

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN



**BÀI TẬP LỚN MÔN PHÁT TRIỂN HỆ THỐNG
THƯƠNG MẠI ĐIỆN TỬ**

Đề tài : App mua bán hàng trực tuyến

Giảng viên : Kim Ngọc Bách

Nhóm bài tập lớn 03

Họ tên Mã sinh viên

Đào Việt Anh B21DCCN003

Nguyễn Giang Linh B21DCCN483

Đoàn Minh Hiền B21DCCN339

Hà Nội – 2025

Lời cảm ơn

Trong suốt quá trình thực hiện đề tài “App mua bán hàng trực tuyến” thuộc môn học Phát triển hệ thống thương mại điện tử, nhóm chúng em đã nhận được rất nhiều sự giúp đỡ và hỗ trợ quý báu.

Trước tiên, nhóm xin gửi lời cảm ơn chân thành và sâu sắc tới thầy Kim Ngọc Bách – giảng viên hướng dẫn, người đã tận tình chỉ bảo, định hướng và hỗ trợ nhóm trong suốt quá trình triển khai đề tài. Những góp ý và nhận xét của thầy là kim chỉ nam giúp nhóm hoàn thiện sản phẩm tốt hơn cả về mặt kỹ thuật lẫn nội dung.

Chúng em cũng xin cảm ơn Khoa Công nghệ Thông tin – Học viện Công nghệ Bru chính Viện thông đã tạo điều kiện học tập và cung cấp môi trường lý tưởng để nhóm có cơ hội vận dụng kiến thức vào thực tế thông qua đề tài này.

Cuối cùng, nhóm xin cảm ơn các thành viên trong nhóm đã cùng nhau làm việc nghiêm túc, phối hợp chặt chẽ, hỗ trợ lẫn nhau để hoàn thành bài tập lớn đúng tiến độ và chất lượng đề ra.

Mặc dù nhóm đã cố gắng hết sức trong quá trình thực hiện, nhưng chắc chắn không thể tránh khỏi những thiếu sót. Nhóm rất mong nhận được sự thông cảm và những góp ý từ thầy cô để có thể cải thiện và hoàn thiện hơn trong các dự án sau.

Mục lục

Phần I: Giới thiệu chung	4
I. Phát biểu bài toán	4
II. Mô tả công việc	4
III. Giới thiệu công nghệ, công cụ sử dụng	5
IV. Lý do lựa chọn công nghệ	5
Chương II: Các công việc triển khai	7
Giai đoạn 1: Thiết kế và xây dựng giao diện ứng dụng di động (Flutter)	7
Giai đoạn 2: Xây dựng hệ thống backend (Django REST Framework)	7
Giai đoạn 3: Thiết kế và quản lý cơ sở dữ liệu (PostgreSQL & Supabase)	8
Giai đoạn 4: Kiểm thử và triển khai	8
Chương III: Kết quả đạt được	10
I. Thiết kế hệ thống	10
II. Thiết kế API	14
III. Giao diện Mobile app	17

Phần I: Giới thiệu chung

I. Phát biểu bài toán

Trong bối cảnh nền kinh tế số ngày càng phát triển, thương mại điện tử đã trở thành một phần thiết yếu trong cuộc sống hàng ngày của người tiêu dùng. Với nhu cầu trao đổi, mua bán hàng hóa ngày một gia tăng, việc xây dựng một nền tảng trực tuyến thuận tiện, dễ sử dụng và đa chức năng là điều vô cùng cấp thiết. Xuất phát từ thực tế đó, dự án "Ứng dụng mua bán hàng hóa đa dạng" được triển khai với mục tiêu tạo ra một hệ sinh thái thương mại điện tử thân thiện, bảo mật và hiệu quả cho người dùng.

Dự án sẽ bao gồm một số các tính năng:

- Thêm sản phẩm của bạn: Thêm các sản phẩm để bán hàng: thông tin sản phẩm, ảnh, nhãn, loại, ...
- Xem sản phẩm của các danh mục khác nhau: Tính năng mua hàng, phân loại sản phẩm
- Duy trì giỏ hàng: Xem các đơn hàng đã thêm, thêm sửa xóa các sản phẩm
- Đánh dấu sản phẩm là mục yêu thích của bạn
- Quản lý địa chỉ của bạn: Sửa địa chỉ để nhận hàng
- Xem lại các sản phẩm bạn mua
- Tìm kiếm sản phẩm: Hỗ trợ tìm kiếm sản phẩm, dễ tìm kiếm sản phẩm cần theo tag
- Quản lý đơn hàng, đổi trả, hoàn tiền: Theo dõi đơn hàng sau khi đặt, có thể đổi trả và hoàn tiền
- Thanh toán: Hỗ trợ các hình thức thanh toán bằng ví điện tử

II. Mô tả công việc

Dự án ứng dụng thương mại điện tử C2C nhằm kết nối người mua và người bán cá nhân hoặc shop nhỏ thông qua nền tảng mobile (Flutter) và backend server (Django REST Framework với PostgreSQL). Hệ thống hỗ trợ các chức năng như xác thực người dùng qua Google/mạng xã hội, đăng và tìm kiếm sản phẩm, quản lý giỏ hàng, đơn hàng, thanh toán bằng ví điện tử, theo dõi và đổi trả hàng. Mô hình doanh thu bao gồm phí giao dịch, quảng cáo ưu tiên và tích hợp thanh toán.

III. Giới thiệu công nghệ, công cụ sử dụng

Mobile App:

- **Flutter:** Framework đa nền tảng để phát triển ứng dụng trên Android và iOS.

Backend:

- **Ngôn ngữ:** Python
- **Framework:** **Django REST Framework (DRF)** – hỗ trợ xây dựng API mạnh mẽ, có tích hợp ORM.

Cơ sở dữ liệu:

- **PostgreSQL:** Hệ quản trị cơ sở dữ liệu quan hệ, ổn định và phù hợp với ứng dụng thương mại điện tử.

Thanh toán:

- **Tích hợp thanh toán:** Hỗ trợ thanh toán trực tuyến qua các phương thức hiện đại.

Công cụ triển khai và phát triển:

- **Git:** Quản lý mã nguồn.
- **Postman:** Kiểm thử API.

IV. Lý do lựa chọn công nghệ

Việc lựa chọn các công nghệ và công cụ trên dựa trên các tiêu chí về hiệu năng, tính phổ biến, cộng đồng hỗ trợ và khả năng mở rộng trong tương lai:

- **Flutter:** Cho phép phát triển ứng dụng đa nền tảng (Android và iOS) từ một codebase, giúp tiết kiệm thời gian và công sức.
- **Django REST Framework:** Cung cấp cách xây dựng API mạnh mẽ, dễ mở rộng, phù hợp với mô hình client-server hiện đại.
- **PostgreSQL:** Hệ quản trị cơ sở dữ liệu quan hệ ổn định, đáng tin cậy, hỗ trợ tốt cho các truy vấn phức tạp và dữ liệu lớn.
- **Cổng thanh toán:** Đáp ứng nhu cầu thanh toán nhanh, tiện lợi, phù hợp xu hướng tiêu dùng hiện nay.
- **Git:** Giúp quản lý mã nguồn, theo dõi thay đổi và hỗ trợ làm việc nhóm hiệu quả.

- **Postman:** Hỗ trợ kiểm thử API nhanh chóng, dễ sử dụng trong quá trình phát triển và kiểm tra hệ thống.

Việc kết hợp các công nghệ trên tạo thành một nền tảng vững chắc, vừa đảm bảo tốc độ phát triển nhanh chóng, vừa đáp ứng tốt yêu cầu kỹ thuật và trải nghiệm người dùng.

Chương II: Các công việc triển khai

Giai đoạn 1: Thiết kế và xây dựng giao diện ứng dụng di động (Flutter)

Flutter là framework mã nguồn mở của Google cho phép xây dựng ứng dụng mobile đa nền tảng. Trong giai đoạn này, em đảm nhận việc phát triển giao diện và các chức năng chính như:

- Giao diện đăng nhập/đăng ký đơn giản, dễ sử dụng.
- Trang chủ hiển thị danh mục sản phẩm và các mặt hàng nổi bật giúp người dùng dễ dàng khám phá.
- Chức năng tìm kiếm và lọc sản phẩm theo từ khóa, loại, giá cả,... hỗ trợ người dùng nhanh chóng tìm được sản phẩm phù hợp.
- Cho phép xem chi tiết sản phẩm: hình ảnh, mô tả, giá, đánh giá từ người bán và người mua.
- Giỏ hàng hỗ trợ thêm, xóa, cập nhật số lượng sản phẩm, thuận tiện cho quá trình mua sắm.
- Tích hợp thanh toán trực tuyến qua ví điện tử nhanh chóng và an toàn.
- Hỗ trợ theo dõi trạng thái đơn hàng theo thời gian thực sau khi đặt mua.
- Trang quản lý thông tin cá nhân, cho phép người dùng cập nhật địa chỉ nhận hàng dễ dàng.
- Tất cả chức năng đều kết nối với backend thông qua API RESTful đảm bảo dữ liệu đồng bộ và chính xác.

Giai đoạn 2: Xây dựng hệ thống backend (Django REST Framework)

Django REST Framework (DRF) là một framework mạnh mẽ để xây dựng API bằng Python. Trong giai đoạn này, backend được chia thành nhiều module nhỏ bao gồm:

- Phát triển API quản lý sản phẩm, bao gồm thêm, sửa, xóa và hiển thị sản phẩm; API danh mục để phân loại sản phẩm theo từng nhóm; API giỏ hàng cho phép người dùng thêm sản phẩm, điều chỉnh số lượng và xem nội dung giỏ hàng; cùng với API đơn hàng để xử lý đặt hàng, cập nhật trạng thái và theo dõi tiến trình vận chuyển.
- Cung cấp API cho chức năng đánh dấu sản phẩm yêu thích, hỗ trợ người dùng lưu lại các mặt hàng quan tâm; API quản lý địa chỉ giao hàng bao gồm thêm, sửa, xóa địa chỉ; và API đổi trả hàng cho phép gửi yêu cầu trả lại hoặc hoàn tiền sau khi mua hàng.

- Xây dựng API thanh toán hỗ trợ kết nối với ví điện tử, giúp xử lý giao dịch nhanh chóng, tiện lợi và an toàn trong quá trình mua sắm.
- Áp dụng cơ chế phân quyền người dùng bằng cách kết hợp JWT và middleware bảo mật, giúp hệ thống xác định vai trò (người dùng thường, người bán, admin), đảm bảo chỉ người dùng hợp lệ mới có quyền truy cập vào các chức năng tương ứng.

Giai đoạn 3: Thiết kế và quản lý cơ sở dữ liệu (PostgreSQL & Supabase)

- PostgreSQL được sử dụng làm hệ quản trị cơ sở dữ liệu chính nhờ khả năng xử lý hiệu quả các truy vấn quan hệ phức tạp, phù hợp với hệ thống thương mại điện tử có nhiều bảng liên kết như sản phẩm, đơn hàng, người dùng,...
- Dữ liệu được ánh xạ qua ORM của Django, giúp quản lý mô hình rõ ràng, dễ phát triển và bảo trì, đồng thời giảm rủi ro khi thao tác trực tiếp với cơ sở dữ liệu.
- Supabase được tích hợp để hỗ trợ lưu trữ ảnh sản phẩm, xác thực phụ và cung cấp khả năng đồng bộ dữ liệu real-time, phục vụ cho các tính năng nâng cao của ứng dụng.
- Hệ thống bao gồm nhiều bảng dữ liệu hỗ trợ đầy đủ chức năng của một nền tảng thương mại điện tử.

Giai đoạn 4: Kiểm thử và triển khai

- Postman được sử dụng để kiểm thử các API trong quá trình phát triển, giúp gửi các yêu cầu HTTP (GET, POST, PUT, DELETE,...) tới server, kiểm tra dữ liệu phản hồi, xác minh tính đúng đắn của các chức năng backend và hỗ trợ phát hiện lỗi sớm.
- Thực hiện kiểm thử tích hợp giữa frontend (Flutter) và backend (Django REST Framework) để đảm bảo dữ liệu được truyền nhận chính xác, các chức năng hoạt động đồng bộ, trải nghiệm người dùng liền mạch và ổn định trên giao diện ứng dụng.
- Tiến hành tối ưu hiệu năng các API, bao gồm việc cải thiện tốc độ truy vấn cơ sở dữ liệu, giới hạn dữ liệu trả về, xử lý phân trang, caching, và đồng thời xây dựng các cơ chế xử lý lỗi toàn diện để đảm bảo hệ thống phản hồi chính xác và bảo mật khi xảy ra sự cố.
- Đóng gói và hoàn thiện tài liệu kỹ thuật, bao gồm mô tả chi tiết các endpoint API, hướng dẫn sử dụng và cấu trúc dữ liệu đầu vào/đầu ra, sau đó chuẩn bị triển khai sản phẩm lên môi trường production, đảm bảo ứng dụng có thể vận hành thực tế một cách ổn định và hiệu quả.

Chương III: Kết quả đạt được

Sau quá trình nghiên cứu và phát triển đề tài “App bán hàng” trong khuôn khổ môn học *Phát triển hệ thống thương mại điện tử*, nhóm đã hoàn thành một hệ thống ứng dụng bán hàng cơ bản với các chức năng chính dành cho người dùng và quản trị viên. Hệ thống bao gồm các phần sau:

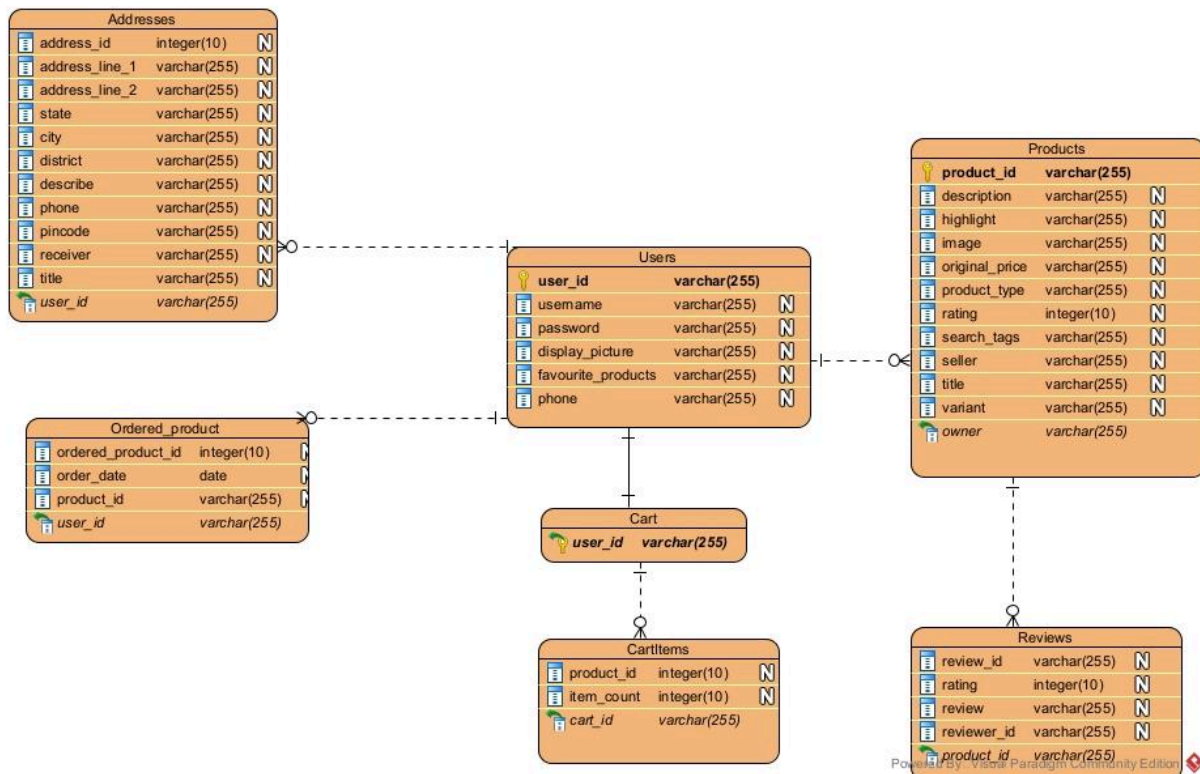
I. Thiết kế hệ thống

1. Sơ đồ hệ thống



2. Thiết kế cơ sở dữ liệu

Hệ thống sử dụng cơ sở dữ liệu quan hệ PostgreSQL để lưu trữ toàn bộ thông tin người dùng, sản phẩm, giỏ hàng, đơn hàng, địa chỉ, đánh giá sản phẩm và mối quan hệ giữa chúng. Việc thiết kế dựa trên mô hình C2C (Consumer to Consumer), hỗ trợ các chức năng mua bán, đánh giá, đề xuất sản phẩm.



2.1. Các bảng chính

a. Bảng User

Bảng lưu trữ thông tin của tất cả người dùng trong hệ thống, bao gồm cả người mua và người bán. Các thông tin như tên người dùng, mật khẩu, email, ảnh đại diện, và số điện thoại được lưu tại đây. Ngoài ra, bảng này còn liên kết với các bảng như Products, Addresses, Cart, Ordered_Product, và Reviews.

Thuộc tính	Kiểu dữ liệu	Ràng buộc
user_id	varchar(255)	PK
username	varchar(255)	Not null, Unique
password	varchar(255)	Not null
display_picture	varchar(255)	
favourite_products	varchar(255)	

phone	varchar(255)	
-------	--------------	--

b. Bảng Product

Bảng này chứa thông tin chi tiết về các sản phẩm được đăng bán trên hệ thống, bao gồm tên, mô tả, ảnh, giá gốc, giá khuyến mãi, loại sản phẩm, người bán (chủ sản phẩm), và các tag tìm kiếm. Mỗi sản phẩm có thể nhận được nhiều đánh giá từ người dùng và cũng có thể được thêm vào giỏ hàng hoặc đặt mua.

Thuộc tính	Kiểu dữ liệu	Ràng buộc
product_id	varchar(255)	PK
description	varchar(255)	
highlight	varchar(255)	
image	varchar(255)	
original_price	varchar(255)	
product_type	varchar(255)	
rating	Integer	
search_tags	varchar(255)	
seller	varchar(255)	
title	varchar(255)	
variant	varchar(255)	
owner	varchar(255)	FK → User

c. Address

Lưu danh sách địa chỉ giao hàng mà người dùng đã thêm vào tài khoản. Một người dùng có thể có nhiều địa chỉ khác nhau để thuận tiện khi đặt hàng. Các thông tin gồm tên người nhận, số điện thoại, địa chỉ chi tiết (tỉnh/thành, quận/huyện, phường/xã, tên đường, tòa nhà...).

Thuộc tính	Kiểu dữ liệu	Ràng buộc
address_id	Integer	PK

address_line_1	varchar(255)	
address_line_2	varchar(255)	
state	varchar(255)	
city	varchar(255)	
district	varchar(255)	
describe	varchar(255)	
phone	varchar(255)	
pincode	varchar(255)	
receiver	varchar(255)	
title	varchar(255)	
user_id	varchar(255)	FK → User

d. Cart

Mỗi người dùng có một giỏ hàng đại diện cho các sản phẩm mà họ đang có ý định mua. Bảng này giúp xác định mối liên hệ giữa người dùng và giỏ hàng của họ.

Thuộc tính	Kiểu dữ liệu	Ràng buộc
user_id	varchar(255)	PK → User

e. CartItems

Bảng lưu trữ các mục (sản phẩm) có trong giỏ hàng của người dùng, cùng với số lượng mỗi sản phẩm. Một người dùng có thể thêm nhiều sản phẩm vào giỏ và cập nhật số lượng tương ứng.

Thuộc tính	Kiểu dữ liệu	Ràng buộc
product_id	integer	FK → Products
item_count	integer	
cart_id	varchar(255)	FK → Cart

f. Ordered_Product

Ghi nhận lịch sử các sản phẩm mà người dùng đã đặt mua. Bảng này lưu thời gian đặt hàng, thông tin sản phẩm, và thông tin người dùng đã mua sản phẩm đó.

Thuộc tính	Kiểu dữ liệu	Ràng buộc
ordered_product_id	integer	Primary Key
order_date	date	
product_id	varchar(255)	FK → Products
user_id	varchar(255)	FK → Users

g. Reviews

Lưu lại các đánh giá của người dùng đối với các sản phẩm đã mua. Mỗi đánh giá bao gồm số sao (rating), nội dung đánh giá (review text), thông tin người đánh giá và sản phẩm được đánh giá. Các đánh giá này giúp hệ thống đề xuất sản phẩm phù hợp với sở thích người dùng.

Thuộc tính	Kiểu dữ liệu	Ràng buộc
review_id	varchar(255)	Primary Key
rating	integer	
review	varchar(255)	
reviewer_id	varchar(255)	FK → Users
product_id	varchar(255)	FK → Products

II. Thiết kế API

2.1. Tổng quan về Django Rest Framework

Backend của hệ thống được xây dựng dựa trên Django REST Framework (DRF) – một thư viện mở rộng của Django, chuyên dùng để phát triển các API RESTful hiện đại, có khả năng mở rộng và bảo trì tốt. DRF cung cấp một bộ công cụ mạnh mẽ và linh hoạt để xử lý các yêu cầu HTTP, ánh xạ dữ liệu từ và tới cơ sở dữ liệu thông qua mô hình ORM (Object-Relational Mapping) của Django, đồng thời hỗ trợ nhiều phương thức xác thực và phân quyền nâng cao.

Việc lựa chọn DRF làm nền tảng backend xuất phát từ nhu cầu xây dựng một hệ thống client-server rõ ràng, nơi backend chỉ đóng vai trò cung cấp dữ liệu, xử lý logic nghiệp vụ và phản hồi dưới dạng JSON để tương tác trực tiếp với ứng dụng frontend viết bằng Flutter. Kiến trúc này mang lại sự tách biệt rõ ràng giữa giao diện người dùng và logic xử lý phía server, giúp dễ dàng mở rộng, bảo trì hoặc thay thế từng phần của hệ thống mà không ảnh hưởng đến phần còn lại. Một số lý do cụ thể khiến DRF được sử dụng trong dự án bao gồm:

- Khả năng xây dựng API nhanh chóng và chuẩn RESTful: DRF hỗ trợ sẵn các view, router, serializer giúp định nghĩa các endpoint API một cách rõ ràng, dễ dàng mở rộng và duy trì.
- Hỗ trợ đầy đủ các thao tác CRUD: Các thao tác cơ bản như tạo, đọc, cập nhật và xóa dữ liệu được DRF xử lý một cách tự động thông qua các lớp ModelViewSet, giúp tiết kiệm thời gian phát triển.
- Tích hợp dễ dàng với hệ ORM của Django: Mọi dữ liệu được ánh xạ thông qua các model Django, giúp quản lý cơ sở dữ liệu hiệu quả, dễ kiểm soát logic nghiệp vụ và mối quan hệ giữa các thực thể như người dùng, sản phẩm, đơn hàng,...
- Hệ thống xác thực và phân quyền linh hoạt: DRF hỗ trợ nhiều hình thức xác thực, bao gồm xác thực bằng Token, JWT, OAuth2,... Trong hệ thống của nhóm, Simple JWT được sử dụng để bảo mật các endpoint, giúp người dùng đăng nhập và truy cập API một cách an toàn. Ngoài ra, phân quyền được triển khai thông qua các lớp permission giúp kiểm soát ai có quyền thực hiện thao tác nào (ví dụ: chỉ admin mới được thêm sản phẩm).
- Khả năng phân trang, lọc và tìm kiếm: DRF hỗ trợ sẵn các tính năng phân trang dữ liệu (pagination), lọc theo tham số query (filter_backends), giúp nâng cao hiệu năng khi truy xuất danh sách sản phẩm hoặc đơn hàng có số lượng lớn.
- Xử lý lỗi rõ ràng, thống nhất: DRF cung cấp các phản hồi lỗi chuẩn hóa theo định dạng JSON, giúp client dễ dàng xử lý khi có lỗi xảy ra (ví dụ: dữ liệu đầu vào không hợp lệ, không có quyền truy cập,...).
- Dễ dàng mở rộng và tùy biến: DRF cho phép tùy biến sâu các lớp view, serializer hoặc permission để phù hợp với các logic nghiệp vụ đặc thù như xác thực qua mạng xã hội, xử lý thanh toán ví điện tử, hoặc xây dựng hệ thống đổi trả hàng hóa.

2.2. Tổng quan về postman

Postman là một công cụ hỗ trợ kiểm thử và phát triển API mạnh mẽ, được sử dụng rộng rãi trong các dự án lập trình backend. Postman cho phép người dùng gửi các yêu cầu HTTP đến server, phân tích phản hồi, kiểm tra logic xử lý API và hỗ trợ tự động hóa các kịch bản kiểm thử.

Các tính năng chính của Postman:

- Hỗ trợ gửi nhiều loại request như GET, POST, PUT, DELETE, v.v.
- Quản lý API theo collection, giúp tổ chức tốt các request liên quan trong cùng một dự án.
- Tạo biến môi trường giúp dễ dàng chuyển đổi giữa các môi trường phát triển, kiểm thử và triển khai.

- Hỗ trợ kiểm thử API tự động bằng các script nhỏ, giúp phát hiện lỗi sớm trong quá trình phát triển.
- Chia sẻ API và tài liệu API cho các thành viên khác trong nhóm. Postman đóng vai trò rất quan trọng trong giai đoạn phát triển và kiểm thử Backend, giúp đảm bảo rằng các API được xây dựng đúng logic và ổn định trước khi tích hợp với Frontend.

2.3. Mục product

API Name	Endpoint	Method	Request Body	Request Parameters	Purpose
Get Product List	/api/products	GET	None	pageNumber, pageSize, sortBy, sortOrder	Lấy danh sách sản phẩm.
Get Product Details	/api/products/{productid}	GET	None	productid	Lấy thông tin chi tiết sản phẩm.
Add Favourite Product	/api/users/favourite/{productid}	POST	None	productid	Thêm sản phẩm vào danh sách yêu thích.
Get Favourite Products	/api/users/favourite	GET	None	None	Lấy danh sách sản phẩm yêu thích.

2.4. Mục cart

API Name	Endpoint	Method	Request Body	Request Parameters	Purpose
Add to Cart	/api/users/cart/{productid}/add	POST	None	productid	Thêm sản phẩm vào giỏ hàng.
Increase Cart Item	/api/users/cart/{productid}/increase	POST	None	productid	Tăng số lượng sản phẩm trong giỏ hàng.
Decrease Cart Item	/api/users/cart/{productid}/decrease	POST	None	productid	Giảm số lượng sản phẩm trong giỏ hàng.
Remove from Cart	/api/users/cart/{productid}/remove	DELETE	None	productid	Xóa sản phẩm khỏi giỏ hàng.
Get Cart Items	/api/users/cart	GET	None	None	Lấy danh sách sản phẩm trong giỏ hàng.

2.5. Mục Users

API Name	Endpoint	Method	Request Body	Request Parameters	Purpose
Login	/api/users/login	POST	email, password	None	Đăng nhập người dùng.
Signup	/api/users/signup	POST	email, password, username, phone, etc.	None	Đăng ký tài khoản mới.
Change Password	/api/users/change-password	POST	current_password, new_password	None	Thay đổi mật khẩu người dùng.
Get Users	/api/users	GET	None	None	Lấy danh sách người dùng.

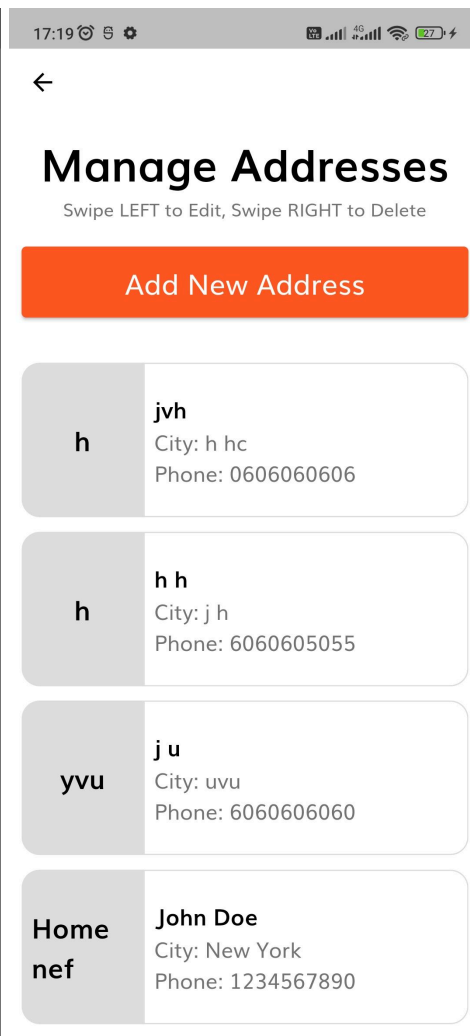
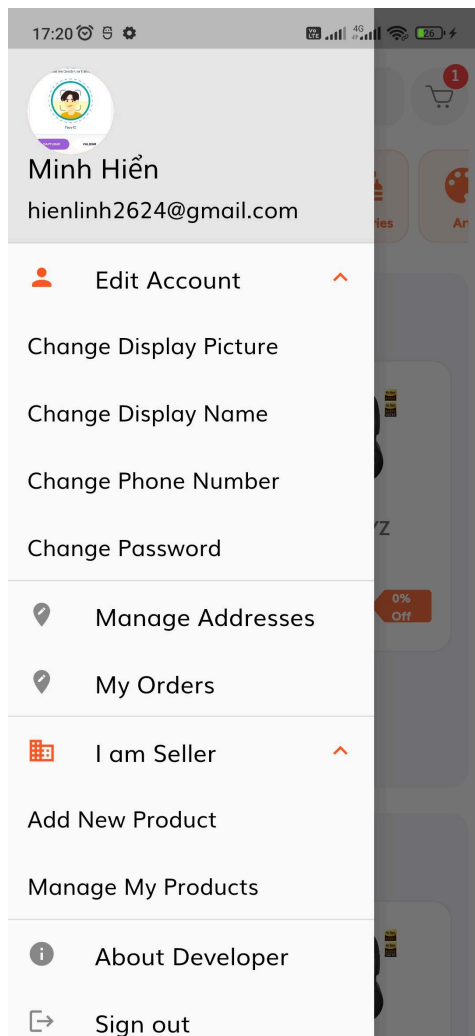
2.6. Mục Address

API Name	Endpoint	Method	Request Body	Request Parameters	Purpose
Update Address	/api/users/address/{addressid}	POST	address details	addressid	Cập nhật địa chỉ của người dùng.
Delete Address	/api/users/address/{addressid}	DELETE	None	addressid	Xóa địa chỉ của người dùng.

2.7. Mục Ordered Product

API Name	Endpoint	Method	Request Body	Request Parameters	Purpose
Add Ordered Product	/api/users/ordered-product	POST	product_id, user_id, etc.	None	Thêm sản phẩm đã đặt hàng.

III. Giao diện Mobile app





Your Products

Swipe LEFT to Edit, Swipe RIGHT to Delete



Áo hoodie nam

100.0 ₫100.0



Your Cart

Proceed to Payment



Tai nghe XYZ

500000.0 ₫500000.0

1

Total
₫ 500000.0

Checkout



Search Result

tai in [All Products](#)



Tai nghe XYZ

₫ 500000.0 0% off
₫500000.0



Fill Product Details

Basic Details

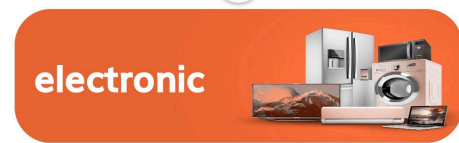
Describe Product


Upload Images



fashion


Search Tags





Tai nghe XYZ

~~₫ 500000.0~~ 0% Off
₫500000.0




Chuột ABC

~~₫ 300000.0~~ 0% Off
₫300000.0




Products You Like




Áo hoodie nam

~~₫ 100.0~~ 0% Off
₫100.0





Tai nghe XYZ

~~₫ 500000.0~~ 0% Off
₫500000.0



Explore All Products





20:22



Register Account

Complete your details or continue
with social media

Email

Enter your email



Password

Enter your password



Confirm Password

Re-enter your password



Sign up

By continuing you confirm that you agree
with our Terms and Conditions

20:22



Welcome to Ecommerce App



Or sign in with your email and password

Email

Enter your email



Password

Enter your password



[Forgot Password](#)

Sign in

Don't have an account? [Sign Up](#)