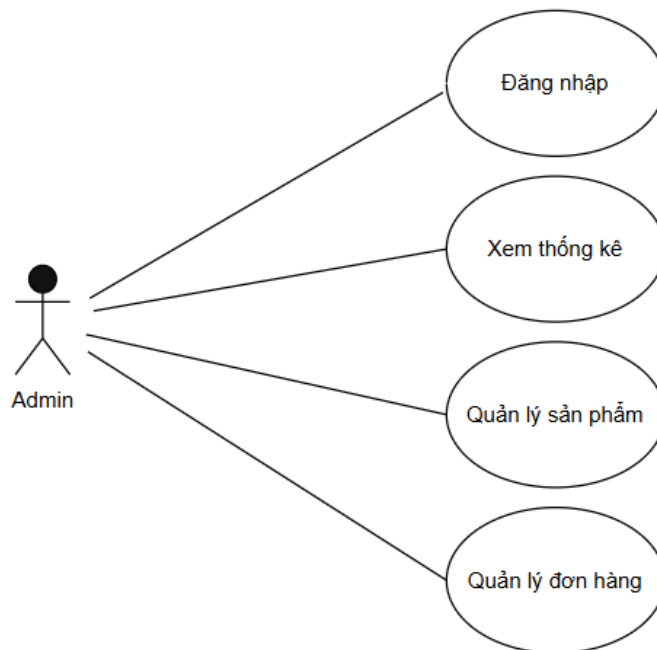


Hình 3. 2 Mô hình vật lý

3.2.3. Sơ đồ use case



Hình 3. 3 Sơ đồ use case Khách hàng



Hình 3. 4 Sơ đồ use case Admin

3.3. Mô tả các bảng trong cơ sở dữ liệu

Bảng brand

Trường	Kiểu dữ liệu	Ý nghĩa
--------	--------------	---------

ID_BRAND	INT	Mã định danh duy nhất cho từng thương hiệu (primary key).
NAME	VARCHAR	Tên thương hiệu, ví dụ: "Apple", "Samsung".
DESCRIPTION	TEXT	Mô tả chi tiết về thương hiệu (có thể là giới thiệu, lịch sử, v.v.).
CREATEAT	DATETIME	Ngày giờ thương hiệu được thêm vào hệ thống.
UPDATEAT	DATETIME	Ngày giờ thương hiệu được cập nhật gần nhất.
ISDELETE	BOOLEAN, TINYINT(1)	Trạng thái xóa mềm

Bảng 3. 1 Bảng brand

Bảng Cart

Trường	Kiểu dữ liệu	Ý nghĩa
ID_CART	int(11)	Mã định danh duy nhất cho giỏ hàng
ID_USER	int(11)	Mã người dùng (foreign key liên kết đến bảng users), xác định ai sở hữu giỏ hàng này.
CODENAME	varchar(255)	Mã hoặc tên định danh cho giỏ hàng, có thể dùng để phân biệt hoặc tạo nhóm nếu cần.
DESCRIPTION	text	Mô tả chi tiết về giỏ hàng (dùng cho admin hoặc tracking).
CREATEAT	datetime	Thời điểm tạo giỏ hàng.
UPDATEAT	datetime	Thời điểm cập nhật giỏ hàng gần nhất.
ISDELETE	tinyint(1)	Trạng thái xóa mềm

Bảng 3. 2 Bảng cart

Bảng cart_item

Trường	Kiểu dữ liệu	Ý nghĩa
ID_CARTITEMS	int(11)	Mã định danh duy nhất của dòng sản phẩm trong giỏ hàng.
ID_PRODUCTDETAILS	int(11)	Mã sản phẩm chi tiết (foreign key tới bảng product_details).
ID_CART	int(11)	Mã giỏ hàng chứa dòng sản phẩm này (foreign key tới bảng cart).
QUANTITY	int(11)	Số lượng sản phẩm người dùng đã thêm vào giỏ.
TOTAL_PRICE	float	Tổng tiền cho dòng sản phẩm này = đơn giá \times số lượng.
CREATEAT	datetime	Ngày tạo bản ghi.
UPDATEAT	datetime	Ngày cập nhật gần nhất của bản ghi.
ISDELETE	tinyint(1)	Trạng thái xóa mềm

*Bảng 3. 3 Bảng cart_item***Bảng category**

Trường	Kiểu dữ liệu	Ý nghĩa
ID_CATEGORY	int(11)	Mã định danh duy nhất cho mỗi danh mục.
NAME_CATEGORY	varchar(255)	Tên danh mục
SLUG	varchar(255)	Đường dẫn thân thiện dùng trong URL
DESCRIPTION	text	Mô tả chi tiết về danh mục (có thể để trống).
PARENTID	int(11)	ID của danh mục cha (dùng để tạo danh mục con – phân cấp cây).
CREATEAT	datetime	Ngày tạo danh mục.
UPDATEAT	datetime	Ngày cập nhật gần nhất.
ISDELETE	tinyint(1)	Đánh dấu xóa mềm

Bảng 3. 4 Bảng category

Bảng comment

Trường	Kiểu dữ liệu	Ý nghĩa
ID_COMMENT	int(11)	Mã định danh duy nhất cho bình luận
ID_PRODUCTDETAILS	int(11)	Mã sản phẩm được bình luận (liên kết tới bảng product_details).
ID_USER	int(11)	Mã người dùng đã bình luận (liên kết tới bảng users).
CONTENT_COMMENT	text	Nội dung bình luận.
STATUS	varchar(255)	Trạng thái xử lý bình luận (ví dụ: pending, approved, rejected).
RATING	float	Điểm đánh giá sản phẩm (từ 1.0 đến 5.0).
ISSHOW	tinyint(1)	Cờ hiển thị bình luận
CREATEAT	datetime	Thời gian tạo bình luận.
UPDATEAT	datetime	Thời gian chỉnh sửa bình luận gần nhất.
ISDELETE	tinyint(1)	Cờ xóa mềm

*Bảng 3. 5 Bảng comment***Bảng orders**

Trường	Kiểu dữ liệu	Ý nghĩa
ID_ORDER	int(11)	Mã đơn hàng
ID_USER	int(11)	Mã người dùng đã tạo đơn hàng (liên kết đến bảng users).
QUANTITY	int(11)	Tổng số lượng sản phẩm trong đơn hàng.
STATUS	varchar(255)	Trạng thái đơn hàng
PAYMENTSTATUS	varchar(255)	Trạng thái thanh toán: paid, unpaid, refunded, v.v.

PAYMENTMETHOD	varchar(255)	Phương thức thanh toán: COD, Credit Card, Momo, ZaloPay,...
TOTALORDERPRICE	float	Tổng giá trị đơn hàng trước khi áp dụng giảm giá/voucher.
DISCOUNTEDVOUCHERAMOUNT	float	Số tiền giảm giá do áp dụng voucher (nếu có).
PRICEAFTERVOUCHER	float	Số tiền khách phải thanh toán sau khi trừ giảm giá.
CREATEAT	datetime	Thời gian tạo đơn hàng.
UPDATEAT	datetime	Thời gian cập nhật đơn hàng lần cuối.
ISDELETE	tinyint(1)	Cờ xóa mềm

Bảng 3. 6 Bảng orders

Bảng order_item

Trường	Kiểu dữ liệu	Ý nghĩa
ID_ORDERITEM	int(11)	Mã mục sản phẩm trong đơn hàng.
ID_PRODUCTDETAILS	int(11)	Mã sản phẩm chi tiết (liên kết đến bảng product_details).
ID_ORDER	int(11)	Mã đơn hàng mà sản phẩm này thuộc về (liên kết đến bảng orders).
QUANTITY	int(11)	Số lượng sản phẩm đã mua trong đơn hàng.
UNIT_PRICE	float	Giá của một đơn vị sản phẩm tại thời điểm đặt hàng.
TOTAL_PRICE	float	Thành tiền của mục này = QUANTITY * UNIT_PRICE.
CREATEAT	datetime	Thời gian tạo mục này.
UPDATEAT	datetime	Thời gian cập nhật mục này lần cuối.
ISDELETE	tinyint(1)	Cờ đánh dấu xóa mềm

Bảng 3. 7 Bảng order_item

Bảng product

Trường	Kiểu dữ liệu	Ý nghĩa
ID_PRODUCT	int(11)	Mã sản phẩm
ID_PROMOTION	int(11)	Mã khuyến mãi liên kết (FK tới bảng promotion, nếu có).
ID_CATEGORY	int(11)	Mã danh mục sản phẩm (FK tới bảng category).
ID_BRAND	int(11)	Mã thương hiệu sản phẩm (FK tới bảng brand).
NAMEPRODUCT	varchar(255)	Tên sản phẩm.
SLUG	varchar(255)	URL-friendly version của tên sản phẩm (ví dụ: dien-thoai-iphone-15).
STATUS	varchar(255)	Trạng thái (ví dụ: "Còn hàng", "Hết hàng").
UNIT	varchar(255)	Đơn vị tính (ví dụ: "chiếc", "bộ", "cái").
METATITLE	varchar(1000)	Tiêu đề cho SEO (hiển thị trên Google).
SHORTDESCRIPTION	varchar(1000)	Mô tả ngắn, hiển thị ngoài danh sách sản phẩm.
DESCRIPTION	text	Mô tả chi tiết sản phẩm.
METADESCRIPTION	varchar(1000)	Mô tả SEO (hiển thị khi chia sẻ link hoặc trong meta tag).
ISDELETE	tinyint(1)	Đánh dấu xóa mềm (1 là đã xóa, 0 hoặc NULL là còn tồn tại).
CREATEAT	datetime	Thời gian tạo sản phẩm.
UPDATEAT	datetime	Thời gian cập nhật gần nhất.
GALLERYPRODUCT	text	Danh sách ảnh (có thể là JSON hoặc chuỗi phân tách bởi ; chứa tên file ảnh).

Bảng 3. 8 Bảng product

Bảng product_details

Trường	Kiểu dữ liệu	Ý nghĩa
ID_PRODUCTDETAILS	int(11)	Mã chi tiết sản phẩm
ID_PRODUCT	int(11)	FK liên kết với bảng product (mỗi product có thể có nhiều product_details).
NAME_PRODUCTDETAILS	varchar(255)	Tên phiên bản sản phẩm (ví dụ: "Laptop Asus - i7 RAM 16GB SSD 512GB").
PRICE_PRODUCTDETAILS	float	Giá bán hiện tại.
SALE_PRODUCTDETAILS	float	Mức giảm giá (phần trăm hoặc số tiền, tùy quy ước hệ thống).
RATING_PRODUCTDETAILS	float	Điểm đánh giá trung bình của khách hàng.
ISSHOW_PRODUCTDETAILS	tinyint(1)	Có hiển thị ra ngoài không? (1: có, 0: ẩn).
AMOUNT_AVAILABLE	int(11)	Số lượng tồn kho.
SPECIFICATION	text	Mô tả kỹ thuật chi tiết (RAM, CPU, màn hình...).
Import_Price	float	Giá nhập của sản phẩm (nội bộ quản lý lợi nhuận).
GALLERYPRODUCT_DETAILS	text	Ảnh đại diện hoặc nhiều ảnh, có thể dạng chuỗi hoặc JSON.
USERUPDATE	varchar(255)	Tên người cập nhật cuối cùng (để kiểm soát thay đổi).
CREATEAT	datetime	Thời gian tạo bản ghi.
UPDATEAT	datetime	Thời gian cập nhật gần nhất.
ISDELETE	tinyint(1)	Xóa mềm

Bảng 3. 9 Bảng product_details

Bảng promotion

Trường	Kiểu dữ liệu	Ý nghĩa
ID_PROMOTION	int(11)	Mã chương trình khuyến mãi
NAME_PROMOTION	varchar(255)	Tên chương trình khuyến mãi (VD: "Tết Sale 2025", "Black Friday").
DISCOUNTRATE_PROMOTION	float	Mức giảm giá (VD: 0.2 nghĩa là giảm 20%).
DESCRIPTION	text	Mô tả chi tiết chương trình (điều kiện áp dụng, sản phẩm áp dụng,...).
STRARDATE	datetime	Ngày bắt đầu chương trình khuyến mãi.
ENDDATE	datetime	Ngày kết thúc chương trình khuyến mãi.
CREATEAT	datetime	Ngày tạo bản ghi.
UPDATEAT	datetime	Ngày cập nhật gần nhất.
ISDELETE	tinyint(1)	Trạng thái xóa mềm

*Bảng 3. 10 Bảng promotion***Bảng role**

Trường	Kiểu dữ liệu	Ý nghĩa
ID_ROLE	int(11)	Mã vai trò
RANK_ROLE	varchar(255)	Cấp độ quyền
NAME_ROLE	varchar(255)	Tên vai trò rõ ràng
DESCRIBES	text	Mô tả chi tiết quyền hạn, chức năng, vai trò.
CREATEAT	datetime	Ngày tạo vai trò.
UPDATEAT	datetime	Ngày cập nhật gần nhất.
ISDELETE	tinyint(1)	Trạng thái xóa mềm

Bảng 3. 11 Bảng role

Bảng user

Trường	Kiểu dữ liệu	Ý nghĩa
ID_USER	int(11)	Khóa chính, định danh duy nhất của người dùng
ID_ROLE	int(11)	Khóa ngoại đến bảng role, xác định quyền của người dùng.
EMAIL	varchar(255)	Email đăng nhập của người dùng
FIRSTNAME	varchar(255)	Họ người dùng.
LASTNAME	varchar(255)	Tên người dùng.
PHONENUMBER	varchar(15)	Số điện thoại người dùng.
CODEADDRESS	varchar(255)	Mã địa chỉ
ADDRESS	varchar(255)	Địa chỉ chi tiết (số nhà, tên đường...).
PASSWORD	varchar(255)	Mật khẩu đã được mã hóa (hash).
CREATEAT	datetime	Thời điểm tạo tài khoản.
UPDATEAT	datetime	Thời điểm cập nhật thông tin gần nhất.
ISDELETE	tinyint(1)	Trạng thái xóa mềm

Bảng 3. 12 Bảng user

3.4. Hướng dẫn cài đặt và chạy dự án

Để triển khai và chạy hệ thống website bán đồ công nghệ, cần tiến hành cài đặt môi trường và cấu hình dự án theo các bước sau:

3.4.1. Cài đặt môi trường

– Cài đặt Node.js (phiên bản LTS khuyến nghị $\geq 18.x$) từ trang chủ: <https://nodejs.org>.

– Cài đặt npm (trình quản lý gói đi kèm Node.js) hoặc yarn.

– Cài đặt MySQL (phiên bản ≥ 8.0) để quản lý cơ sở dữ liệu.

– Cài đặt Git để quản lý và lấy mã nguồn dự án.

– Link dự án: <https://github.com/BaoQuocZero/TN-DA21TTA-110121007->

[NguyenLamQuocBao-ShopDienTu.git](https://github.com/BaoQuocZero/TN-DA21TTA-110121007-NguyenLamQuocBao-ShopDienTu.git).

3.4.2. Cấu hình cơ sở dữ liệu

Khởi động MySQL và tạo cơ sở dữ liệu:

```
CREATE DATABASE kltn CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci;
```

Đảm bảo thông tin tài khoản MySQL trùng khớp với cấu hình trong file .env:

User: root

Password: 123456

Nhập cơ sở dữ liệu trong thư mục src\Database

3.4.3. Cài đặt và chạy Backend (NodeJS Express)

Di chuyển vào thư mục backend: src\node-app

Cài đặt các thư viện cần thiết:

```
npm install
```

Tạo file .env trong thư mục backend với nội dung:

PORT=8081

URL_REACT=http://localhost:3000

HOST_NAME=localhost

NODE_ENV=development

DB_HOST=localhost

DB_PORT=3306

DB_USER=root

DB_NAME=kltn

DB_PASSWORD=123456

SECRETKEYADMIN=kltn

EMAIL_OTP= <email gửi thư cho khách>

PASSWORD_OTP= <Mật khẩu email gửi thư>

Khởi chạy server:

```
npm run dev
```

Backend chạy tại địa chỉ: http://localhost:8081.

3.4.4. Cài đặt và chạy Frontend (ReactJS)

Di chuyển vào thư mục frontend: src\react-app

Cài đặt các thư viện cần thiết:

```
npm install
```

Tạo file .env trong thư mục frontend với nội dung:

```
REACT_APP_URL_SERVER= http://localhost:8081
```

```
REACT_APP_GOOGLE_CLIENT_ID=386034289331-  
e7cdf9e2167phuqu1u7qvo0lcsc6isqi.apps.googleusercontent.com
```

```
Client_secret=GOCSXP-aTq02Wf-w4YcdNoE-cQvO_cr_liT
```

Khởi chạy ứng dụng React:

```
npm start
```

Frontend chạy tại địa chỉ: <http://localhost:3000>.

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

4.1. Các chức năng chính của người dùng

Giao diện trang chủ

Là giao diện đầu tiên khi người dùng truy cập website.

Hiển thị danh sách sản phẩm nổi bật, khuyến mãi, danh mục sản phẩm, banner quảng cáo.

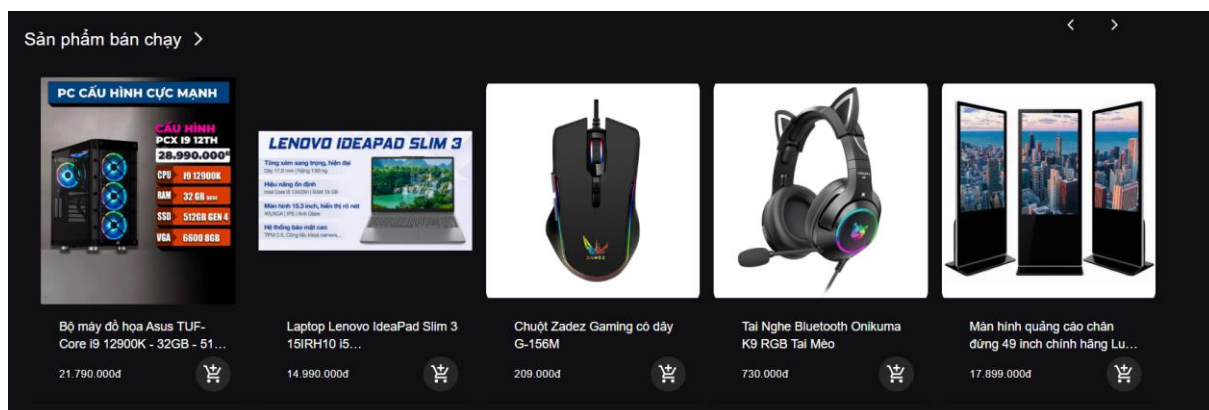
Cung cấp thanh tìm kiếm và điều hướng đến các chức năng chính như: giỏ hàng, đăng nhập, tài khoản, yêu thích.

Gây ấn tượng đầu tiên với người dùng.

Giúp người dùng nhanh chóng tiếp cận các sản phẩm và khuyến mãi quan trọng.



Hình 4. 1 Giao diện trang chủ (slider)



Hình 4. 2 Giao diện trang chủ Product Carousel

Giao diện giỏ hàng

Hiển thị danh sách sản phẩm người dùng đã chọn mua.

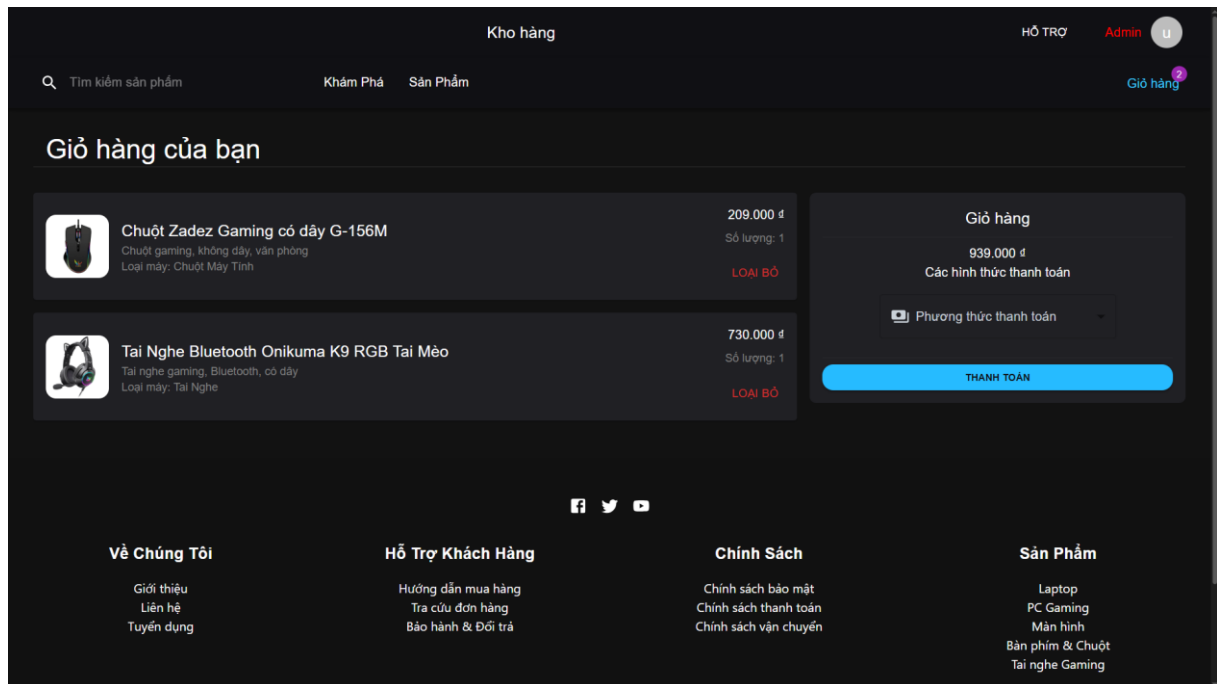
Cho phép cập nhật số lượng, xóa sản phẩm khỏi giỏ.

Tính tổng tiền hàng, vận chuyển, chiết khấu (nếu có).

Cung cấp nút Thanh toán để chuyển sang quy trình đặt hàng.

Hỗ trợ trải nghiệm mua sắm liền mạch.

Là bước trung gian quan trọng giữa xem sản phẩm và thanh toán.



Hình 4. 3 Giao diện footer

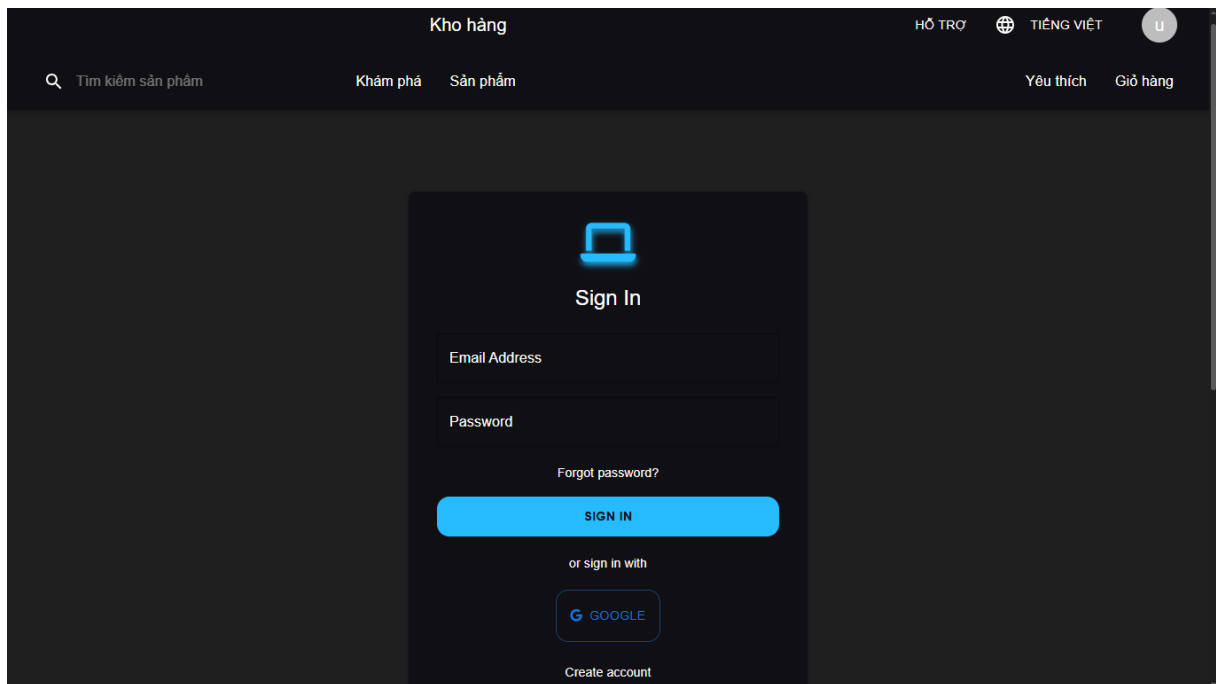
Giao diện đăng nhập

Cho phép người dùng đăng nhập vào hệ thống bằng email và mật khẩu.

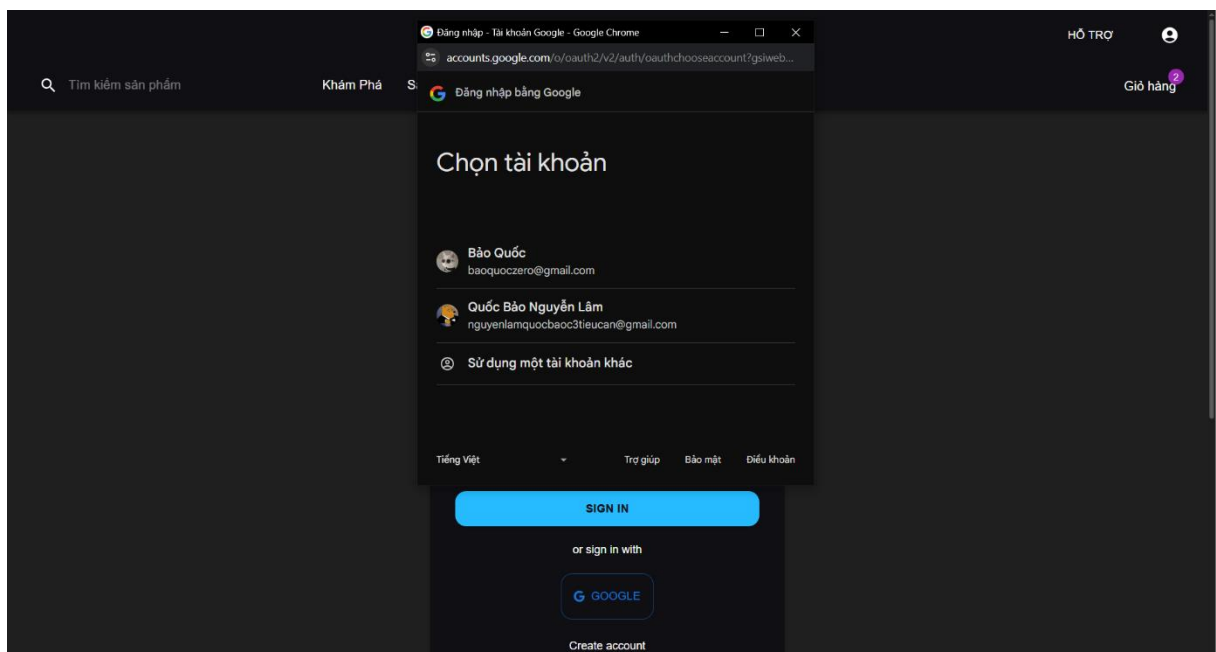
Kiểm tra xác thực thông tin và điều hướng đến trang người dùng hoặc admin (nếu là quản trị viên).

Bảo vệ tài khoản người dùng và phân quyền truy cập hệ thống.

Là bước bắt buộc để sử dụng các chức năng như mua hàng, đánh giá sản phẩm



Hình 4. 4 Giao diện đăng nhập



Hình 4. 5 Giao diện đăng nhập qua google

Giao diện đăng ký

Cho phép người dùng tạo tài khoản mới với các thông tin: tên, email, mật khẩu, số điện thoại, địa chỉ.

Kiểm tra hợp lệ dữ liệu đầu vào, xác minh mật khẩu.

Mở rộng hệ thống người dùng.

Là tiền đề để thực hiện các chức năng cá nhân hóa (giỏ hàng, đơn hàng, bình luận...).

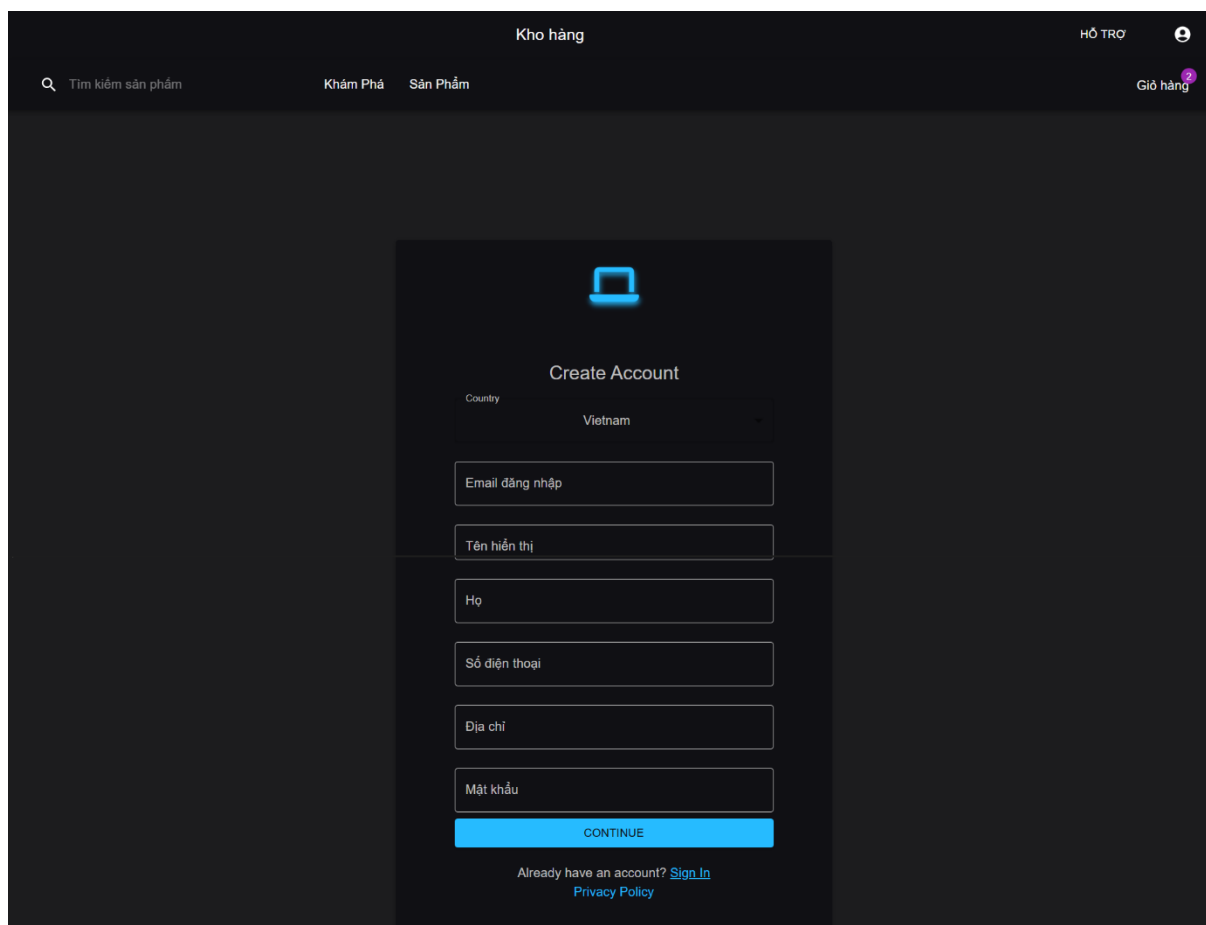
Giao diện đăng ký

Cho phép người dùng tạo tài khoản mới với các thông tin: tên, email, mật khẩu, số điện thoại, địa chỉ.

Kiểm tra hợp lệ dữ liệu đầu vào, xác minh mật khẩu.

Mở rộng hệ thống người dùng.

Là tiền đề để thực hiện các chức năng cá nhân hóa (giỏ hàng, đơn hàng, bình luận...).



Hình 4. 6 Giao diện đăng ký

4.2. Các chức năng chính của admin

Giao diện thống kê

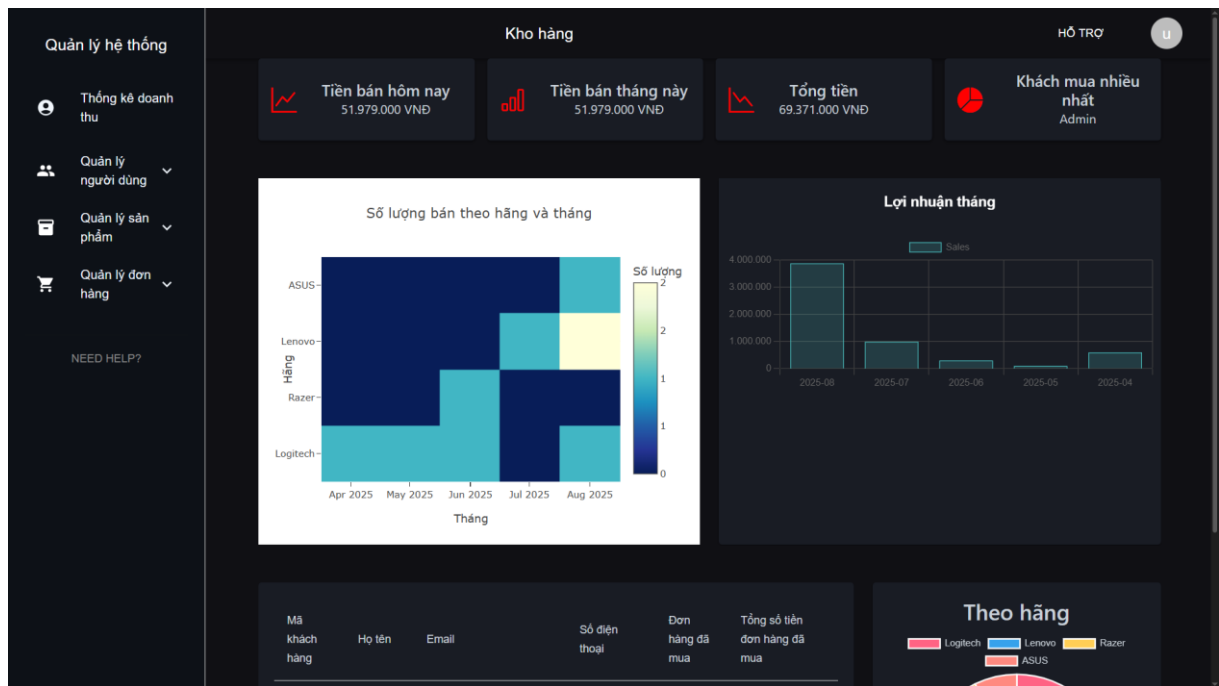
Hiển thị số lượng đơn hàng, doanh thu, số người dùng, sản phẩm đã bán,... theo ngày, tháng, năm.

Biểu đồ Heatmap (Số lượng bán theo hãng và tháng)

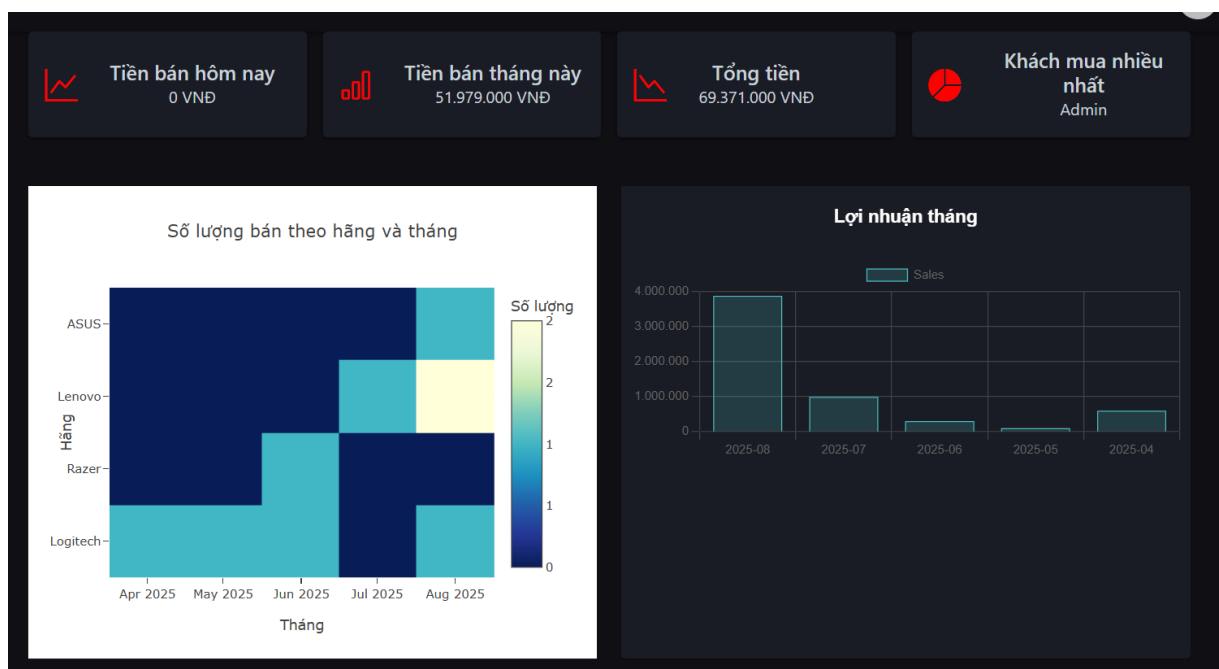
Cho phép lọc theo thời gian, loại sản phẩm, trạng thái đơn hàng.

Giúp quản trị viên theo dõi hiệu quả kinh doanh.

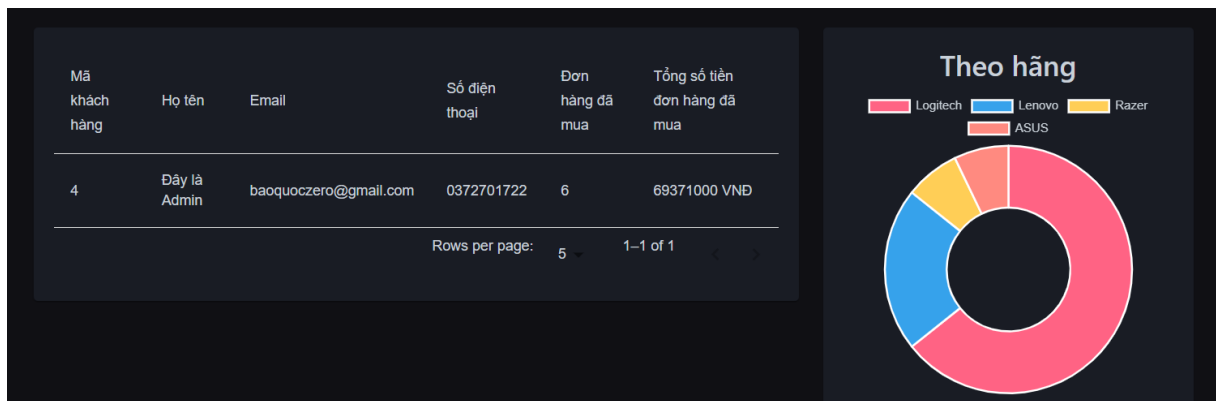
Hỗ trợ đưa ra các quyết định kinh doanh (tăng khuyến mãi, nhập hàng...).



Hình 4. 7 Giao diện thống kê



Hình 4. 8 Giao diện thống kê Heatmap



Hình 4. 9 Giao diện thống kê Biểu đồ tròn

Giao diện quản lý người dùng

Hiển thị danh sách người dùng trong hệ thống.

Hỗ trợ tìm kiếm, phân trang, lọc theo vai trò (admin, khách hàng).

Cho phép xem chi tiết, sửa thông tin hoặc khóa tài khoản người dùng.

Giúp quản trị viên kiểm soát người dùng.

Quản lý hệ thống

Thống kê doanh thu

Quản lý người dùng

Danh sách người dùng

Quản lý sản phẩm

Quản lý đơn hàng

NEED HELP?

Kho hàng

HỖ TRỢ

TIẾNG VIỆT

U

Quản lý người dùng

Họ Tên	Email	Số điện thoại	Vai Trò	Trạng thái	Actions
<div><div></div></div> Bảo Nguyễn	baoquoczero@gmail.com	0372701722	Admin	Đang hoạt động	<div>CHỈNH SỬA</div>
<div><div></div></div> Bảo Nguyễn	baoquocone@gmail.com	0372701722	User	Đang hoạt động	<div>CHỈNH SỬA</div>

Hình 4. 10 Giao diện quản lý người dùng

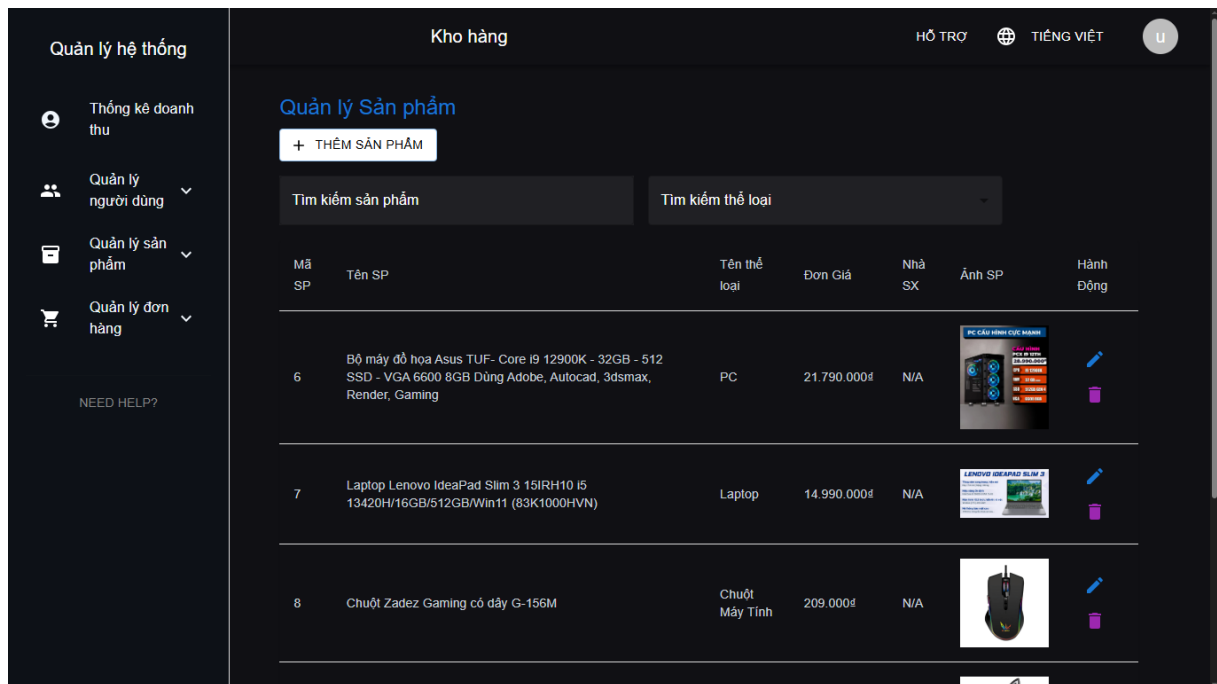
Giao diện quản lý sản phẩm

Danh sách các sản phẩm hiện có trong hệ thống: tên, giá, số lượng, trạng thái.

Cho phép thêm mới, cập nhật thông tin, ẩn/hiện hoặc xóa sản phẩm.

Hỗ trợ upload hình ảnh sản phẩm.

Là giao diện cốt lõi để cập nhật hàng hóa trên sàn thương mại.



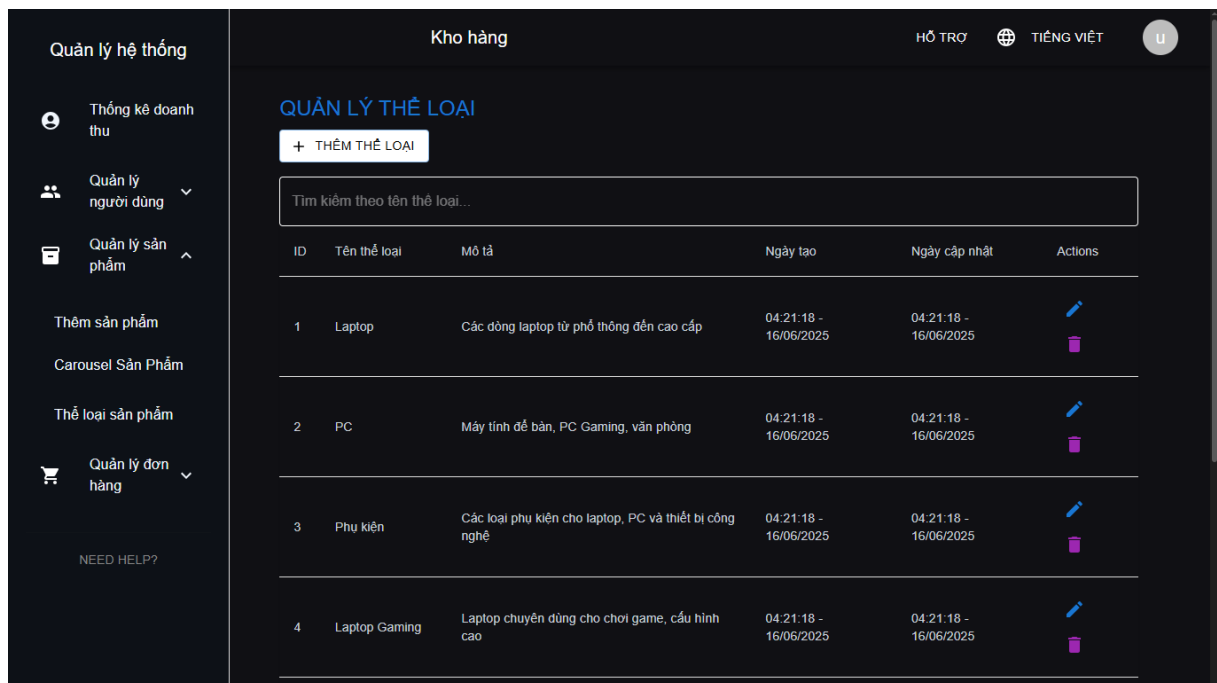
Hình 4. 11 Giao diện quản lý sản phẩm

Giao diện quản lý loại sản phẩm

Hiển thị danh sách các loại sản phẩm (Laptop, Điện thoại, Gaming Gear...).

Hỗ trợ thêm, sửa, xóa hoặc sắp xếp danh mục.

Có thể phân loại theo cha – con (danh mục cha – danh mục con).



Hình 4. 12 Giao diện quản lý loại sản phẩm

Giao diện quản lý đơn hàng

Hiển thị tất cả đơn hàng trong hệ thống: người đặt, tổng tiền, trạng thái, phương thức thanh toán.

Cho phép cập nhật trạng thái đơn hàng: chờ xử lý, đang giao, đã giao, hủy.
Hỗ trợ tìm kiếm theo tên người dùng, thời gian đặt, trạng thái đơn.

Quản lý hệ thống

Thống kê doanh thu

Quản lý người dùng

Quản lý sản phẩm

Quản lý đơn hàng

Đang chờ xác nhận

Lịch sử đơn hàng

Phương Thức Thanh Toán

NEED HELP?

Kho hàng

HỖ TRỢ

U

DANH SÁCH ĐƠN HÀNG ĐANG XỬ LÝ

ID Đơn Hàng	Người dùng	Số điện thoại	Tỉnh thành	Tổng Tiền	Trạng Thái	Ngày Tạo	Ngày Cập Nhật	Chi Tiết	Hành động
52	Đầy là Admin	0372701722	WBXJ+46R, Đường D5, Phường 5, Trà Vinh, Việt Nam	14.990.000đ	Đang chờ xác nhận	17/07/2025 09:28	N/A		XÁC NHẬN HỦY
53	Đầy là Admin	0372701722	WBXJ+46R, Đường D5, Phường 5, Trà Vinh, Việt Nam	939.000đ	Đang chờ xác nhận	17/06/2025 09:28	N/A		XÁC NHẬN HỦY
53	Đầy là Admin	0372701722	WBXJ+46R, Đường D5, Phường 5, Trà Vinh, Việt Nam	939.000đ	Đang chờ xác nhận	17/06/2025 09:28	N/A		XÁC NHẬN HỦY
54	Đầy là Admin	0372701722	WBXJ+46R, Đường D5, Phường 5, Trà Vinh, Việt Nam	209.000đ	Đang chờ xác nhận	17/05/2025 09:29	N/A		XÁC NHẬN HỦY
55	Đầy là Admin	0372701722	WBXJ+46R, Đường D5, Phường 5, Trà Vinh, Việt Nam	1.254.000đ	Đang chờ xác nhận	17/04/2025 09:29	N/A		XÁC NHẬN HỦY
			WBXJ+46R, Đường D5, Phường 5, Trà Vinh, Việt Nam		Đang chờ xác nhận	17/06/2025 09:28	N/A		XÁC NHẬN HỦY

Hình 4. 13 Giao diện quản lý đơn hàng

Giao diện lịch sử đơn hàng

Hiện thị tất cả các đơn hàng mà người dùng đã đặt.

Cho phép người dùng xem chi tiết từng đơn: sản phẩm, số lượng, tổng tiền, trạng thái.

Quản lý hệ thống

Thống kê doanh thu

Quản lý người dùng

Quản lý sản phẩm

Quản lý đơn hàng

Đang chờ xác nhận

Lịch sử đơn hàng

Phương Thức Thanh Toán

NEED HELP?

Kho hàng

HỖ TRỢ

U

DANH SÁCH ĐƠN HÀNG GIAO DỊCH THÀNH CÔNG

ID Đơn Hàng	Người dùng	Số điện thoại	Tỉnh thành	Tổng Tiền	Trạng Thái	Ngày Tạo	Ngày Cập Nhật	Chi Tiết	Trạng thái thanh toán
57	Đây là Admin	0372701722	W&JX+46R, Đường D5, Phường 5, Trà Vinh, Việt Nam	36.989.000đ	Đang chờ xác nhận	17/08/2025 09:29	N/A		CHƯA THANH TOÁN
57	Đây là Admin	0372701722	W&JX+46R, Đường D5, Phường 5, Trà Vinh, Việt Nam	36.989.000đ	Đang chờ xác nhận	17/08/2025 09:29	N/A		CHƯA THANH TOÁN
57	Đây là Admin	0372701722	W&JX+46R, Đường D5, Phường 5, Trà Vinh, Việt Nam	36.989.000đ	Đang chờ xác nhận	17/08/2025 09:29	N/A		CHƯA THANH TOÁN
56	Đây là Admin	0372701722	W&JX+46R, Đường D5, Phường 5, Trà Vinh, Việt Nam	14.990.000đ	Đang chờ xác nhận	17/08/2025 09:29	N/A		CHƯA THANH TOÁN
52	Đây là Admin	0372701722	W&JX+46R, Đường D5, Phường 5, Trà Vinh, Việt Nam	14.990.000đ	Đang chờ xác nhận	17/07/2025 09:28	N/A		CHƯA THANH TOÁN

Hình 4. 14 Giao diện lịch sử đơn hàng

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Sau quá trình nghiên cứu và phát triển, đề tài "Xây dựng website bán đồ công nghệ" đã hoàn thành với các chức năng cơ bản của một hệ thống thương mại điện tử. Website cho phép người dùng thực hiện các thao tác như xem sản phẩm, tìm kiếm, thêm vào giỏ hàng, đặt hàng và theo dõi đơn hàng. Đồng thời, hệ thống cũng cung cấp giao diện quản trị cho phép quản lý sản phẩm, đơn hàng và người dùng.

Việc sử dụng ReactJS cho frontend đã giúp xây dựng một giao diện hiện đại, phản hồi nhanh, nâng cao trải nghiệm người dùng. NodeJS (Express) ở phía backend đảm bảo khả năng xử lý hiệu quả các yêu cầu và quản lý dữ liệu. Cơ sở dữ liệu được thiết kế hợp lý, hỗ trợ tốt cho các nghiệp vụ cần thiết.

Thông qua việc thực hiện đề tài, em đã có cơ hội củng cố và vận dụng kiến thức đã học, rèn luyện kỹ năng lập trình web fullstack, phân tích thiết kế hệ thống, cũng như khả năng tự học và giải quyết vấn đề thực tiễn.

5.2. Hạn chế

- Giao diện người dùng còn đơn giản, chưa tối ưu cho thiết bị di động (mobile responsive).
- Chưa tích hợp chức năng thanh toán trực tuyến qua cổng thanh toán thực tế (ví dụ: Momo, VNPay, Paypal,...).
- Chưa triển khai chức năng gửi email thông báo đơn hàng hoặc xác nhận tài khoản.
- Hệ thống bảo mật mới ở mức cơ bản, chưa có các biện pháp nâng cao như mã hóa JWT, phân quyền nâng cao,...
- Chưa có hệ thống phân tích, thống kê nâng cao về doanh thu, lượt truy cập, hành vi người dùng.

5.3. Hướng phát triển

- Thiết kế giao diện responsive phù hợp với nhiều loại thiết bị (điện thoại, máy tính bảng,...).
- Tích hợp các cổng thanh toán trực tuyến để thực hiện thanh toán thật.
- Thêm chức năng đánh giá, nhận xét sản phẩm và hiển thị xếp hạng đánh giá.
- Phát triển chức năng gửi email xác nhận đơn hàng và khôi phục mật khẩu.
- Xây dựng hệ thống quản lý khuyến mãi, mã giảm giá.
- Áp dụng các công nghệ như JWT, OAuth2 để nâng cao bảo mật hệ thống.

- Triển khai hệ thống phân tích dữ liệu, giúp quản trị viên theo dõi xu hướng mua sắm và tối ưu kinh doanh.

DANH MỤC TÀI LIỆU THAM KHẢO

Phạm Thị Phương Nam, Tài liệu giảng dạy môn “Hệ quản trị Cơ sở dữ liệu”, Trà Vinh, 2024.

Phạm Thị Phương Nam, Tài liệu giảng dạy môn “Cơ sở dữ liệu”, Trà Vinh, 2024.

Phạm Thị Trúc Mai, Tài liệu giảng dạy môn "Thiết kế web", Trà Vinh, 2024.

Phạm Thị Trúc Mai, Tài liệu giảng dạy môn "Phát triển ứng dụng web với mã nguồn mở", Trà Vinh, 2025.

Phạm Minh Dương, Tài liệu giảng dạy môn “Phân tích và thiết kế hệ thống thông tin”, Trà Vinh, 2024.

Tim Berners-Lee, "W3SCHOOL," 30 12 2024. [Online]. Available:
www.w3schools.com.