

TRƯỜNG ĐẠI HỌC PHENIKAA
TRƯỜNG CÔNG NGHỆ THÔNG TIN PHENIKAA



BÁO CÁO KẾT THÚC HỌC PHẦN
Thiết kế Website thương mại điện tử

NGUYỄN MẠNH ĐẠT
22010220@st.phenikaa-uni.edu.vn

Khoa Công nghệ Thông tin

Giảng viên hướng dẫn:

Ths. Vũ Quang Dũng

Bộ môn:

Phân tích thiết kế phần mềm

HÀ NỘI, 3/2025

Lời cảm ơn

Em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy Ths. Vũ Quang Dũng là người đã tận tình hướng dẫn và giúp đỡ chúng em trong suốt quá trình thực hiện đồ án môn học này. Những kiến thức và kinh nghiệm quý báu mà thầy đã chia sẻ không chỉ giúp chúng em hoàn thiện bài làm mà còn mở rộng tầm nhìn và nâng cao hiểu biết của chúng em về lĩnh vực nghiên cứu này.

Tóm tắt nội dung đồ án

Đồ án này phát triển một website thương mại điện tử chuyên mua bán các vật phẩm ảo như key bản quyền phần mềm (Windows, Office, Photoshop), tài khoản số (Netflix, Facebook, Spotify, Canva, v.v.). Vấn đề đặt ra là hiện nay, thị trường giao dịch các loại tài khoản số và key bản quyền còn nhiều bất cập, thiếu nền tảng tập trung để đảm bảo giao dịch an toàn, minh bạch giữa người mua và người bán. Phương pháp thực hiện bao gồm xây dựng một hệ thống web với các chức năng chính như quản lý tài khoản, sản phẩm, đơn hàng và thanh toán trực tuyến.

Công cụ sử dụng bao gồm ReactJS (Vite, Redux) cho frontend, Node.js + Express.js cho backend, MongoDB làm cơ sở dữ liệu, và JWT Authentication để bảo mật.

Kết quả đạt được là một nền tảng hoạt động ổn định, đáp ứng nhu cầu mua bán tài khoản số một cách an toàn và tiện lợi. Hệ thống hỗ trợ người mua đặt hàng nhanh chóng, người bán dễ dàng đăng bán sản phẩm, và quản trị viên kiểm soát nội dung, giao dịch.

Về tính thực tế, website có thể áp dụng ngay vào thị trường, mở rộng với tích hợp cổng thanh toán, hỗ trợ nhiều loại tài khoản số hơn. Qua đồ án, sinh viên đã nâng cao kỹ năng lập trình web, thiết kế hệ thống, xử lý bảo mật và tối ưu hiệu suất website.

Sinh viên thực hiện

Ký và ghi rõ họ tên

Mục lục

CHƯƠNG 1. MỞ ĐẦU	1
1.1 Tổng quan về đồ án	1
1.1.1 Mục tiêu và phạm vi của đồ án	1
1.1.2 Giới thiệu về mô hình kinh doanh bán hàng trực tuyến.....	2
1.1.3 Đối tượng người dùng mục tiêu	2
CHƯƠNG 2. PHÂN TÍCH YÊU CẦU	4
2.1 Yêu cầu chức năng	4
2.1.1 Quản lý người dùng (đăng ký, đăng nhập, quản lý tài khoản).....	4
2.1.2 Quản lý sản phẩm	6
2.1.3 Giỏ hàng và thanh toán.....	7
2.1.4 Tìm kiếm và lọc sản phẩm	9
2.1.5 Quản lý đơn hàng	9
2.2 Yêu Cầu Phi Chức Năng	11
2.2.1 Bảo mật thông tin tài khoản và dữ liệu người dùng	11
2.2.2 Trải nghiệm người dùng và giao diện	11
CHƯƠNG 3. THIẾT KẾ HỆ THỐNG	13
3.1 Kiến trúc hệ thống	13
3.1.1 Sơ đồ kiến trúc tổng thể MERN Stack	13
3.1.2 Mô hình client-server	15
3.1.3 Luồng dữ liệu và tương tác	16
3.2 Thiết kế cơ sở dữ liệu.....	18
3.2.1 Mô hình dữ liệu MongoDB (các collection)	18
3.2.2 Mối quan hệ giữa các collection	20
CHƯƠNG 4. TRIỂN KHAI HỆ THỐNG	23
4.1 Backend.....	23
4.1.1 Cấu trúc thư mục Node.js và Express	23
4.1.2 Models (MongoDB Schemas)	24
4.1.3 Controllers & Routes.....	25
4.1.4 Middleware.....	25

4.1.5	Tích hợp cổng thanh toán PayOS	27
4.2	Frontend Development.....	27
4.2.1	Cấu trúc thư mục React.....	28
4.2.2	Phân cấp component.....	28
4.2.3	Quản lý trạng thái với Redux	28
4.2.4	UI/UX Design	29
4.3	Tích hợp và kiểm thử	29
4.3.1	Unit Tests	29
4.3.2	Integration Tests.....	29
4.3.3	Quy trình tích hợp	30
4.3.4	Tích hợp Front-end với Back-end	30
	CHƯƠNG 5. CÁC CHỨC NĂNG CHÍNH	31
5.1	Giao diện trang chủ	31
5.2	Đăng ký	32
5.3	Đăng nhập	32
5.4	Xem sản phẩm.....	33
5.5	Xem thông tin chi tiết về sản phẩm.....	33
5.6	Xem giỏ hàng	35
5.7	Thanh toán.....	35
5.8	Xem đơn thông tin các đơn hàng	35
5.9	Xem trang cá nhân	36
5.10	Nạp tiền	37
5.11	Chức năng quản lý cho Seller	38
5.12	Chức năng quản lý của admin.....	40
5.12.1	Quản lý người dùng	41
5.12.2	Quản lý tài khoản số (Sản phẩm)	42
5.12.3	Quản lý giao dịch & thanh toán.....	42
5.12.4	Quản lý khiếu nại & báo cáo vi phạm	43
	CHƯƠNG 6. HƯỚNG PHÁT TRIỂN	44
6.1	Mở rộng danh mục sản phẩm.....	44
6.2	Cải thiện trải nghiệm người dùng	44

6.3	Hệ thống đánh giá và bảo mật.....	44
6.4	Tích hợp nhiều phương thức thanh toán	44
6.5	Xây dựng hệ thống Affiliate & Referral	44
CHƯƠNG 7. TỔNG KẾT		46

CHƯƠNG 1. MỞ ĐẦU

1.1 Tổng quan về đồ án

1.1.1 Mục tiêu và phạm vi của đồ án

Đồ án này tập trung vào việc phát triển một nền tảng thương mại điện tử toàn diện chuyên về giao dịch các loại tài khoản trực tuyến phổ biến như Facebook, Netflix, Spotify, và các dịch vụ số khác. Mục tiêu chính của dự án là tạo ra một hệ thống an toàn, tiện lợi và đáng tin cậy cho cả người bán và người mua, đồng thời giải quyết những thách thức đặc thù trong thị trường tài khoản trực tuyến.

Các mục tiêu cụ thể của dự án bao gồm:

1. Xây dựng một nền tảng hoàn chỉnh sử dụng công nghệ MERN Stack (MongoDB, Express.js, React, Node.js) để đảm bảo hiệu suất cao, khả năng mở rộng và trải nghiệm người dùng mượt mà.
2. Phát triển hệ thống quản lý tài khoản, danh mục sản phẩm, và các cơ chế bảo mật để đảm bảo tính toàn vẹn của thông tin và giao dịch.
3. Thiết kế cơ chế giao tài khoản tự động sau khi thanh toán, giảm thiểu sự can thiệp thủ công và tăng tính tiện lợi cho người dùng.
4. Triển khai hệ thống thanh toán đa dạng, tích hợp nhiều phương thức phổ biến tại Việt Nam và quốc tế.
5. Xây dựng hệ thống báo cáo và thống kê chi tiết cho người quản trị, giúp nắm bắt tình hình kinh doanh và đưa ra quyết định phù hợp.

Phạm vi của đồ án bao gồm:

- Phát triển đầy đủ cả frontend và backend của nền tảng.
- Thiết kế và triển khai cơ sở dữ liệu phi quan hệ với MongoDB.
- Xây dựng API RESTful cho tất cả các chức năng của hệ thống.
- Tích hợp các cổng thanh toán PayOS, hỗ trợ thanh toán bảo mật.
- Triển khai các biện pháp bảo mật cho dữ liệu người dùng và thông tin tài khoản.
- Tối ưu hóa hiệu suất và trải nghiệm người dùng trên nhiều thiết bị.

1.1.2 Giới thiệu về mô hình kinh doanh bán hàng trực tuyến

Mô hình kinh doanh bán tài khoản trực tuyến là một phân khúc đặc biệt trong thị trường thương mại điện tử, tập trung vào việc cung cấp quyền truy cập vào các dịch vụ số thông qua việc chuyển giao thông tin đăng nhập. Lĩnh vực này đã phát triển mạnh mẽ trong những năm gần đây, phản ánh nhu cầu ngày càng tăng của người dùng đối với các dịch vụ giải trí, mạng xã hội và tiện ích trực tuyến.

Các hình thức kinh doanh chính trong mô hình này bao gồm:

1. **Bán tài khoản Premium/VIP:** Cung cấp tài khoản có đặc quyền cao cấp cho các dịch vụ streaming như Netflix, Spotify, hay Disney+, thường với giá thấp hơn so với đăng ký trực tiếp.
2. **Chia sẻ tài khoản:** Cho phép nhiều người dùng chia sẻ chi phí của một tài khoản cao cấp, tận dụng khả năng đa người dùng của nhiều nền tảng.
3. **Tài khoản theo thời hạn:** Cung cấp quyền truy cập trong một khoảng thời gian nhất định, từ một tháng đến một năm, với nhiều lựa chọn giá cả.
4. **Tài khoản mạng xã hội:** Bán các tài khoản Facebook, Instagram, Twitter với lượng người theo dõi hoặc tương tác cao cho mục đích marketing.

Dự án này nhằm mục đích hợp pháp hóa và chuẩn hóa thị trường này thông qua:

- Xác minh nguồn gốc hợp pháp của các tài khoản
- Bảo vệ thông tin cá nhân của người bán và người mua
- Đảm bảo tính minh bạch trong giao dịch
- Cung cấp các chính sách bảo hành và hỗ trợ sau bán hàng
- Tuân thủ các quy định pháp luật liên quan đến thương mại điện tử và dịch vụ số

Hệ thống sẽ thu lợi nhuận từ các nguồn:

- Phí hoa hồng từ mỗi giao dịch thành công
- Gói đăng ký cao cấp cho người bán (để tiếp cận nhiều khách hàng hơn)
- Dịch vụ quảng cáo và đẩy sản phẩm
- Phí xác minh tài khoản cao cấp cho người dùng

1.1.3 Đối tượng người dùng mục tiêu

Nền tảng này nhắm đến nhiều nhóm đối tượng người dùng khác nhau, mỗi nhóm với những nhu cầu và mong đợi riêng biệt:

1. **Người tiêu dùng cá nhân:**

- Người dùng tìm kiếm giải pháp tiết kiệm cho các dịch vụ giải trí trực tuyến
- Người mới làm quen với các nền tảng số và muốn thử nghiệm trước khi cam kết dài hạn
- Người dùng không có khả năng thanh toán quốc tế hoặc gặp khó khăn trong việc đăng ký trực tiếp
- Người tiêu dùng quan tâm đến việc tiết kiệm chi phí thông qua việc chia sẻ tài khoản

2. **Người bán tài khoản:**

- Cá nhân muốn kiếm thêm thu nhập từ việc bán lại các tài khoản không sử dụng
- Đại lý chính thức của các dịch vụ số tìm kiếm kênh phân phối mới
- Nhà cung cấp chuyên nghiệp với số lượng lớn tài khoản các loại

3. **Người quản trị và vận hành nền tảng:**

- Quản trị viên hệ thống giám sát hoạt động
- Nhân viên chăm sóc khách hàng và hỗ trợ kỹ thuật
- Đội ngũ marketing và phát triển kinh doanh

Hiểu rõ về các đối tượng người dùng này sẽ giúp định hướng thiết kế giao diện, xây dựng tính năng và phát triển các chính sách phù hợp với từng nhóm. Nền tảng sẽ chú trọng vào việc tạo ra trải nghiệm người dùng trực quan, an toàn và thân thiện, đồng thời đáp ứng các nhu cầu đặc thù của mỗi nhóm đối tượng.

Trong quá trình phát triển, dự án sẽ thực hiện các khảo sát và phân tích nhu cầu người dùng để không ngừng cải thiện và tối ưu hóa nền tảng, đảm bảo đáp ứng tốt nhất mong đợi của các đối tượng mục tiêu.

CHƯƠNG 2. PHÂN TÍCH YÊU CẦU

2.1 Yêu cầu chức năng

2.1.1 Quản lý người dùng (đăng ký, đăng nhập, quản lý tài khoản)

Hệ thống cần hỗ trợ các chức năng quản lý người dùng toàn diện, đảm bảo trải nghiệm mượt mà và an toàn cho người dùng từ khi bắt đầu tạo tài khoản đến khi sử dụng các tính năng nâng cao:

1. Đăng ký tài khoản:

- Đăng ký bằng email và mật khẩu
- Đăng ký thông qua tài khoản mạng xã hội (OAuth với Google, Facebook)
- Xác minh email thông qua liên kết kích hoạt
- Tùy chọn đăng ký làm người mua hoặc người bán (với quy trình xác minh khác nhau)
- Thu thập thông tin cá nhân cần thiết (tên, số điện thoại, địa chỉ)

2. Đăng nhập và xác thực:

- Đăng nhập với email/tên đăng nhập và mật khẩu
- Hỗ trợ đăng nhập thông qua OAuth (Google, Facebook)
- Xác thực hai yếu tố (2FA) cho tài khoản cao cấp
- Khôi phục mật khẩu qua email
- Duy trì phiên đăng nhập an toàn với JWT

3. Quản lý hồ sơ người dùng:

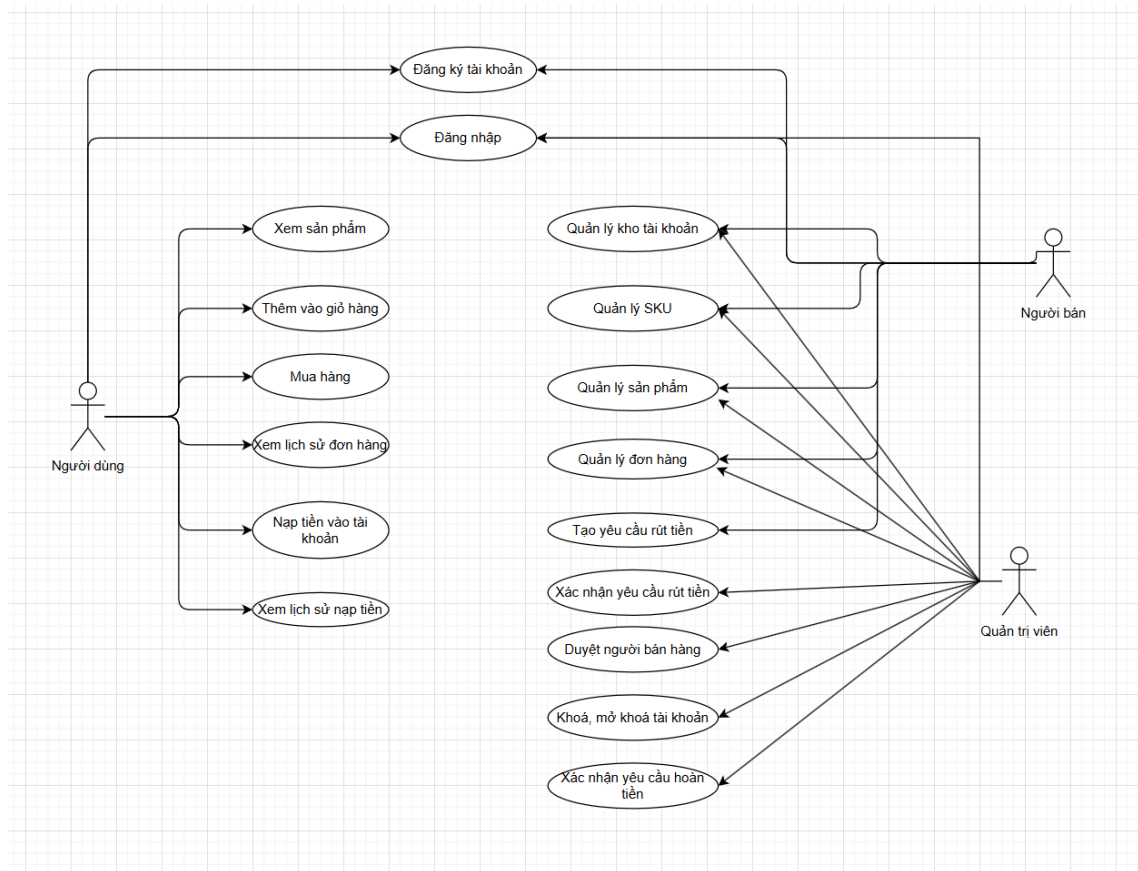
- Xem và chỉnh sửa thông tin cá nhân
- Thay đổi mật khẩu
- Quản lý địa chỉ giao hàng/thanh toán
- Liên kết/hủy liên kết với tài khoản mạng xã hội
- Quản lý phương thức thanh toán
- Xem lịch sử giao dịch và đơn hàng

4. Phân quyền và vai trò:

- Người dùng thông thường (mua hàng)

- Người bán (bán tài khoản, quản lý kho)
- Quản trị viên (quản lý toàn bộ hệ thống)

Sơ đồ usecase của hệ thống:



Hình ảnh 1: Minh họa usecase của hệ thống

Các tác nhân trong sơ đồ:

- Người dùng
- Người bán
- Quản trị viên

2.1.2 Quản lý sản phẩm

Hệ thống cần có khả năng quản lý hiệu quả nhiều loại tài khoản số khác nhau, với các đặc điểm và yêu cầu quản lý riêng biệt:

1. Thêm và quản lý sản phẩm:

- Giao diện thêm sản phẩm (tài khoản) mới
- Nhập thông tin đăng nhập an toàn
- Tải lên hình ảnh minh họa
- Chọn danh mục và thuộc tính
- Thiết lập giá và chính sách

2. Phân loại sản phẩm:

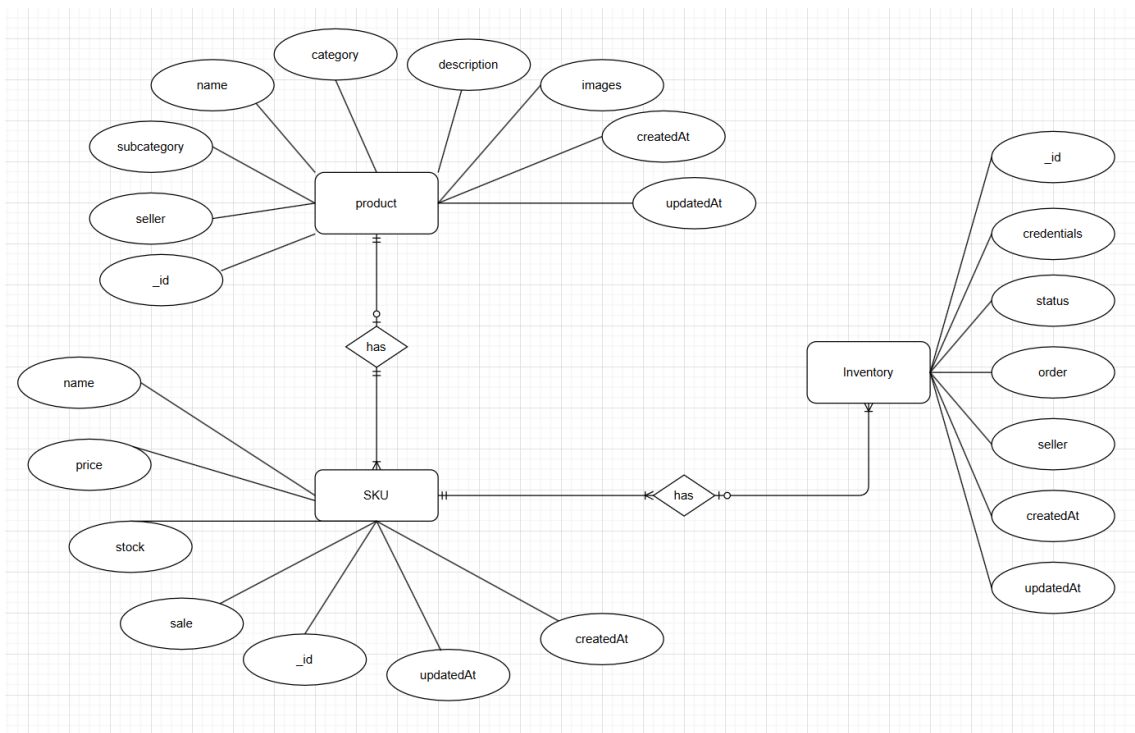
- Tạo và quản lý danh mục (mạng xã hội, streaming, công cụ, v.v.)
- Thiết lập thuộc tính cho từng loại tài khoản (thời hạn, số người dùng, v.v.)
- Nhóm sản phẩm theo nhà cung cấp, loại dịch vụ
- Gắn thẻ và từ khóa cho sản phẩm

3. Quản lý kho và tình trạng:

- Theo dõi số lượng tài khoản có sẵn
- Tự động cập nhật trạng thái sau khi bán
- Cảnh báo khi số lượng tồn kho thấp
- Kiểm tra tính hợp lệ của tài khoản định kỳ

4. Quản lý giá và khuyến mãi:

- Thiết lập giá bán
- Tạo mã giảm giá và chương trình khuyến mãi
- Giá theo gói (1 tháng, 3 tháng, 1 năm)



Hình ảnh 2: Sơ đồ thực thể của sản phẩm

2.1.3 Giỏ hàng và thanh toán

Hệ thống cần có quy trình giỏ hàng và thanh toán đơn giản nhưng đầy đủ tính năng, đảm bảo trải nghiệm mua sắm suôn sẻ:

1. Quản lý giỏ hàng:

- Thêm/xóa sản phẩm khỏi giỏ hàng
- Cập nhật số lượng sản phẩm
- Tính toán tổng tiền tự động
- Lưu giỏ hàng cho phiên đăng nhập sau
- Áp dụng mã giảm giá

2. Quy trình thanh toán:

- Form thanh toán với thông tin cần thiết
- Lựa chọn phương thức thanh toán
- Xác nhận đơn hàng trước khi thanh toán
- Hiện thị tóm tắt đơn hàng
- Xử lý giao dịch an toàn

3. Phương thức thanh toán:

- Chuyển khoản ngân hàng (PayOS)
- Bảo mật thông tin thanh toán

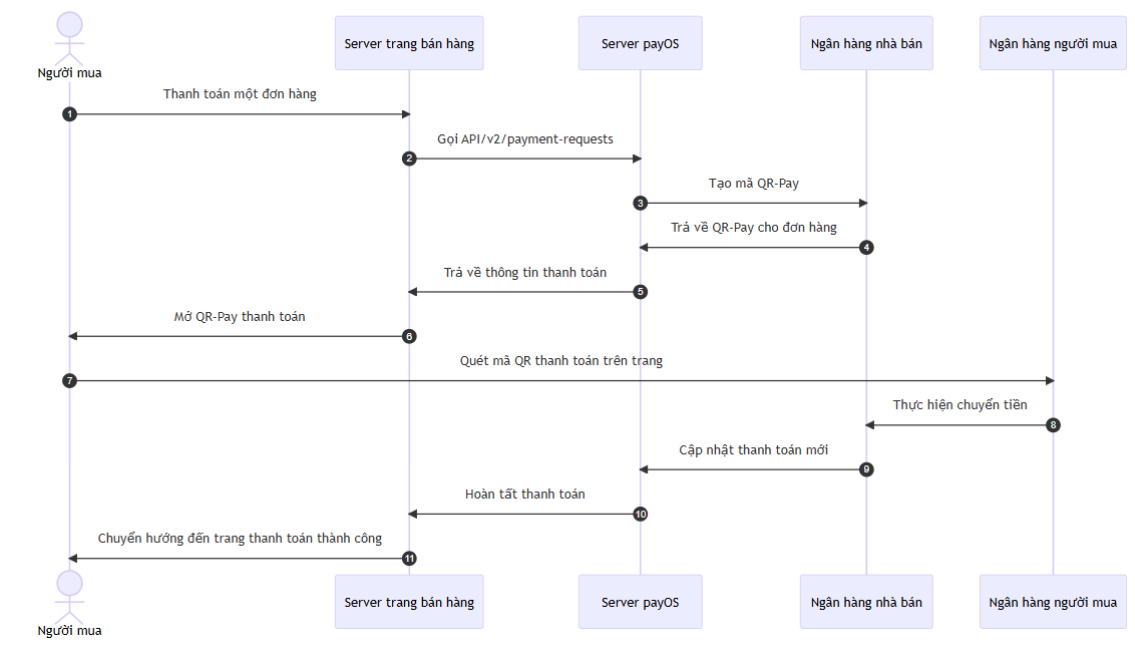
4. Xử lý sau thanh toán:

- Cập nhật trạng thái đơn hàng
- Tạo hóa đơn điện tử
- Cập nhật kho sản phẩm

5. Xử lý lỗi và hoàn tiền:

- Xử lý thanh toán không thành công
- Quy trình hoàn tiền
- Xử lý tranh chấp và khiếu nại
- Chính sách đảm bảo và bảo hành

Sequence Diagram cho quy trình nạp tiền bằng PayOS:



2.1.4 Tìm kiếm và lọc sản phẩm

Hệ thống cần có khả năng tìm kiếm và lọc hiệu quả để người dùng dễ dàng tìm được sản phẩm phù hợp:

1. Chức năng tìm kiếm:

- Tìm kiếm theo từ khóa
- Tìm kiếm theo danh mục
- Tìm kiếm nâng cao với nhiều tiêu chí

2. Bộ lọc sản phẩm:

- Lọc theo danh mục
- Lọc theo giá

3. Sắp xếp kết quả:

- Sắp xếp theo giá (tăng/giảm)
- Sắp xếp theo độ phổ biến
- Sắp xếp theo thời gian đăng
- Sản phẩm mới nhất

4. Hiển thị kết quả:

- Hiển thị dạng lưới hoặc danh sách
- Phân trang kết quả
- Hiển thị thông tin tóm tắt
- Xem nhanh thông tin chi tiết
- Thêm vào giỏ hàng

2.1.5 Quản lý đơn hàng

Hệ thống cần có khả năng quản lý đơn hàng hiệu quả từ khi tạo đến khi hoàn thành:

1. Tạo và xử lý đơn hàng:

- Tạo đơn hàng mới từ giỏ hàng
- Tạo mã đơn hàng duy nhất
- Ghi nhận thông tin người mua và sản phẩm
- Tự động tính tổng tiền và thuế/phí

2. Theo dõi trạng thái đơn hàng:

- Các trạng thái: Chờ thanh toán, Đã thanh toán, Đang xử lý, Đã giao, Hoàn thành, Hủy
- Hiện thị timeline trạng thái đơn hàng
- Thông báo khi trạng thái thay đổi
- Cập nhật trạng thái theo thời gian thực

3. Giao tài khoản:

- Gửi thông tin đăng nhập qua email an toàn
- Hiện thị thông tin tài khoản trong trang đơn hàng
- Hướng dẫn sử dụng tài khoản
- Hỗ trợ kỹ thuật sau bán hàng

4. Quản lý hóa đơn:

- Tạo hóa đơn điện tử
- Lưu trữ lịch sử đơn hàng

5. Xử lý vấn đề và hoàn tiền:

- Báo cáo vấn đề với đơn hàng
- Yêu cầu hoàn tiền/đổi tài khoản
- Xử lý tranh chấp
- Chính sách bảo hành và đảm bảo
- Đánh giá mức độ hài lòng

State Diagram để thể hiện các trạng thái khác nhau của đơn hàng và cách chuyển đổi giữa các trạng thái

2.2 Yêu Cầu Phi Chức Năng

2.2.1 Bảo mật thông tin tài khoản và dữ liệu người dùng

1. Bảo mật dữ liệu:

- Mã hóa thông tin đăng nhập và mật khẩu (cả lưu trữ và truyền tải)
- Kiểm soát truy cập dữ liệu theo vai trò

2. Xác thực và ủy quyền:

- JWT (JSON Web Tokens) cho xác thực API
- Chính sách mật khẩu mạnh
- Quản lý phiên đăng nhập an toàn

3. Phòng chống tấn công:

- Bảo vệ chống SQL Injection
- Bảo vệ chống Cross-Site Scripting (XSS)

4. Kiểm soát và giám sát:

- Ghi nhật ký (logging) toàn diện các hoạt động

5. Quản lý sự cố:

- Thông báo vi phạm cho người dùng bị ảnh hưởng
- Quy trình đánh giá và cải thiện bảo mật

2.2.2 Trải nghiệm người dùng và giao diện

Giao diện người dùng cần được thiết kế trực quan, thân thiện và hiệu quả để tạo trải nghiệm tích cực cho người dùng:

1. Thiết kế giao diện:

- Giao diện hiện đại, sạch sẽ, dễ sử dụng
- Thiết kế responsive cho mọi thiết bị
- Bảng màu và typography nhất quán
- Tương thích đa trình duyệt

2. Trải nghiệm người dùng:

- Điều hướng trực quan và dễ dàng
- Số lượng click tối thiểu để hoàn thành tác vụ

- Thông báo phản hồi rõ ràng cho người dùng
- Hỗ trợ các mẫu UI/UX phổ biến
- Trải nghiệm mượt mà, không bị gián đoạn

3. **Khả năng tiếp cận:**

- Độ tương phản màu sắc phù hợp
- Hỗ trợ điều hướng bằng bàn phím
- Văn bản thay thế cho hình ảnh

4. **Tương tác và phản hồi:**

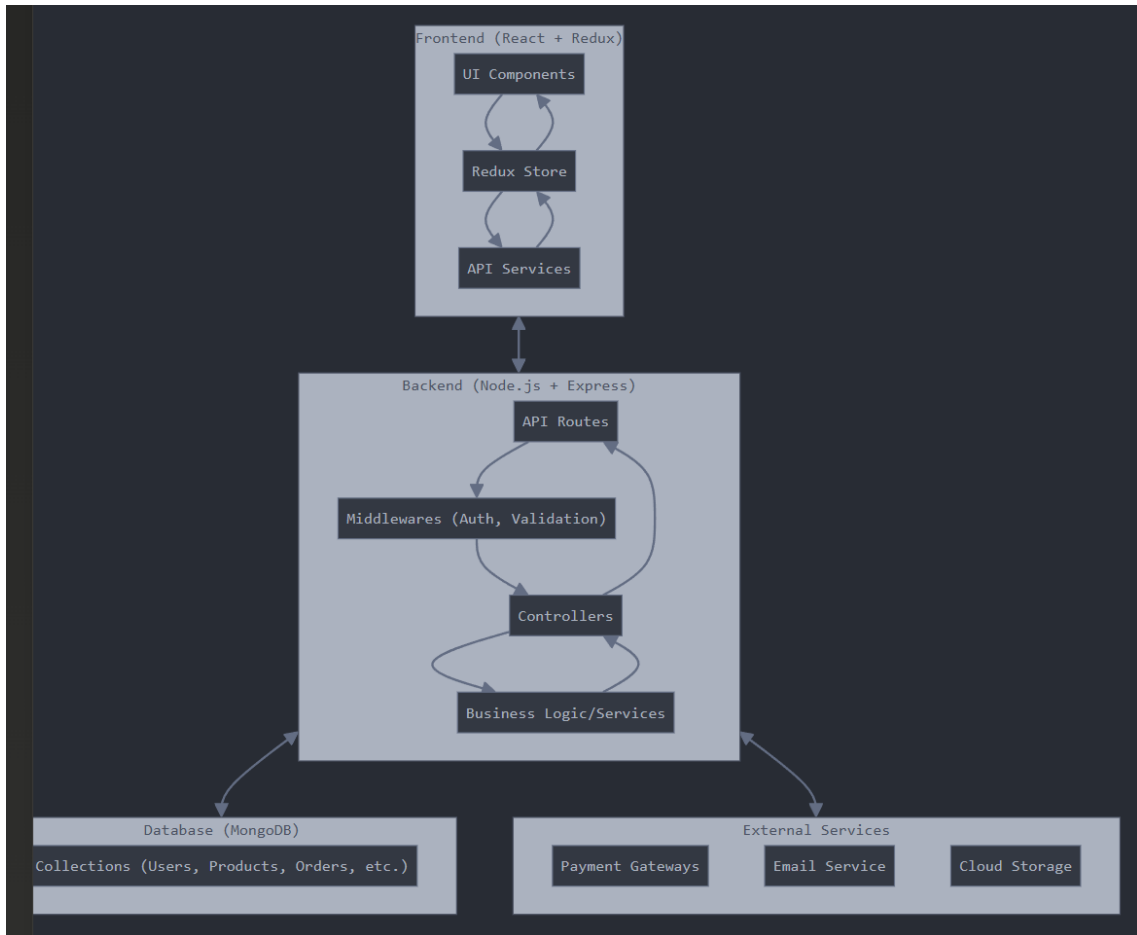
- Animation và chuyển động tinh tế
- Hiệu ứng loading thích hợp
- Thông báo lỗi và thành công rõ ràng
- Form validation trực quan

CHƯƠNG 3. THIẾT KẾ HỆ THỐNG

3.1 Kiến trúc hệ thống

3.1.1 Sơ đồ kiến trúc tổng thể MERN Stack

Kiến trúc MERN Stack trong hệ thống thương mại điện tử bán tài khoản số và key bản quyền được tổ chức theo mô hình phân lớp như sau:



Hình ảnh 3: Minh họa kiến trúc của một dự án sử dụng MERN stack

Kiến trúc MERN Stack của hệ thống được thiết kế để phân tách rõ ràng giữa các thành phần, giúp dễ dàng phát triển, bảo trì và mở rộng. Mỗi thành phần có vai trò riêng biệt:

1. Frontend (React + Redux):

- UI Components: Thành phần giao diện người dùng, hiển thị dữ liệu và tương tác với người dùng
- Redux Store: Quản lý trạng thái toàn cục của ứng dụng
- API Services: Các service gọi API đến backend

2. Backend (Node.js + Express):

- API Routes: Định nghĩa các endpoint của API
- Middlewares: Xử lý xác thực, phân quyền, và validation trước khi request đến controller
- Controllers: Xử lý logic điều khiển, nhận request và trả về response
- Services: Chứa logic nghiệp vụ phức tạp

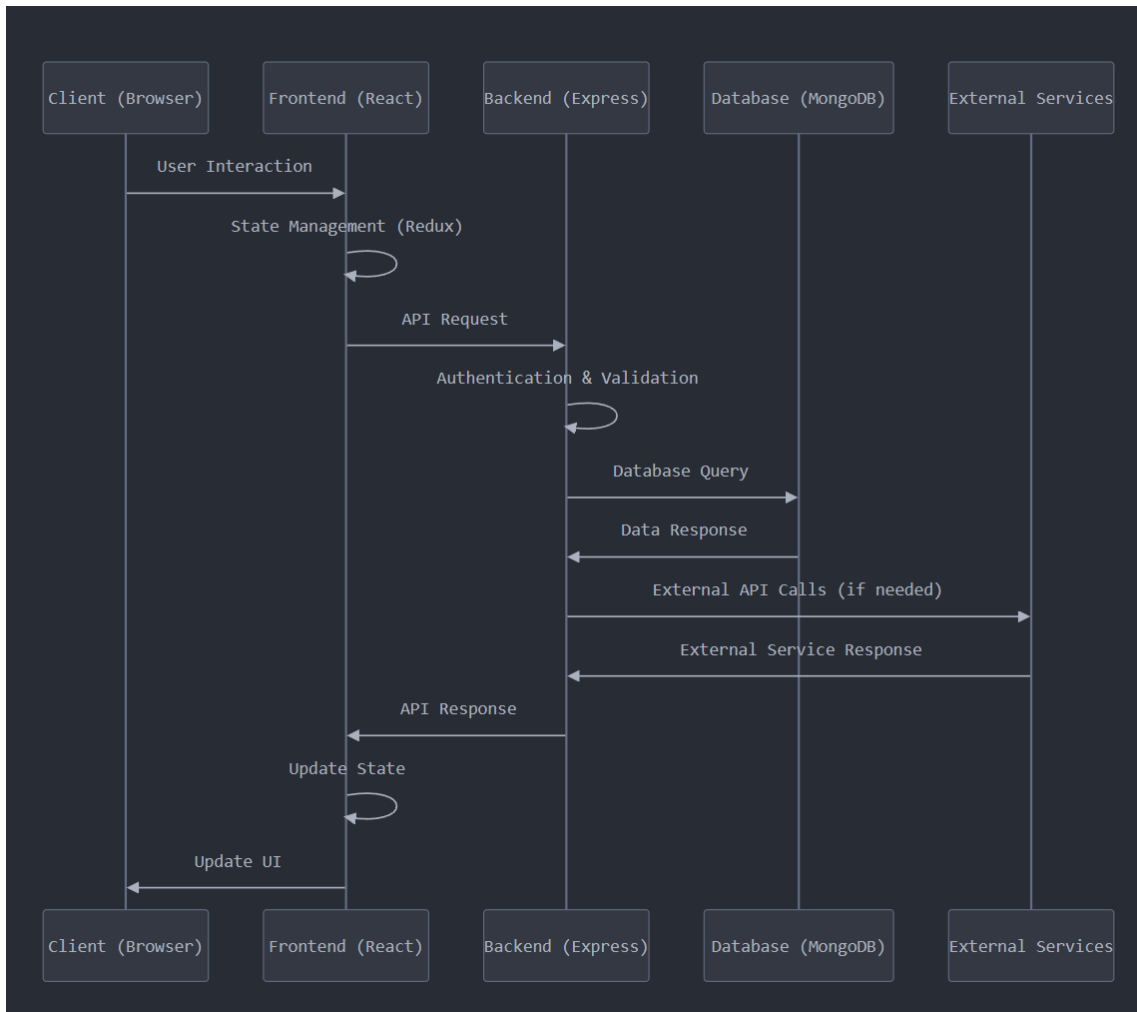
3. Database (MongoDB):

- Lưu trữ dữ liệu dưới dạng các collection như Users, Products, Orders...

4. External Services:

- Payment Gateways: Cổng thanh toán như Stripe, PayPal
- Email Service: Dịch vụ gửi email xác nhận, thông báo
- Cloud Storage: Lưu trữ hình ảnh và tài liệu

3.1.2 Mô hình client-server



Hình ảnh 4: Minh họa mô hình Client-Server

Mô hình client-server trong hệ thống được thiết kế để tối ưu hóa việc phân chia trách nhiệm và truyền tải dữ liệu hiệu quả:

1. Tách biệt Frontend và Backend:

- Frontend (React) hoạt động độc lập trên trình duyệt, tập trung vào trải nghiệm người dùng
- Backend (Express) chịu trách nhiệm xử lý logic nghiệp vụ, tương tác với cơ sở dữ liệu và các dịch vụ bên ngoài

2. Giao tiếp qua REST API:

- Frontend và Backend giao tiếp thông qua RESTful API
- Dữ liệu được trao đổi dưới dạng JSON, dễ dàng xử lý ở cả hai phía

3. Stateless Backend:

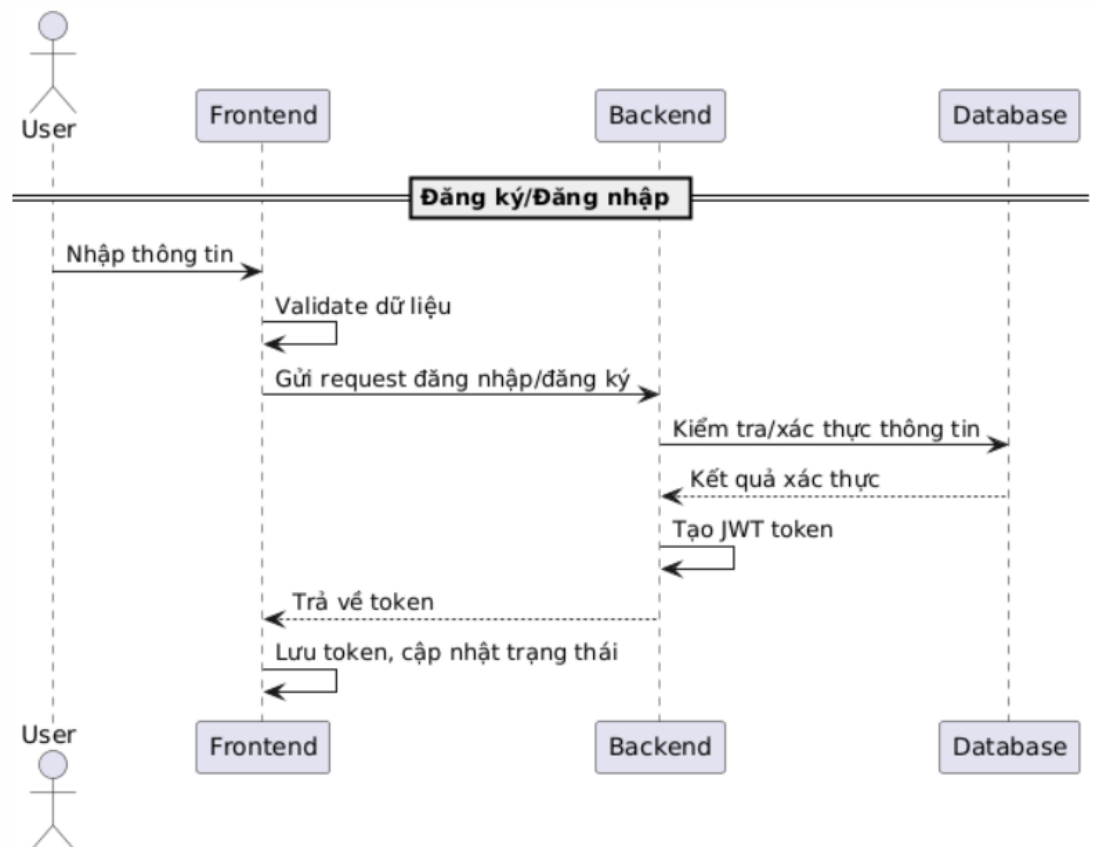
- Backend không lưu trữ trạng thái của phiên làm việc, giúp dễ dàng mở rộng quy mô
- Xác thực người dùng thông qua JWT (JSON Web Tokens)

3.1.3 Luồng dữ liệu và tương tác

Luồng dữ liệu điển hình trong hệ thống bán tài khoản số diễn ra như sau:

1. Đăng ký/Đăng nhập:

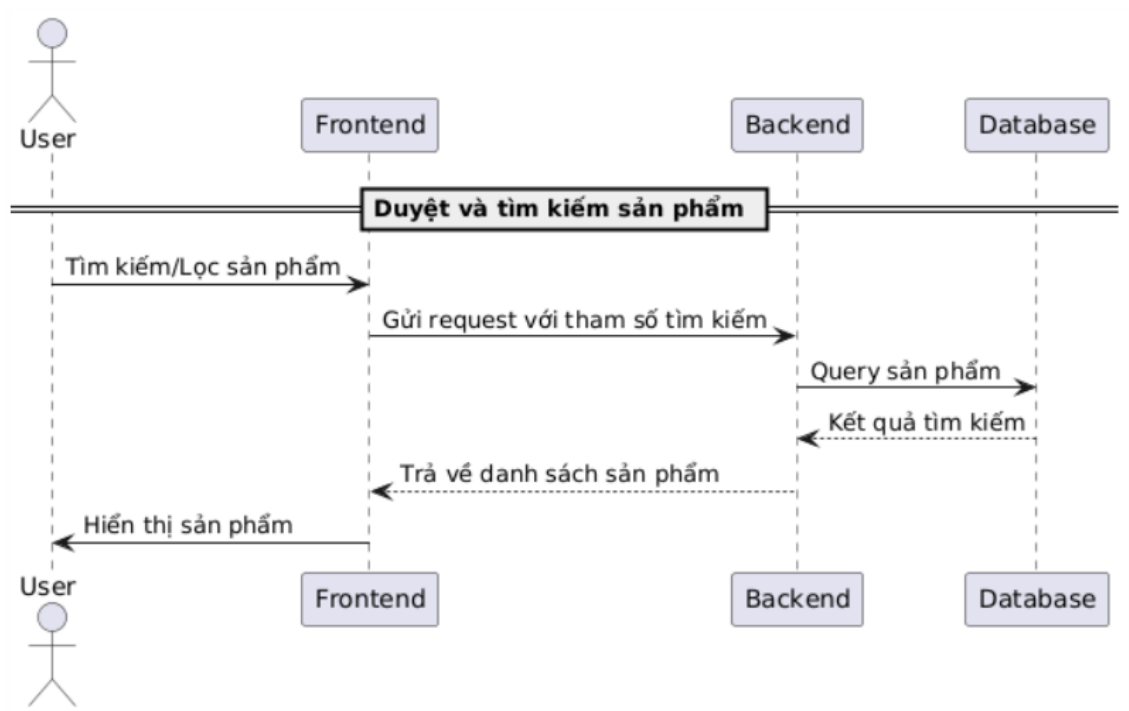
- Người dùng nhập thông tin → Frontend validate → Gửi request đến Backend
- Backend xác thực → Tạo JWT → Trả về token
- Frontend lưu token và cập nhật trạng thái đăng nhập



Hình ảnh 5: Minh họa quy trình tạo tài khoản

2. Duyệt và tìm kiếm sản phẩm:

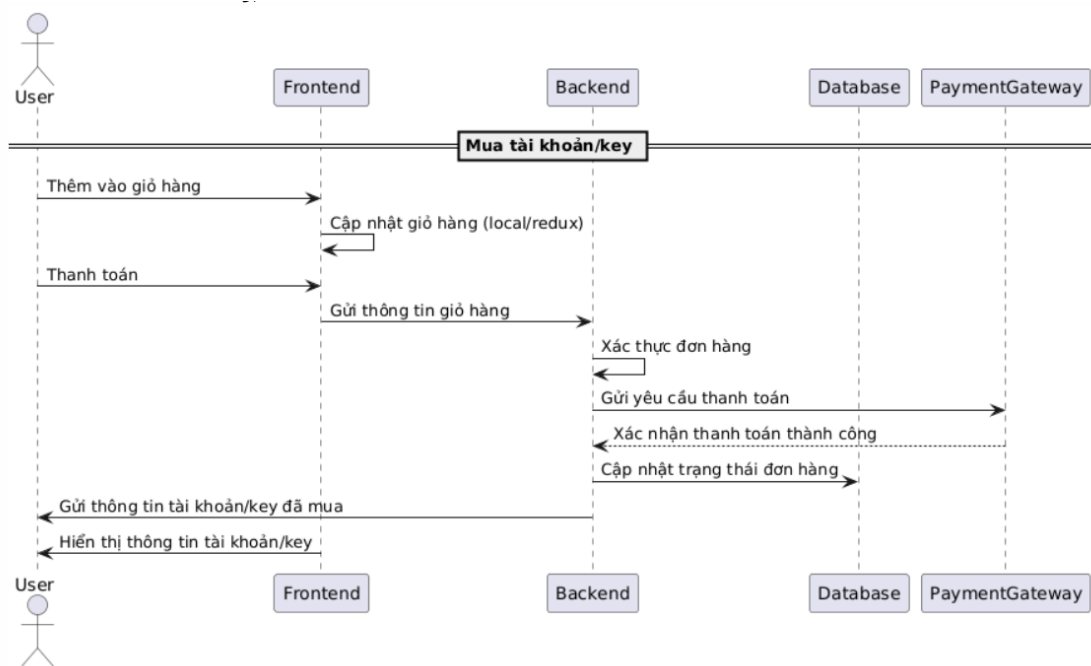
- Người dùng tìm kiếm/lọc → Frontend gửi request với tham số tìm kiếm
- Backend query database → Trả về kết quả → Frontend hiển thị



Hình ảnh 6: Minh họa quy trình duyệt và tìm kiếm sản phẩm

3. Mua tài khoản/key:

- Người dùng thêm vào giỏ hàng → Frontend cập nhật giỏ hàng (lưu local/redux)
- Người dùng thanh toán → Frontend gửi thông tin giỏ hàng đến Backend
- Backend xác thực → Gọi cổng thanh toán → Xử lý thanh toán thành công
- Backend cập nhật trạng thái đơn hàng → Gửi thông tin tài khoản/key đến người dùng
- Frontend hiển thị thông tin tài khoản/key đã mua



Hình ảnh 7: Minh họa quy trình mua hàng

3.2 Thiết kế cơ sở dữ liệu

3.2.1 Mô hình dữ liệu MongoDB (các collection)

Mô hình dữ liệu MongoDB được thiết kế để phù hợp với đặc thù của hệ thống bán tài khoản số và key bản quyền:

1. User Collection:

- Lưu trữ thông tin người dùng bao gồm email, mật khẩu (đã được hash), thông tin cá nhân (username)
- Phân quyền thông qua trường role (admin, user)

2. Product Collection:

- Chứa thông tin cơ bản (tên, mô tả, hình ảnh)
- Kết nối với danh mục và danh mục phụ
- Liên kết với người bán (người dùng có vai trò là người bán)
- Có nhiều SKU (đơn vị lưu kho) cho các biến thể khác nhau

3. SKU Collection:

Xử lý các biến thể cụ thể của sản phẩm:

- Liên kết với sản phẩm gốc
- Theo dõi giá, hàng tồn kho và số liệu bán hàng

- Duy trì ràng buộc duy nhất cho mỗi sản phẩm (không trùng tên SKU)

4. Inventory Collection

Lưu trữ thông tin xác thực kỹ thuật số thực tế:

- Tham chiếu SKU
- Theo dõi trạng thái thông tin xác thực (có sẵn, đã bán, không hợp lệ)
- Liên kết đến đơn hàng khi đã bán
- Liên kết với người bán

5. Category Collection:

- Phân loại sản phẩm (Netflix, Spotify, Windows, Office...)

6. Order Collection:

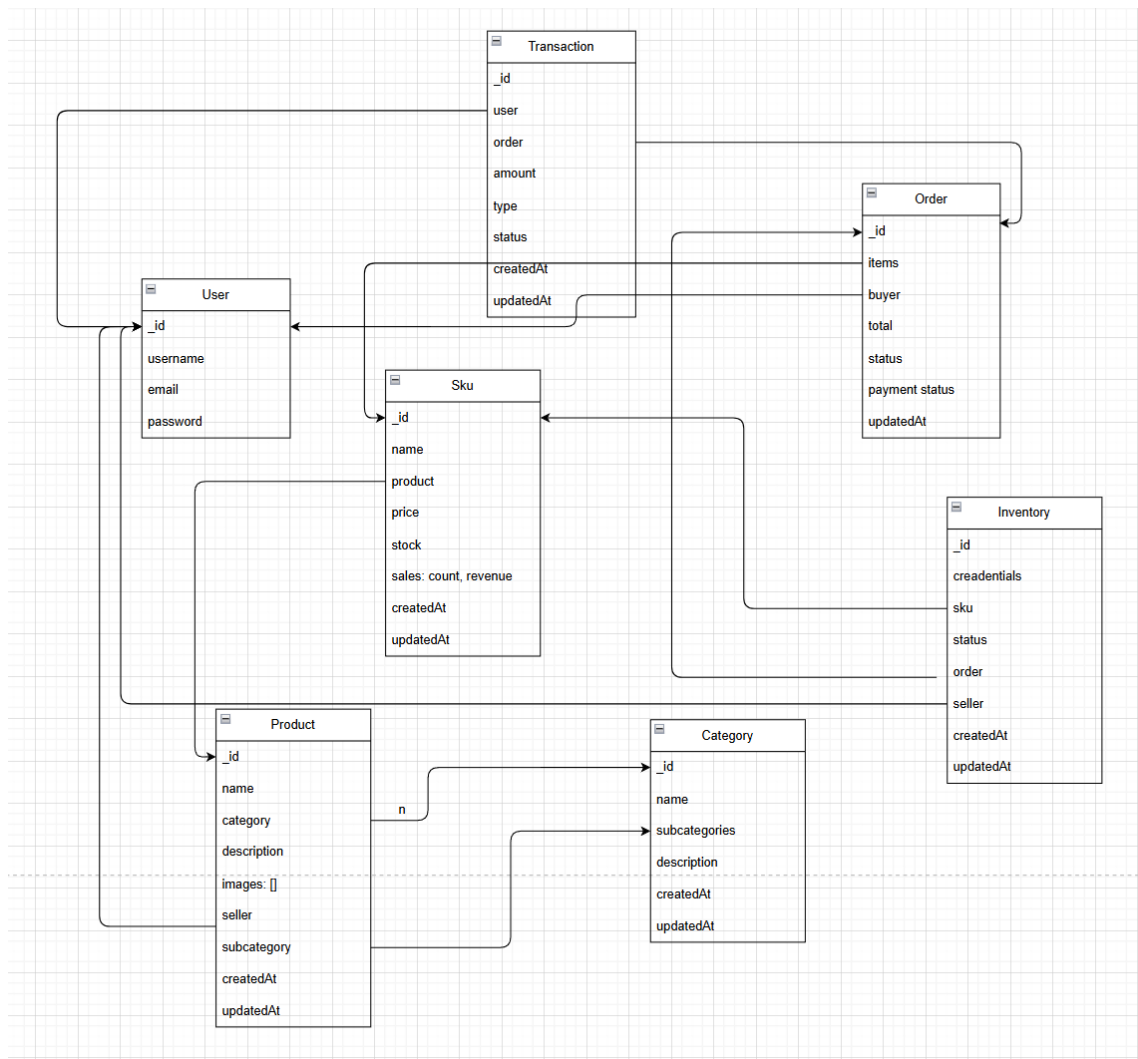
Ghi lại các giao dịch mua hàng:

- Lưu trữ đơn hàng và trạng thái
- items là mảng chứa các sản phẩm trong đơn hàng
- Theo dõi phương thức thanh toán và trạng thái

7. Transaction Collection:

Lưu trữ hồ sơ tài chính:

- Các loại: mua hàng, hoàn tiền, rút tiền, gửi tiền
- Theo dõi trạng thái (đang chờ xử lý, đã hoàn tất, không thành công)
- Tham chiếu đơn hàng và người dùng
- Hỗ trợ xử lý thanh toán qua PayOS



Hình ảnh 8: Mô hình các collection trong MongoDB

3.2.2 Mỗi quan hệ giữa các collection

1. Collection User

Mối quan hệ:

- **User** → **Product**: Quan hệ 1-n (một User có thể tạo nhiều Product nếu là seller)
- **User** → **Order**: Quan hệ 1-n (một User mua nhiều Order)
- **User** → **Transaction**: Quan hệ 1-n (một User có nhiều Transaction)
- **User** → **Inventory**: Quan hệ 1-n (một User quản lý nhiều Inventory nếu là seller)

2. Collection Product

- **Product** → **User**: Quan hệ n-1 (nhiều Product thuộc về một User/seller)
- **Product** → **Category**: Quan hệ n-1 (nhiều Product thuộc một Category)

- **Product** → **SKU**: Quan hệ 1-n (một Product có nhiều SKU)

3. Collection SKU (Stock Keeping Unit)

Mối quan hệ:

- **SKU** → **Product**: Quan hệ n-1 (nhiều SKU thuộc về một Product)
- **SKU** → **Inventory**: Quan hệ 1-n (một SKU có thể có nhiều item Inventory)
- **SKU** → **Order**: Quan hệ n-n (nhiều SKU có thể xuất hiện trong nhiều Order)

4. Collection Inventory

Mối quan hệ:

- **Inventory** → **SKU**: Quan hệ n-1 (nhiều Inventory item thuộc về một SKU)
- **Inventory** → **User**: Quan hệ n-1 (nhiều Inventory thuộc một seller/User)
- **Inventory** → **Order**: Quan hệ n-1 (nhiều Inventory có thể được bán trong một Order)

5. Collection Order

Mối quan hệ:

- **Order** → **User**: Quan hệ n-1 (nhiều Order thuộc về một User/buyer)
- **Order** → **SKU**: Quan hệ n-n (nhiều Order chứa nhiều SKU khác nhau)
- **Order** → **Inventory**: Quan hệ 1-n (một Order có thể bán nhiều Inventory items)
- **Order** → **Transaction**: Quan hệ 1-n (một Order có thể tạo ra nhiều Transaction)

6. Collection Transaction

Mối quan hệ:

- **Transaction** → **User**: Quan hệ n-1 (nhiều Transaction thuộc về một User)
- **Transaction** → **Order**: Quan hệ n-1 (nhiều Transaction có thể liên quan đến một Order)

7. Collection Category

Mối quan hệ:

- **Category** → **Product**: Quan hệ 1-n (một Category có nhiều Product)

Phân tích Môi quan hệ Chi tiết

Môi quan hệ Phức tạp:

1. Quy trình Bán hàng:

User (seller) → tạo Product → tạo nhiều SKU → upload nhiều Inventory items

User (buyer) → tạo Order → chọn nhiều SKU → mua Inventory items → tạo Transaction

2. Quy trình Cập nhật Hàng tồn kho:

Khi Order được tạo → giảm stock của các SKU liên quan

Cập nhật trạng thái Inventory từ 'available' sang 'sold'

Liên kết Inventory với Order thông qua trường order

3. Quy trình Quản lý Tài chính:

User thực hiện giao dịch → tạo Transaction

Transaction liên kết với User và có thể liên kết với Order

Cập nhật balance của User dựa trên Transaction

CHƯƠNG 4. TRIỂN KHAI HỆ THỐNG

4.1 Backend

4.1.1 Cấu trúc thư mục Node.js và Express

Sử dụng phương án phân tách theo chức năng (feature-based structure), một cách tiếp cận hiện đại và dễ bảo trì:

backend-lab/

```
|— src/
|   |— config/      # Cấu hình ứng dụng (database, JWT, etc.)
|   |— models/      # Mongoose schemas
|   |— controllers/  # Xử lý logic nghiệp vụ
|   |— routes/       # Định tuyến API endpoints
|   |— middlewares/  # Middleware xác thực và phân quyền
|   |— utils/        # Các hàm tiện ích
|   └— server.js     # Điểm khởi đầu ứng dụng
```

Cấu trúc thư mục được thiết kế theo nguyên tắc phân tách trách nhiệm rõ ràng:

1. **Controllers:** Xử lý HTTP requests và gửi responses, gọi đến services để thực hiện logic nghiệp vụ
2. **Models:** Định nghĩa cấu trúc dữ liệu và tương tác với MongoDB
3. **Routes:** Định nghĩa các API endpoints và kết nối chúng với controllers
4. **Services:** Chứa logic nghiệp vụ phức tạp, tách biệt khỏi controllers
5. **Middleware:** Xử lý các tác vụ như xác thực, xử lý lỗi trước khi đến controllers
6. **Config:** Lưu trữ cấu hình ứng dụng, kết nối cơ sở dữ liệu
7. **Utils:** Các hàm tiện ích sử dụng trong toàn bộ ứng dụng

Cấu trúc này đạt được sự phân tách mối quan tâm (separation of concerns) rõ ràng, giúp code dễ đọc, dễ mở rộng và dễ thay đổi.

4.1.2 Models (MongoDB Schemas)

Sử dụng Mongoose làm ODM (Object Document Mapper) với các schema được thiết kế tỉ mỉ:

Các Schema Chính:

1. User Schema

```
const userSchema = new mongoose.Schema({
  email: { type: String, required: true, unique: true, trim: true },
  password: { type: String, required: true },
  role: { type: String, enum: ['user', 'seller', 'admin'], default: 'user' },
  username: { type: String, required: true, unique: true },
  balance: { type: Number, default: 0 },
  status: { type: String, enum: ['active', 'suspended', 'deleted'], default: 'active' }
}, { timestamps: true });
```

- Hỗ trợ phân quyền 3 cấp: người dùng, người bán, quản trị viên
- Quản lý số dư tài khoản cho thanh toán nội bộ
- Quản lý trạng thái tài khoản (active, suspended, deleted)

2. Product Schema

- Lưu thông tin sản phẩm: tên, hình ảnh, mô tả
- Liên kết với danh mục, người bán
- Có index trên trường seller để tối ưu truy vấn
- Hàm middleware pre('save') để xác thực danh mục và danh mục con

3. SKU Schema:

- Quản lý các biến thể của sản phẩm
- Theo dõi số lượng, giá cả và thống kê bán hàng
- Có ràng buộc unique trên cặp (product, name)
- Phương thức tính để lấy thống kê bán hàng

4. Order Schema

- Lưu thông tin đơn hàng, các sản phẩm được mua
- Quản lý trạng thái đơn hàng và phương thức thanh toán
- Có các index tối ưu cho buyer và items.sku

5. Inventory Schema

- Lưu trữ thông tin tài khoản số (credentials) của các sản phẩm
- Theo dõi trạng thái: available, sold, invalid
- Liên kết với SKU, Order và người bán

6. Transaction Schema

- Ghi lại các giao dịch tài chính: mua hàng, hoàn tiền, nạp/rút tiền
- Theo dõi trạng thái giao dịch
- Hỗ trợ metadata cho các thông tin bổ sung

4.1.3 Controllers & Routes

Controllers được tổ chức theo chức năng, xử lý logic nghiệp vụ và tương tác với database:

1. Auth Controller:

- Xử lý đăng ký, đăng nhập, làm mới token
- Sử dụng bcrypt để mã hóa mật khẩu
- Tạo và quản lý JWT tokens

2. Product Controller:

- CRUD operations cho sản phẩm
- Xử lý upload hình ảnh (base64)
- Quản lý SKUs liên quan

3. Order Controller:

- Tạo đơn hàng mới
- Xử lý logic thanh toán
- Cấp tài khoản số cho người mua

4. Seller Controller:

- Quản lý sản phẩm của người bán
- Xem thống kê bán hàng
- Xử lý yêu cầu rút tiền

5. Admin Controller:

- Quản lý người dùng
- Xem thống kê hệ thống
- Phê duyệt giao dịch rút tiền

4.1.4 Middleware

1. Auth Middleware:

```

export const verifyToken = (req, res, next) => {
  try {
    const token = req.headers.authorization?.split(' ')[1];
    if (!token) {
      return res.status(403).json({
        message: 'No token provided',
        code: 'TOKEN_MISSING'
      });
    }
    const decoded = jwt.verify(token, JWT_ACCESS_SECRET);
    req.user = decoded;
    next();
  } catch (error) {
    return res.status(401).json({
      message: 'Unauthorized',
      code: 'UNAUTHORIZED'
    });
  }
};

```

- Xác thực JWT token
- Thêm thông tin người dùng vào req.user cho các middleware/controller tiếp theo xử lý

2. Role Middleware:

```

export const requireRole = (roles) => {
  return (req, res, next) => {
    if (!roles.includes(req.user.role)) {
      return res.status(403).json({ message: 'Permission denied' });
    }
    next();
  };
};

```

- Kiểm tra quyền truy cập dựa trên vai trò người dùng
- Sử dụng closure pattern để tạo middleware có thể cấu hình

3. Các Middleware Khác

- cors để xử lý Cross-Origin Resource Sharing

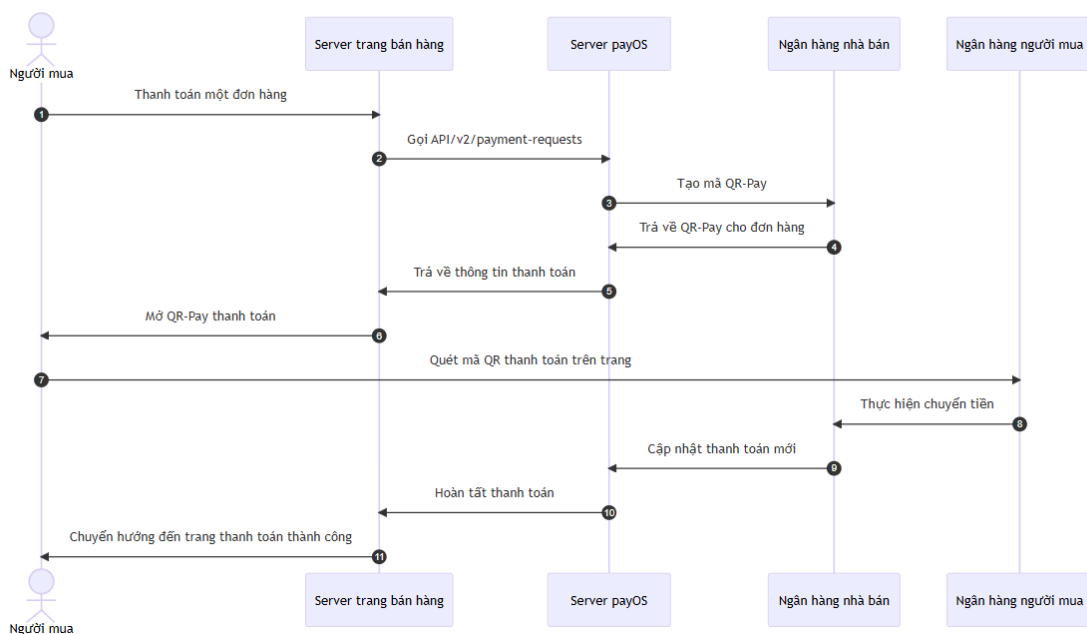
- `express.json` và `express.urlencoded` để parse request body
- `cookieParser` để xử lý cookies

4.1.5 Tích hợp cổng thanh toán PayOS

Dự án tích hợp PayOS để xử lý nạp tiền giúp bảo đảm tính bảo mật, nhanh chóng và thuận tiện cho người dùng.

Quá trình thanh toán bao gồm:

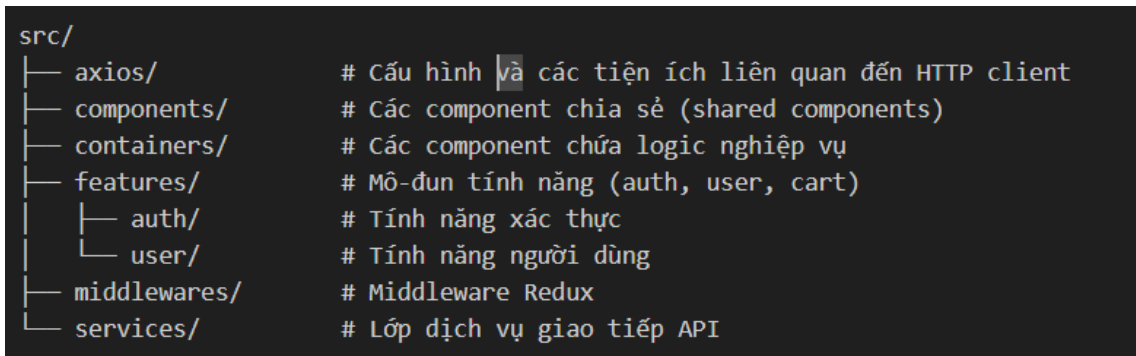
1. Tạo payment link thông qua PayOS API
2. Lưu thông tin transaction trong hệ thống
3. Xử lý callback từ cổng thanh toán
4. Cập nhật trạng thái đơn hàng và số dư người dùng



Hình ảnh 9: Quy trình hoạt động với PayOS

4.2 Frontend Development

4.2.1 Cấu trúc thư mục React



Hình ảnh 10: Cấu trúc thư mục

- Tách biệt rõ ràng giữa logic UI và logic nghiệp vụ
- Dễ dàng mở rộng khi thêm tính năng mới
- Mô-đun hóa giúp nhiều developer làm việc đồng thời trên các phần khác nhau

4.2.2 Phân cấp component

1. Container Components: Quản lý logic nghiệp vụ và trạng thái (App.js, SellerDashboard.js)
2. Presentational Components: Hiển thị UI và xử lý tương tác người dùng (ProductCard, UserHeader)
3. Modal Components: Xử lý các tương tác cụ thể (NewProductModal)

Ưu điểm:

- Tái sử dụng component tốt hơn
- Dễ dàng bảo trì và kiểm thử
- Giảm sự phụ thuộc giữa các phần khác nhau của ứng dụng

4.2.3 Quản lý trạng thái với Redux

Cách hoạt động:

1. **Action Creators:** Sử dụng createAsyncThunk để xử lý các thao tác bất đồng bộ như gọi API
2. **Reducers:** Định nghĩa trong createSlice để xử lý các thay đổi trạng thái
3. **Middleware:** authMiddleware xử lý các token xác thực và quản lý phiên đăng nhập

Luồng xác thực:

- Khi đăng nhập thành công, token được lưu trữ và sử dụng cho các yêu cầu API
- refreshToken tự động cập nhật token khi hết hạn
- Quản lý logout và xóa thông tin người dùng

Ưu điểm:

- Quản lý trạng thái tập trung, dễ debug
- Middleware linh hoạt cho các tác vụ phức tạp
- Redux Toolkit giảm boilerplate code

4.2.4 UI/UX Design

Dự án sử dụng nhiều kỹ thuật để xây dựng giao diện người dùng:

1. **CSS/SCSS tùy chỉnh:** Mỗi component có style riêng
2. **Thư viện UI components:**
 - React Slick cho các carousel
 - Thông báo Toast cho phản hồi người dùng
3. **Material Symbols:** Cho các biểu tượng và yếu tố trực quan

Ưu điểm:

- Giao diện nhất quán trên toàn ứng dụng
- Mô-đun hóa CSS giúp tránh xung đột style
- Tích hợp linh hoạt với các thư viện UI khác nhau

4.3 Tích hợp và kiểm thử

4.3.1 Unit Tests

Các đơn vị kiểm thử cần tập trung vào:

- **Redux reducers và actions:** Đảm bảo các actions như [login](#), [register](#), [logout](#) hoạt động đúng
- **Các hàm tiện ích:** Như các hàm xử lý và chuyển đổi dữ liệu
- **Các components riêng lẻ:** Kiểm thử rendering và xử lý sự kiện

4.3.2 Integration Tests

Kiểm thử tích hợp tập trung vào:

- **Luồng xác thực hoàn chỉnh:** Đăng ký → Đăng nhập → Đăng xuất
- **Giao tiếp giữa Redux và React components:** Đảm bảo UI phản hồi đúng khi trạng thái thay đổi
- **Middleware hoạt động:** Kiểm tra authMiddleware xử lý token đúng cách

4.3.3 Quy trình tích hợp

Để đảm bảo chất lượng và sự ổn định của hệ thống, chúng ta nên thiết lập quy trình tích hợp liên tục với:

- **GitHub Actions/Jenkins:** Tự động hóa việc kiểm thử, xây dựng và triển khai khi có thay đổi trong mã nguồn
- **Quy trình làm việc:**
 - Phát triển trên nhánh tính năng
 - Tạo Pull Request
 - Tự động chạy kiểm thử
 - Code Review
 - Gộp vào nhánh chính khi đạt yêu cầu

4.3.4 Tích hợp Front-end với Back-end

Trong [authService.js](#) và [axios.js](#):

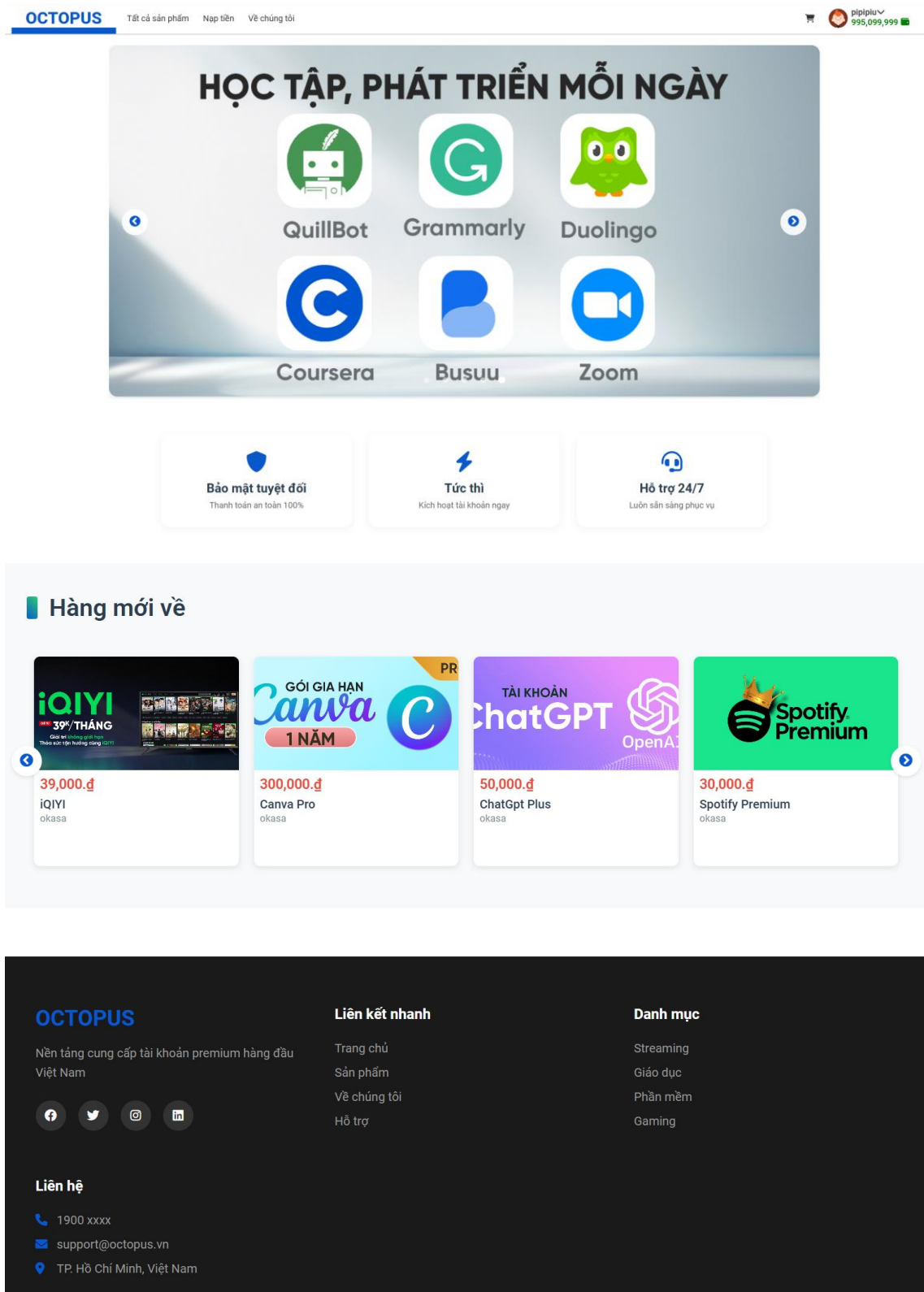
- **Tách biệt lớp dịch vụ:** Tất cả API calls được đóng gói trong các service riêng biệt
- **Xử lý token và refresh:** Hệ thống đã có cơ chế tự động làm mới token khi hết hạn
- **Middleware xử lý authentication:** [authMiddleware.js](#) quản lý token sau mỗi thao tác đăng nhập/dăng xuất

Chiến lược tích hợp này rất tốt vì:

1. Dễ dàng thay đổi API endpoint khi cần
2. Xử lý lỗi tập trung
3. Tách biệt logic gọi API và UI logic

CHƯƠNG 5. CÁC CHỨC NĂNG CHÍNH

5.1 Giao diện trang chủ



5.2 Đăng ký

OCTOPUS

Tất cả sản phẩm

Nạp tiền

Về chúng tôi

Đăng nhập

Đăng ký tài khoản

Nhập thông tin tài khoản

Email

Mật khẩu

Xác nhận mật khẩu

Username

Đăng ký tài khoản

[Đã có tài khoản? Vui lòng Đăng nhập](#)

5.3 Đăng nhập

OCTOPUS

Tất cả sản phẩm

Nạp tiền

Về chúng tôi

Đăng nhập

Xin chào

Vui lòng nhập thông tin của bạn:

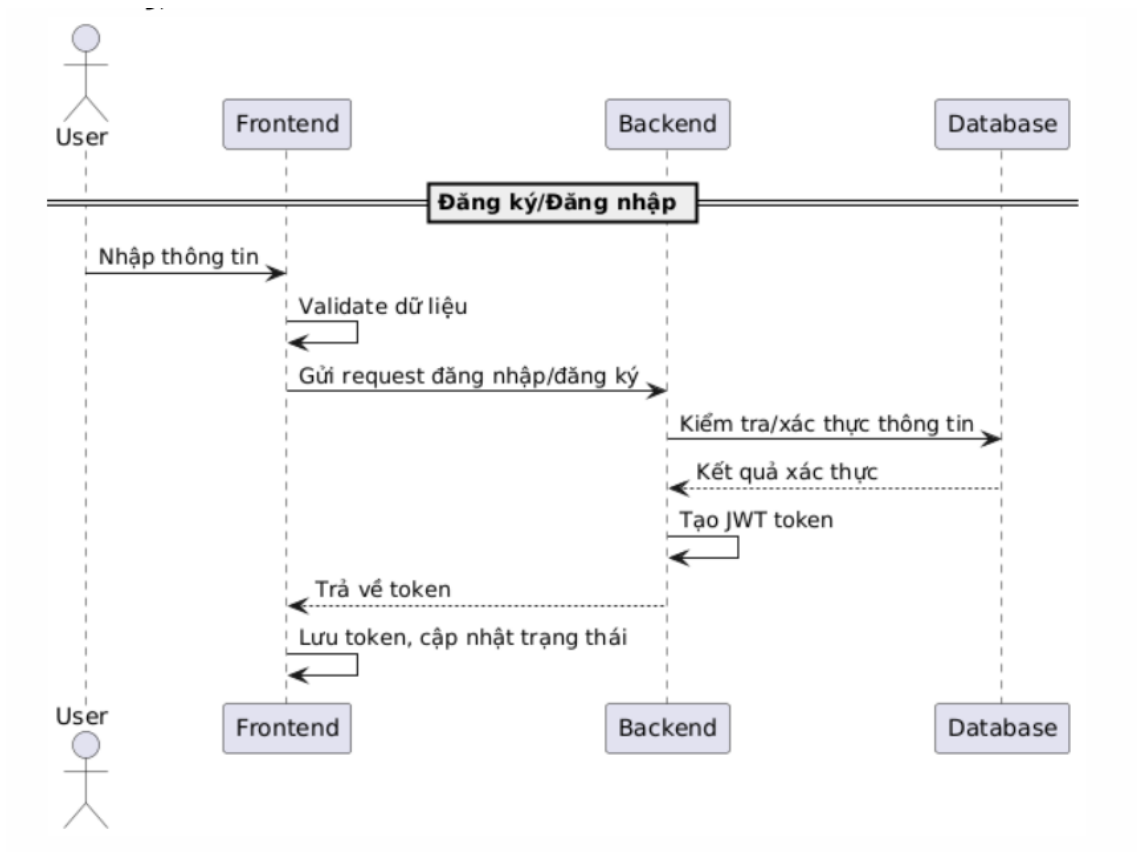
Tài khoản

Mật khẩu

[Quên mật khẩu](#)

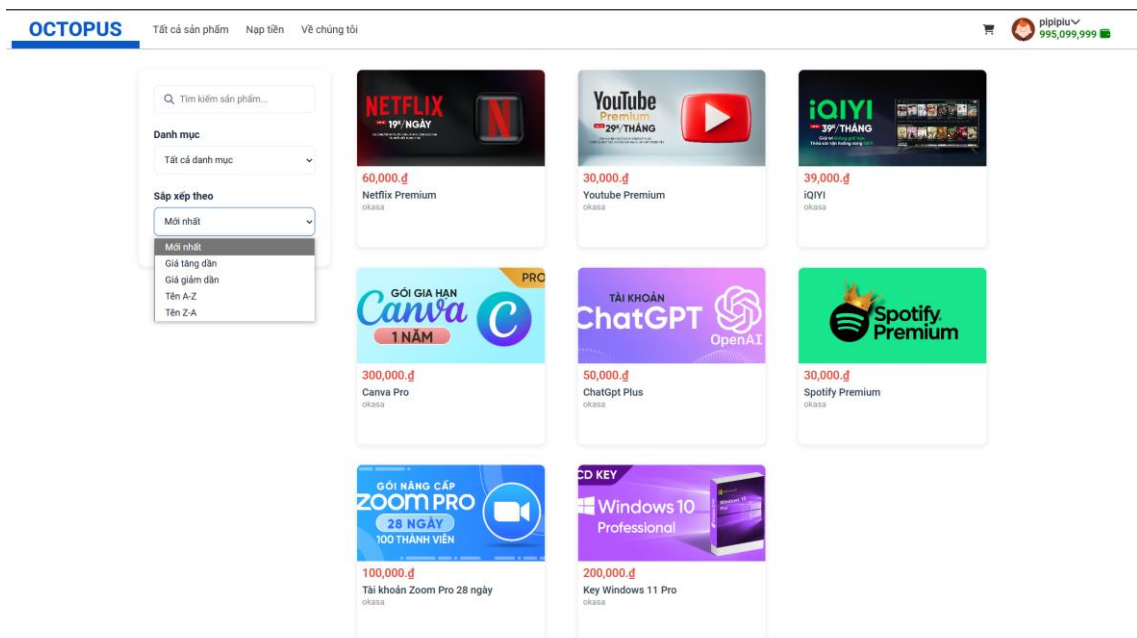
Đăng nhập

[Không có tài khoản? Đăng ký ngay](#)



Sơ đồ chức năng đăng nhập

5.4 Xem sản phẩm



5.5 Xem thông tin chi tiết về sản phẩm



Netflix Premium

Cung cấp bởi **okasa**

60,000 đ

Chọn gói:

1080 - 1 Tháng - 60,000đ

Số lượng:

— +

☒ Hết hàng

Thêm vào giỏ hàng

Mua ngay



Bảo mật thanh toán
An toàn tuyệt đối



Giao dịch tức thì
Nhận hàng ngay lập tức



Hỗ trợ 24/7
Luôn sẵn sàng hỗ trợ

Mô tả sản phẩm

Tài khoản Netflix Premium có rất nhiều tính năng hấp dẫn như:

- Xem phim chất lượng 4k, chất lượng đường truyền nhanh.
- Hỗ trợ xem trên các thiết bị PC/TV/Mobile.
- Tải phim để có thể xem ngoại tuyến.
- Nội dung đa dạng, xem phim đầy đủ phụ đề tiếng Việt.

Đây là tài khoản tạo sẵn và được chia làm nhiều User, mỗi khách hàng sẽ được cấp 1 User riêng trong tài khoản để sử dụng.


Sau khi mua hàng, bạn sẽ nhận được ngay thông tin tài khoản đăng nhập

5.6 Xem giỏ hàng

Giỏ hàng

☐ Chọn tất cả

☐



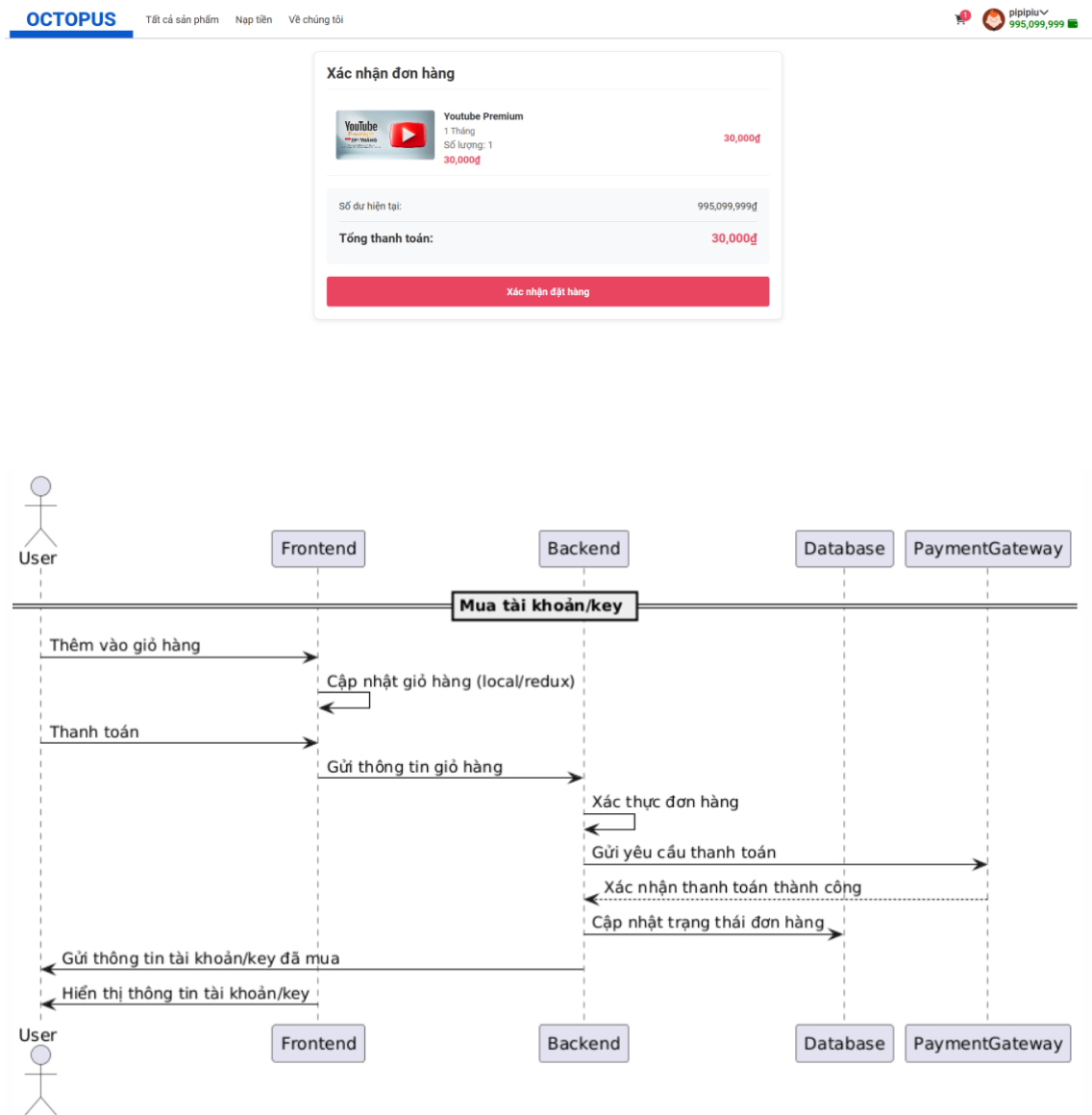
Youtube Premium
1 Tháng
30,000đ

1 x

Tổng cộng (0 sản phẩm)
0đ

Thanh toán (0)

5.7 Thanh toán



5.8 Xem đơn thông tin các đơn hàng

Đơn hàng của tôi

Tổng số đơn hàng: 15 Trang 1/2

- Tất cả
- Hoàn thành
- Đang xử lý
- Đã hủy

#67ca8b4f09fcb4954e147e59	12:59 07/03/2025	Hoàn thành
Netflix Premium	1080 - 1 Tháng × 1	60,000đ
Tổng tiền: 60,000đ	Xem chi tiết	

#67b5afc5750afe0cef48b3d3	17:17 19/02/2025	Hoàn thành
Netflix Premium	4k - 1 Tháng × 1	100,000đ
Tổng tiền: 100,000đ	Xem chi tiết	

Chi tiết đơn hàng #67ca8b4f09fcb4954e147e59

Hoàn thành

12:59 07/03/2025 #67ca8b4f09fcb4954e147e61

Thông tin thanh toán

Tổng tiền

60,000đ

Trạng thái thanh toán

Hoàn thành

Netflix Premium

1080 - 1 Tháng

60,000đ

Tài khoản


katrinabl3q2@outlook.com

Mật khẩu

.....

5.9 Xem trang cá nhân

Thông tin tài khoản



Đổi ảnh đại diện

active

Tên người dùng

pipipiu

Email

pi@bmail.com

Số dư tài khoản

995,099,999đ

Vai trò

User

Ngày tham gia

11/01/2025

5.10 Nạp tiền

Nạp tiền vào tài khoản

Thanh toán an toàn qua PayOS

Số dư hiện tại

995.039.999 đ

Số tiền nạp tối thiểu

10.000đ

Số tiền muốn nạp

0

VND

50.000đ

100.000đ

200.000đ

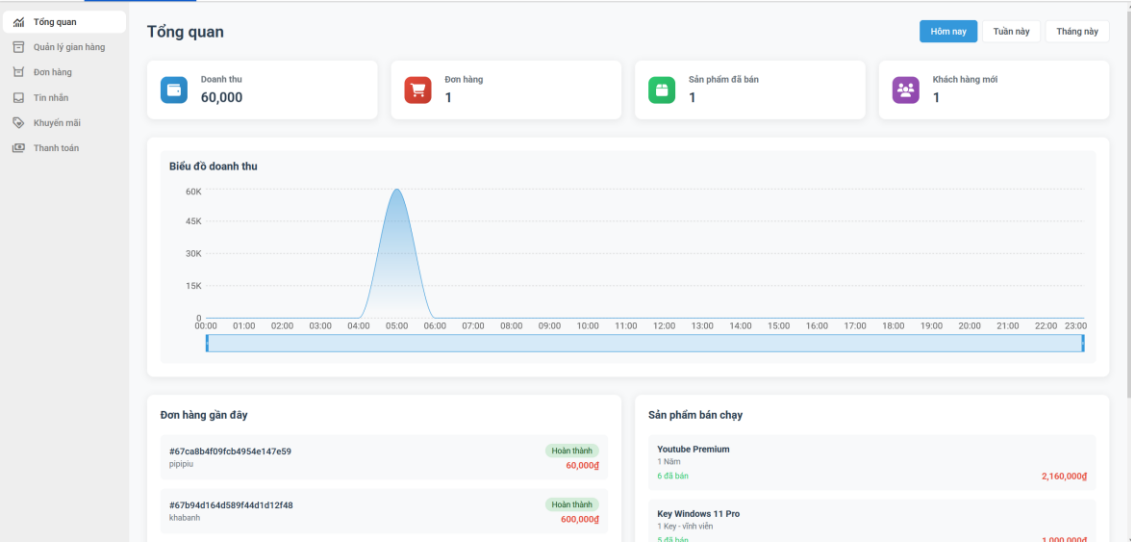
500.000đ



Nạp tiền ngay

5.11 Chức năng quản lý cho Seller

- Quản lý danh sách tài khoản cần bán
- Thêm/sửa/xóa tài khoản
- Xem lịch sử bán hàng
- Quản lý đơn hàng từ người mua
- Rút tiền từ doanh thu



Sản phẩm

+ Thêm sản phẩm

HÌNH ẢNH	TÊN SẢN PHẨM	DOANH SỐ	DOANH THU	KHO HÀNG	THAO TÁC
	Netflix Premium Tài khoản Netflix Premium có rất nhiều tính năng hấp dẫn như - Xem phim ch...	7	500,000 đ	1	
	Youtube Premium Với Youtube Premium, bạn có thể xem hàng triệu video mà không bị quảng c...	8	2,220,000 đ	22	
	iQIYI iQIYI Cao cấp là gói dịch vụ cao cấp của nền tảng iQIYI – một trong những đi...	0	0 đ	3	
	Canva Pro Quy trình nhận hàng: Đây là gói gia hạn Canva Pro chính chủ có thời hạn sử ...	1	300,000 đ	2	
	ChatGpt Plus ChatGPT Plus là bản nâng cấp cao cấp cho trình tạo văn bản trí tuệ nhân tạo ...	0	0 đ	4	
	Spotify Premium Nghe nhạc không có quảng cáo. Nghe podcast, nhạc nằm trong mục premium.	1	360,000 đ	18	
	Tài khoản Zoom Pro 28 ngày Bỏ giới hạn, thêm tới đa 100 thành viên	0	0 đ	4	
	Key Windows 11 Pro Kích hoạt windows vĩnh viễn, sử dụng tất cả các tính năng của Windows	5	1,000,000 đ	35	

Quản lý đơn hàng

2

Tất cả Hoàn thành Đang xử lý Đã hoàn tiền Đã báo cáo << < 1 2 > >>

Mã đơn	Ngày đặt	Sản phẩm	Tổng tiền	Trạng thái	Thao tác
#67ca8b4f09fcb4954e147e59	12:59:43 7/3/2025	Netflix Premium (1080 - 1 Tháng x1)	60,000đ	Hoàn thành	
#67b94d164d589f44d1d12f48	11:05:42 22/2/2025	Key Windows 11 Pro (1 Key - vĩnh viễn x3)	600,000đ	Hoàn thành	
#67b8873c151170b8b1e61411	21:01:32 21/2/2025	Youtube Premium (1 Năm x1)	360,000đ	Hoàn thành	
#67b5afc5750afe0cef48b3d3	17:17:41 19/2/2025	Netflix Premium (4k - 1 Tháng x1)	100,000đ	Hoàn thành	
#67b59e64750afe0cef48b36d	16:03:32 19/2/2025	Youtube Premium (1 Năm x1)	360,000đ	Hoàn thành	
#67b4942c67ab8f39c9146024	21:07:40 18/2/2025	Youtube Premium (1 Năm x1)	360,000đ	Hoàn thành	
#67b40e637e53c98ca2b4274d	11:36:51 18/2/2025	Netflix Premium (1080 - 1 Tháng x1)	60,000đ	Hoàn thành	
#67b1570ede4a6e8db611e5a74	10:10:06 16/2/2025	Youtube Premium (1 Năm x1)	360,000đ	Hoàn thành	
#67b0a206c80bf6f989ac93f0	21:17:42 15/2/2025	Youtube Premium (1 Năm x1)	360,000đ	Hoàn thành	

Tổng yêu cầu

3

70,000đ

Đang chờ xử lý

1

10,000đ

Đã hoàn thành

1

50,000đ

Đã từ chối

1

10,000đ

Tạo yêu cầu rút tiền

Tất cả trạng thái

10 / trang

Mã yêu cầu	Số tiền	Trạng thái	Thời gian	Ghi chú
67b98d1b86d3e57e4e732580	50,000đ	Đã hoàn thành	22/2/2025	-
67b98cc20f591ec3911902e2	10,000đ	Đã từ chối	22/2/2025	-
67b98cab0f591ec3911902df	10,000đ	Chờ xử lý	22/2/2025	-

Tạo yêu cầu rút tiền

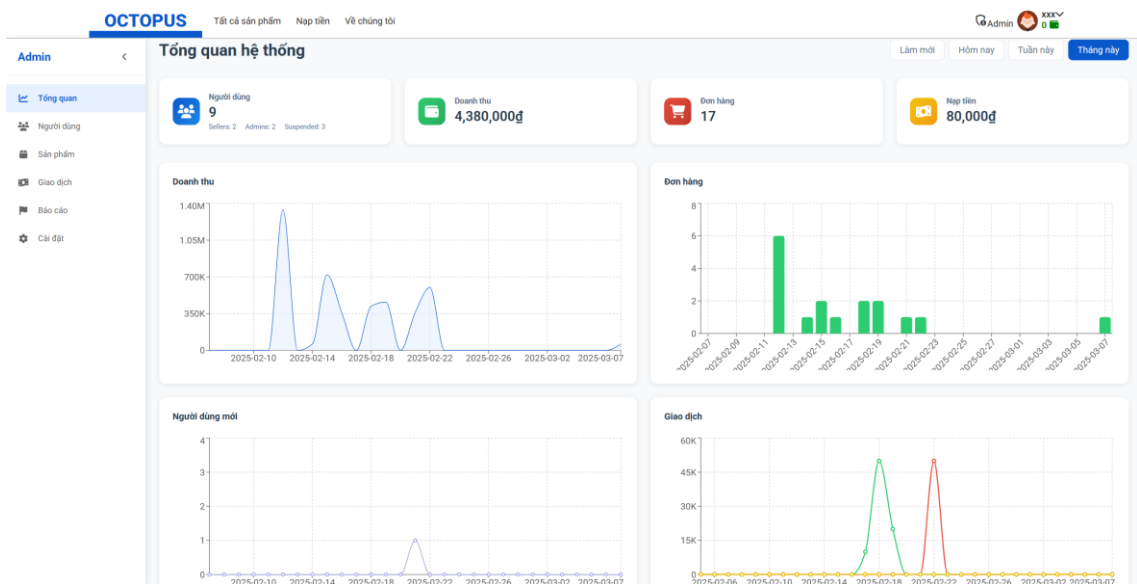
Số tiền muốn rút

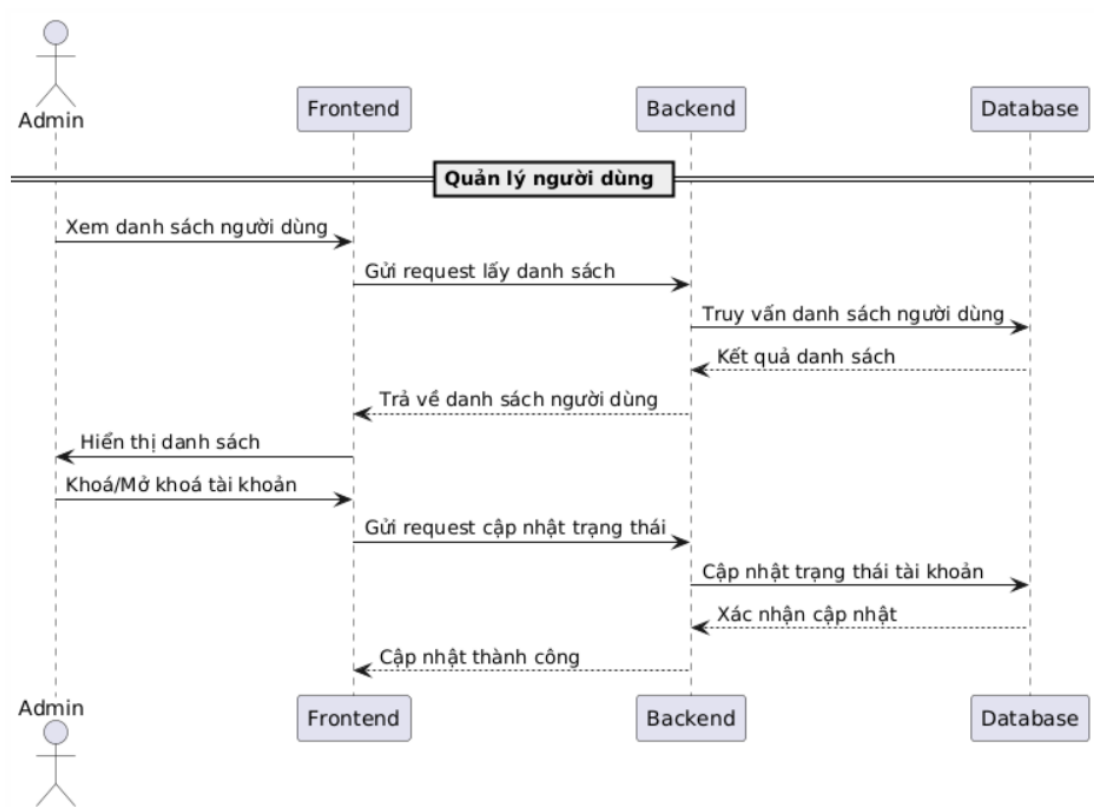
Nhập số tiền...

Xác nhận

Hủy

5.12 Chức năng quản lý của admin





5.12.1 Quản lý người dùng

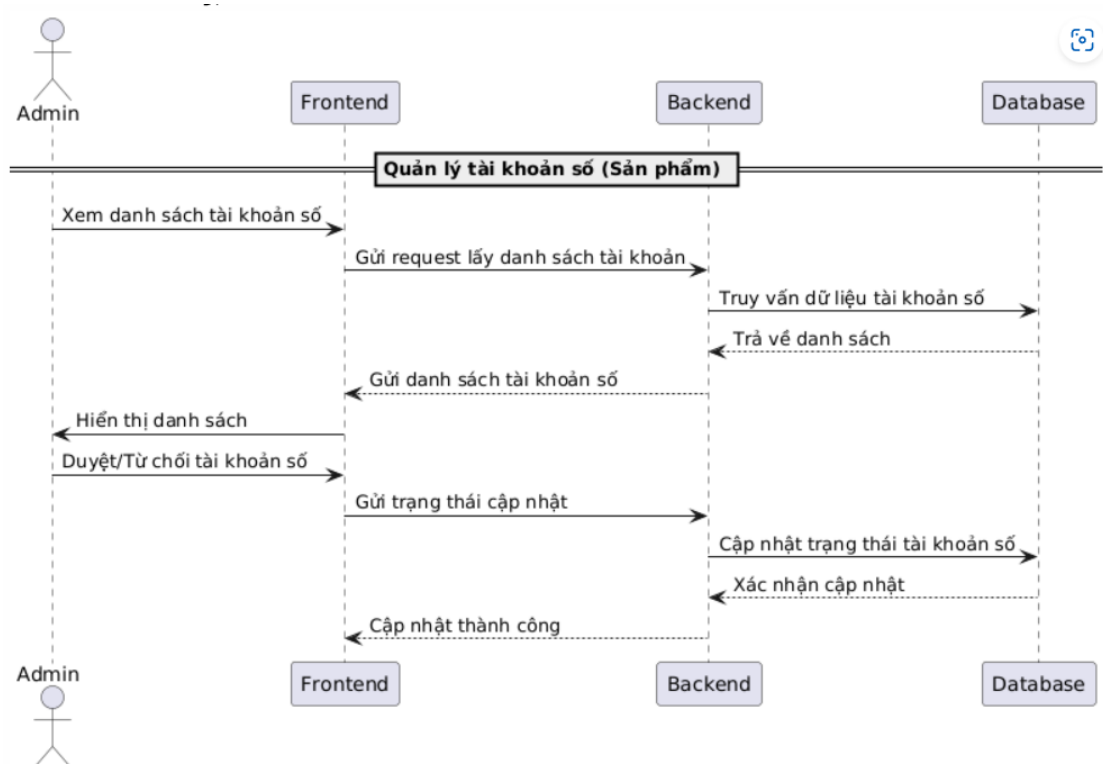
- Xem danh sách tất cả người dùng (Admin, Seller, Buyer).
- Tìm kiếm, lọc người dùng theo vai trò hoặc trạng thái.
- Khoá/Mở khoá tài khoản người dùng khi vi phạm.
- Cập nhật thông tin người dùng (nếu cần).

Username	Email	Vai trò	Trạng thái	Số dư	Tổng chi tiêu	Số đơn hàng	Ngày tạo	Thao tác
olalaa	ooos@gmail.com	User	Đã khoá	0đ	0đ	0	21/2/2025	✎ 🔒
trewtre	motjiji@gmail.com	User	Đã khoá	0đ	0đ	0	11/1/2025	✎ 🔒
mujigng	mujin@gmail.com	Seller	Đã khoá	0đ	0đ	0	11/1/2025	✎ 🔒
mokaka	mokaka@gmail.com	User	Hoạt động	0đ	0đ	0	11/1/2025	✎ 🔒
modji	modi@gmail.com	User	Hoạt động	0đ	0đ	0	11/1/2025	✎ 🔒
ojin	ojin@gmail.com	User	Hoạt động	0đ	0đ	0	11/1/2025	✎ 🔒
motcongmot	motcongmot@gmail.com	User	Hoạt động	0đ	0đ	0	11/1/2025	✎ 🔒
pipiliu	pi@bmail.com	User	Hoạt động	995,039,999đ	3,420,000đ	15	11/1/2025	✎ 🔒
okasa	okasa@gmail.com	Seller	Hoạt động	960,000đ	0đ	0	11/1/2025	✎ 🔒
xxx	test@gmail.com	Admin	Hoạt động	0đ	0đ	0	11/1/2025	✎ 🔒

5.12.2 Quản lý tài khoản số (Sản phẩm)

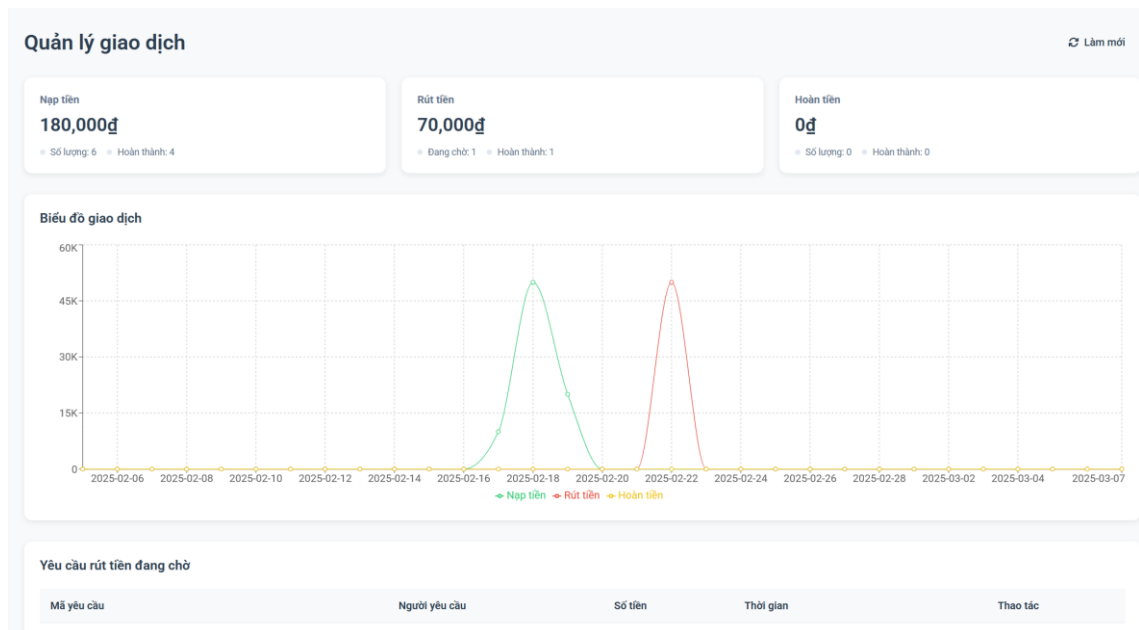
- Xem danh sách tài khoản được Seller đăng bán.
- Duyệt hoặc từ chối tài khoản số nếu vi phạm chính sách.
- Xóa tài khoản số vi phạm hoặc bị báo cáo.

Username	Email	Vai trò	Trạng thái
olalaa	ooos@gmail.com	User <div>User</div>	Đã khóa
trewtre	motijji@gmail.com	User <div>Seller</div>	Đã khóa



5.12.3 Quản lý giao dịch & thanh toán

- Theo dõi danh sách đơn hàng, trạng thái thanh toán.
- Xử lý tranh chấp giữa người mua và người bán.
- Kiểm soát dòng tiền (Seller rút tiền, hoàn tiền Buyer khi cần).
- Xem báo cáo doanh thu của hệ thống.



5.12.4 Quản lý khiếu nại & báo cáo vi phạm

- Xem danh sách khiếu nại từ Buyer/Seller.
- Phân tích nội dung khiếu nại và đưa ra quyết định.
- Cảnh báo Seller nếu có hành vi gian lận.
- Khóa tài khoản Seller nếu vi phạm nhiều lần.

CHƯƠNG 6. HƯỚNG PHÁT TRIỂN

6.1 Mở rộng danh mục sản phẩm

Hiện tại, hệ thống tập trung vào việc mua bán tài khoản số (Facebook, Telegram, Netflix, Canva, v.v.). Trong tương lai, nền tảng có thể mở rộng để hỗ trợ:

- Bán **dịch vụ số** (thuê tài khoản, đăng ký membership theo tháng).
- Bán **mã giảm giá, gift card, khóa học online**.
- Hỗ trợ mua bán tài khoản game theo cấp độ.

Mục tiêu: Đáp ứng đa dạng nhu cầu của người dùng, thu hút nhiều người mua và người bán hơn.

6.2 Cải thiện trải nghiệm người dùng

- Ứng dụng công nghệ AI để gợi ý sản phẩm phù hợp với nhu cầu của từng người dùng.
- Tích hợp chatbot hỗ trợ tự động để tư vấn khách hàng nhanh chóng.
- Tối ưu giao diện UI/UX giúp thao tác tìm kiếm, thanh toán mượt mà hơn.

Mục tiêu: Tăng tỷ lệ chuyển đổi, giữ chân khách hàng lâu dài.

6.3 Hệ thống đánh giá và bảo mật

- Hệ thống đánh giá người bán để đảm bảo chất lượng tài khoản số.
- Cơ chế ký quỹ: Tài khoản chỉ được chuyển giao sau khi người mua xác nhận sử dụng thành công.
- Xác thực hai lớp (2FA) để bảo vệ tài khoản người dùng khỏi bị đánh cắp.

Mục tiêu: Tạo niềm tin cho người mua, đảm bảo tính minh bạch và an toàn.

6.4 Tích hợp nhiều phương thức thanh toán

Hiện tại hệ thống có thể hỗ trợ thanh toán qua **chuyển khoản ngân hàng, ví điện tử**. Trong tương lai, có thể bổ sung:

- Thanh toán qua crypto (USDT, BTC, ETH) để tiếp cận thị trường quốc tế.
- Hỗ trợ trả góp hoặc thanh toán theo kỳ hạn cho những tài khoản có giá trị cao.

Mục tiêu: Tăng sự tiện lợi và tiếp cận nhiều khách hàng hơn.

6.5 Xây dựng hệ thống Affiliate & Referral

- Tạo chương trình giới thiệu: Người dùng giới thiệu bạn bè sẽ nhận được hoa hồng từ mỗi giao dịch thành công.

- **Hệ thống cộng tác viên (Affiliate)** giúp mở rộng hệ thống người bán, tăng trưởng nhanh chóng.

Mục tiêu: Thúc đẩy sự phát triển tự nhiên, giúp dự án mở rộng thị phần mà không tốn quá nhiều chi phí quảng cáo.

CHƯƠNG 7. TỔNG KẾT

Dự án Octopus được xây dựng với mục tiêu tạo ra một nền tảng giao dịch tài khoản số minh bạch, an toàn và tiện lợi. Với sự phát triển mạnh mẽ của thị trường tài khoản số, nhu cầu mua bán tài khoản game, dịch vụ streaming (Netflix, YouTube, Canva), hay các tài khoản mạng xã hội (Facebook, Telegram) ngày càng tăng cao. Octopus không chỉ đơn thuần là một nền tảng mua bán, mà còn là một hệ sinh thái kết nối giữa người mua và người bán một cách nhanh chóng và hiệu quả.

Từ những ngày đầu phát triển, chúng tôi đã tập trung vào việc xây dựng một hệ thống vững chắc với công nghệ MERN Stack (MongoDB, Express.js, React.js, Node.js), đảm bảo hiệu suất cao, bảo mật tốt, và trải nghiệm người dùng mượt mà. Hệ thống đăng nhập bảo mật bằng JWT Authentication, dữ liệu được tối ưu với MongoDB, và giao diện người dùng linh hoạt nhờ React.js kết hợp Redux. Không chỉ dừng lại ở tính năng mua bán đơn giản, Octopus còn tích hợp giỏ hàng, thanh toán qua nhiều phương thức, và quản lý giao dịch một cách chặt chẽ.

Trong quá trình phát triển, nhóm chúng tôi đã không ngừng tối ưu hệ thống, lắng nghe phản hồi từ người dùng để cải thiện giao diện, nâng cao trải nghiệm tìm kiếm sản phẩm, cũng như triển khai các biện pháp bảo mật giúp giảm thiểu rủi ro gian lận. Chúng tôi hiểu rằng sự tin tưởng của khách hàng là yếu tố cốt lõi, do đó, hệ thống đánh giá người bán, quản lý tranh chấp, và cơ chế ký quỹ sẽ được phát triển nhằm đảm bảo mọi giao dịch diễn ra một cách công bằng và an toàn.

Hướng tới tương lai, Octopus không chỉ dừng lại ở việc phục vụ thị trường trong nước mà còn hướng đến mở rộng ra quốc tế, hỗ trợ nhiều ngôn ngữ và phương thức thanh toán như crypto (BTC, USDT, ETH). Chúng tôi cũng đang nghiên cứu tích hợp AI để đề xuất sản phẩm thông minh, cải thiện bảo mật với xác thực hai lớp (2FA), và triển khai chương trình Affiliate Marketing nhằm mở rộng quy mô người dùng.

Với một nền tảng công nghệ vững chắc, mô hình kinh doanh rõ ràng, và tiềm năng phát triển mạnh mẽ, Octopus hứa hẹn sẽ trở thành một trong những nền tảng giao dịch tài khoản số hàng đầu, mang lại trải nghiệm mua bán nhanh chóng, minh bạch và đáng tin cậy cho cộng đồng người dùng.

CHƯƠNG 8. TÀI LIỆU THAM KHẢO

1. Chatgpt
2. Claude
3. plantuml.com