BỘ THÔNG TIN VÀ TRUYỀN THÔNG HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

----o0o-----



PHÁT TRIỂN HỆ THỐNG THƯƠNG MẠI ĐIỆN TỬ ĐỀ TÀI: XÂY DỰNG WEBSITE THƯƠNG MẠI ĐIỆN TỬ

Giảng viên: Kim Ngọc Bách

Nhóm môn học :02

Nhóm bài tập lớn: 20

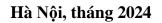
Thành viên của nhóm

Họ và tên Mã sinh viên

Trần Thế Tỏa B21DCCN711

Ngô Quang Phúc B21DCCN099

Trần Tuấn Phúc B21DCCN100



1 Tổng quan về dự án Web/app

1.1 Tổng quan về dự án

Dự án này là một nền tảng thương mại điện tử tiên tiến, được thiết kế để cung cấp giải pháp mua sắm trực tuyến toàn diện, đáp ứng nhu cầu ngày càng cao của người tiêu dùng trong thời đại số hóa. Nền tảng cho phép người dùng dễ dàng duyệt, tìm kiếm và mua sắm sản phẩm từ nhiều danh mục đa dạng như thời trang, điện tử, gia dụng, thực phẩm, và nhiều lĩnh vực khác. Với mục tiêu mang lại trải nghiệm mua sắm thuận tiện, an toàn và phong phú, nền tảng không chỉ tập trung vào việc cải thiện trải nghiệm người dùng mà còn đảm bảo tính bảo mật trong giao dịch và chất lượng dịch vụ.

Dự án được xây dựng để kết nối trực tiếp giữa doanh nghiệp và người tiêu dùng cuối theo mô hình B2C (Business-to-Consumer). Điều này cho phép các doanh nghiệp, từ quy mô nhỏ đến lớn, tiếp cận một lượng lớn khách hàng tiềm năng một cách hiệu quả, trong khi người tiêu dùng được hưởng lợi từ sự đa dạng sản phẩm, giá cả cạnh tranh và dịch vụ hỗ trợ tận tâm. Hệ sinh thái thương mại điện tử này được tối ưu hóa để đáp ứng các tiêu chuẩn hiện đại, tích hợp các công nghệ tiên tiến như trí tuệ nhân tạo (AI) để cá nhân hóa trải nghiệm mua sắm, cùng với các phương thức thanh toán an toàn và giao hàng nhanh chóng. Mục tiêu cuối cùng là tạo ra một nền tảng mua sắm trực tuyến bền vững, nơi doanh nghiệp và người tiêu dùng có thể tương tác một cách hiệu quả, thúc đẩy sự phát triển của thị trường thương mại điện tử.

1.2 Mô hình kinh doanh

Mô hình kinh doanh của dự án dựa trên nền tảng thương mại điện tử B2C, đóng vai trò trung gian kết nối giữa các doanh nghiệp cung cấp sản phẩm/dịch vụ và người tiêu dùng cuối. Nền tảng cung cấp một không gian trực tuyến nơi người dùng có thể truy cập vào hàng triệu sản phẩm từ nhiều nhà cung cấp khác nhau, được phân loại rõ ràng theo danh mục để dễ dàng tìm kiếm và lựa chọn. Các doanh nghiệp tham gia nền tảng có thể tận dụng cơ sở hạ tầng kỹ thuật số, bao gồm hệ thống quản lý sản phẩm, thanh toán trực tuyến và dịch vụ khách hàng, để tiếp cận khách hàng mà không cần xây dựng một hệ thống riêng.

Đối với người tiêu dùng, nền tảng mang đến một trải nghiệm mua sắm liền mạch với giao diện thân thiện, công cụ tìm kiếm thông minh, và các tính năng như đánh giá sản phẩm, so sánh giá cả, và gợi ý cá nhân hóa. Ngoài ra, nền tảng còn tích hợp các dịch vụ hậu cần như giao hàng nhanh và chính sách đổi trả linh hoạt, giúp tăng cường sự hài lòng của khách hàng. Bằng cách tạo ra một hệ sinh thái thương mại điện tử toàn diện, dự án không chỉ đáp ứng nhu cầu mua sắm mà còn xây dựng lòng tin và sự trung thành từ cả phía người mua và người bán.

1.3 Mô hình doanh thu

Nền tảng thương mại điện tử này được thiết kế để tạo ra doanh thu thông qua hai nguồn chính: phí giao dịch và quảng cáo sản phẩm. Cả hai nguồn này không chỉ đảm bảo tính bền vững tài chính của nền tảng mà còn hỗ trợ việc cải thiện dịch vụ và mở rộng quy mô hoạt động.

Phí giao dịch:Nền tảng thu một khoản phí giao dịch nhỏ, được tính dựa trên phần trăm giá trị của mỗi đơn hàng thành công. Cụ thể, khi người mua hoàn tất việc thanh toán cho một sản phẩm hoặc dịch vụ thông qua nền tảng, một phần doanh thu từ giao dịch này sẽ được trích lại để đóng góp vào chi phí vận hành, bảo trì hệ thống, và nâng cấp các tính năng mới. Tỷ lệ phí giao dịch được thiết kế để cạnh tranh, đảm bảo không gây áp lực tài chính quá lớn cho người bán, đồng thời vẫn mang lại nguồn thu ổn định cho nền tảng.

Quảng cáo sản phẩm:Nguồn doanh thu thứ hai đến từ các dịch vụ quảng cáo sản phẩm, cho phép người bán nâng cao khả năng tiếp cận khách hàng thông qua các vị trí hiển thị nổi bật trên nền tảng

Dịch vụ quảng cáo không chỉ mang lại doanh thu mà còn tạo động lực cho người bán cải thiện chất lượng sản phẩm và dịch vụ của họ để cạnh tranh trên nền tảng. Đồng thời, người mua được hưởng lợi từ việc khám phá các sản phẩm chất lượng cao và phù hợp hơn với nhu cầu của họ, nhờ vào các thuật toán gợi ý thông minh kết hợp với quảng cáo.

Cả hai nguồn doanh thu này được thiết kế để bổ trợ lẫn nhau, tạo ra một mô hình tài chính cân bằng và bền vững. Trong tương lai, nền tảng có thể mở rộng

thêm các nguồn thu khác, chẳng hạn như cung cấp các gói thành viên cao cấp cho người bán với nhiều lợi ích hơn, hoặc tích hợp các dịch vụ giá trị gia tăng như tài chính hỗ trợ hoặc phân tích dữ liệu thị trường.

1.4 Mục tiêu của đề tài

Mục tiêu của dự án là xây dựng một nền tảng thương mại điện tử B2C (Business-to-Consumer) hiện đại, cung cấp giải pháp mua sắm trực tuyến toàn diện, kết nối hiệu quả giữa doanh nghiệp và người tiêu dùng cuối. Cụ thể, dự án hướng đến:

- 1. **Cung cấp trải nghiệm mua sắm tối ưu**: Mang lại một nền tảng trực tuyến thuận tiện, an toàn và đa dạng, cho phép người dùng dễ dàng duyệt, tìm kiếm và mua sắm sản phẩm từ nhiều danh mục khác nhau, với giao diện thân thiện và các tính năng thông minh như gợi ý cá nhân hóa, so sánh giá, và đánh giá sản phẩm.
- 2. Đảm bảo tính bền vững và an toàn: Xây dựng một nền tảng đáng tin cậy với các phương thức thanh toán an toàn, chính sách đổi trả linh hoạt, và cơ chế bảo mật cao để bảo vệ thông tin người dùng và giao dịch.
- 3. **Thúc đẩy tăng trưởng kinh doanh**: Hỗ trợ các doanh nghiệp mở rộng thị trường thông qua các công cụ như quảng cáo sản phẩm, phân tích dữ liệu, và tích hợp dịch vụ hậu cần, đồng thời tạo nguồn doanh thu bền vững cho nền tảng thông qua phí giao dịch và dịch vụ quảng cáo.

2 Cơ sở lý thuyết, công nghệ áp dụng

2.1 Cơ sở lý thuyết

Dự án nền tảng thương mại điện tử được xây dựng dựa trên một số khái niệm và nguyên tắc lý thuyết cốt lõi, đảm bảo ứng dụng không chỉ đáp ứng các yêu cầu chức năng mà còn dễ dàng mở rộng, bảo trì và tối ưu hóa hiệu suất. Các khái niệm lý thuyết chính được áp dụng bao gồm:

2.1.1 Kiến trúc MVC (Model-View-Controller)

Kiến trúc MVC là một mô hình thiết kế phần mềm phổ biến, được sử dụng để

tách biệt các mối quan tâm trong ứng dụng, giúp tổ chức mã nguồn một cách rõ ràng và dễ quản lý. Cụ thể:

- Model: Đại diện cho dữ liệu và logic nghiệp vụ của ứng dụng, chịu trách nhiệm quản lý cấu trúc dữ liệu, tương tác với cơ sở dữ liệu và xử lý các quy tắc kinh doanh. Trong nền tảng thương mại điện tử, Model bao gồm các thực thể như sản phẩm, đơn hàng, tài khoản người dùng và giỏ hàng.
- View: Là giao diện người dùng, hiển thị dữ liệu từ Model đến người dùng cuối. View đảm bảo rằng thông tin được trình bày một cách trực quan, dễ hiểu, và đáp ứng trên nhiều thiết bị. Ví dụ, giao diện danh mục sản phẩm, trang chi tiết sản phẩm, hoặc giỏ hàng đều thuộc về View.
- Controller: Đóng vai trò trung gian, xử lý các yêu cầu từ người dùng, tương tác với Model để lấy hoặc cập nhật dữ liệu, sau đó gửi dữ liệu phù hợp đến View để hiển thị. Controller đảm bảo rằng các hành động của người dùng, như thêm sản phẩm vào giỏ hàng hoặc thực hiện thanh toán, được xử lý chính xác.
 Việc áp dụng MVC giúp cải thiện khả năng bảo trì mã nguồn, cho phép các nhà phát triển làm việc trên từng thành phần độc lập mà không ảnh hưởng đến các phần khác, đồng thời hỗ trợ mở rộng ứng dụng khi cần tích hợp các tính năng mới.

2.1.2 Thiết kế API RESTful

Nền tảng sử dụng các nguyên tắc RESTful (Representational State Transfer) để thiết kế các API, đảm bảo giao tiếp giữa frontend và backend hiệu quả, trực quan và dễ mở rộng. Các đặc điểm chính của API RESTful trong dự án bao gồm:

- Giao tiếp không trạng thái: Mỗi yêu cầu từ client đến server đều chứa đầy đủ thông tin cần thiết, không phụ thuộc vào trạng thái của yêu cầu trước đó, giúp tăng khả năng mở rộng và giảm tải cho server.
- URL dựa trên tài nguyên: Các tài nguyên như sản phẩm, đơn hàng, hoặc người dùng được biểu diễn thông qua các URL rõ ràng, ví dụ: /products để truy xuất dah sách sản phẩm hoặc /products/{id} để lấy thông tin sản phẩm cụ thể.
- Phương thức HTTP tiêu chuẩn: Sử dụng các phương thức GET (lấy dữ liệu), POST (tạo mới), PUT (cập nhật), và DELETE (xóa) để thực hiện các thao tác trên tài nguyên, giúp API dễ hiểu và sử dụng.

Định dạng dữ liệu JSON: API trả về dữ liệu ở định dạng JSON, đảm bảo tính tương thích với nhiều nền tảng và thiết bị.
Thiết kế RESTful giúp các nhà phát triển tích hợp dễ dàng với các ứng dụng khách (web, mobile) và hỗ trợ mở rộng trong tương lai, chẳng hạn như tích hợp với các dịch vu bên thứ ba.

2.1.3 Thiết kế đáp ứng (Responsive Design)

Để đáp ứng nhu cầu sử dụng đa nền tảng, ứng dụng được thiết kế đáp ứng, đảm bảo hoạt động mượt mà trên các thiết bị có kích thước màn hình khác nhau, từ điện thoại thông minh, máy tính bảng đến máy tính để bàn. Các kỹ thuật chính bao gồm:

- Bố cục lưới linh hoạt (Flexible Grid Layout): Sử dụng các đơn vị tương đối như phần trăm hoặc rem/em thay vì pixel cố định, giúp giao diện tự động điều chỉnh theo kích thước màn hình.
- Truy vấn phương tiện (Media Queries): Áp dụng các quy tắc CSS khác nhau dựa trên đặc điểm thiết bị, chẳng hạn như độ phân giải hoặc chiều rộng màn hình, để tối ưu hóa giao diện.
- Đồ họa vector có thể mở rộng (SVG): Sử dụng SVG cho các biểu tượng và hình ảnh, đảm bảo chất lượng hiển thị không bị giảm khi thay đổi kích thước. Thiết kế đáp ứng không chỉ nâng cao trải nghiệm người dùng mà còn cải thiện SEO (tối ưu hóa công cụ tìm kiếm), giúp nền tảng tiếp cận nhiều khách hàng hơn.

2.1.4 4.Bảo mật

Bảo mật là yếu tố cốt lõi của nền tảng thương mại điện tử, đặc biệt khi xử lý thông tin nhạy cảm như dữ liệu cá nhân và giao dịch tài chính. Các biện pháp bảo mật được áp dụng bao gồm:

• Xác thực (Authentication): Sử dụng các phương pháp như JSON Web Token

(JWT) để xác minh danh tính người dùng, đảm bảo chỉ những người dùng hợp lệ mới có thể truy cập tài nguyên.

- Ůy quyền (Authorization): Kiểm soát quyền truy cập dựa trên vai trò (ví dụ: người mua, người bán, quản trị viên), ngăn chặn truy cập trái phép vào các tính năng hoặc dữ liệu nhạy cảm.
- Xác thực dữ liệu (Data Validation): Kiểm tra và làm sạch dữ liệu đầu vào để
 ngăn chặn các cuộc tấn công như SQL Injection hoặc Cross-Site Scripting
 (XSS).
- Mã hóa dữ liệu: Áp dụng giao thức HTTPS và mã hóa dữ liệu nhạy cảm (như thông tin thanh toán) để bảo vệ trong quá trình truyền tải và lưu trữ.
 Những biện pháp này đảm bảo rằng nền tảng không chỉ an toàn mà còn xây dựng được lòng tin từ người dùng và đối tác.

2.2 Công nghệ áp dụng

Dự án tận dụng một bộ công nghệ hiện đại để xây dựng một nền tảng thương mại điện tử mạnh mẽ, hiệu quả và dễ mở rộng. Các công nghệ được sử dụng được chia thành các hạng mục sau:

2.2.1 Công nghệ Frontend

React.js

React.js là một thư viện JavaScript mã nguồn mở, được sử dụng để xây dựng giao diện người dùng động và hiệu quả, đặc biệt phù hợp với các ứng dụng một trang (Single Page Application - SPA). Trong dự án, React.js được sử dụng để:

- Tạo các thành phần giao diện có thể tái sử dụng, như danh sách sản phẩm, giỏ hàng, và trang thanh toán.
- Quản lý trạng thái ứng dụng thông qua các công cụ như React Context hoặc
 Redux, đảm bảo dữ liệu được đồng bộ hóa mà không cần tải lại trang.

 Tăng tốc độ hiển thị giao diện nhờ cơ chế Virtual DOM, giảm thiểu thao tác trực tiếp trên DOM thực.

React.js giúp nâng cao trải nghiệm người dùng với giao diện mượt mà và phản hồi nhanh chóng, đồng thời hỗ trợ phát triển nhanh các tính năng mới.

TypeScript

TypeScript là một siêu tập của JavaScript, bổ sung tính năng kiểm tra kiểu tĩnh để giảm thiểu lỗi trong quá trình phát triển. Việc sử dụng TypeScript trong dự án mang lại các lợi ích sau:

- Phát hiện lỗi sớm: Các lỗi liên quan đến kiểu dữ liệu được phát hiện ngay trong giai đoạn biên dịch, giúp cải thiện chất lượng mã.
- Cải thiện khả năng bảo trì: Các định nghĩa kiểu rõ ràng giúp mã dễ đọc và dễ hiểu hơn, đặc biệt trong các dự án lớn.
- Tích hợp tốt với React: TypeScript hỗ trợ định nghĩa kiểu cho các thành phần React, props và state, đảm bảo tính nhất quán trong toàn bộ ứng dụng.
 TypeScript là lựa chọn lý tưởng để xây dựng một frontend đáng tin cậy và dễ mở rộng.

Tailwind CSS

Tailwind CSS là một framework CSS tiện ích-first (utility-first), cho phép xây dựng giao diện người dùng trực tiếp trong mã HTML bằng cách sử dụng các lớp CSS nhỏ gọn và linh hoạt. Trong dự án, Tailwind CSS được sử dụng nhằm mục đích:

Tăng tốc quá trình phát triển giao diện thông qua việc sử dụng các lớp tiện ích, giúp lập trình viên không cần viết nhiều mã CSS thủ công.

Tạo giao diện tùy chỉnh linh hoạt, không bị ràng buộc bởi các thành phần giao diện cố định như một số thư viện UI khác.

Hỗ trợ responsive design hiệu quả nhờ hệ thống breakpoint tích hợp đảm bảo giao diện hoạt động tốt trên mọi thiết bị.

Dễ dàng mở rộng và đồng bộ thiết kế thông qua cấu hình theme cho phép tùy chỉnh màu sắc, font chữ, spacing, v.v.

Việc sử dụng Tailwind CSS giúp giao diện của ứng dụng mang tính cá nhân hóa cao, tối ưu hiệu suất và duy trì được sự nhất quán trong thiết kế mà không cần viết CSS thủ công nhiều.

2.2.2 Công nghệ Backend

Node.js

Node.js là một nền tảng chạy JavaScript phía server, dựa trên công cụ V8 của Google Chrome. Trong dự án, Node.js được sử dụng làm nền tảng chính cho backend với các ưu điểm:

Xử lý bất đồng bộ và phi luồng (non-blocking I/O): Giúp xử lý đồng thời nhiều request, phù hợp cho các ứng dụng thương mại điện tử có lượng người dùng truy cập lớn.

Hiệu suất cao: Sử dụng event-driven architecture giúp tối ưu tốc độ phản hồi và hiệu quả sử dụng tài nguyên.

Hệ sinh thái phong phú: Thư viện npm khổng lồ cho phép tích hợp các tính năng như xác thực, kết nối cơ sở dữ liệu, thanh toán,...

Node.js giúp phát triển backend một cách nhanh chóng, linh hoạt và dễ dàng mở rộng quy mô trong tương lai.

Express.js

Express.js là một framework web nhẹ chạy trên nền Node.js, được sử dụng để xây dựng các API RESTful trong dự án. Các lợi ích của Express bao gồm: Cấu trúc gọn nhẹ, dễ tiếp cận: Hỗ trợ định tuyến và xử lý request/response rõ ràng. Tích hợp middleware dễ dàng: Cho phép xử lý xác thực, log, xử lý lỗi, upload tệp,... một cách hiệu quả.

Hỗ trợ mô hình RESTful đầy đủ: Xây dựng API phục vụ frontend (React/Vue/...) hoặc ứng dụng mobile một cách dễ dàng.

Express.js đóng vai trò trung gian chính xử lý logic nghiệp vụ, kết nối cơ sở dữ liệu và trả dữ liêu về cho client.

Mongoose (ODM cho MongoDB)

Mongoose là một thư viện Object Data Modeling (ODM) dành cho MongoDB và Node.js, giúp quản lý dữ liệu dễ dàng hơn. Trong dự án, Mongoose được dùng để:

Định nghĩa schema rõ ràng: Các đối tượng như User, Product, Order được mô hình hóa rõ ràng với cấu trúc chặt chẽ.

Hỗ trợ quan hệ dữ liệu: Mặc dù MongoDB là NoSQL, Mongoose vẫn hỗ trợ populate, reference để xử lý các mối quan hệ như một-nhiều (1-N), nhiều-nhiều (N-N).

Tích hợp validation và middleware: Giúp kiểm soát dữ liệu đầu vào và thực hiện logic trung gian như mã hóa mật khẩu, xác minh,...

Mongoose giúp tương tác với MongoDB hiệu quả và giảm thiểu lỗi do thao tác thủ công với cơ sở dữ liệu.

JSON Web Token (JWT)

JWT được sử dụng trong dự án để xác thực người dùng và bảo vệ các API. Lý do sử dụng:

Xác thực không trạng thái (stateless): Token được lưu trữ trên client và gửi kèm trong mỗi request, không cần lưu session trên server.

Phân quyền linh hoạt: Có thể nhúng thông tin vai trò người dùng (admin, customer, seller...) vào token để kiểm soát quyền truy cập.

An toàn và hiệu quả: Token được ký bằng secret key hoặc RSA, đảm bảo dữ liệu không bị giả mạo.

JWT kết hợp với middleware trong Express giúp bảo vệ toàn bộ hệ thống API một cách hiệu quả.

2.2.3 Cơ sở dữ liệu: MongoDB

MongoDB là hệ quản trị cơ sở dữ liệu NoSQL phổ biến, được sử dụng để

lưu trữ dữ liệu trong định dạng JSON (BSON). Ưu điểm của MongoDB trong dự án:

Lưu trữ linh hoạt: Không yêu cầu schema cố định, dễ dàng mở rộng và thêm trường mới trong tương lai.

Hiệu suất cao: Hỗ trợ truy vấn nhanh và tối ưu hóa cho các thao tác đọc ghi lớn. Khả năng mở rộng tốt: Dễ dàng scale theo chiều ngang khi ứng dụng phát triển với lượng dữ liệu lớn.

Tích hợp tốt với Mongoose: Giúp thao tác CRUD nhanh chóng và có kiểm soát. MongoDB là nền tảng chính lưu trữ dữ liệu sản phẩm, đơn hàng, người dùng, và toàn bộ thông tin liên quan đến hoạt động thương mại điện tử.

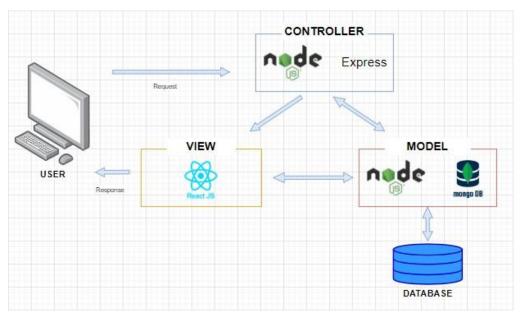
2.2.4 Quản lý phiên bản(Git)

Git là hệ thống quản lý phiên bản phân tán, được sử dụng để theo dõi các thay đổi trong mã nguồn trong suốt quá trình phát triển. Các tính năng chính bao gồm:

- Cộng tác nhóm: Cho phép nhiều nhà phát triển làm việc đồng thời trên cùng một dự án mà không gây xung đột.
- Quản lý lịch sử mã: Lưu trữ mọi phiên bản của mã nguồn, hỗ trợ quay lại các phiên bản trước nếu cần.

Tích hợp với CI/CD: Kết hợp với các công cụ như Jenkins hoặc GitHub Actions để tự động hóa quy trình triển khai.

Git đảm bảo rằng mã nguồn của dự án được tổ chức tốt, dễ dàng quản lý và bảo trì.



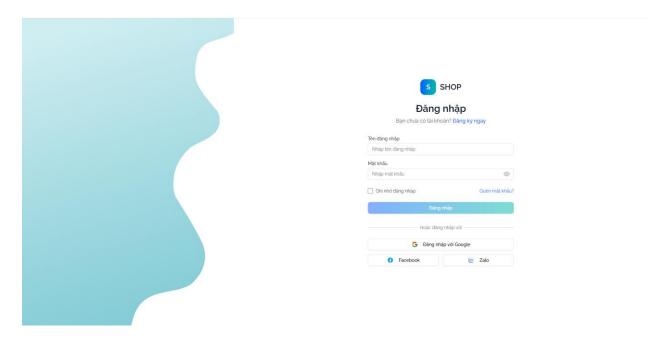
Hình 2.1 Kiến trúc tổng Thể của hệ thống

3 Liệt kê, mô tả các tính năng sẽ thiết kế và triển khai

3.1 Tính năng người dùng

3.1.1 Đăng ký và Đăng nhập người dùng:

- No table of figures entries found. Form đăng nhập với các trường cho username và mật khẩu trên trang Login.
- Form đăng ký với các trường cho họ tên người dùng, username, email và mật khẩu trên trang Register.
- Thông báo lỗi nếu thông tin đăng nhập không chính xác.



Hình 3.1Đăng nhập

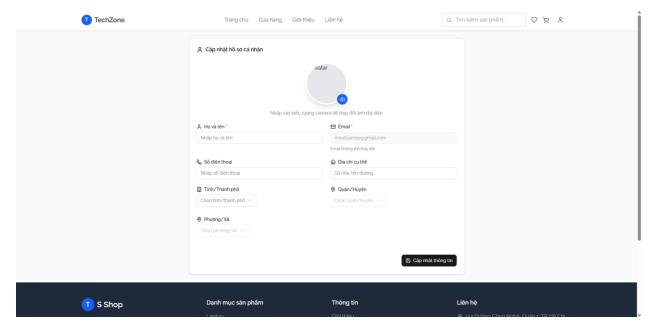




Hình 3.2Đăng kí

3.1.2 Quản lý hồ sơ:

- Thông tin cá nhân như tên, email...
- Nút để cập nhật thông tin.
- Thông báo thành công hoặc lỗi khi cập nhật.

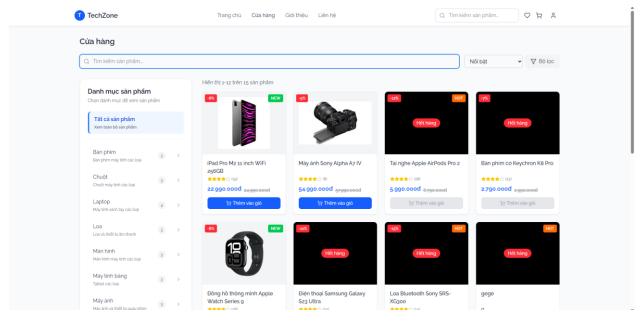


Hình 3.3Profile

Hình 2: Giao diện Profile

3.1.3 Duyệt và Tìm kiếm sản phẩm:

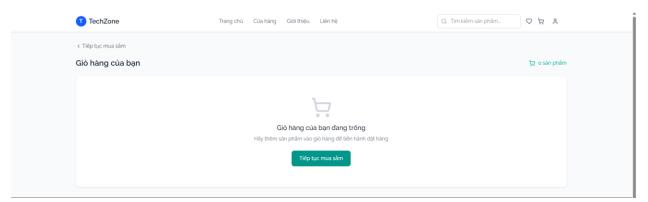
- Danh sách sản phẩm với hình ảnh, tên, giá, và đánh giá.
- Thanh tìm kiếm và bộ lọc để tìm kiếm sản phẩm theo danh mục, giá cả
- Nút để thêm sản phẩm vào giỏ hàng



Hình 3.4Tìm kiếm

3.1.4 Giỏ hàng:

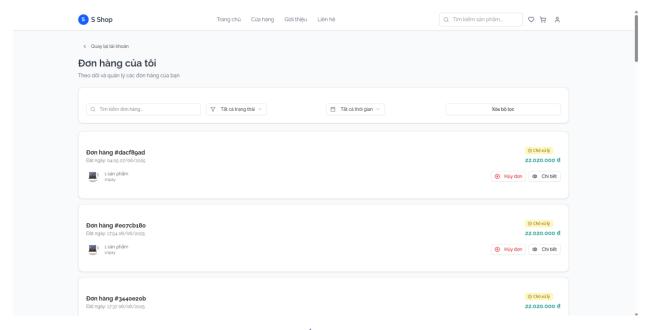
- Danh sách sản phẩm đã thêm vào giỏ hàng với thông tin chi tiết (hình ảnh, tên, giá, số lượng).
- Tổng giá trị đơn hàng và nút để tiến hành thanh toán.
- Tùy chọn để cập nhật số lượng hoặc xóa sản phẩm khỏi giỏ hàng



Hình 3.5Giỏ hàng

3.1.5 Quản lý đơn hàng:

- Danh sách các đơn hàng đã đặt với trạng thái (đang xử lý, đã giao, đã hủy) trên trang
- Chi tiết đơn hàng khi nhấp vào từng đơn hàng
- Tùy chọn để theo dõi trạng thái đơn hàng.

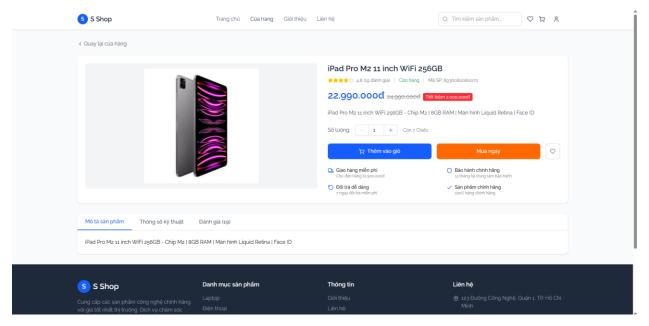


Hình 3.6 Chi tiết đơn hàng

3.1.6 Xem chi tiết sản phẩm:

Trang: ProductDetail

- Form để người dùng để lại đánh giá và xếp hạng cho sản phẩm.
- Danh sách các đánh giá từ người dùng khác.

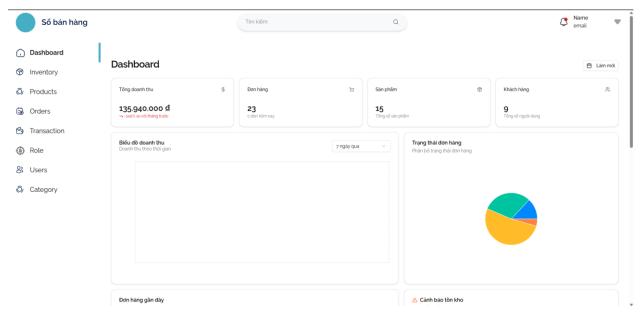


Hình 3.7: Chi tiết sản phẩm

3.2 Tính năng quản trị viên

3.2.1 Dashboard

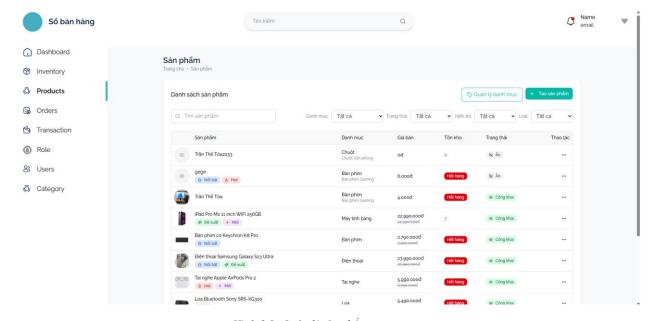
Biểu đồ và thống kê về doanh số, hoạt động của người dùng và mức tồn kho.



Hình 3.8:Dashboard

3.2.2 Quản lý sản phẩm:

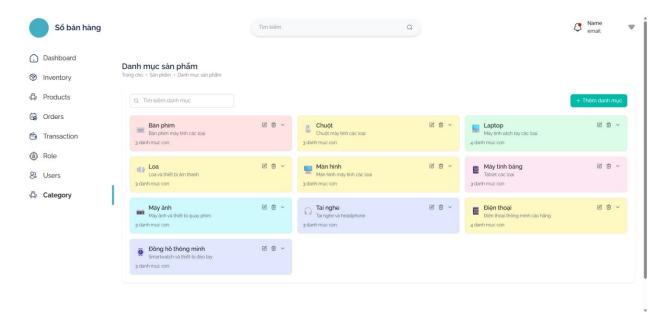
- Danh sách sản phẩm với tùy chọn thêm, sửa, xóa sản phẩm trên trang
- Form để nhập thông tin chi tiết sản phẩm trên trang ProductForm.



Hình 3.9: Quản lí sản phẩm

3.2.3 Quản lý danh mục:

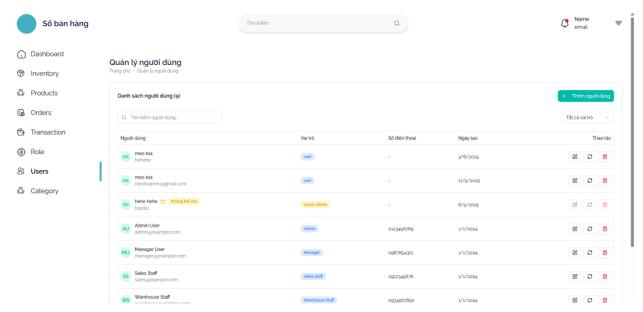
■ Danh sách các danh mục sản phẩm với tùy chọn thêm, sửa, xóa danh mục.



Hình 3.10: Quản lí danh mục

3.2.4 Quản lý người dùng:

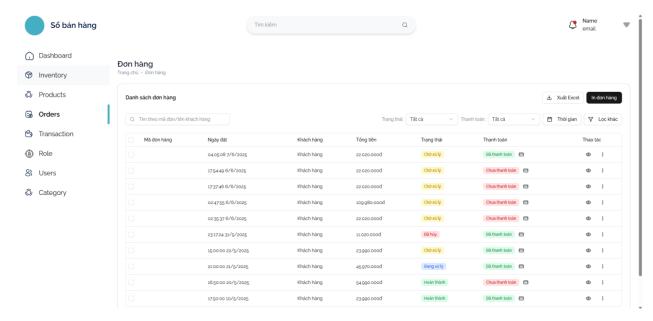
■ Danh sách người dùng với tùy chọn xem chi tiết hoặc xóa tài khoản



Hình 3.11: Quản lí người dùng

3.2.5 Quản lý đơn hàng:

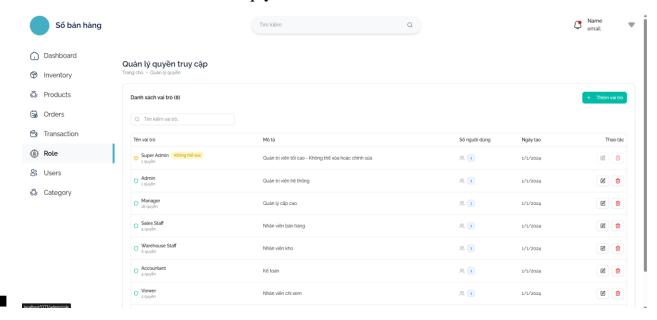
 Danh sách tất cả các đơn hàng với tùy chọn xử lý trả hàng và yêu cầu của khách hàng.



Hình 3.12:Quản lí đơn hàng

3.2.6 Quản lý quyền truy cập:

Chình sửa chi tiết các quyền

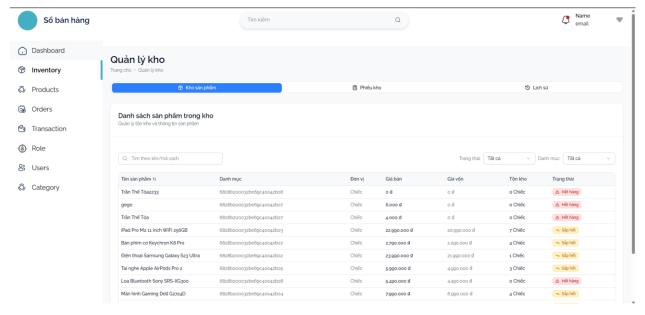


Hình 3.13: Quyền truy cập

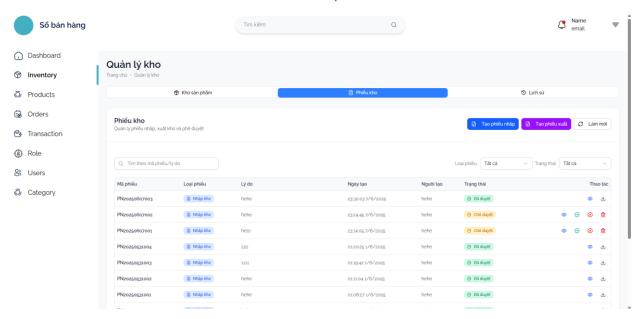
3.2.7 Quản lí kho

- Xem lịch sử nhập kho
- -Xem số lượng thay đổi từng sản phẩm
- -Xem thông tin tồn kho sản phẩm

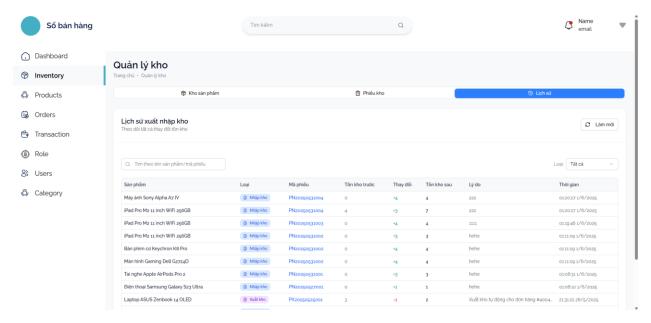
-Tạo/ phê duyệt phiếu nhập xuất kho



Hình 3.14: Tồn kho sản phẩm



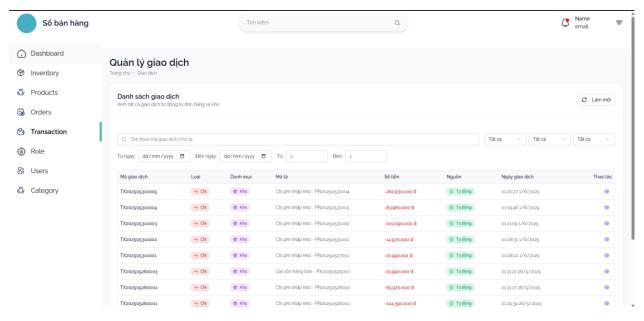
Hình 3.15: Phiếu kho



Hình 3.16: Lịch sử kho

3.2.8 Quản lí giao dịch

Xem thông tin các giao dịch thu vào chi



Hình 3.17: Giao dịch

3.3 Tính năng chung

3.3.1 Giao diện người dùng (UI/UX):

- Thiết kế giao diện đáp ứng trên nhiều thiết bị.
- Đảm bảo tương thích với nhiều trình duyệt (Chrome, Firefox, Safari, Edge).

3.3.2 Công cụ tìm kiếm:

Tìm kiếm cơ bản và nâng cao với bộ lọc đa tiêu chí (danh mục, giá, đánh giá). Sắp xếp kết quả tìm kiếm theo nhiều tiêu chí (mới nhất, giá thấp đến cao, đánh giá cao nhất).

4 Hướng Phát triển trong tương lai

• Tùy chọn đa dạng hóa đăng nhập:

 Hỗ trợ thêm các phương thức đăng nhập như Facebook, Apple ID, hoặc SMS OTP.

• Đa ngôn ngữ:

Hỗ trợ đa ngôn ngữ để phục vụ nhiều đối tượng người dùng hơn.

• Tích hợp AI:

- O Tích hợp AI hỗ trợ người dùng, giảm tải suport cho admin
- $\circ \;\;$ Gợi ý sản phẩm theo mùa , theo thời tiết