

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC TÂY ĐÔ
KHOA KỸ THUẬT – CÔNG NGHỆ**



Trí tuệ - Sáng tạo - Năng động – Đổi mới

**NIÊN LUẬN 2
NỀN TẢNG GIAO DỊCH XE MÁY**

**CHUYÊN NGÀNH
CÔNG NGHỆ THÔNG TIN**

**SINH VIÊN THỰC HIỆN
QUÁCH TRƯỜNG PHÚC - 227060168**

Cần Thơ, tháng 4, năm 2025

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC TÂY ĐÔ
KHOA KỸ THUẬT – CÔNG NGHỆ**



Trí tuệ - Sáng tạo - Năng động – Đổi mới

**NIÊN LUẬN 1
NỀN TẢNG GIAO DỊCH XE MÁY**

**CHUYÊN NGÀNH
CÔNG NGHỆ THÔNG TIN**

**SINH VIÊN THỰC HIỆN
QUÁCH TRƯỜNG PHÚC - 227060168**

**GIÁO VIÊN HƯỚNG DẪN
THS. NGUYỄN CHÍ CƯỜNG**

Cần thơ, tháng 4, năm 2025

MỤC LỤC

MỤC LỤC HÌNH ẢNH.....	iii
MỤC LỤC BẢNG.....	iv
LỜI CẢM ƠN	v
LỜI CAM ĐOAN.....	vi
TÓM TẮT	vii
ABSTRACT	viii
CHƯƠNG I. MỞ ĐẦU.....	1
1.1. Đặt vấn đề.....	1
1.2. Mục tiêu cần đạt được	1
1.3. Phạm vi nghiên cứu.....	2
1.3.1. Thị trường xe máy trực tuyến	2
1.3.2. Công nghệ phát triển hệ thống.....	2
1.3.3. Quản lý xe máy và giao dịch.....	2
1.3.4. Tìm kiếm và lọc sản phẩm	2
1.3.5. Người dùng và tối ưu hoá.....	2
1.3.6. Phạm vi địa lý và mở rộng nền tảng.....	2
1.4. Phương pháp nghiên cứu.....	3
1.4.1 Nghiên cứu tài liệu và cơ sở lý thuyết	3
1.4.2 Phân tích và khảo sát hệ thống thực tế.....	3
1.4.3 Thiết kế hệ thống và mô hình dữ liệu	3
1.4.4 Kiểm thử và đánh giá	3
1.5. Hướng giải quyết.....	3
1.6. Kế hoạch thực hiện.....	4
CHƯƠNG II. CƠ SỞ LÝ THUYẾT	6
2.1. Mô hình kinh doanh C2C	6
2.2. Mô hình hoá dữ liệu.....	7
2.2.1. Thực thể và thuộc tính	7
2.2.2. MCD.....	8
2.2.3. MLD	8
2.2.4. MPD	9
2.3. Định nghĩa Website.....	9
2.3. API (Application Programming Interface)	10
2.3.1. Khái niệm API	10
2.3.2. Cách hoạt động của API	10
2.3.3. Các dạng Web API.....	11

2.3.4. Giao thức API	12
2.4. JSON Web Token (JWT)	12
2.3.1. Định nghĩa	12
2.3.2. Lợi ích mang lại	12
2.3.3. Cấu trúc của một JWT	13
2.3.4. Cách thức hoạt động	14
2.3.5. Lý do sử dụng	16
2.5. Công nghệ được sử dụng	16
2.2.1. Môi trường phát triển	16
2.2.2. Công cụ phát triển	17
2.2.3. Giao diện người dùng	17
2.2.4. Xử lý dữ liệu	19
2.2.5. Cơ sở dữ liệu	19
2.2.6. Quản lý thư viện và gói phần mềm	20
CHƯƠNG III. KẾT QUẢ THỰC HIỆN	21
3.1. Quản lý dữ liệu	21
3.2. Giao diện và chức năng	29
3.2.1. Trang chủ	29
3.2.2. Điều hướng người dùng	30
3.2.2. Đăng ký và đăng nhập	31
3.2.3. Tìm kiếm và lọc xe	34
3.2.4. Đăng bán xe	36
3.2.5. Quản lý xe máy	38
3.2.6. Chi tiết xe máy	41
3.2.7. Nhắn tin	42
3.2.8. Giao dịch	44
3.2.9. Đánh giá uy tín	46
3.2.10. Cá nhân hoá người dùng	47
KẾT LUẬN – ĐÁNH GIÁ	50
TÀI LIỆU THAM KHẢO	52

MỤC LỤC HÌNH ẢNH

Hình 2. 1 Kết quả của giải mã một JWT ví dụ	14
Hình 3. 1 Mô hình MCD của hệ thống	21
Hình 3. 2 Mô hình MLD	22
Hình 3. 3 Mô hình MPD	22
Hình 3. 4 Sơ đồ cơ sở dữ liệu.....	23
Hình 3. 5 Trang chủ.....	29
Hình 3. 6 Các nội dung của trang chủ.....	29
Hình 3. 7 Menu điều hướng người dùng	30
Hình 3. 8 Form đăng ký	31
Hình 3. 9 Form đăng nhập.....	31
Hình 3. 10 Lưu đồ chức năng đăng ký tài khoản.....	33
Hình 3. 11 Lưu đồ chức năng đăng nhập.....	33
Hình 3. 12 Trang tìm kiếm và lọc sản phẩm.....	34
Hình 3. 13 Ví dụ khi sử dụng thanh lọc sản phẩm	35
Hình 3. 14 Lưu đồ của chức năng tìm kiếm và lọc xe.....	35
Hình 3. 15 Trang bán xe	36
Hình 3. 16 Lưu đồ chức năng đăng bán.....	37
Hình 3. 17 Trang quản lý xe máy đã đăng	38
Hình 3. 18 Các hành động để tương tác với xe đăng bán.....	38
Hình 3. 19 Chỉnh sửa thông tin xe đã đăng bán.....	39
Hình 3. 20 Chỉnh sửa thông tin xe đã đăng bán.....	39
Hình 3. 21 Lưu đồ chức năng chỉnh sửa thông tin xe đã đăng	40
Hình 3. 22 Trang thông tin chi tiết của xe.....	41
Hình 3. 23 Bên dưới của trang thông tin chi tiết xe	42
Hình 3. 24 Giao diện của chức năng nhắn tin.....	42
Hình 3. 25 Chi tiết của cuộc hội thoại	43
Hình 3. 26 Khi nhắn tin	43
Hình 3. 27 Lưu đồ chức năng nhắn tin	44
Hình 3. 28 Trang giao dịch xe máy bao gồm cả biểu đồ thống kê.....	44
Hình 3. 29 Những giao dịch của người dùng.....	45
Hình 3. 30 Lưu đồ chức năng giao dịch	46
Hình 3. 31 Mục đánh giá uy tín của người dùng khác.....	46
Hình 3. 32 Hiện thị ra những đánh giá của người đó.....	46
Hình 3. 33 Lưu đồ chức năng đánh giá người dùng.....	47
Hình 3. 34 Trang cá nhân	47
Hình 3. 35 Nút tải lên của ảnh bìa.....	48
Hình 3. 36 Chỉnh sửa thông tin cá nhân	48
Hình 3. 37 Lưu đồ chức năng chỉnh sửa thông tin người dùng	49

MỤC LỤC BẢNG

Bảng 3. 1 Chi tiết thực thể USER.....	24
Bảng 3. 2 Chi tiết thực thể MOTORCYCLE.....	25
Bảng 3. 3 Chi tiết thực thể BRAND.....	26
Bảng 3. 4 Chi tiết thực thể TRANSACTION	27
Bảng 3. 5 Chi tiết thực thể MESSAGE	28
Bảng 3. 6 Chi tiết thực thể REPUTATION.....	28

LỜI CẢM ƠN

Lời đầu tiên, xin gửi lời cảm ơn chân thành tới Thầy Nguyễn Chí Cường, người đã tận tình hướng dẫn và chỉ bảo trong suốt quá trình thực hiện bài niên luận này. Thầy đã giúp vượt qua nhiều khó khăn và cung cấp những kiến thức quý báu, góp phần quan trọng vào việc hoàn thành nghiên cứu này. Bên cạnh đó, xin bày tỏ lòng biết ơn sâu sắc tới các thầy cô trong Khoa Kỹ Thuật Công Nghệ, Trường Đại học Tây Đô, vì đã tạo điều kiện thuận lợi trong quá trình học tập và nghiên cứu. Cảm ơn gia đình, cha mẹ và những người thân yêu đã luôn động viên, khích lệ cả về tinh thần lẫn vật chất trong suốt chặng đường học tập. Cuối cùng, xin cảm ơn bạn bè đã luôn hỗ trợ và đóng góp ý kiến quý báu, giúp hoàn thiện bài niên luận này. Những lời động viên và sự giúp đỡ của mọi người chính là nguồn động lực to lớn giúp hoàn thành tốt công việc nghiên cứu. Xin chân thành cảm ơn!

LỜI CAM ĐOAN

Tôi xin cam đoan rằng Niên luận này là do chính tôi thực hiện, không sao chép dưới bất kỳ hình thức nào hay thuê hoặc nhờ người khác thực hiện.

Dữ liệu và kết quả phân tích trong Niên luận đảm bảo tính chính xác, khách quan và trung thực, không có bất kỳ sự ngụy tạo và điều chỉnh kết quả nghiên cứu bằng sự chủ quan của tác giả.

Cần Thơ, ngày.....tháng 4 năm 2025

Sinh viên thực hiện

Quách Trường Phúc

TÓM TẮT

RevNow là một nền tảng giao dịch trực tuyến được tạo ra nhằm mục đích phục vụ cho những người đang có nhu cầu tìm mua xe phù hợp yêu cầu của bản thân, và những người bán xe, những đại lý muốn mở rộng cách tiếp cận khách hàng. Nền tảng này cung cấp cho 2 loại xe để giao dịch chính là xe mới và xe cũ.

Để sử dụng được chức năng bán và mua người dùng cần đăng nhập tài khoản của mình vào để nền tảng quản lý. Nền tảng cung cấp 2 cách thức đăng nhập là đăng nhập bằng email và mật khẩu là cách truyền thống, và cách đăng nhập bằng tài khoản Google. Khi chưa có tài khoản người dùng có thể dễ dàng đăng ký tài khoản hoặc cũng có thể chọn đăng ký bằng tài khoản Google.

Người dùng có thể dễ dàng tìm kiếm chiếc xe mong muốn của mình thông qua chức năng lọc và tìm kiếm xe máy của hệ thống. Và khi có nhu cầu bán chiếc xe của bản thân thì người dùng có thể đăng bán chiếc xe của mình bằng biểu mẫu lên nền tảng, sau khi đã đăng bán rồi thì người bán khi cần sửa lại thông tin của xe đã đăng bán thì đều có thể vào chỉnh sửa bất kỳ lúc nào.

Các giao dịch giữa các người dùng đều sẽ được lưu lại để phục vụ mục đích thống kê cũng như là có thể xem lại lịch sử giao dịch của bản thân người dùng.

Ngoài ra nền tảng còn cung cấp một chức năng để các người dùng có thể dễ dàng nhắn tin với nhau là chức năng nhắn tin.

ABSTRACT

RevNow is an online trading platform designed to meet the needs of individuals looking to purchase motorbikes that suit their preferences, as well as sellers and dealerships seeking to expand their customer reach. The platform supports transactions for two main categories of motorbikes: new and used.

To access the buying and selling features, users are required to log into their accounts for management purposes. RevNow offers two login methods: the traditional email and password method, and a more convenient option using a Google account. For new users, account registration is simple and can also be completed using a Google account.

Users can easily search for their desired motorbikes through the platform's advanced filtering and search functions. If users wish to sell their motorbikes, they can submit a listing using an online form provided by the system. After a listing is posted, sellers can edit their information at any time as needed.

All transactions between users are recorded to support data tracking and allow users to review their personal transaction history.

In addition, the platform includes a built-in messaging feature that enables users to communicate with each other easily and efficiently..

ĐÁNH GIÁ KẾT QUẢ THỰC HIỆN NIÊN LUẬN, TIỂU LUẬN, KHOÁ LUẬN

(Học kỳ: II, Năm 2025)

TÊN ĐỀ TÀI: NỀN TẢNG GIAO DỊCH XE MÁY

GIÁO VIÊN HƯỚNG DẪN:

STT	HỌ VÀ TÊN	MSCB
1	Nguyễn Chí Cường	

SINH VIÊN THỰC HIỆN:

STT	HỌ VÀ TÊN	MSSV	THƯỜNG (Tối đa 1,0 điểm)	ĐIỂM
1	Quách Trường Phúc	227060168		

I. HÌNH THỨC (Tối đa 1,5 điểm)

Bìa (tối đa 0,5 điểm)

Bố cục (tối đa 1 điểm)

II. NỘI DUNG (Tối đa 5 điểm)

Giới thiệu (tối đa 0.5 điểm)

Lý thuyết (tối đa 1.0 điểm)

Ứng dụng (tối đa 3 điểm)

Kết luận (tối đa 0,5 điểm)

III. CHƯƠNG TRÌNH DEMO (Tối đa 3.5 điểm)

Giao diện thân thiện với người dùng (0.5 điểm)

Hướng dẫn sử dụng (0.5 điểm)

Kết quả thực hiện đúng với kết quả của phần ứng dụng (2.5 điểm)

Ghi chú:

- Điểm trong khung “sinh viên thực hiện” là điểm kết quả cuối cùng của từng sinh viên trong quá trình thực hiện Niên luận.../tiểu luận/khoá luận.
- Nếu sinh viên demo chương trình và trả lời vấn đáp không đạt yêu cầu của giáo viên hướng dẫn thì sinh viên sẽ nhận điểm F cho học phần này.

Cần Thơ, ngày tháng năm 2025.

GIÁO VIÊN CHẤM

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Cần Thơ, ngày tháng năm 2025

Giáo viên hướng dẫn

.....

CHƯƠNG I. MỞ ĐẦU

1.1. Đặt vấn đề

– Trong bối cảnh công nghệ kỹ thuật số, mạng internet phát triển mạnh mẽ như hiện nay thì nhu cầu giao dịch trực tuyến cũng ngày càng trở nên là thiết yếu trong nhiều thị trường khác nhau. Đối với thị trường xe máy thì rất ít nền tảng tồn tại để phục vụ cho nhu cầu này. Người dùng khi thực hiện giao dịch truyền thống sẽ gặp một số vấn đề như: thiếu sự minh bạch (xe cũ nhưng tua đồng hồ Odometer ít lại), khó khăn trong việc tìm kiếm chiếc xe mong muốn, khó giao tiếp được với người khác.

– Hiện nay nếu người dùng muốn mua, bán xe máy thì thường phải thực hiện thông qua các mạng xã hội, các diễn đàn hoặc nền tảng thương mại tổng hợp trực tuyến. Những nơi mà không phải chuyên để bán xe nên những tính năng hỗ trợ mua bán xe máy thường là không có, hoặc đơn giản, rất ít không phù hợp cho xe máy vì cần tính pháp lý và kỹ thuật.

– Từ hiện trạng đó những ý tưởng về RevNow bắt đầu được hình thành nhằm giải quyết tất cả vấn đề còn tồn đọng trong thị trường giao dịch xe máy. Cung cấp cho người mua tính năng thiết thực như chức năng lọc và tìm kiếm xe máy nâng cao, đối với người mua cũng được cung cấp cho tính năng đăng bán xe của mình chi tiết hơn, có thể nhập thông tin cụ thể hơn. Ngoài ra còn có chức năng nhắn tin giữa các người dùng với nhau giúp cho việc kết nối, giao tiếp trở nên dễ dàng hơn.

1.2. Mục tiêu cần đạt được

– Cung cấp tính năng đăng bán cho người bán một cách dễ dàng và chi tiết, bao gồm các thông tin kỹ thuật, tình trạng, giấy tờ, giá cả. Đồng thời đối với người cần mua thì sẽ cung cấp cho họ tính năng lọc và tìm kiếm nâng cao hơn, giúp người mua dễ dàng tìm kiếm chiếc xe hợp nhu cầu, sở thích.

– Tích hợp nhắn tin cho các người dùng để dễ dàng trao đổi, thương lượng và giải quyết thắc mắc.

– Tạo hệ thống đánh giá uy tín người dùng, cho phép người dùng đánh giá người dùng khác để tăng tính minh bạch và giúp người mua tìm được người bán uy tín.

– Tìm cách để xác thực người dùng hiệu quả hơn, giảm thiểu rủi ro gian lận, đảm bảo các bên đều là người thật việc thật.

– Giao diện trực quan, dễ sử dụng nhưng cũng phải đẹp, nổi bật hơn các nền tảng khác.

1.3. Phạm vi nghiên cứu

1.3.1. Thị trường xe máy trực tuyến

- Nghiên cứu nhu cầu và thị trường xe máy để từ đó xác định các xu hướng tiêu dùng và các vấn đề thường gặp trong giao dịch mua bán xe máy.
- Tham khảo và rút ra các điểm yếu của các nền tảng khác.

1.3.2. Công nghệ phát triển hệ thống

- Tìm hiểu về các công nghệ frontend hiện đại như React và TailwindCSS, cũng như cách để xây dựng giao diện thân thiện, dễ sử dụng, và cách tối ưu hóa trải nghiệm người dùng.
- Nghiên cứu, tìm tòi về Supabase để bảo mật và xác thực người dùng và triển khai hệ thống backend hiệu quả. Và nghiên cứu về PostgreSQL và cách sử dụng trong Supabase để quản lý dữ liệu người dùng và các giao dịch mua bán xe máy.

1.3.3. Quản lý xe máy và giao dịch

- Tìm cách để quản lý xe máy một cách thuận tiện, tối ưu và đồng đều giữa các xe máy, bảo đảm không có dữ liệu ngoài luồng.
- Thiết kế quy trình đăng bán và mua cho đến khi giao dịch một cách trơn tru, nhanh nhẹn để tối ưu hoá trải nghiệm người dùng.

1.3.4. Tìm kiếm và lọc sản phẩm

- Nghiên cứu việc phát triển các tính năng tìm kiếm nâng cao và bộ lọc sản phẩm, cho phép người dùng dễ dàng tìm kiếm các sản phẩm xe máy dựa trên nhiều tiêu chí khác nhau như giá cả, hãng xe, loại xe, tình trạng, phân khối, và số km đã đi.

1.3.5. Người dùng và tối ưu hoá

- Nghiên cứu và thử nghiệm các phương pháp tối ưu hóa trải nghiệm người dùng, bao gồm giao diện thân thiện, dễ sử dụng, cùng với việc tối ưu hóa tốc độ tải trang và khả năng phản hồi của hệ thống.
- Xây dựng hệ thống đánh giá độ uy tín của người bán, giúp người mua có thể đưa ra quyết định mua hàng an toàn và đáng tin cậy.

1.3.6. Phạm vi địa lý và mở rộng nền tảng

- Ban đầu, nền tảng sẽ phục vụ cho thị trường xe máy ở châu Âu, nhưng sẽ có kế hoạch mở rộng và áp dụng mô hình giao dịch cho các quốc gia khác bao gồm cả nước Việt Nam.
- Phạm vi mở rộng sẽ bao gồm việc tích hợp các ngôn ngữ khác nhau và điều chỉnh nền tảng để phù hợp với các yêu cầu và quy định địa phương của các quốc gia khác nhau.

1.4. Phương pháp nghiên cứu

1.4.1 Nghiên cứu tài liệu và cơ sở lý thuyết

– Tìm hiểu các tài liệu học thuật, bài báo, báo cáo và tài liệu hướng dẫn chính thức liên quan đến:

- + Mô hình thương mại điện tử C2C (Customer to Customer).
- + Hành vi người dùng khi mua bán xe máy trực tuyến.
- + Các xu hướng thiết kế giao diện người dùng (UI/UX) trong các nền tảng hiện đại.
- + Các công nghệ được sử dụng trong dự án như React, TailwindCSS, Supabase, PostgreSQL, cùng các thư viện hỗ trợ React.

1.4.2 Phân tích và khảo sát hệ thống thực tế

– Thực hiện khảo sát và phân tích một số nền tảng mua bán xe máy phổ biến hiện nay tại Việt Nam như Chợ Tốt, Muaban.net, OKXE, Và các nền tảng nước ngoài như Cycletrader,... nhằm:

- + Rút ra điểm mạnh, điểm yếu.
- + Xác định nhu cầu và hành vi người dùng.
- + Tìm hiểu cách thức các nền tảng này tổ chức luồng giao dịch và giao diện người dùng.

1.4.3 Thiết kế hệ thống và mô hình dữ liệu

- Áp dụng các mô hình thiết kế như:
- + Mô hình MCD (Model Conceptual Data) để thiết kế cơ sở dữ liệu trên PostgreSQL.

1.4.4 Kiểm thử và đánh giá

- Kiểm thử chức năng (Functional Testing): Đảm bảo các tính năng như đăng nhập, đăng bài, tìm kiếm, nhắn tin, lọc sản phẩm,... hoạt động đúng yêu cầu.
- Kiểm thử giao diện (UI Testing): Đảm bảo tính nhất quán, thân thiện với người dùng và hoạt động tốt trên nhiều thiết bị.
- Kiểm thử hiệu năng (Performance Testing): Kiểm tra tốc độ phản hồi, khả năng xử lý dữ liệu lớn.
- Thu thập phản hồi từ người dùng thử nghiệm để tiếp tục cải tiến.

1.5. Hướng giải quyết

– Thiết kế giao diện người dùng trực quan và thân thiện, cho phép người dùng dễ dàng đăng tin, tìm kiếm, và quản lý các giao dịch. Giao diện sẽ được xây dựng trên nền tảng React kết hợp với TailwindCSS, đảm bảo tính linh hoạt, responsive và dễ sử dụng trên nhiều thiết bị. Các chức năng chính bao gồm: đăng ký/dăng nhập, đăng bán xe máy, tìm kiếm và lọc xe máy theo nhiều tiêu chí, giao dịch xe máy, cùng với hệ thống nhắn tin tích hợp để kết nối người mua và người bán.

– Sử dụng Supabase làm nền tảng backend kết hợp với cơ sở dữ liệu PostgreSQL để lưu trữ và quản lý thông tin người dùng, sản phẩm và giao dịch. Hệ thống backend sẽ đảm bảo tính bảo mật, xác thực người dùng hiệu quả (đăng nhập qua Google và email) và xử lý giao dịch nhanh chóng. Các API được xây dựng sẽ hỗ trợ các thao tác CRUD cho người dùng và sản phẩm, đồng thời tích hợp các thư viện React như Supabase và React Query để tối ưu hóa hiệu suất và quản lý trạng thái ứng dụng.

– Triển khai các tính năng hỗ trợ giao dịch trực tuyến hiệu quả, bao gồm:

+ Hệ thống nhắn tin thời gian thực giúp người mua và người bán trao đổi thông tin, thương lượng và hẹn gặp kiểm tra xe.

+ Cơ chế đánh dấu sản phẩm "đã bán" sau khi giao dịch hoàn tất, đảm bảo thông tin được cập nhật liên tục và chính xác.

+ Tích hợp hệ thống đánh giá độ uy tín, cho phép người dùng đánh giá người dùng khác, góp phần xây dựng một môi trường giao dịch an toàn và đáng tin cậy.

– Hệ thống sẽ hiển thị kết quả giao dịch và các thông tin chi tiết về sản phẩm một cách rõ ràng và dễ hiểu, hỗ trợ người dùng theo dõi lịch sử giao dịch và đánh giá sản phẩm. Các kết quả, bao gồm thông tin sản phẩm, đánh giá uy tín và trạng thái giao dịch, sẽ được lưu trữ và xuất ra dưới dạng báo cáo thống kê, giúp người dùng và quản trị viên có cái nhìn tổng quan về hoạt động của nền tảng.

1.6. Kế hoạch thực hiện

– Tuần 1: Tìm hiểu và nghiên cứu các công nghệ cần thiết cho dự án bao gồm Vite, React, TailwindCSS cho frontend, Supabase cho backend và cơ sở dữ liệu. Ngoài ra, các thư viện hỗ trợ như React Query, React Router, Axios,... cũng được xem xét và học cách tích hợp.

– Tuần 2: Tiến hành thiết kế kiến trúc tổng thể của hệ thống, phác thảo giao diện người dùng với các trang chức năng chính như: trang chủ, trang tìm kiếm, trang chi tiết sản phẩm, trang đăng bán, trang đăng nhập/đăng ký,... Đồng thời lên bố cục cơ sở dữ liệu và cách tổ chức các bảng dữ liệu trên Supabase.

– Tuần 3: Bắt đầu lập trình giao diện người dùng với React và TailwindCSS. Xây dựng các phần cơ bản như trang chủ, hệ thống điều hướng (navigation), trang đăng nhập và đăng ký người dùng.

– Tuần 4: Triển khai tính năng đăng bán sản phẩm, cho phép người dùng điền thông tin chi tiết về xe máy và tải lên hình ảnh. Xây dựng phần hiển thị danh sách sản phẩm với chức năng lọc theo các tiêu chí như hãng xe, loại xe, giá, tình trạng,...

– Tuần 5: Kết nối với Supabase để xử lý chức năng xác thực người dùng (authentication), lưu trữ và truy vấn dữ liệu sản phẩm. Tích hợp các API để hiển thị dữ liệu động từ cơ sở dữ liệu lên giao diện người dùng.

– Tuần 6: Xây dựng hệ thống nhắn tin thời gian thực giữa người mua và người bán. Phát triển chức năng đánh giá độ uy tín người dùng sau mỗi giao dịch. Bổ sung chức năng đánh dấu sản phẩm đã bán để cập nhật trạng thái sản phẩm.

– Tuần 7: Thực hiện kiểm thử hệ thống để phát hiện và sửa các lỗi phát sinh. Tối ưu hiệu năng ứng dụng, đảm bảo tính ổn định và mượt mà trong quá trình sử dụng. Kiểm tra tương thích trên nhiều thiết bị và trình duyệt.

– Tuần 8: Hoàn thiện phần báo cáo , trình bày lại quy trình phát triển, phân tích hệ thống, giao diện, tính năng và đánh giá kết quả đạt được. Chuẩn bị nội dung đề báo cáo, và nộp sản phẩm hoàn chỉnh.

CHƯƠNG II. CƠ SỞ LÝ THUYẾT

2.1. Mô hình kinh doanh C2C

– C2C, hay "Consumer to Consumer", là một mô hình kinh doanh cho phép các cá nhân tự mình trao đổi hàng hóa và dịch vụ thông qua các nền tảng trực tuyến. Khác với mô hình truyền thống, trong đó doanh nghiệp cung cấp sản phẩm cho người tiêu dùng, mô hình C2C tạo điều kiện cho việc giao dịch trực tiếp giữa người tiêu dùng với nhau qua các website hoặc ứng dụng chuyên dụng kết nối các cá nhân.

– Đặc điểm của mô hình này:

+ Giao dịch trực tiếp giữa cá nhân.

+ Trong mô hình này, giao dịch được thực hiện trực tiếp giữa người mua và người bán, không cần qua trung gian sản xuất hay bán lẻ. Việc trao đổi thông tin, thương lượng giá cả và thanh toán được thực hiện trực tiếp giữa các cá nhân, giúp cắt giảm chi phí và tối ưu hóa hiệu quả giao dịch.

+ Nền tảng trực tuyến: Các giao dịch C2C thường diễn ra trên các sàn giao thương điện tử, trang web rao vặt hay qua các mạng xã hội. Nhờ vậy, người bán có thể tiếp cận với một lượng lớn khách hàng tiềm năng, trong khi người mua có cơ hội dễ dàng so sánh và lựa chọn sản phẩm từ nhiều nguồn khác nhau.

+ Đa dạng về mặt sản phẩm: Mô hình C2C cho phép giao dịch nhiều loại sản phẩm, từ hàng mới đến hàng đã qua sử dụng, từ hàng thủ công đến sản phẩm công nghiệp. Quá trình đăng bán sản phẩm được tối giản, không cần qua nhiều thủ tục phức tạp, qua đó tạo ra sự đa dạng cho thị trường và đáp ứng được nhiều nhu cầu của người tiêu dùng.

+ Giá cả cạnh tranh: Vì loại bỏ các chi phí trung gian, giá bán trong mô hình C2C thường cạnh tranh hơn so với các mô hình kinh doanh truyền thống. Người mua có thể dễ dàng so sánh giá giữa các nhà bán khác nhau, từ đó tiết kiệm chi phí và mua được sản phẩm với giá ưu đãi nhất.

+ Tính mở và dễ tiếp cận: Bất kỳ ai chỉ cần có kết nối Internet cũng có thể tham gia vào mô hình C2C. Việc đăng ký tài khoản và bắt đầu bán hàng được thực hiện nhanh chóng, đơn giản và thường miễn phí, tạo ra một sân chơi bình đẳng cho mọi cá nhân với cơ hội kinh doanh rộng lớn.

– Lợi ích mà mô hình này mang lại:

+ Mở rộng phạm vi tiếp cận khách hàng: Mô hình C2C loại bỏ rào cản địa lý, cho phép người bán dễ dàng tiếp cận khách hàng trên toàn cầu thông qua Internet. Các nền tảng thương mại điện tử như Shopee, Lazada, Tiki, ... thu hút lượng truy cập lớn hàng ngày, giúp sản phẩm được phơi bày trước một lượng khách hàng tiềm năng đáng kể, từ đó tăng cơ hội giao dịch thành công.

+ Tiết kiệm chi phí vận hành: Với mô hình này, người bán không phải chịu các chi phí cố định như thuê mặt bằng, quản lý cửa hàng hay chi trả tiền lương cho nhân viên. Việc bán hàng trực tuyến chủ yếu đòi hỏi một khoản đầu tư ban đầu cho

website, quảng cáo và chi phí vận chuyển, giúp giảm thiểu rủi ro tài chính và tối ưu hóa quy trình giao dịch cho cả hai bên.

+ Cơ hội kinh doanh cho mọi đối tượng: C2C mở ra cánh cửa cho bất kỳ ai mong muốn tham gia vào hoạt động kinh doanh mà không cần nhiều vốn hay kinh nghiệm. Chỉ cần có sản phẩm và kết nối Internet, cá nhân có thể trở thành người bán trực tuyến. Điều này tạo điều kiện cho việc tạo thêm nguồn thu nhập cho từng người cũng như giúp các doanh nghiệp nhỏ tiếp cận thị trường rộng lớn, từ đó tăng doanh số và lợi nhuận.

+ Trải nghiệm mua sắm thuận tiện và đa dạng: Người mua có thể dễ dàng mua sắm bất cứ lúc nào và từ bất cứ đâu, so sánh giá và lựa chọn sản phẩm ưng ý nhất. Đồng thời, người bán cũng có thể quản lý đơn hàng và giao dịch một cách hiệu quả thông qua các công cụ trực tuyến, đồng hành cùng đó là sự đa dạng về sản phẩm với các mức giá phù hợp với nhiều nhu cầu và sở thích khác nhau.

+ Thúc đẩy sự cạnh tranh và đổi mới: Mô hình C2C không giới hạn người bán chỉ hoạt động trong một lĩnh vực cụ thể, giúp họ có thể đa dạng hóa danh mục sản phẩm hoặc dịch vụ. Các nền tảng liên tục cập nhật xu hướng thị trường, tạo điều kiện cho người bán nắm bắt nhanh chóng nhu cầu của khách hàng và cung cấp các sản phẩm phù hợp, từ đó gia tăng lợi thế cạnh tranh.

2.2. Mô hình hoá dữ liệu

2.2.1. Thực thể và thuộc tính

– Thực thể là một đối tượng hoặc khái niệm cụ thể trong thế giới thực mà chúng ta cần lưu trữ thông tin. Mỗi thực thể thường đại diện cho một bảng (table) trong cơ sở dữ liệu, ví dụ như SinhVien, GiaoDich, NhanVien hay SanPham. Thực thể bao gồm nhiều thuộc tính, là các đặc điểm hoặc thông tin cụ thể mô tả thực thể đó. Thuộc tính chính là các cột (column) trong bảng, chẳng hạn như tên, ngày sinh, địa chỉ, số điện thoại, mã số định danh, v.v. Mỗi thực thể có thể có một hoặc nhiều thuộc tính, và các thuộc tính này mang dữ liệu cụ thể về các đối tượng trong thực tế.

– Một yếu tố cốt lõi trong việc thiết kế cơ sở dữ liệu là xác định khóa chính (Primary Key) và khóa phụ (Foreign Key). Khóa chính là một thuộc tính, hoặc tập hợp các thuộc tính, dùng để định danh duy nhất mỗi bản ghi (record) trong một bảng. Không có hai bản ghi nào được phép có cùng giá trị ở khóa chính, và giá trị của khóa chính không được để trống (NOT NULL). Ví dụ, trong bảng SinhVien, mã sinh viên (MaSV) có thể được chọn làm khóa chính vì mỗi sinh viên có một mã duy nhất. Việc xác định đúng khóa chính là rất quan trọng để đảm bảo tính toàn vẹn dữ liệu và giúp truy vấn dữ liệu hiệu quả hơn.

– Ngược lại, khóa phụ là một thuộc tính trong bảng này nhưng tham chiếu tới khóa chính của một bảng khác. Mục đích của khóa phụ là để thể hiện mối quan hệ giữa các bảng với nhau. Ví dụ, trong bảng GiaoDich có thể có thuộc tính MaKH là khóa phụ, liên kết tới bảng KhachHang để chỉ ra rằng giao dịch đó thuộc về khách

hàng nào. Khóa phụ giúp duy trì mối liên kết logic giữa các thực thể, hỗ trợ việc xây dựng các hệ thống dữ liệu có tính liên kết chặt chẽ và chính xác.

- Mỗi thuộc tính đều phải có 1 kiểu dữ liệu, đây là định dạng mà giá trị của thuộc tính đó được lưu trữ trong cơ sở dữ liệu. Việc xác định đúng kiểu dữ liệu không chỉ giúp tối ưu hóa việc lưu trữ mà còn đảm bảo độ chính xác và hiệu quả khi truy vấn dữ liệu. Một số kiểu dữ liệu phổ biến gồm có: INT hoặc INTEGER cho số nguyên, VARCHAR(n) cho chuỗi ký tự có độ dài tối đa n ký tự, TEXT cho chuỗi dài, DATE cho ngày tháng, TIMESTAMP cho ngày giờ đầy đủ, BOOLEAN cho giá trị đúng/sai. Ngoài ra, có thể sử dụng FLOAT hoặc DECIMAL cho các giá trị số thực, đặc biệt khi cần lưu trữ số có phần thập phân như giá tiền hoặc các đơn vị đo lường chính xác.

2.2.2. MCD

- Mô hình dữ liệu khái niệm (Conceptual Data Model) là một mô hình trừu tượng và tổng quan về dữ liệu trong một hệ thống thông tin. Tập trung vào việc mô tả các khái niệm, mối quan hệ và luồng thông tin chính trong một hệ thống, đồng thời chi tiết hóa về cách dữ liệu được lưu trữ hoặc triển khai.

- Mô hình này không phụ thuộc vào bất kỳ hệ quản trị cơ sở dữ liệu cụ thể nào và không liên quan đến cấu trúc vật lý hay các đặc thù kỹ thuật. Mà sẽ tạo ra một cấu trúc dữ liệu trừu tượng được sử dụng để diễn giải và trình bày các khái niệm và mối quan hệ giữa chúng trong hệ thống.

- Các yếu tố có trong mô hình dữ liệu khái niệm này như thực thể (entities), thuộc tính (attributes), mối quan hệ (relationships) và các ràng buộc (constraints) để mô tả dữ liệu và các mối quan hệ chính trong hệ thống. Được biểu diễn bằng cách sử dụng các biểu đồ Entity-Relationship (ER) hoặc các biểu đồ tương tự.

- Ví dụ, trong một mô hình dữ liệu khái niệm cho hệ thống quản lý bán hàng, sẽ có các thực thể như “Khách hàng” (Customers), “Sản phẩm” (Products), “Đơn hàng” (Orders) và các mối quan hệ như “Một khách hàng có thể đặt nhiều đơn hàng” hoặc “Một đơn hàng có chứa nhiều sản phẩm”.

- Mô hình dữ liệu khái niệm giúp thiết kế ban đầu của hệ thống thông tin, làm rõ các khái niệm quan trọng và mối quan hệ giữa chúng. Nó cung cấp một cơ sở chung để hiểu và trao đổi thông tin với các bên liên quan trong quá trình phát triển hệ thống. Mô hình dữ liệu khái niệm sau đó có thể được dùng như một cơ sở để phát triển các mô hình dữ liệu logic (Logical Data Model) và mô hình dữ liệu vật lý (Physical Data Model).

2.2.3. MLD

- Mô hình dữ liệu logic (Logical Data Model) là một biểu đồ hoặc mô tả trừu tượng về cấu trúc dữ liệu và quan hệ giữa chúng mà không phụ thuộc vào hệ quản trị cơ sở dữ liệu (DBMS) cụ thể. Mô hình này tập trung vào cách dữ liệu được tổ chức,

mô tả các thực thể (entities), các thuộc tính (attributes) và mối quan hệ (relationships) giữa các thực thể đó.

- Các thành phần có trong mô hình dữ liệu logic như: Các thực thể (entities), thuộc tính (attributes), khóa (keys), và mối quan hệ (relationships) để mô tả dữ liệu và mối quan hệ giữa các thực thể đó.

- Ví dụ, một mô hình dữ liệu logic cho một hệ thống quản lý thư viện có thể có các thực thể như “Sách” (Books), “Tác giả” (Authors), “Thể loại” (Genres), và các mối quan hệ như “Một sách có thể có nhiều tác giả” hoặc “Một sách thuộc một thể loại duy nhất”.

2.2.4. MPD

- Mô hình dữ liệu vật lý (Physical Data Model) là mô hình tập trung vào các khía cạnh về cấu trúc lưu trữ dữ liệu trong cơ sở dữ liệu. Định nghĩa các đối tượng vật lý như bảng, cột, chỉ mục và quan hệ giữa chúng. Cung cấp chi tiết về cách dữ liệu được lưu trữ trên ổ đĩa hoặc hệ thống lưu trữ.

- Mô hình này là một phần của quá trình thiết kế cơ sở dữ liệu trong hệ thống thông tin, cách dữ liệu được tổ chức, lưu trữ và truy cập trong một hệ thống cụ thể.

- Một mô hình dữ liệu vật lý có thể chỉ định kiểu dữ liệu và kích thước của mỗi cột, chỉ mục để truy vấn dễ dàng, có tính ràng buộc để đảm bảo tính toàn vẹn dữ liệu, các thông số về việc lưu trữ dữ liệu như dung lượng ổ đĩa, phân vùng, v.v.

- Ví dụ, trong mô hình dữ liệu vật lý, một bảng “Nhân viên” có thể được định nghĩa với các cột như “ID nhân viên” (kiểu số nguyên), “Họ” (kiểu chuỗi), “Tên” (kiểu chuỗi), và chỉ mục trên cột “ID nhân viên” để tìm kiếm nhanh hơn.

2.3. Định nghĩa Website

- Website là một hệ thống thông tin số hóa được xây dựng bằng các ngôn ngữ lập trình, kịch bản, và công nghệ web (HTML, CSS, JavaScript, v.v.), hoạt động như một giao diện tương tác giữa tổ chức/cá nhân với người dùng thông qua môi trường Internet. Cấu trúc của website bao gồm nhiều trang con (web pages) liên kết với nhau bằng siêu liên kết (hyperlink), được lưu trữ trên máy chủ web (web server) và phân phối đến người dùng thông qua giao thức HTTP/HTTPS. Mỗi website được định danh duy nhất bằng tên miền (domain name) và địa chỉ IP (Internet Protocol), cho phép truy cập từ mọi thiết bị có kết nối mạng thông qua trình duyệt (Chrome, Firefox, Safari).

- Cấu trúc cơ bản của một Website bao gồm:

- 2.1. Giao diện người dùng (Frontend): Đây là lớp trình bày trực tiếp tiếp xúc với người dùng cuối, đóng vai trò như cầu nối giữa hệ thống và người sử dụng thông qua các thành phần quen thuộc như HTML để xây dựng khung sườn nội dung, CSS để tạo nên diện mạo trực quan và sinh động, còn JavaScript đảm nhiệm việc xử lý các thao tác tương tác một cách linh hoạt và mượt mà. Một điểm nổi bật của giao diện người dùng chính là việc áp dụng thiết kế đáp ứng (Responsive Design) — một hệ

thông minh có khả năng tự điều chỉnh bố cục và hiển thị sao cho phù hợp với mọi kích thước màn hình, từ máy tính để bàn cho đến điện thoại di động, dựa trên các quy tắc do nhà thiết kế định sẵn. Bên cạnh đó, giao diện còn được tối ưu bằng các kỹ thuật như tương thích trình duyệt (Cross-Browser Compatibility) nhằm đảm bảo trải nghiệm nhất quán trên các nền tảng trình duyệt khác nhau, và tối ưu hiệu năng (Performance Optimization) với các giải pháp như tải chậm (Lazy Loading) và phân tách mã nguồn (Code Splitting) để rút ngắn thời gian tải trang, nâng cao trải nghiệm người dùng.

2.2. Xử lý dữ liệu (Backend): Đây là nơi vận hành các chức năng cốt lõi của hệ thống, đảm nhận vai trò của lớp nghiệp vụ (Business Logic Layer) và lớp dữ liệu (Data Layer), hoạt động phía sau hậu trường để xử lý các yêu cầu từ người dùng. Thành phần chính trong lớp này là máy chủ (Server) – nơi tiếp nhận và xử lý các yêu cầu được gửi đến từ giao diện người dùng thông qua các điểm cuối API (API Endpoints), từ đó phản hồi lại dữ liệu phù hợp. Hệ thống còn tích hợp cơ sở dữ liệu (Database) – bộ nhớ trung tâm lưu trữ toàn bộ thông tin quan trọng như dữ liệu người dùng, sản phẩm, đơn hàng và giao dịch. Bên cạnh đó, yếu tố xác thực và bảo mật cũng được chú trọng, với các cơ chế như quản lý đăng nhập, phân quyền truy cập, mã hóa dữ liệu và bảo vệ thông tin cá nhân, đảm bảo hệ thống vận hành an toàn và đáng tin cậy.

2.3. API (Application Programming Interface)

2.3.1. Khái niệm API

- API là một tập hợp các quy tắc hoặc giao thức giúp các ứng dụng phần mềm giao tiếp với nhau để trao đổi dữ liệu, tính năng và chức năng.

- API đơn giản hóa và tăng tốc quá trình phát triển ứng dụng và phần mềm bằng cách cho phép các nhà phát triển tích hợp dữ liệu, dịch vụ và khả năng từ các ứng dụng khác, thay vì phải phát triển chúng từ đầu. APIs cũng cung cấp cho chủ sở hữu ứng dụng một cách đơn giản và an toàn để làm cho dữ liệu và chức năng của ứng dụng có sẵn cho các phòng ban trong tổ chức của họ. Chủ sở hữu ứng dụng cũng có thể chia sẻ hoặc tiếp thị dữ liệu và chức năng cho các đối tác kinh doanh hoặc bên thứ ba.

- API cho phép chia sẻ chỉ những thông tin cần thiết, giữ kín các chi tiết hệ thống nội bộ khác, điều này giúp cải thiện bảo mật hệ thống. Các máy chủ hoặc thiết bị không cần phải tiết lộ toàn bộ dữ liệu — APIs cho phép chia sẻ các gói dữ liệu nhỏ, liên quan đến yêu cầu cụ thể.

2.3.2. Cách hoạt động của API

- Để dễ hình dung về giao tiếp giữa API thì nên xem nó như quá trình trao đổi yêu cầu và phản hồi giữa client (ứng dụng yêu cầu) và server (máy chủ phản hồi). Ứng dụng gửi yêu cầu gọi là client, và server sẽ trả về kết quả tương ứng. API chính là “cầu nối” giúp hai bên này giao tiếp được với nhau.

– Một ví dụ đơn giản để hiểu cách API hoạt động là thông qua dịch vụ thanh toán từ bên thứ ba. Ví dụ, khi mua hàng trên một website thương mại điện tử, có thể bạn sẽ thấy nút "Thanh toán qua Momo" hoặc một dịch vụ khác. Tính năng này được xây dựng dựa trên API để kết nối đến hệ thống thanh toán đó.

– Quy trình cụ thể thường diễn ra như sau:

+ Khi người dùng bấm vào nút thanh toán, một API call (lời gọi API) sẽ được gửi đi để lấy thông tin cần thiết. Đây chính là yêu cầu (request). Yêu cầu này sẽ đi từ ứng dụng đến web server thông qua URI (Uniform Resource Identifier) của API và thường sẽ kèm theo HTTP verb (như GET, POST...), header, và đôi khi là request body (nội dung yêu cầu).

+ Khi API nhận được yêu cầu hợp lệ từ website sản phẩm, nó sẽ chuyển tiếp yêu cầu đó đến hệ thống bên thứ ba, ví dụ hệ thống thanh toán như Momo.

+ Server của hệ thống thanh toán sẽ xử lý và gửi phản hồi (response) lại cho API với dữ liệu được yêu cầu.

+ API sau đó sẽ chuyển dữ liệu phản hồi đó về lại cho ứng dụng gốc, tức là website bán hàng ban đầu.

2.3.3. Các dạng Web API

– Web API cho phép truyền dữ liệu và chức năng qua internet bằng giao thức HTTP.

– Ngày nay, phần lớn API đều là Web API. Web API là một dạng của Remote API nghĩa là API này tương tác với các tài nguyên bên ngoài thông qua các giao thức chuẩn, và cung cấp dữ liệu hoặc chức năng của ứng dụng ra bên ngoài qua Internet.

– Bốn loại Web API phổ biến:

+ Open APIs (API mở) Là các API mã nguồn mở mà bất kỳ ai cũng có thể truy cập qua giao thức HTTP. Còn được gọi là Public APIs (API công khai), chúng thường có tài liệu rõ ràng, bao gồm các endpoint (đường dẫn gọi API), định dạng yêu cầu và phản hồi cụ thể.

+ Partner APIs (API đối tác) Là API được thiết kế để kết nối với các đối tác kinh doanh chiến lược. Thông thường, các lập trình viên có thể truy cập thông qua cổng API công khai (public developer portal), nhưng cần trải qua quy trình đăng ký, phê duyệt và nhận thông tin đăng nhập để sử dụng.

+ Internal APIs (API nội bộ) Đây là các API chỉ được sử dụng trong nội bộ công ty, không công khai ra bên ngoài. Mục đích của chúng là để nâng cao hiệu suất làm việc và hỗ trợ giao tiếp giữa các nhóm phát triển trong tổ chức.

+ Composite APIs (API tổng hợp) Composite API kết hợp nhiều API khác nhau vào một lời gọi duy nhất. Tức là chỉ cần một lần gọi API, bạn có thể nhận được dữ liệu từ nhiều nguồn khác nhau. Loại API này đặc biệt hữu ích trong kiến trúc microservices, nơi một tác vụ đơn lẻ có thể cần dữ liệu từ nhiều dịch vụ nhỏ khác nhau.

2.3.4. Giao thức API

- Có rất nhiều giao thức API khác được phát triển như SOAP, RPC, XML-RPC, JSON-RPC, gRPC, WebSocket, GraphQL. Nhưng dự án chỉ sử dụng REST (representational state transfer) nên sẽ tập trung vào giao thức này.
- REST là một tập hợp các nguyên tắc kiến trúc dùng cho Web API. Các REST API (còn gọi là RESTful API) là những API tuân theo các nguyên tắc này.
- REST API sử dụng các HTTP method phổ biến như:
 - + GET – để lấy dữ liệu
 - + PUT – để cập nhật dữ liệu
 - + DELETE – để xóa dữ liệu
 - + HEAD – để lấy metadata
- REST tổ chức dữ liệu dưới dạng tài nguyên (resources), và mỗi tài nguyên sẽ có một địa chỉ duy nhất gọi là URI. Khi client muốn lấy dữ liệu, nó chỉ cần gửi yêu cầu (request) đến đúng URI của tài nguyên đó.
- Ngoài ra REST API có đặc điểm là stateless (Không trạng thái) tức không lưu trạng thái của client giữa các lần gọi API. Mỗi yêu cầu từ client đều độc lập và phải chứa đầy đủ thông tin để server xử lý.

2.4. JSON Web Token (JWT)

2.3.1. Định nghĩa

– JSON Web Token (JWT) là một tiêu chuẩn mở (RFC 7519) được dùng để để truyền tải thông tin một cách an toàn giữa các bên dưới dạng đối tượng JSON. Thông tin này có thể được xác minh và tin cậy bởi vì nó được ký số. JWT có thể được ký bằng cách sử dụng một khóa bí mật (với thuật toán HMAC) hoặc một cặp khóa công khai/riêng tư sử dụng RSA hoặc ECDSA.

– Các token được ký có thể xác minh tính toàn vẹn của các thông tin (claims) chứa bên trong, trong khi các token được mã hóa giúp che giấu những thông tin đó khỏi các bên khác. Khi các token được ký sử dụng cặp khóa công khai/riêng tư, chữ ký cũng chứng nhận rằng chỉ riêng bên nắm giữ khóa riêng mới là bên đã thực hiện ký.

2.3.2. Lợi ích mang lại

– Ủy quyền (Authorization): Đây là tình huống phổ biến nhất khi sử dụng JWT. Khi người dùng đã đăng nhập, mỗi yêu cầu tiếp theo sẽ bao gồm JWT, cho phép người dùng truy cập vào các tuyến, dịch vụ và tài nguyên được phép với token đó. Single Sign On là một tính năng hiện nay sử dụng JWT rất phổ biến nhờ vào dung lượng nhỏ và khả năng dễ dàng sử dụng trên nhiều miền khác nhau.

– Trao đổi thông tin (Information Exchange): JSON Web Tokens là một cách hiệu quả để truyền tải thông tin một cách an toàn giữa các bên. Vì JWT có thể được ký — ví dụ, sử dụng cặp khóa công khai/riêng tư — bạn có thể chắc chắn rằng người

gửi là đúng như họ khai báo. Ngoài ra, vì chữ ký được tính dựa trên phần tiêu đề và nội dung, bạn cũng có thể xác minh rằng nội dung chưa bị thay đổi.

2.3.3. Cấu trúc của một JWT

– Một JWT sẽ có cấu trúc chia làm 3 phần tách nhau bởi dấu chấm (.), là: Header.Payload.Signature vì vậy một JWT thường là xxxxx.yyyyy.zzzzz

– **Header** phần đầu tiên của token bao gồm hai thành phần chính là loại của token tức là JWT, và thuật toán để ký, chẳng hạn như HMAC SHA256 hoặc RSA

– Ví dụ: Ta có một file JSON như sau:

```
1. {  
2.   "alg": "HS256",  
3.   "typ": "JWT"  
4. }  
5.
```

– Sau đó, đoạn JSON này sẽ được mã hóa theo định dạng Base64Url để tạo thành phần đầu tiên của JWT.

– **Payload** là phần thứ hai của token, chứa các claim. Claim là các phát biểu về một thực thể (thường là người dùng) và dữ liệu bổ sung. Có ba loại claim: claim đã đăng ký, claim công khai và claim riêng.

+ Claim đã đăng ký (Registered claims): Đây là tập hợp các claim được định nghĩa sẵn, không bắt buộc nhưng được khuyến khích sử dụng để cung cấp các thông tin hữu ích và có khả năng tương tác cao. Một số claim tiêu biểu là: iss (người phát hành), exp (thời gian hết hạn), sub (chủ thể), aud (người nhận), và một số claim khác.

+ Claim công khai (Public claims): Những claim này có thể được định nghĩa tự do bởi những người sử dụng JWT. Tuy nhiên, để tránh trùng lặp, chúng nên được đăng ký trong IANA JSON Web Token Registry hoặc được định nghĩa dưới dạng URI chứa không gian tên chống trùng lặp.

+ Claim riêng (Private claims): Đây là các claim do người dùng tự định nghĩa nhằm chia sẻ thông tin giữa các bên đã thống nhất sử dụng chúng, và không thuộc nhóm claim đã đăng ký hay công khai.

– Ví dụ:

```
1. {  
2.   "sub": "1234567890",  
3.   "name": "John Doe",  
4.   "admin": true  
5. }  
6.
```

– Payload này sẽ được mã hóa theo định dạng Base64Url để tạo thành phần thứ hai của JSON Web Token.

– **Signature** là phần cuối cùng của token. Để tạo phần chữ ký, bạn cần lấy header đã mã hóa, payload đã mã hóa, một chuỗi bí mật (secret), thuật toán được chỉ định trong header, và thực hiện ký dữ liệu đó.

– Ví dụ, nếu muốn sử dụng thuật toán HMAC SHA256, chữ ký sẽ được tạo theo cách sau:

```

1. HMACSHA256(
2.   base64UrlEncode(header) + "." +
3.   base64UrlEncode(payload),
4.   secret)
5.

```

– Chữ ký được dùng để xác minh rằng nội dung của thông điệp không bị thay đổi trong quá trình truyền đi và, trong trường hợp các token được ký bằng khóa riêng, nó cũng có thể xác minh rằng người gửi JWT đúng là người mà họ tuyên bố.

– **Kết hợp ba phần lại** ta được kết quả cuối cùng là ba chuỗi được mã hóa theo định dạng Base64-URL và được ngăn cách bởi dấu chấm (.). Cấu trúc này có thể dễ dàng được truyền trong môi trường HTML và HTTP, đồng thời gọn nhẹ hơn so với các tiêu chuẩn dựa trên XML như SAML.

– Ví dụ về một JWT có chứa header và payload như đã trình bày trước đó, và được ký bằng một chuỗi bí mật:

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c

– Khi giải mã ta được kết quả sau:

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```

Signature Verified

SHARE JWT

Hình 2. 1 Kết quả của giải mã một JWT ví dụ

2.3.4. Cách thức hoạt động

– Trong quá trình xác thực, khi người dùng đăng nhập thành công bằng thông tin đăng nhập của họ, một JSON Web Token sẽ được trả về. Vì token đóng vai trò như thông tin xác thực, nên cần đặc biệt cẩn trọng để tránh các vấn đề bảo mật. Nói chung, không nên giữ token lâu hơn mức cần thiết.

14

– Cũng không nên lưu trữ dữ liệu phiên (session) nhạy cảm trong bộ nhớ trình duyệt (như localStorage hoặc sessionStorage) do thiếu tính bảo mật.

– Bất cứ khi nào người dùng muốn truy cập một tuyến (route) hoặc tài nguyên được bảo vệ, trình duyệt (user agent) nên gửi JWT, thường là trong header Authorization sử dụng chuẩn Bearer. Nội dung của header sẽ như sau:

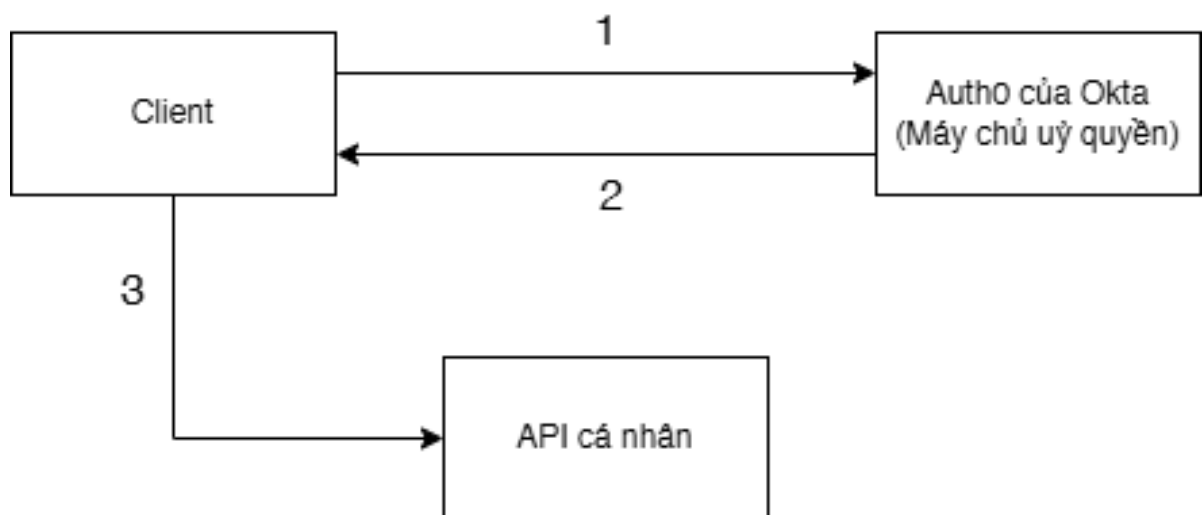
– Authorization: Bearer <token>

– Trong một số trường hợp, đây có thể là một cơ chế ủy quyền không trạng thái (stateless authorization). Các tuyến được bảo vệ trên server sẽ kiểm tra JWT hợp lệ trong header Authorization, và nếu tồn tại và hợp lệ, người dùng sẽ được phép truy cập các tài nguyên bảo vệ. Nếu JWT chứa các dữ liệu cần thiết, có thể giảm bớt việc truy vấn cơ sở dữ liệu cho một số thao tác — tuy nhiên điều này không phải lúc nào cũng đúng.

– Lưu ý rằng nếu bạn gửi token JWT qua HTTP header, bạn nên đảm bảo token không quá lớn. Một số máy chủ không chấp nhận header vượt quá 8 KB. Nếu bạn đang cố gắng nhồi nhét quá nhiều thông tin vào JWT — ví dụ như toàn bộ danh sách quyền của người dùng — thì có thể cần một giải pháp thay thế, chẳng hạn như Auth0 Fine-Grained Authorization.

– Nếu token được gửi trong header Authorization, thì vấn đề CORS (Cross-Origin Resource Sharing) sẽ không xảy ra vì không sử dụng cookie.

– Sơ đồ sau đây cho thấy cách JWT được lấy và sử dụng để truy cập các API hoặc tài nguyên:



– Giải thích:

+ Client yêu cầu quyền truy cập từ máy chủ ủy quyền (authorization server). Điều này được thực hiện thông qua một trong các luồng ủy quyền khác nhau.

+ Khi quyền truy cập được cấp, máy chủ cấp quyền trả về một token truy cập (access token) cho ứng dụng.

+ Client sử dụng token truy cập để truy cập tài nguyên bảo vệ (ví dụ như một API).

2.3.5. Lý do sử dụng

- Những lợi ích của JSON Web Tokens (JWT) khi so với Simple Web Tokens (SWT) và Security Assertion Markup Language Tokens (SAML).
- Vì JSON ít chi tiết hơn XML, khi được mã hóa, kích thước của nó cũng nhỏ hơn, làm cho JWT gọn nhẹ hơn SAML. Điều này khiến JWT trở thành một lựa chọn tốt để truyền tải trong môi trường HTML và HTTP.
- Về mặt bảo mật, SWT chỉ có thể được ký đối xứng bằng một khóa bí mật chung sử dụng thuật toán HMAC. Tuy nhiên, JWT và SAML token có thể sử dụng cặp khóa công khai/riêng tư dưới dạng chứng chỉ X.509 để ký. Việc ký XML với XML Digital Signature mà không tạo ra các lỗi hỏng bảo mật khó hiểu là rất khó khăn khi so với sự đơn giản của việc ký JSON.
- Các trình phân tích JSON (JSON parsers) phổ biến trong hầu hết các ngôn ngữ lập trình vì chúng ánh xạ trực tiếp tới các đối tượng. Ngược lại, XML không có ánh xạ tự nhiên từ tài liệu đến đối tượng. Điều này làm cho việc làm việc với JWT dễ dàng hơn so với các khai báo SAML.
- Về mặt sử dụng, JWT được sử dụng ở quy mô Internet. Điều này nhấn mạnh sự dễ dàng trong việc xử lý token JSON Web ở phía client trên nhiều nền tảng, đặc biệt là trên di động.

2.5. Công nghệ được sử dụng

2.2.1. Môi trường phát triển

- Hệ điều hành được sử dụng trong quá trình phát triển dự án là Windows 10 phiên bản 64-bit. Đây là nền tảng được lựa chọn nhờ khả năng tương thích rộng rãi với các công cụ lập trình hiện đại, hỗ trợ đa nhiệm mạnh mẽ, cùng tính năng bảo mật nâng cao như Windows Defender và Windows Sandbox.
- Về kiểm thử giao diện và chức năng, trình duyệt Firefox được ưu tiên sử dụng nhờ bộ công cụ dành cho nhà phát triển (DevTools) toàn diện, bao gồm chế độ Responsive Design Mode để tối ưu hóa hiển thị trên nhiều thiết bị, tính năng debug JavaScript chi tiết, và khả năng phân tích hiệu suất trang web thông qua Network Monitor và Performance Tool. Firefox cũng được đánh giá cao về việc tuân thủ các tiêu chuẩn web mới nhất (HTML5, CSS3), đảm bảo tính tương thích chéo cho ứng dụng.
- Công đoạn biên dịch và xây dựng ứng dụng được tối ưu hóa thông qua sự kết hợp giữa Node.js runtime environment và Vite build tool Node.js đóng vai trò nền tảng quản lý gói thư viện thông qua npm (Node Package Manager), đồng thời hỗ trợ thực thi các script server-side. Trong khi đó, Vite – công cụ build front-end thế hệ mới – giúp tăng tốc độ phát triển nhờ cơ chế Hot Module Replacement (HMR) cho phép cập nhật mã nguồn ngay lập tức mà không cần reload trang, cùng khả năng tối ưu hóa tài nguyên tĩnh (CSS, JavaScript, hình ảnh) thông qua tích hợp Rollup và ES

Modules. Sự kết hợp này không chỉ rút ngắn thời gian build mà còn hỗ trợ phát triển các ứng dụng Single-Page Application (SPA) một cách linh hoạt.

2.2.2. Công cụ phát triển

- **Figma** được sử dụng làm công cụ chính để thiết kế giao diện người dùng (UI/UX). Nền tảng này cho phép tạo wireframe, prototype tương tác, và thiết kế chi tiết với tính năng Auto Layout và Design Systems, giúp duy trì tính nhất quán về màu sắc, typography, và component trên mọi màn hình. Khả năng cộng tác thời gian thực (real-time collaboration) của Figma tối ưu hóa quy trình làm việc nhóm, cho phép các thành viên cùng chỉnh sửa, comment trực tiếp, và kiểm thử usability ngay trên trình duyệt.

- **Visual Studio Code** được lựa chọn làm môi trường soạn thảo mã nguồn chính nhờ hệ sinh thái extension phong phú (ESLint, Prettier, GitLens) và tích hợp sâu với các công cụ phát triển. Editor này hỗ trợ đa ngôn ngữ (JavaScript, TypeScript, Python), debugger trực quan, và terminal tích hợp, giúp tăng tốc độ code, kiểm tra lỗi, và triển khai ứng dụng liền mạch. Tính năng Live Share còn cho phép lập trình viên pair programming từ xa, nâng cao hiệu suất làm việc nhóm.

- **Git** kết hợp **GitHub** được triển khai để quản lý phiên bản mã nguồn (version control). Với quy trình Git Flow, hệ thống được tổ chức theo các nhánh chính (main, develop), nhánh feature, và nhánh hotfix, đảm bảo kiểm soát thay đổi mã nguồn chặt chẽ. GitHub đóng vai trò trung tâm để review code (pull request), theo dõi lỗi (issues), và tích hợp CI/CD (GitHub Actions) nhằm tự động hóa kiểm thử, build, và deploy. Kho lưu trữ đám mây còn hỗ trợ backup dữ liệu an toàn và chia sẻ mã nguồn với cộng đồng mã nguồn mở.

- **PowerDesigner** được ứng dụng để thiết kế mô hình cơ sở dữ liệu (Database Modeling). Công cụ này cho phép xây dựng sơ đồ ERD (Entity-Relationship Diagram), ánh xạ quan hệ giữa các thực thể, và tự động sinh script SQL tương thích với hệ quản trị cơ sở dữ liệu như MySQL, PostgreSQL. Tính năng reverse engineering của PowerDesigner giúp phân tích cấu trúc database từ mã nguồn có sẵn, đồng thời tối ưu hóa hiệu năng thông qua phân tích ràng buộc và chỉ mục.

- **draw.io (diagrams.net)** được sử dụng để thiết kế lưu đồ thuật toán (flowchart), sơ đồ use case, và kiến trúc hệ thống. Công cụ miễn phí này cung cấp thư viện hình khối đa dạng, tích hợp trực tiếp với Google Drive, GitHub, và Confluence để lưu trữ, chia sẻ tài liệu. Nhờ giao diện trực quan và khả năng xuất file đa định dạng (PNG, PDF, SVG), draw.io trở thành giải pháp tối ưu để mô tả logic nghiệp vụ và tương tác giữa các thành phần trong hệ thống.

2.2.3. Giao diện người dùng

- **ReactJS** (phiên bản 19.0.0) được lựa chọn làm thư viện JavaScript trung tâm để xây dựng giao diện người dùng (UI) dựa trên mô hình component-based architecture. Mỗi component trong React là một đơn vị độc lập, kết hợp logic

(JavaScript), giao diện (JSX), và styling (CSS-in-JS), giúp tái sử dụng mã nguồn và quản lý trạng thái ứng dụng thông qua React Hooks (useState, useEffect, useContext). Virtual DOM của React tối ưu hiệu suất bằng cách chỉ cập nhật những phần thay đổi trên giao diện, giảm thiểu tương tác trực tiếp với DOM thật. Đặc biệt, phiên bản 19.0.0 tích hợp Concurrent Mode và Automatic Batching, cho phép xử lý tác vụ không đồng bộ mượt mà hơn và tối ưu hóa quá trình render.

- **TailwindCSS** (phiên bản 4.0.12) được triển khai như một utility-first CSS framework, thay thế phương pháp viết CSS truyền thống. Thay vì định nghĩa lớp CSS tùy chỉnh, Tailwind cung cấp hàng trăm utility class (ví dụ: flex, pt-4, text-center) để thiết kế trực tiếp trong HTML/JSX thông qua các attribute. Framework này đi kèm với tính năng Just-in-Time (JIT) Compiler, tự động tạo ra các lớp CSS động dựa trên mã nguồn, giúp giảm kích thước file build và hỗ trợ tùy biến theme (màu sắc, spacing, breakpoint) thông qua file tailwind.config.js. Sự kết hợp giữa Tailwind và PostCSS (qua plugin @tailwindcss/vite) đảm bảo quá trình build tối ưu và loại bỏ CSS không sử dụng (PurgeCSS).

- **Vite** (phiên bản 6.2.0) đóng vai trò là công cụ build front-end thế hệ mới, thay thế Webpack truyền thống. Vite tận dụng ES Modules (ESM) để phân chia ứng dụng thành các module nhỏ, cho phép trình duyệt tải song song các tài nguyên mà không cần bundle toàn bộ. Điều này giúp thời gian khởi động dev server giảm xuống dưới 1 giây, kể cả với dự án lớn. Tính năng Hot Module Replacement (HMR) của Vite cập nhật component React ngay lập tức khi có thay đổi mã nguồn mà không làm mất trạng thái ứng dụng. Ngoài ra, Vite tích hợp sẵn TypeScript, SCSS, và hỗ trợ tối ưu hóa production build thông qua Rollup (tree-shaking, code-splitting).

- Ngoài ra còn có các thư viện React khác như:

- + react-modal (3.16.3): Quản lý modal/popup với khả năng kiểm soát accessibility (ARIA attributes) và tương thích SSR.

- + react-router (7.3.0): Xử lý client-side routing với cơ chế dynamic routing, hỗ trợ nested routes và lazy loading components

- + @tanstack/react-query (5.67.2): Thư viện quản lý data fetching và caching, tự động xử lý stale data, retry khi gặp lỗi, và đồng bộ hóa dữ liệu giữa các tab.

- + react-multi-carousel (2.8.5): Xây dựng carousel đa năng với tính năng responsive breakpoints, infinite scroll, và custom arrows/dots.

- + react-paginate (8.3.0): Phân trang dữ liệu server-side/client-side với giao diện có thể tùy chỉnh bằng CSS.

- + chart.js + react-chartjs-2 (4.4.8 + 5.3.0): Render biểu đồ (line, bar, pie) tương tác, hỗ trợ real-time updates và tùy chỉnh tooltip/legend.

- + dompurify (3.2.5): Làm sạch HTML đầu vào (XSS sanitization) trước khi render bằng dangerouslySetInnerHTML.

- + browser-image-compression (2.0.2): Nén hình ảnh trước khi upload để giảm dung lượng (định dạng WebP/AVIF).

2.2.4. Xử lý dữ liệu

– **Supabase** (phiên bản SDK 2.49.1) được triển khai như một nền tảng Backend-as-a-Service mã nguồn mở, cung cấp giải pháp thay thế toàn diện cho Firebase với ưu điểm vượt trội về tính linh hoạt và khả năng tùy biến. Khác với Firebase (sử dụng NoSQL Firestore), Supabase xây dựng trên nền tảng PostgreSQL hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mạnh mẽ, hỗ trợ SQL nguyên bản, transactions ACID, và stored procedures. Kiến trúc này cho phép phát triển ứng dụng phức tạp với quan hệ dữ liệu đa chiều, đồng thời tận dụng tính năng Row Level Security (RLS) để kiểm soát truy cập dữ liệu ở cấp độ từng hàng, đảm bảo an ninh thông qua chính sách role-based.

– Các dịch vụ cốt lõi của Supabase:

+ **Authentication:** Hỗ trợ đa phương thức xác thực: Email/Mật khẩu, OAuth (Google, Facebook, GitHub), Magic Links, và WebAuthn (FIDO2). Quản lý phiên người dùng (JWT tokens) tự động, tích hợp với Supabase Storage và Realtime để đồng bộ quyền truy cập.

+ **PostgreSQL Database:** Tương tác dữ liệu thông qua @supabase/supabase-js SDK:

```
1. // Truy vấn dữ liệu với RLS
2. const { data, error } = await supabase
3.   .from('users')
4.   .select('id, name')
5.   .eq('role', 'admin');
```

+ **Realtime Database:** Theo dõi thay đổi dữ liệu theo thời gian thực thông qua WebSocket:

```
1. const channel = supabase
2.   .channel('orders')
3.   .on('postgres_changes', { event: 'INSERT', schema: 'public' }, handleNewOrder)
4.   .subscribe();
```

+ **Storage:** Lưu trữ file (hình ảnh, video) trong buckets, hỗ trợ CDN tích hợp để tối ưu tốc độ tải. Upload/download file với quyền truy cập được kiểm soát bởi RLS:

```
1. await supabase.storage.from('avatars').upload('user1.jpg', file);
2. const url = supabase.storage.from('avatars').getPublicUrl('user1.jpg');
```

+ **Auto-generated APIs:** REST API và GraphQL API tự động sinh từ cấu trúc database, kèm theo tài liệu Swagger/OpenAPI. Tích hợp với React Query hoặc Axios để fetch dữ liệu từ client.

2.2.5. Cơ sở dữ liệu

– **PostgreSQL** được Supabase tích hợp như hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) nền tảng, đóng vai trò trung tâm trong việc lưu trữ và xử lý dữ liệu có cấu trúc của ứng dụng. Khác với các database NoSQL, PostgreSQL tuân thủ nghiêm ngặt các nguyên tắc ACID (Atomicity, Consistency, Isolation, Durability), đảm bảo tính toàn vẹn dữ liệu ngay cả trong các giao dịch phức tạp. Supabase mở rộng sức mạnh của PostgreSQL thông qua hỗ trợ real-time, Row Level Security (RLS), và khả

năng tích hợp trực tiếp với dịch vụ xác thực người dùng, biến nó thành một giải pháp toàn diện cho cả persistence layer và business logic.

2.2.6. Quản lý thư viện và gói phần mềm

– **npm (Node Package Manager)** đóng vai trò trung tâm trong việc quản lý vòng đời thư viện của dự án. Với phiên bản 9.x+ (đi kèm Node.js 18.x), npm không chỉ cài đặt các dependency thông qua lệnh `npm install`, mà còn hỗ trợ version locking (tệp `package-lock.json`) để đảm bảo mọi thành viên trong nhóm sử dụng chính xác phiên bản thư viện đã xác định, tránh xung đột do semantic versioning (SemVer). Quy trình làm việc tuân thủ nguyên tắc deterministic build, cho phép tái tạo chính xác môi trường phát triển trên mọi hệ thống.

– Tập `package.json` hoạt động như bản thiết kế tổng thể của dự án, bao gồm:

+ **Dependencies**: Danh sách thư viện runtime (React, Axios) được khai báo dưới trường `dependencies`.

+ **DevDependencies**: Công cụ hỗ trợ phát triển (ESLint, Prettier, Vite) được khai báo trong `devDependencies`, chỉ cài đặt trong môi trường local.

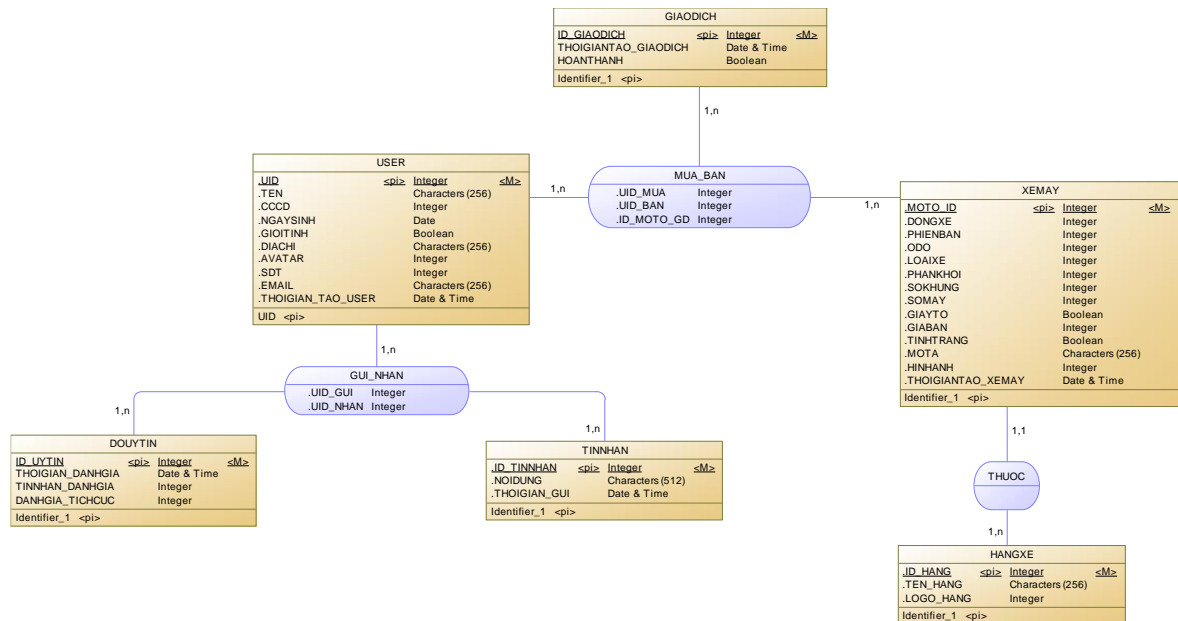
+ **Scripts**: Tập lệnh tự động hóa (start server, build production, kiểm tra lint).

+ **Engines**: Ràng buộc phiên bản Node.js/npm tối thiểu (`"node": ">=18.0.0"`).

+ **Proxy**: Cấu hình reverse proxy để tránh lỗi CORS khi gọi API từ client.

CHƯƠNG III. KẾT QUẢ THỰC HIỆN

3.1. Quản lý dữ liệu



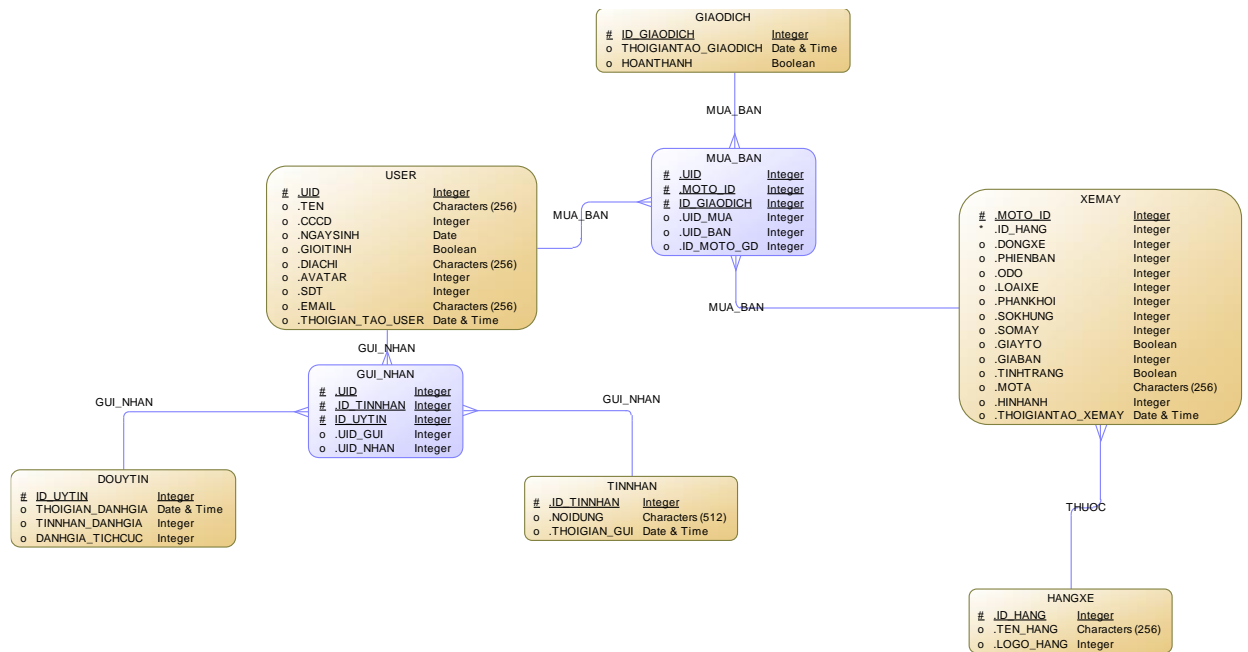
Hình 3. 1 Mô hình MCD của hệ thống

– Trong hệ thống này, mỗi quan hệ giữa xe máy và hãng sản xuất được thiết kế theo kiểu "một-nhiều": mỗi chiếc xe máy cụ thể sẽ thuộc về một hãng sản xuất duy nhất, nhưng ngược lại, mỗi hãng sản xuất sẽ có thể có nhiều mẫu xe máy khác nhau được đăng bán trên hệ thống.

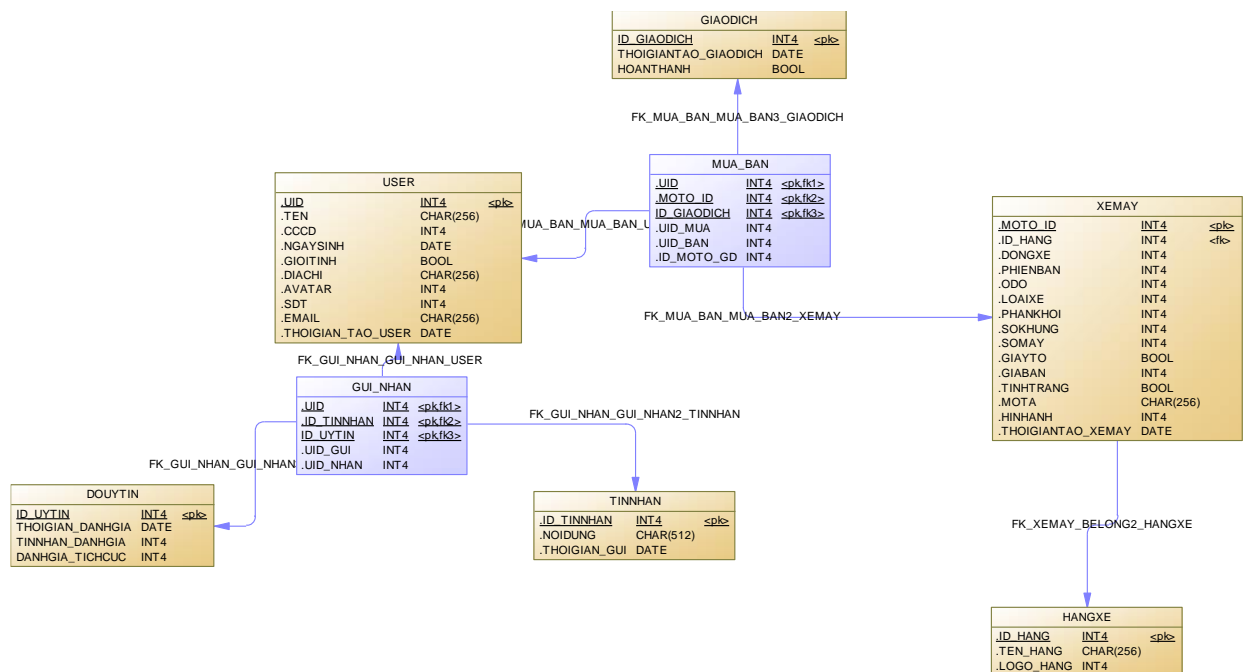
– Tất cả người dùng đã đăng ký đều có quyền thực hiện các giao dịch mua bán xe máy. Khi một người dùng quyết định mua một chiếc xe, hệ thống sẽ tạo ra một bản ghi giao dịch. Ở phía người mua, giao dịch này sẽ được ghi nhận với trạng thái "đang mua". Tương tự, ở phía người bán chiếc xe đó, một bản ghi giao dịch tương ứng cũng sẽ được tạo ra với trạng thái "đang bán". Khi quá trình mua bán giữa hai người dùng hoàn tất, người bán sẽ có quyền đánh dấu một thuộc tính có tên "HOANTHANH" thành "true" trong bản ghi giao dịch, để hệ thống nhận biết rằng giao dịch này đã thành công.

– Các người dùng với nhau có thể giao tiếp trực tiếp thông qua chức năng gửi tin nhắn. Để xác định rõ ràng vai trò của từng người trong cuộc trò chuyện, mỗi tin nhắn sẽ được gắn với mã định danh duy nhất của người gửi (sử dụng thuộc tính "UID_GUI") và mã định danh duy nhất của người nhận (sử dụng thuộc tính "UID_NHAN").

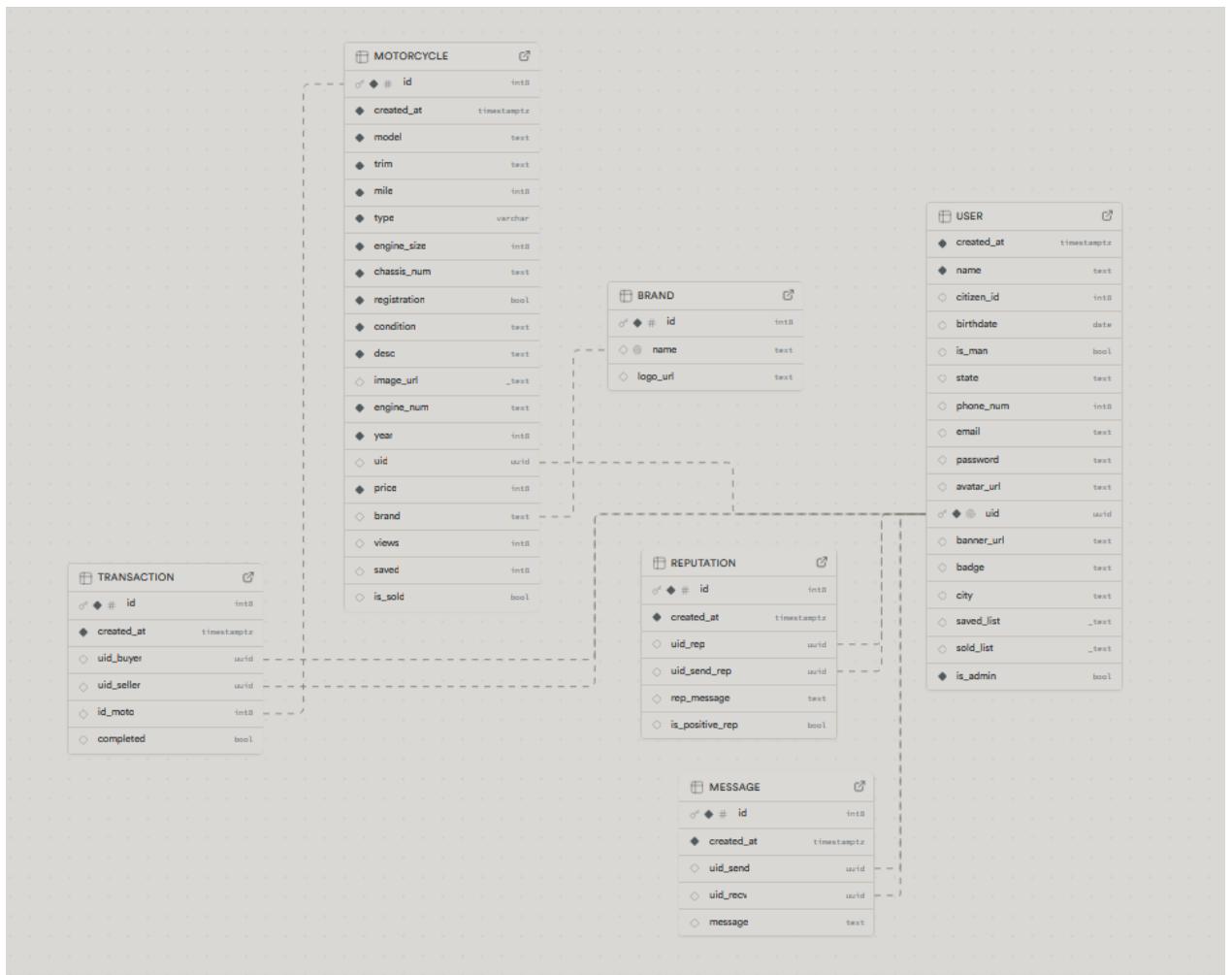
– Cách hoạt động của chức năng đánh giá người dùng là tương tự với chức năng nhắn tin.



Hình 3. 2 Mô hình MLD



Hình 3. 3 Mô hình MPD



Hình 3. 4 Sơ đồ cơ sở dữ liệu

– **USER (Người dùng)**: Trong quá trình đăng ký tài khoản, mỗi người dùng bắt buộc phải cung cấp đầy đủ các thông tin cá nhân bao gồm: địa chỉ Email, họ tên, số điện thoại, số căn cước công dân, ngày sinh, lựa chọn giới tính, cùng với một mật khẩu để đảm bảo tính bảo mật cho tài khoản. Ngay sau khi hoàn tất đăng ký, hệ thống xác thực Supabase Auth sẽ tự động khởi tạo và gán cho người dùng một ID ngẫu nhiên duy nhất, được sử dụng để định danh trong toàn bộ hệ thống. Bên cạnh các thông tin cơ bản, mỗi tài khoản người dùng còn được liên kết với một số thuộc tính phụ nhằm cá nhân hóa hồ sơ người dùng, bao gồm: liên kết ảnh đại diện (avatar), liên kết banner trang cá nhân, cũng như thông tin về thành phố (city) và tiểu bang (state) nơi người dùng đang sinh sống.

Thực thể USER

THUỘC TÍNH	KIỂU DỮ LIỆU	NULL?	DỮ LIỆU MẶC ĐỊNH	KHOÁ CHÍNH?	KHOÁ PHỤ?
uid	uuid			true	
created_at	timestampz		now()		
name	text				
citizen_id	int8	true			
birthdate	date	true			
is_man	bool	true			
state	text	true			
city	text	true			
phone_num	int8	true			
email	text				
password	text	true			
avatar_url	text				
banner_url	text				
is_admin	bool		false		

Bảng 3. 1 Chi tiết thực thể USER

Trong đó có:

+ uid: Đây là mã định danh duy nhất cho mỗi người dùng trong hệ thống. Thuộc tính này sử dụng kiểu dữ liệu uuid (Universally Unique Identifier), một kiểu dữ liệu đặc biệt được thiết kế để tạo ra các chuỗi mã ngẫu nhiên, đảm bảo không có sự trùng lặp giữa các người dùng.

+ created_at: Thuộc tính này lưu trữ thời điểm mà thông tin người dùng được tạo ra trong hệ thống. Kiểu dữ liệu được sử dụng là timestampz, cho phép ghi lại cả thời gian và múi giờ. Giá trị mặc định của Thuộc tính này là now(), nghĩa là hệ thống sẽ tự động ghi lại thời điểm hiện tại khi một bản ghi người dùng mới được tạo.

+ name: Đây là tên đầy đủ của người dùng.

+ citizen_id: Lưu trữ số căn cước công dân của người dùng.

+ birthdate: Để ghi lại ngày tháng năm sinh của người dùng.

+ is_man: Dùng để xác định giới tính của người dùng. Nó sử dụng kiểu dữ liệu boolean, với giá trị true biểu thị là nam giới và false biểu thị là nữ giới.

+ state: Lưu trữ thông tin về bang mà người dùng đang sinh sống

+ city: Lưu trữ thông tin về thành phố mà người dùng đang sinh sống.

+ phone_num: Lưu trữ số điện thoại của người dùng.

+ email: Lưu trữ địa chỉ email của người dùng.

+ password: Lưu trữ mật khẩu của người dùng. Khi người dùng đăng ký tài khoản mới, mật khẩu sẽ được mã hóa bằng các thuật toán bảo mật trước khi được lưu trữ vào cơ sở dữ liệu, nhằm bảo vệ thông tin người dùng.

+ avatar_url: Lưu trữ đường dẫn URL đến ảnh đại diện (avatar) của người dùng, được lưu trữ trên bucket của Supabase.

+ banner_url: Lưu trữ đường dẫn URL đến ảnh bìa (banner) của người dùng,

tương tự cũng được lưu trữ trên bucket Supabase.

+ **is_admin**: Thuộc tính này xác định xem người dùng có phải là quản trị viên hệ thống hay không. Nó sử dụng kiểu dữ liệu boolean, với giá trị true biểu thị là quản trị viên và false biểu thị là người dùng thông thường.

– **MOTORCYCLE (Xe máy)**: Khi người dùng tiến hành đăng bán một chiếc xe máy, họ sẽ cần cung cấp các thông tin bắt buộc như kiểu xe, hãng xe, mẫu xe, và phiên bản của xe đó. Những thông tin này được lựa chọn từ tập dữ liệu có sẵn, nhờ đó hệ thống có thể tự động bổ sung các thuộc tính phụ như kích thước động cơ mà người dùng không cần phải nhập lại. Ngoài ra, người dùng còn phải nhập thêm các thông tin chi tiết khác bao gồm: số khung, số máy, ngày sản xuất, số dặm (mile) đã di chuyển, cùng với mô tả chi tiết, giá bán và tình trạng giấy tờ pháp lý. Hệ thống cũng có cơ chế phân loại xe dựa trên số dặm: nếu người dùng nhập 0 mile, hệ thống sẽ tự động xác định đây là xe mới, còn nếu nhập một giá trị lớn hơn 5 mile, xe sẽ được phân loại là đã qua sử dụng. Mỗi chiếc xe máy trong hệ thống đều được gắn kèm một số thuộc tính phụ để hỗ trợ việc quản lý và đánh giá hiệu quả hiển thị. Trong đó, lượt xem (views) phản ánh mức độ quan tâm của người dùng đối với chiếc xe đó, trong khi trạng thái đã bán sẽ được sử dụng để đánh dấu những chiếc xe đã giao dịch thành công, giúp hệ thống ẩn khỏi danh sách bán nhằm tránh gây nhầm lẫn cho người dùng khác.

Thực thể MOTORCYCLE

THUỘC TÍNH	Kiểu Dữ Liệu	NULL?	Dữ Liệu Mặc Định	Khoá Chính?	Khoá Phụ?
id	int8			true	
created_at	timestampz		now()		
uuid	uid				uid -> USER
type	text				
brand	text				name -> BRAND
model	text				
trim	text				
mile	int8				
engine_size	int8				
chassis_num	text	true			
engine_num	text	true			
registration	boolean				
condition	text				
desc	text	true			
image_url	text [array]				
year	int8				
price	int8				
views	int8		0		
is_sold	boolean		false		

Bảng 3. 2 Chi tiết thực thể MOTORCYCLE

Trong đó có:

+ id: Mã định danh duy nhất cho mỗi chiếc xe máy được đăng bán trong hệ thống.

+ created_at: Trường này lưu trữ thời điểm chiếc xe máy được đăng bán lên hệ thống.

+ uid: Mã định danh duy nhất của người bán chiếc xe máy. Trường này có kiểu dữ liệu uuid (Universally Unique Identifier) là khóa phụ (foreign key) liên kết với trường uid trong thực thể người dùng (User), cho biết ai là người đăng bán chiếc xe này.

+ type: Kiểu dáng hoặc loại xe máy, ví dụ như naked bike, sport bike, cruiser, scooter, adventure,...

+ brand: Tên hãng sản xuất xe máy. Là khóa phụ (foreign key) có mối quan hệ liên kết với tên hãng trong thực thể BRAND.

+ model: Tên mẫu xe cụ thể của hãng.

+ trim: Phiên bản cụ thể của mẫu xe.

+ mile: Số dặm (mile) mà chiếc xe đã đi được.

+ engine_size: Kích thước động cơ của xe, được đo bằng đơn vị cc (centimet khối).

+ chassis_num: Số khung (VIN - Vehicle Identification Number) của xe.

+ engine_num: Số máy của xe.

+ registration: Cho biết chiếc xe có giấy tờ đăng ký hợp lệ hay không.

+ condition: Tình trạng của xe là mới (New) hay đã qua sử dụng (Used).

+ desc: Mô tả chi tiết về chiếc xe do người bán cung cấp.

+ image_url: Lưu trữ URL của (các) hình ảnh mà người bán đã tải lên khi đăng bán xe. Để lưu trữ nhiều hình ảnh, kiểu dữ liệu thường là một mảng (array) các chuỗi văn trong đó mỗi phần tử của mảng là một URL dẫn đến một hình ảnh.

+ year: Năm sản xuất của chiếc xe.

+ price: Giá bán của chiếc xe.

+ views: Số lượt xem trang chi tiết của chiếc xe.

+ is_sold: Trạng thái bán (True là đã bán, false là chưa bán)

– **BRAND (Hãng xe)**: Mỗi xe máy trên thị trường đều thuộc về một hãng sản xuất. Để dễ quản lý thì hãng xe cũng sẽ là 1 thực thể riêng thay vì chỉ là 1 thuộc tính.

Thực thể BRAND

THUỘC TÍNH	Kiểu Dữ Liệu	NULL?	DỮ LIỆU MẠC ĐỊNH	KHOÁ CHÍNH?	KHOÁ PHỤ?
id	int8			true	
name	text				
logo_url	text				

Bảng 3. 3 Chi tiết thực thể BRAND

Trong đó có:

+ id: Mã định danh duy nhất cho mỗi hãng xe máy.

+ name: Tên của hãng xe.

+ logo_url: Lưu trữ url logo hãng.

– **TRANSACTION (Giao dịch)**: Mỗi khi diễn ra một giao dịch mua bán xe máy trên hệ thống, một bản ghi giao dịch sẽ được tạo ra nhằm lưu trữ đầy đủ các thông tin cần thiết. Bản ghi này bao gồm: ID giao dịch, ID của người bán, ID của người mua, và ID của chiếc xe được giao dịch – tất cả đều được liên kết thông qua các khóa định danh duy nhất để đảm bảo tính chính xác và dễ dàng truy xuất. Bên cạnh đó, mỗi giao dịch còn đi kèm với một trạng thái giao dịch, dùng để phản ánh tiến độ hoặc kết quả của giao dịch đó, ví dụ như: đang chờ xử lý, đã hoàn tất, hoặc đã hủy. Hệ thống quản lý trạng thái này không chỉ giúp theo dõi quá trình giao dịch mà còn đảm bảo tính minh bạch và rõ ràng cho cả hai bên mua – bán.

Thực thể TRANSACTION

THUỘC TÍNH	KIỂU DỮ LIỆU	NULL?	DỮ LIỆU MẶC ĐỊNH	KHOÁ CHÍNH?	KHOÁ PHỤ?
id	int8			true	
created_at	timestampz				
uid_buyer	uuid				uid -> USER
uid_seller	uuid				uid -> USER
id_moto	int8				id -> MOTORCYCLE
completed	bool		false		

Bảng 3. 4 Chi tiết thực thể TRANSACTION

Trong đó có:

+ id: Mã định danh duy nhất cho mỗi giao dịch.

+ created_at: Thời gian tạo ra giao dịch.

+ uid_buyer: uid người mua. Là khoá phụ liên kết với thuộc tính uid của thực thể USER.

+ uid_seller: uid người bán. Là khoá phụ liên kết với thuộc tính uid của thực thể USER

+ id_moto: id của chiếc xe máy được giao dịch. Là khoá phụ liên kết với thuộc tính id của thực thể MOTORCYCLE

+ completed: Trạng thái giao dịch (true là hoàn thành, false là đang chờ)

– **MESSAGE (Tin nhắn)**: Hệ thống cho phép người dùng liên lạc trực tiếp với nhau thông qua tính năng gửi tin nhắn. Mỗi tin nhắn được gửi đi sẽ bao gồm các thông tin quan trọng như: ID tin nhắn, ID của người gửi, ID của người nhận, cùng với nội dung tin nhắn. Các tin nhắn này được lưu trữ trong cơ sở dữ liệu, cho phép người dùng dễ dàng xem lại lịch sử trò chuyện, đồng thời đảm bảo việc trao đổi thông tin giữa hai bên được thực hiện một cách nhanh chóng, tiện lợi và có tổ chức. Chức năng này góp phần hỗ trợ đắc lực cho quá trình thương lượng và mua bán xe giữa người dùng với nhau.

Thực thể MESSAGE

THUỘC TÍNH	KIỂU DỮ LIỆU	NULL?	DỮ LIỆU MẶC ĐỊNH	KHOÁ CHÍNH?	KHOÁ PHỤ?
id	int8			true	
created_at	timestampz				
uid_sender	uuid				uid -> USER
uid_recv	uuid				uid -> USER
message	text				

Bảng 3. 5 Chi tiết thực thể MESSAGE

Trong đó có:

- + id: Mã định danh duy nhất cho mỗi tin nhắn.
- + created_at: Thời gian tạo ra tin nhắn.
- + uid_sender: uid người gửi tin nhắn. Là khoá phụ liên kết với thuộc tính uid của thực thể USER.
- + uid_recv: uid người nhận tin nhắn. Là khoá phụ liên kết với thuộc tính uid của thực thể USER
- + message: Nội dung tin nhắn được gửi đi.

– **REPUTATION (Độ uy tín người dùng)**: Người dùng có thể gửi đánh giá cho nhau, góp phần phản ánh mức độ tin cậy và chất lượng tương tác của các bên tham gia. Mỗi đánh giá được ghi nhận sẽ bao gồm: ID của đánh giá, ID người gửi đánh giá, ID người nhận đánh giá, nội dung chi tiết của đánh giá, và một thuộc tính xác định liệu đây có phải là đánh giá tích cực hay không. Những đánh giá này sẽ được hệ thống tổng hợp lại để tính toán và thể hiện mức độ uy tín của người dùng trên hồ sơ cá nhân, giúp các bên tham khảo trước khi tiến hành giao dịch trong tương lai.

Thực thể REPUTATION

THUỘC TÍNH	KIỂU DỮ LIỆU	NULL?	DỮ LIỆU MẶC ĐỊNH	KHOÁ CHÍNH?	KHOÁ PHỤ?
id	int8			true	
created_at	timestampz				
uid_rep	uuid				uid -> user
uid_send_rep	uuid				uid -> user
rep_message	text				

Bảng 3. 6 Chi tiết thực thể REPUTATION

Trong đó có:

+ id: Mã định danh duy nhất cho mỗi tin nhắn.

+ created_at: Thời gian tạo ra tin nhắn.

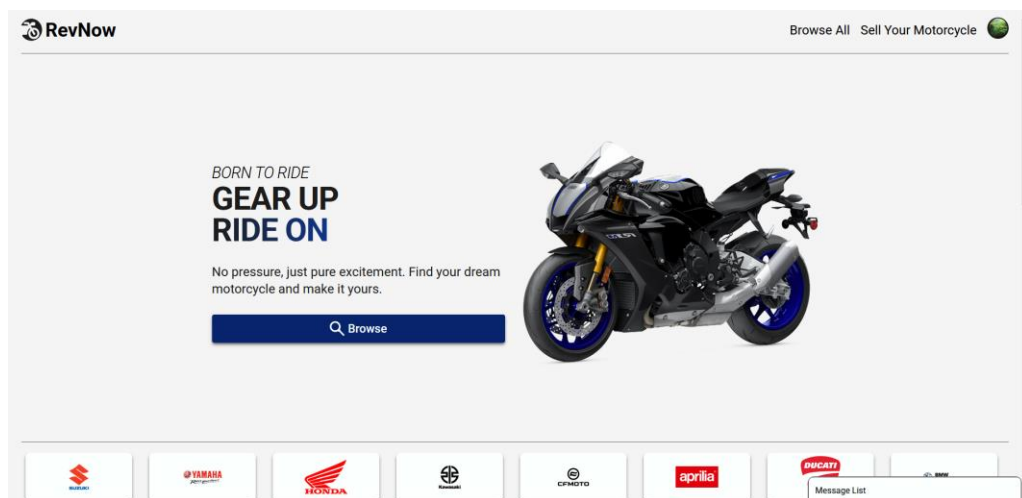
+ uid_rep: uid người nhận đánh giá. Là khoá phụ liên kết với thuộc tính uid của thực thể USER.

+ uid_send_rep: uid người gửi đánh giá. Là khoá phụ liên kết với thuộc tính uid của thực thể USER

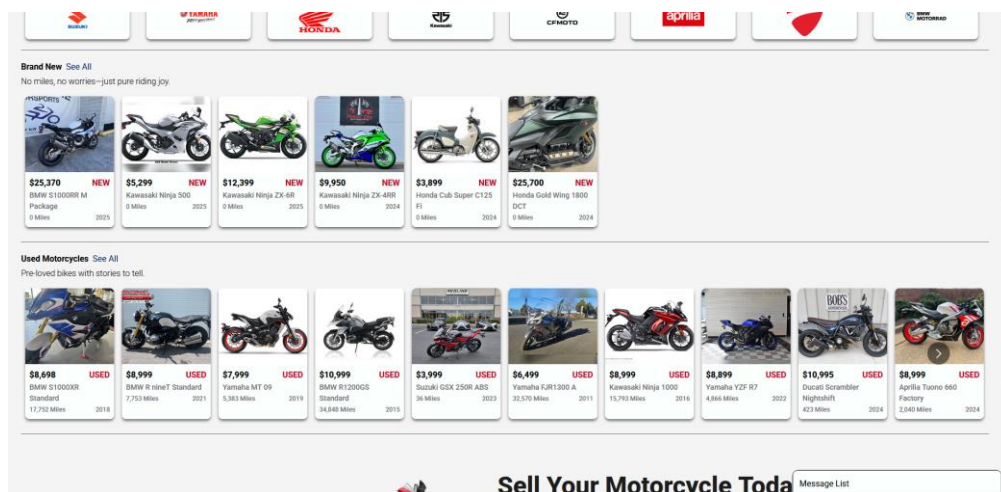
+ rep_message: Nội dung đánh giá được gửi đi cho người được đánh giá.

3.2. Giao diện và chức năng

3.2.1. Trang chủ



Hình 3. 5 Trang chủ



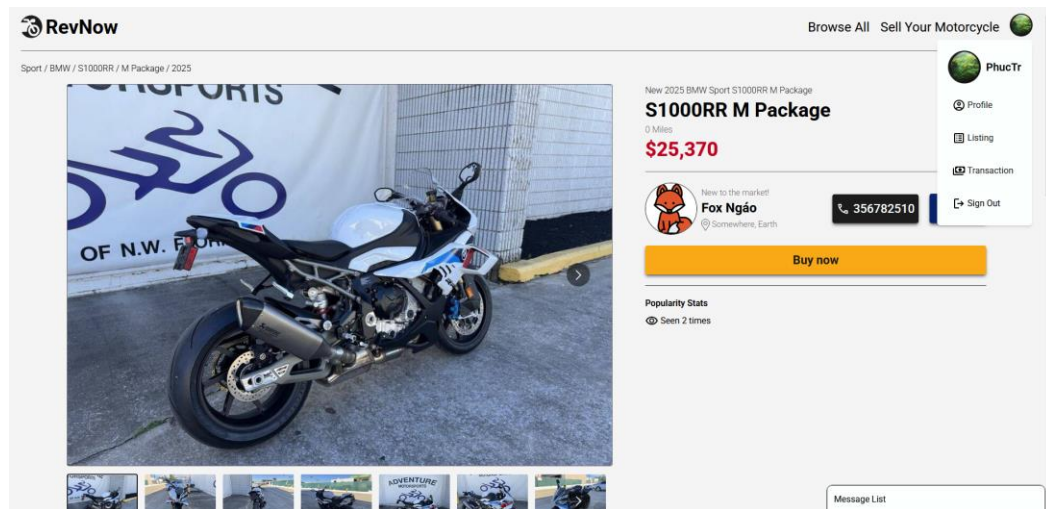
Hình 3. 6 Các nội dung của trang chủ

– Ngay tại phần đầu trang, người dùng có thể dễ dàng tiếp cận khu vực lọc theo hãng xe. Tại đây, logo của các hãng xe máy phổ biến và được ưa chuộng như Honda, Yamaha, Suzuki... được hiển thị một cách trực quan. Chỉ với một thao tác nhấn chuột vào logo của một hãng xe cụ thể, người dùng sẽ được tự động chuyển hướng đến trang kết quả tìm kiếm, nơi bộ lọc đã được áp dụng sẵn để hiển thị tất cả

các mẫu xe thuộc về hãng đó hiện đang được rao bán trên hệ thống.

– Bên cạnh đó, trang chủ còn cung cấp một phương thức phân loại xe rõ ràng dựa trên tình trạng sử dụng, với hai tùy chọn nổi bật là Xe mới và Xe cũ. Mục Xe mới tập trung vào các phương tiện chưa từng qua sử dụng, được đảm bảo có số ki-lô-mét lăn bánh rất thấp, thường dưới 5km, và đi kèm với chính sách bảo hành đầy đủ từ nhà sản xuất. Ngược lại, mục Xe cũ giới thiệu các xe đã qua sử dụng, với thông tin chi tiết về số Mile thực tế đã đi nhằm cung cấp cho người mua những dữ liệu quan trọng để đánh giá tình trạng xe một cách kỹ lưỡng trước khi đưa ra quyết định mua.

3.2.2. Điều hướng người dùng



Hình 3. 7 Menu điều hướng người dùng

– Để hỗ trợ người dùng trong việc điều hướng thì một nút điều hướng đơn giản cũng được tích hợp.

– Khi người dùng nhấn vào ảnh đại diện của mình, một menu điều hướng sẽ xuất hiện, cho phép truy cập nhanh đến các chức năng quan trọng trong tài khoản.

– Trong menu này, người dùng có thể truy cập vào:

- + Trang cá nhân thông qua mục Profile.
- + Theo dõi và chỉnh sửa các xe đã đăng bán trong phần Listing.
- + Kiểm tra tình trạng các giao dịch tại mục Transaction.
- + Sign Out để người dùng có thể đăng xuất khỏi tài khoản.

3.2.2. Đăng ký và đăng nhập

Sign Up / Log In

Find Your Passion

Choose a way to sign up

or

Email

Name

Phone Number


Citizen ID

Birthdate

Gender

Password

☐ Show Password



Hình 3. 8 Form đăng ký

Sign Up / Log In

Find Your Passion


Choose a way to log in

or

Email

Password

☐ Show Password



Hình 3. 9 Form đăng nhập

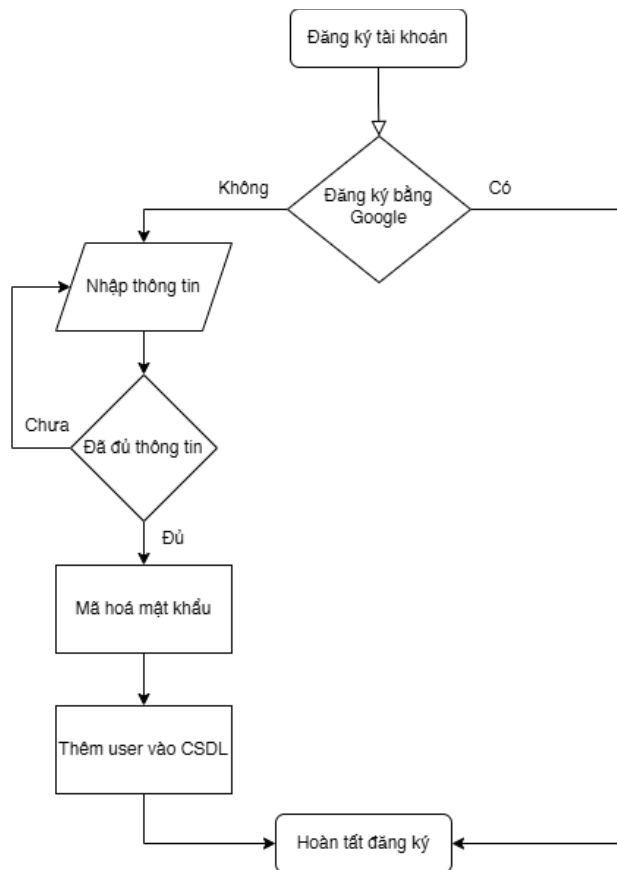
– Để đảm bảo an toàn và bảo mật trong quá trình truy cập, hệ thống xác thực người dùng được xây dựng dựa trên nền tảng Supabase Auth, sử dụng JSON Web Token (JWT) để quản lý phiên đăng nhập. Hệ thống hỗ trợ hai phương thức xác thực phổ biến là đăng nhập truyền thống bằng email và mật khẩu, cùng với đăng nhập qua tài khoản Google thông qua giao thức OAuth 2.0. Toàn bộ quá trình xác thực đều được xử lý tự động và bảo mật, thông qua các cơ chế như mã hóa mật khẩu, xác thực token, và đồng bộ hóa dữ liệu người dùng theo thời gian thực.

– Trong quá trình đăng ký tài khoản, người dùng cần nhập đầy đủ các thông tin cá nhân bắt buộc như: email, mật khẩu, họ tên, số điện thoại, số căn cước công dân (CMND/CCCD), ngày sinh và giới tính. Trước khi lưu trữ vào cơ sở dữ liệu, mật khẩu sẽ được mã hóa bằng thuật toán băm kèm muối (hash with salt) để đảm bảo rằng ngay cả khi dữ liệu bị rò rỉ, thông tin nhạy cảm của người dùng vẫn được bảo vệ. Sau khi đăng ký thành công, Supabase sẽ tự động tạo một ID người dùng duy nhất (UUID) và đăng nhập người dùng vào hệ thống mà không cần thực hiện lại thao tác đăng nhập.

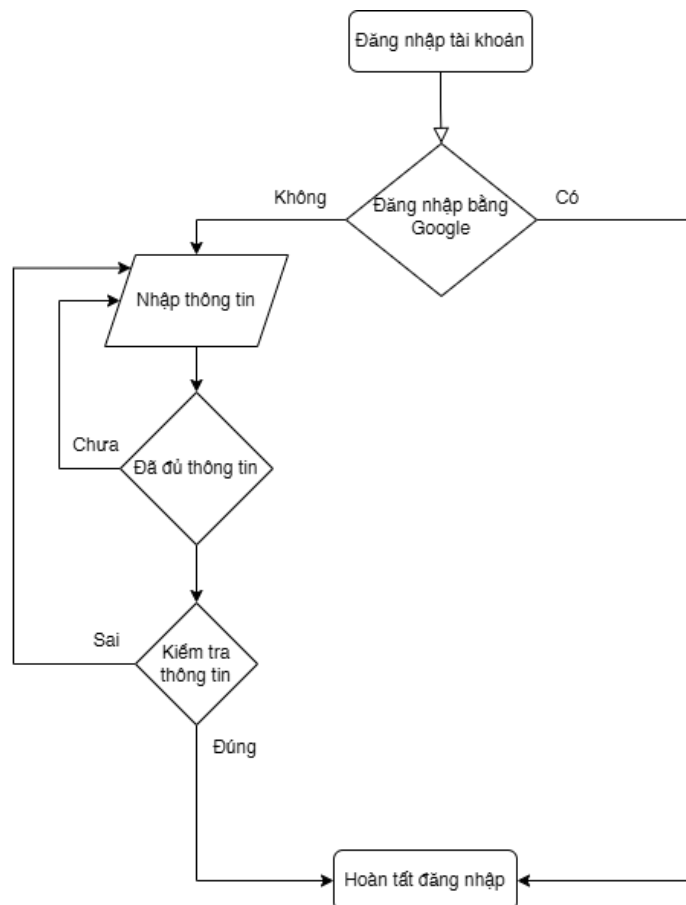
– Đối với quá trình đăng nhập truyền thống, khi người dùng nhập email và mật khẩu, hệ thống sẽ kiểm tra tính hợp lệ của thông tin này bằng cách đối chiếu với cơ sở dữ liệu Supabase. Nếu thông tin chính xác, Supabase sẽ tạo ra một JWT đại diện cho phiên đăng nhập và gửi về phía trình duyệt người dùng. JWT này sẽ được sử dụng trong các lần truy cập tiếp theo để xác minh danh tính mà không cần yêu cầu người dùng đăng nhập lại. Ngược lại, nếu thông tin không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi cụ thể như “Email không tồn tại” hoặc “Mật khẩu không đúng” để người dùng dễ dàng xử lý.

– Bên cạnh đó, hệ thống cũng hỗ trợ đăng ký và đăng nhập bằng Google. Khi người dùng chọn phương thức này, hệ thống sẽ chuyển hướng đến trang đăng nhập của Google. Sau khi người dùng xác nhận quyền truy cập, Google sẽ trả về các thông tin cơ bản như email, tên người dùng và ảnh đại diện. Hệ thống sẽ kiểm tra xem tài khoản Google đó đã tồn tại trong cơ sở dữ liệu chưa. Nếu chưa, một bản ghi người dùng mới sẽ được tạo tự động dựa trên các thông tin nhận được. Quá trình xác thực sau đó cũng sẽ được xử lý thông qua JWT như phương thức đăng nhập truyền thống.

– Sau khi đăng nhập thành công (dù là bằng email hay qua Google), người dùng sẽ được chuyển đến một trang callback trung gian để xác minh lại phiên làm việc. Tại đây, JWT sẽ được kiểm tra thêm một lần nữa để đảm bảo hợp lệ. Hệ thống cũng đồng bộ lại các thông tin người dùng với cơ sở dữ liệu nếu có thay đổi, sau đó chuyển hướng người dùng đến trang chủ của ứng dụng. Nếu có bất kỳ lỗi nào xảy ra trong quá trình xử lý callback, người dùng sẽ được chuyển về lại trang đăng nhập kèm theo thông báo lỗi cụ thể để tiện xử lý.

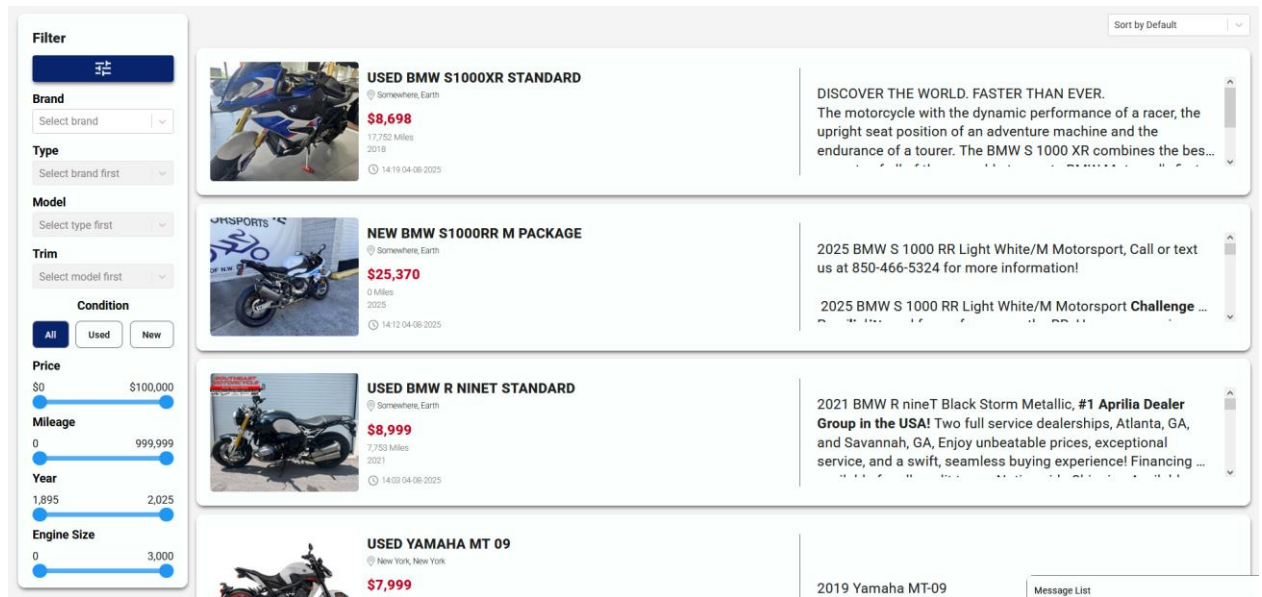


Hình 3. 10 Lưu đồ chức năng đăng ký tài khoản



Hình 3. 11 Lưu đồ chức năng đăng nhập

3.2.3. Tìm kiếm và lọc xe



Hình 3. 12 Trang tìm kiếm và lọc sản phẩm

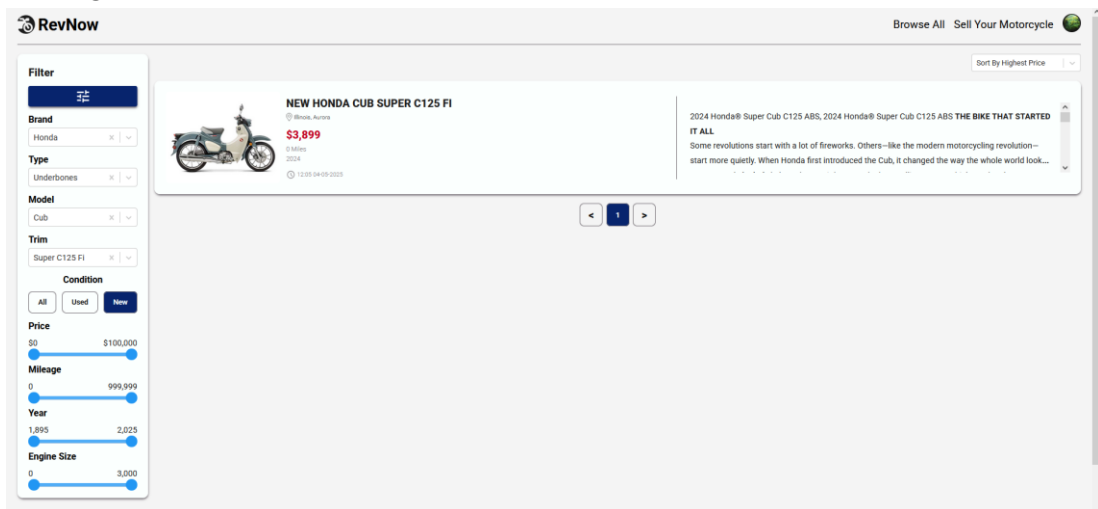
– Hệ thống tìm kiếm xe máy được thiết kế với bộ lọc nâng cao, cho phép người dùng tinh chỉnh kết quả theo nhiều tiêu chí khác nhau một cách linh hoạt và chính xác. Ở phía bên trái giao diện, người dùng có thể bắt đầu bằng việc chọn hãng xe từ danh sách dropdown liệt kê đầy đủ các thương hiệu có trong hệ thống. Cấu trúc lọc được tổ chức theo thứ tự phụ thuộc từ trên xuống, bao gồm các tầng: hãng xe (Brand), loại xe hay dáng xe (Type) như Naked bike, Sport bike, Cruiser, Scooter hoặc Adventure, tiếp theo là mẫu xe (Model) và phiên bản (Trim). Mỗi lựa chọn ở tầng trên sẽ ảnh hưởng trực tiếp đến danh sách của tầng dưới; khi người dùng chọn một hãng xe cụ thể, hệ thống sẽ tự động cập nhật danh sách loại xe tương ứng, và các model và phiên bản cũng sẽ thay đổi phù hợp với các lựa chọn trước đó. Nếu tại bất kỳ tầng nào không có dữ liệu phù hợp, các tầng kế tiếp sẽ bị làm rỗng và tạm thời vô hiệu hóa, nhằm ngăn chặn lỗi lựa chọn không hợp lệ và đảm bảo tính nhất quán trong quá trình lọc.

– Ngoài các bộ lọc dạng dropdown, hệ thống còn tích hợp các bộ lọc theo khoảng giá trị sử dụng thanh trượt (range slider) cho các tiêu chí như giá bán, số ki-lô-mét đã đi, năm sản xuất và dung tích xi-lanh (từ 50cc đến 3000cc). Các thanh trượt này được thiết kế thân thiện với người dùng, hiển thị rõ giá trị hai đầu với định dạng số dễ đọc (ví dụ: “1,000” thay vì “1000”) và có cơ chế debounce 500ms để hạn chế số lần cập nhật URL không cần thiết khi người dùng thao tác. Người dùng cũng có thể lọc theo tình trạng xe với ba tùy chọn: tất cả, đã qua sử dụng và mới. Nếu không chọn tình trạng cụ thể nào, hệ thống mặc định hiển thị toàn bộ kết quả phù hợp, đảm bảo tính linh hoạt trong trải nghiệm tìm kiếm.

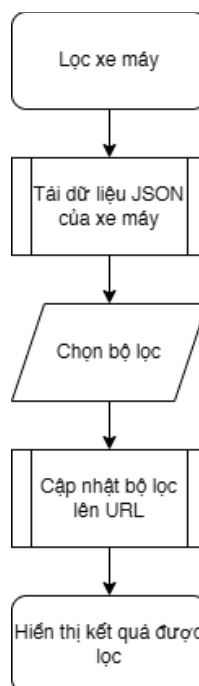
– Toàn bộ thay đổi từ các bộ lọc được phản ánh trực tiếp trên URL thông qua các query parameters. Nhờ vậy, người dùng có thể sao chép, đánh dấu (bookmark) hoặc truy cập lại một URL đã được lọc mà không bị mất cấu hình tìm kiếm trước đó.

Đồng thời, các tham số không có giá trị sẽ tự động bị loại bỏ để giữ cho URL luôn gọn gàng và hợp lệ. Dữ liệu cấu hình về các hãng xe, model và phiên bản được lấy từ một file JSON có cấu trúc rõ ràng, giúp hệ thống truy xuất nhanh chóng và chính xác.

– Kết quả tìm kiếm được hiển thị ở khu vực trung tâm dưới dạng các thẻ (card), mỗi thẻ trình bày một chiếc xe phù hợp với các tiêu chí đã chọn. Các thẻ này bao gồm hình ảnh đại diện của xe, tên hãng sản xuất, model, năm sản xuất, giá bán, số ki-lô-mét đã đi (nếu có), cùng với nút “xem chi tiết” để truy cập vào trang thông tin chi tiết của xe đó. Các kết quả được tự động phân trang, với tối đa mười xe trên mỗi trang và thanh điều hướng nằm ở cuối danh sách. Phía bên phải giao diện tìm kiếm là khu vực sắp xếp kết quả, nơi người dùng có thể lựa chọn sắp xếp theo nhiều tiêu chí như giá bán (tăng hoặc giảm dần), năm sản xuất, số ki-lô-mét đã đi hoặc thời gian đăng tin.



Hình 3. 13 Ví dụ khi sử dụng thanh lọc sản phẩm



Hình 3. 14 Lưu đồ của chức năng tìm kiếm và lọc xe

3.2.4. Đăng bán xe

The screenshot shows a web form titled "Sell Your Motorcycle". At the top, there is a small disclaimer: "Use with caution: Web form protection and first class customer service, you'll be protected every step of the way." The form is organized into several sections. The first section contains dropdown menus for "Type", "Brand", "Model", and "Trims". Below these are input fields for "Stock Number", "VIN", "Year", and "Mileage". A "Condition" section has two radio buttons: "Used" (labeled "Pre-owned motorcycle with stories to tell") and "New" (labeled "Brand new motorcycle without any blemish"). Below the condition section is a "Description" field with a rich text editor toolbar. The next section contains "Price" and "Registration" input fields. There is an "Upload Image" button with a plus icon. At the bottom is a large black button labeled "Sell My Motorcycle".

Hình 3. 15 Trang bán xe

– Nhằm hỗ trợ người dùng trong việc giao dịch và giới thiệu sản phẩm của mình đến cộng đồng, hệ thống cung cấp chức năng đăng bán xe máy với giao diện trực quan và thao tác đơn giản. Người dùng có thể dễ dàng tạo mới hoặc chỉnh sửa thông tin liên quan đến chiếc xe cần bán, đồng thời hệ thống đảm bảo rằng mọi dữ liệu đều được kiểm tra hợp lệ và lưu trữ một cách an toàn. Việc xử lý hình ảnh cũng được tối ưu nhằm mang lại trải nghiệm mượt mà và hiệu quả trong quá trình tải lên.

– Khi người dùng truy cập vào trang đăng bán, hệ thống sẽ kiểm tra phiên đăng nhập hiện tại thông qua Supabase Auth để xác định danh tính. Nếu người dùng chưa đăng nhập, họ sẽ được chuyển hướng đến trang đăng nhập hoặc tài khoản. Trong trường hợp người dùng đã đăng nhập nhưng thông tin cá nhân còn thiếu (CMND/CCCD, số điện thoại, email hoặc địa chỉ), hệ thống sẽ yêu cầu cập nhật đầy đủ thông tin này trước khi tiếp tục thao tác đăng bán nhằm đảm bảo tính xác thực và quản lý hợp lệ.

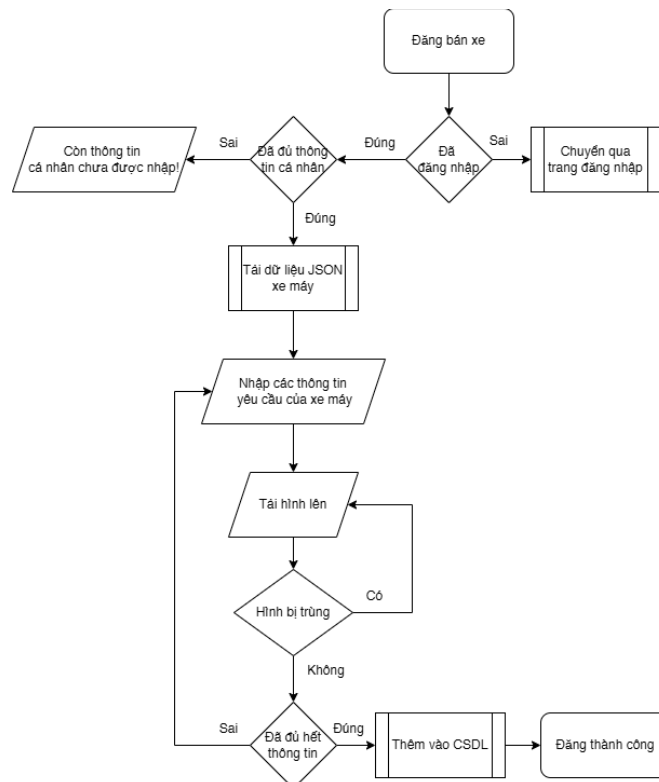
– Để hỗ trợ người dùng điền thông tin xe một cách dễ dàng, hệ thống sử dụng dữ liệu có cấu trúc (JSON) giúp lọc và hiển thị các lựa chọn tương ứng theo từng bước. Khi người dùng chọn loại xe (ví dụ: Naked, Sport), danh sách các hãng xe phù hợp sẽ được lọc và hiển thị tự động. Việc chọn hãng sẽ tiếp tục dẫn đến danh sách model tương ứng, và sau đó là danh sách trim. Khi người dùng chọn trim, hệ thống sẽ tự động điền giá trị dung tích động cơ (engine size) dựa trên dữ liệu có sẵn. Về tình trạng xe, hệ thống sẽ tự động xác định dựa vào số km đã đi: nếu giá trị này lớn hơn 5, xe được phân loại là "Used", còn nếu bằng 0 hoặc không có giá trị thì xe được coi là "New".

– Trong quá trình chọn hình ảnh, hệ thống hỗ trợ người dùng chọn nhiều ảnh cùng lúc và xử lý chúng một cách thông minh. Ảnh trùng lặp sẽ được loại bỏ dựa trên tên hoặc nội dung, và những ảnh được chọn sẽ được hiển thị dưới dạng thumbnail để người dùng dễ dàng xem trước và xóa nếu cần. Trước khi thực hiện tải lên, toàn bộ

ảnh sẽ được nén nhằm tối ưu dung lượng lưu trữ và giảm thời gian tải. Quá trình xử lý này đảm bảo hiệu suất cao, kể cả khi người dùng tải lên nhiều ảnh cùng lúc.

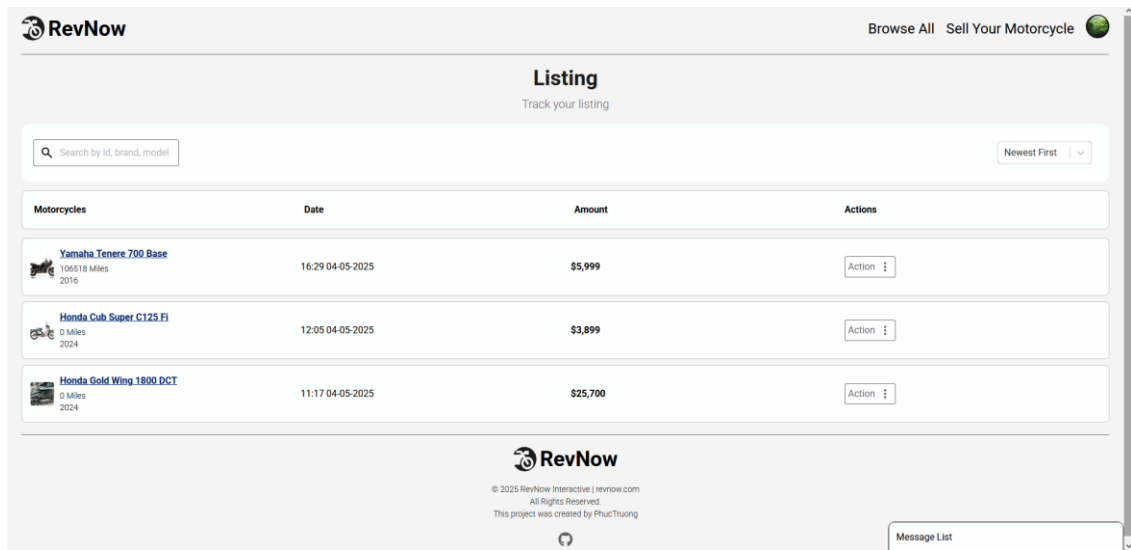
– Sau khi người dùng hoàn tất thông tin và chọn ảnh, hệ thống sẽ kiểm tra lại các trường bắt buộc, bao gồm loại xe, hãng, model, trim, năm sản xuất, số khung, số máy, tình trạng xe, mô tả và giá bán. Nếu bất kỳ trường nào bị thiếu, hệ thống sẽ hiển thị cảnh báo rõ ràng để người dùng bổ sung. Tiếp theo, các ảnh đã chọn được tải lên Supabase Storage, lưu trong bucket motorcycle-media, có cấu trúc thư mục theo uid của người dùng. Tên ảnh được chuẩn hóa dưới định dạng [uid]-[timestamp]-[tên-gốc] để đảm bảo không bị trùng. Sau khi ảnh được tải thành công, URL công khai của từng ảnh sẽ được lấy để lưu cùng dữ liệu xe.

– Cuối cùng, toàn bộ thông tin kỹ thuật của xe, mô tả và danh sách URL ảnh sẽ được thêm vào bảng MOTORCYCLE trong cơ sở dữ liệu. Nếu quá trình lưu thành công, hệ thống sẽ tự động tải lại trang hoặc chuyển hướng người dùng đến bài đăng mới để kiểm tra lại kết quả. Trong trường hợp xảy ra lỗi, hệ thống có cơ chế xử lý rõ ràng: nếu thiếu dữ liệu, người dùng sẽ được cảnh báo cụ thể (ví dụ: “Vui lòng nhập số km đã đi”); nếu xảy ra lỗi tải ảnh, hệ thống sẽ thông báo “Lỗi tải ảnh” và không gửi dữ liệu; nếu không thể kết nối đến Supabase, sẽ hiển thị thông báo “Lỗi hệ thống, vui lòng thử lại” đồng thời ghi log nội bộ để hỗ trợ quá trình kiểm tra và sửa lỗi sau này

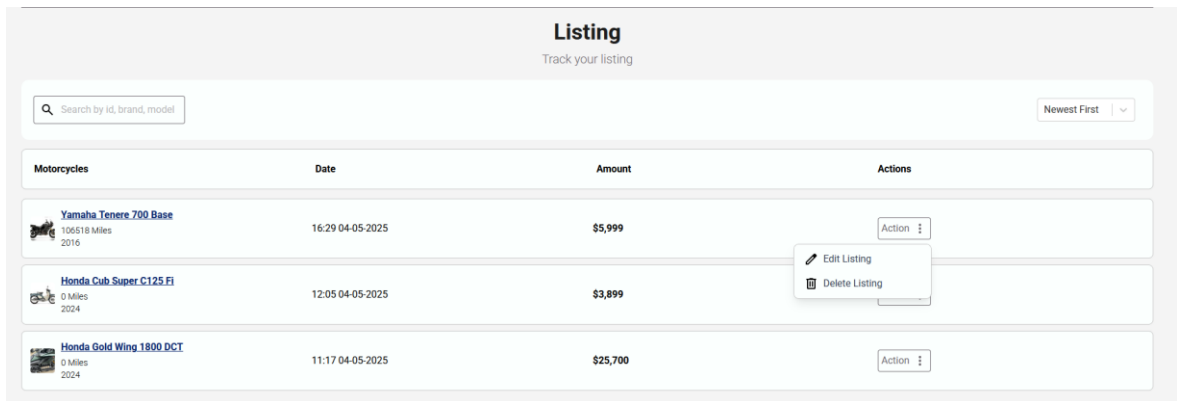


Hình 3. 16 Lưu đồ chức năng đăng bán

3.2.5. Quản lý xe máy



Hình 3. 17 Trang quản lý xe máy đã đăng



Hình 3. 18 Các hành động để tương tác với xe đã bán

– Trang quản lý xe máy đóng vai trò như một trung tâm điều khiển, nơi người dùng có thể dễ dàng theo dõi toàn bộ danh sách các xe đã đăng bán. Mỗi tin đăng hiển thị kèm các thông tin cơ bản như hãng xe, model, giá bán, tình trạng xe, giúp người dùng nhanh chóng nắm bắt tổng quan. Giao diện còn tích hợp các tiện ích tìm kiếm và sắp xếp kết quả theo nhiều tiêu chí như thời gian đăng (mới nhất, cũ nhất), giá bán (cao đến thấp hoặc ngược lại), hỗ trợ người dùng truy xuất và quản lý nội dung hiệu quả. Mỗi xe trong danh sách đều đi kèm một nút "Action", nơi tập hợp các tùy chọn cần thiết, bao gồm chỉnh sửa (Edit Listing) hoặc xóa bỏ tin đăng nếu không còn nhu cầu.

– Khi người dùng nhấp vào tùy chọn chỉnh sửa, hệ thống sẽ chuyển hướng đến trang Edit Listing riêng biệt cho chiếc xe đó. Giao diện chỉnh sửa được thiết kế trực quan, thân thiện, cho phép cập nhật toàn bộ các thông tin đã nhập trước đó tương tự như lúc đăng bài mới. Hệ thống sẽ lấy id từ URL và uid của người dùng để truy vấn dữ liệu từ bảng MOTORCYCLE trên Supabase. Dựa trên kết quả truy vấn, form sẽ được tự động điền các trường dữ liệu bao gồm hãng xe, loại xe, model, phiên bản,

năm sản xuất, số khung, số máy, dung tích động cơ, số km đã đi, giá bán, mô tả, tình trạng xe và danh sách các hình ảnh đã đăng. Đồng thời, hệ thống khởi tạo các dropdown tương ứng cho hãng, model và trim dựa trên loại xe hiện tại, đảm bảo dữ liệu luôn đồng bộ và hợp lệ theo cấu trúc phụ thuộc tầng.

– Trong quá trình chỉnh sửa, hệ thống kiểm tra các trường bắt buộc như hãng xe, model, trim, năm sản xuất, số km, giá bán và mô tả. Nếu thiếu bất kỳ thông tin nào, giao diện sẽ hiển thị cảnh báo rõ ràng, ngăn người dùng cập nhật dữ liệu không hợp lệ. Người dùng cũng có thể cập nhật hình ảnh đi kèm bằng cách sử dụng nút "Upload Image", cho phép chọn và tải lên hình ảnh mới. Các ảnh mới sẽ được xử lý nén bằng thư viện hỗ trợ, đặt tên theo định dạng [uid]/[timestamp]-[filename] để tránh trùng lặp và dễ quản lý trên Supabase Storage. Những hình ảnh cũ cũng có thể được xóa riêng lẻ thông qua nút delete tương ứng, đồng thời hệ thống hỗ trợ hoàn tác nếu gặp lỗi trong quá trình thao tác.

The screenshot shows a web form titled "Sell Your Motorcycle" with a subtitle "List with confidence. With fraud protection and first class customer service, you'll be protected every step of the way." The form contains several dropdown menus and text input fields. The "Type" dropdown is set to "Touring", "Brand" to "Yamaha", "Model" to "Tenere 700", and "Trims" to "Base". The "Stock Number" and "VIN" fields both contain "0". The "Year" field contains "2016" and the "Mileage" field contains "106,518". The "Condition" section has two radio buttons: "Used" (selected) and "New". The "Description" field is empty. At the bottom right, there is a "Message List" button.

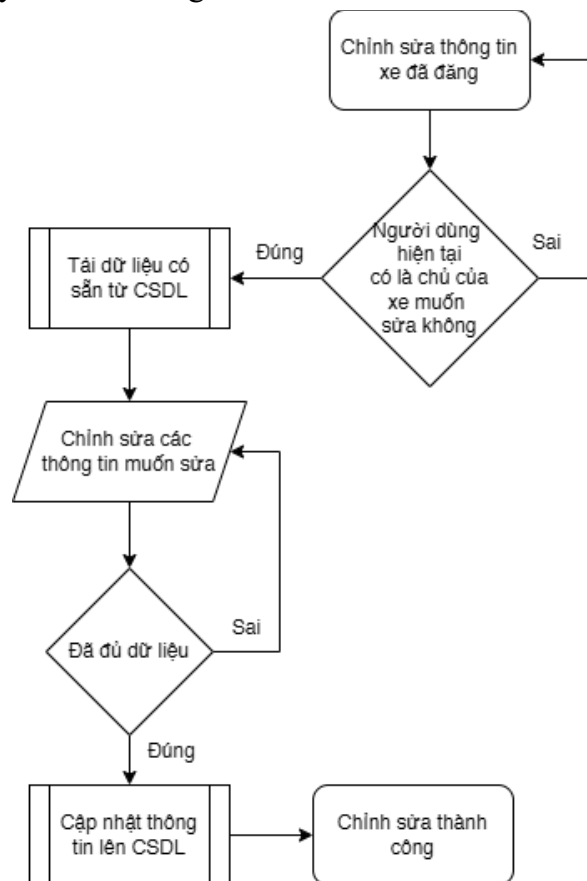
Hình 3. 19 Chỉnh sửa thông tin xe đã đăng bán

The screenshot shows the "Description" section of the form. It features a rich text editor with the text "TRANSCONTINENTAL ADVENTURE" and "Powerful inline twin, cruise control, D Mode, traction control, ABS, and adjustable seat." Below the editor is a "Price" field set to "\$5,999" and a "Registration" checkbox that is checked. There is an "Upload Image" button and a gallery of 12 images showing various motorcycles. At the bottom, there is an "Update Listing" button. A "Message List" button is visible at the bottom right.

Hình 3. 20 Chỉnh sửa thông tin xe đã đăng bán

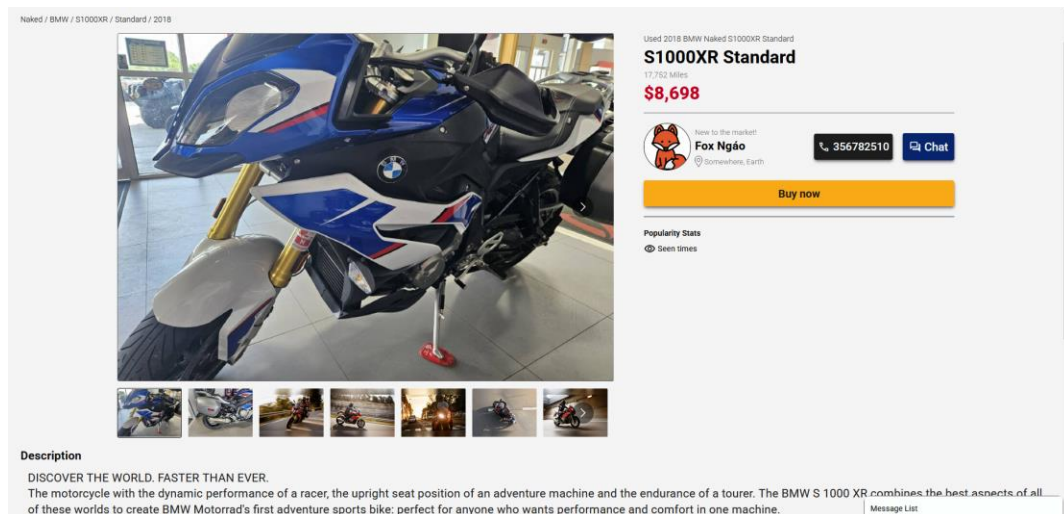
– Sau khi hoàn tất chỉnh sửa, người dùng nhấn xác nhận để gửi toàn bộ dữ liệu mới lên Supabase. Hệ thống sẽ thực hiện truy vấn cập nhật với điều kiện khớp id và uid nhằm đảm bảo chỉ người dùng sở hữu bài đăng mới có quyền thay đổi nội dung. Nếu cập nhật thành công, người dùng sẽ được chuyển hướng đến trang chi tiết để xem lại thông tin mới nhất của bài đăng. Trong trường hợp xảy ra lỗi, hệ thống sẽ hiển thị thông báo cụ thể và ghi log chi tiết để hỗ trợ quá trình kiểm tra, xử lý sau này.

– Trong suốt quá trình thao tác, hệ thống luôn cung cấp phản hồi trực quan như biểu tượng loading khi đang xử lý, xem trước ảnh mới trước khi tải lên, cho phép người dùng hoàn tác nếu thao tác xóa thất bại, và tự động khôi phục ảnh nếu gặp lỗi mạng. Các dropdown về hãng, model, trim được cập nhật động theo dữ liệu hiện tại để đảm bảo tính hợp lệ trong quá trình chỉnh sửa. Chức năng này không chỉ giúp người dùng dễ dàng cập nhật nội dung bài đăng một cách linh hoạt, mà còn đảm bảo dữ liệu luôn được lưu trữ đúng chuẩn, nhất quán và an toàn, góp phần nâng cao trải nghiệm và sự tin cậy của hệ thống.



Hình 3. 21 Lưu đồ chức năng chỉnh sửa thông tin xe đã đăng

3.2.6. Chi tiết xe máy



Hình 3. 22 Trang thông tin chi tiết của xe

– Các hình ảnh được hiển thị trong một băng chuyền (carousel). Phía dưới là một thanh chứa các ảnh thu nhỏ (thumbnail), cho phép người dùng nhanh chóng chuyển đổi giữa các hình ảnh.

– Ngay bên dưới phần hình ảnh là khu vực hiển thị các thông tin cơ bản nhất của xe. Các chi tiết như hãng sản xuất, model, phiên bản, năm sản xuất, số ki-lô-mét đã đi (đối với xe đã qua sử dụng), tình trạng xe (mới hoặc cũ) và giá bán với định dạng tiền tệ rõ ràng bằng đô la Mỹ (\$).

– Tiếp theo là phần giới thiệu về người bán, bao gồm ảnh đại diện, tên người bán (được liên kết đến trang cá nhân), các thông tin liên hệ như địa chỉ khu vực, số điện thoại và một nút chat trực tiếp.

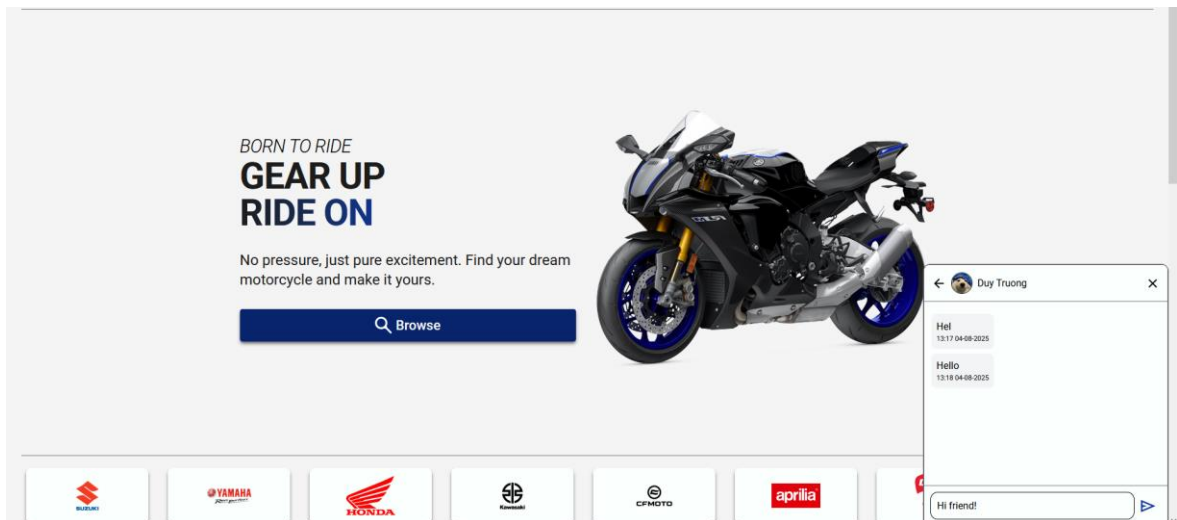
– Chức năng quan trọng nhất của trang là nút "Mua ngay". Khi người dùng nhấp vào nút này, hệ thống sẽ tự động tạo một giao dịch mua mới, đồng thời kiểm tra xem giao dịch tương tự đã tồn tại hay chưa và tiến hành xác thực người dùng. Bên cạnh đó, hệ thống cũng tự động ghi nhận và tăng số lượt xem của chiếc xe, cập nhật thông tin này vào cơ sở dữ liệu và hiển thị công khai như một chỉ số về mức độ quan tâm của cộng đồng đối với chiếc xe đó.

– Phần mô tả chi tiết là nơi người bán có thể cung cấp thông tin đầy đủ và chi tiết hơn về chiếc xe, giúp người mua có cái nhìn sâu sắc hơn về tình trạng, đặc điểm và các ưu điểm của xe. Nội dung được trình bày rõ ràng, dễ đọc và dễ hiểu.

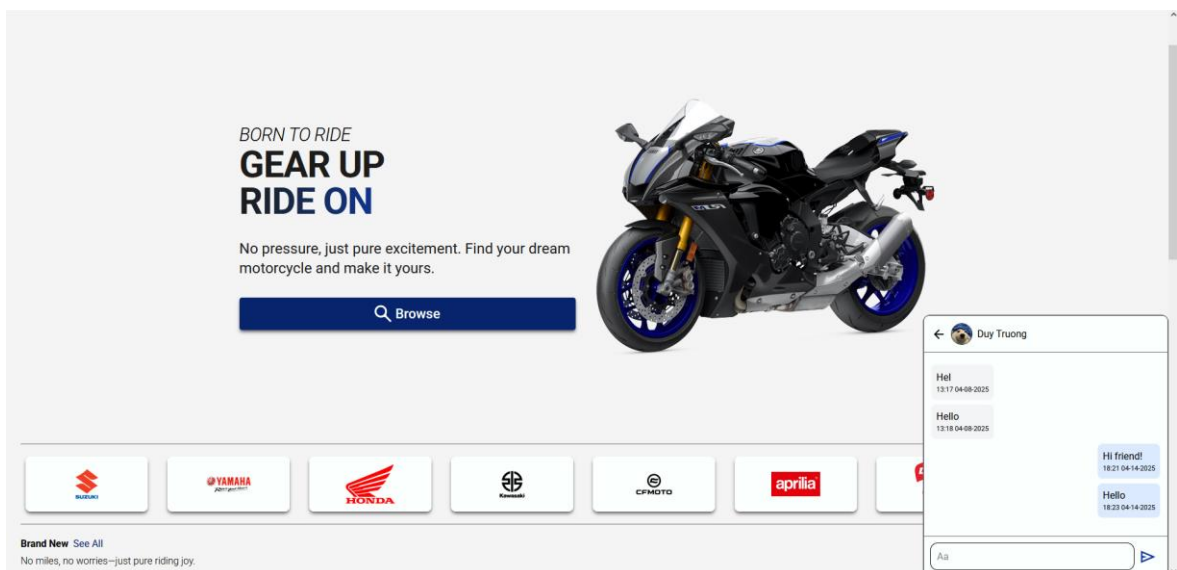
– Cuối cùng, ở phía dưới trang là phần gợi ý các xe khác đến từ cùng một người bán. Các xe này được hiển thị dưới dạng băng chuyền, cho phép người dùng dễ dàng khám phá thêm các sản phẩm khác mà người bán đang có.

thời, trạng thái đăng nhập của người dùng cũng được theo dõi để xác thực trước khi gửi hoặc nhận tin nhắn, đảm bảo tính bảo mật và quyền truy cập hợp lệ.

– Để tối ưu hiệu suất, hệ thống chỉ tải danh sách tin nhắn khi người dùng chọn một cuộc trò chuyện cụ thể, giúp giảm thiểu việc sử dụng tài nguyên không cần thiết. Danh sách người dùng (danh bạ) và nội dung chi tiết của cuộc trò chuyện được xử lý riêng biệt, kèm theo kỹ thuật caching nhằm tăng tốc độ phản hồi và cải thiện hiệu năng tổng thể của ứng dụng.

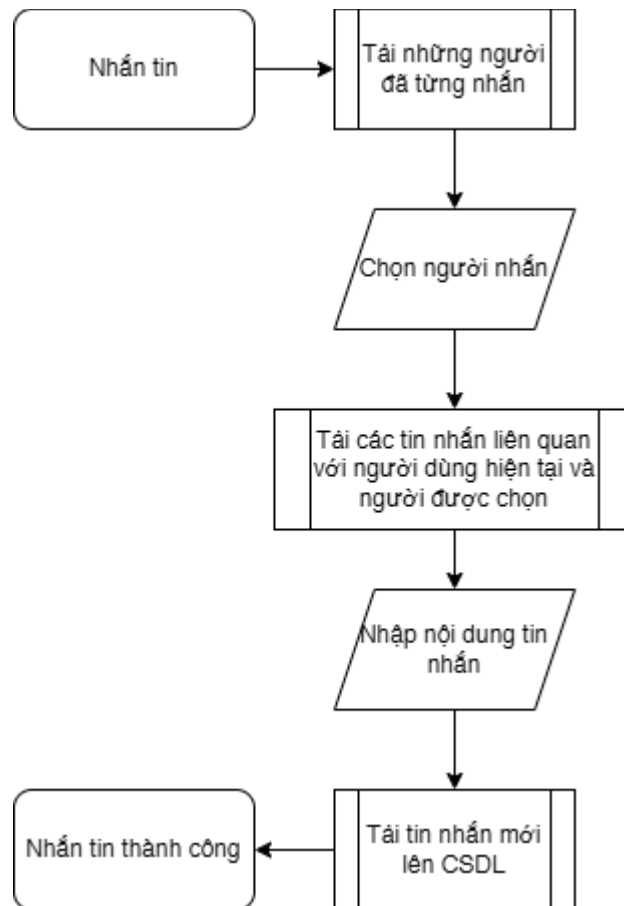


Hình 3. 25 Chi tiết của cuộc hội thoại



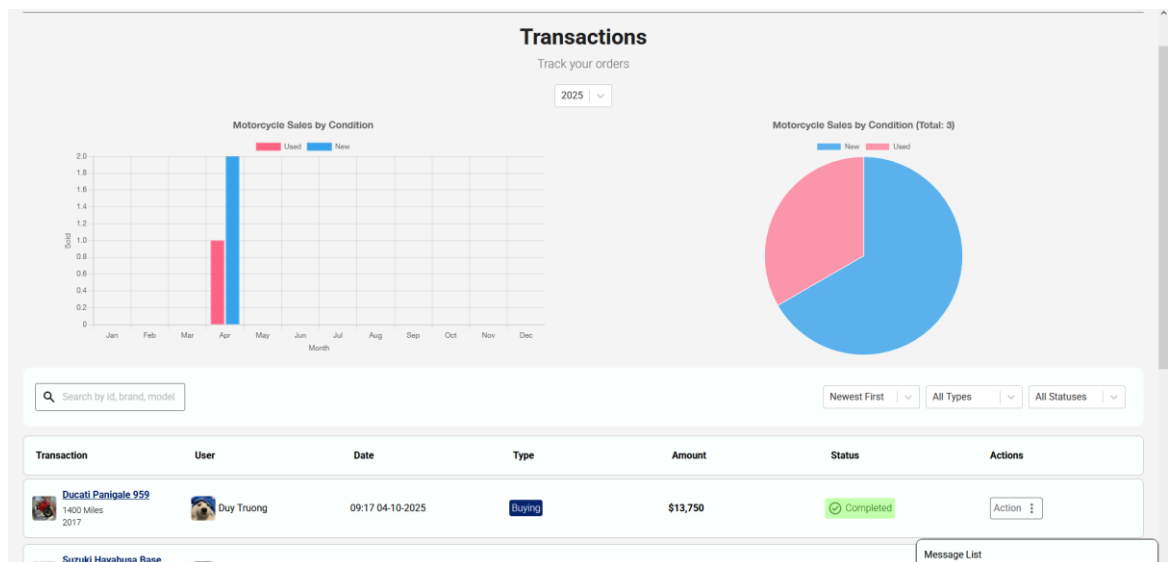
Hình 3. 26 Khi nhận tin

– Về bảo mật, chỉ những tin nhắn liên quan đến người dùng hiện tại mới được hiển thị. Hệ thống kiểm tra quyền truy cập trước khi hiển thị bất kỳ nội dung nào, đồng thời toàn bộ dữ liệu trong quá trình gửi và nhận đều được mã hóa nhằm bảo vệ thông tin cá nhân.



Hình 3. 27 Lưu đồ chức năng nhắn tin

3.2.8. Giao dịch



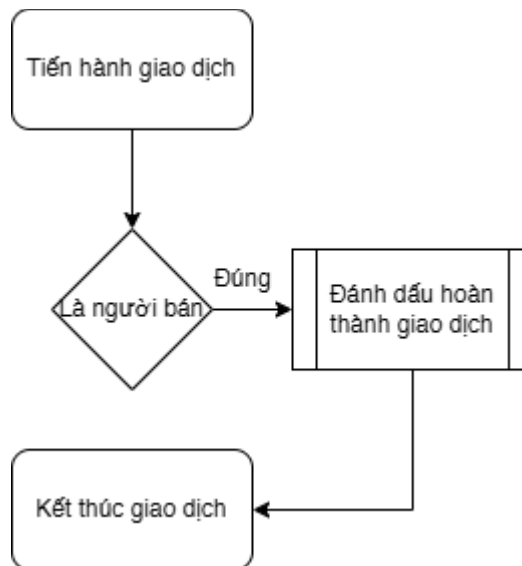
Hình 3. 28 Trang giao dịch xe máy bao gồm cả biểu đồ thống kê

Transaction	User	Date	Type	Amount	Status	Actions
Ducati Panigale 959 1400 Miles 2017	Duy Truong	09:17 04-10-2025	Buying	\$13,750	Completed	Action
Suzuki Hayabusa Base 3100 Miles 2009	Viet Nam Number 1	15:09 04-07-2025	Selling	\$6,900	Completed	Action
Yamaha YZF R3 0 Miles 2025	Viet Nam Number 1	14:42 04-07-2025	Selling	\$5,499	Completed	Action
Kawasaki Ninja 650 0 Miles 2025	hakimochida	16:50 04-05-2025	Selling	\$8,899	Completed	Action

RevNow
© 2025 RevNow Interactive | revnow.com
All Rights Reserved.
This project was created by PhucTruong

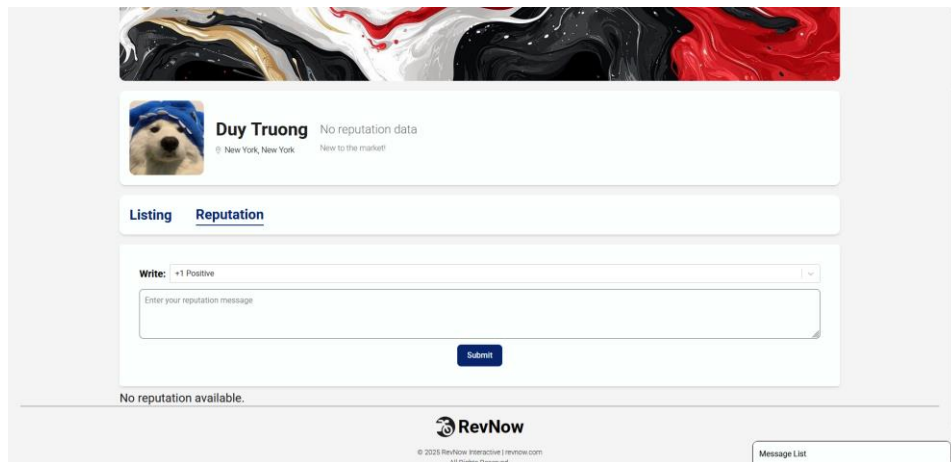
Hình 3. 29 Những giao dịch của người dùng

– Tại trang Giao dịch, người dùng có thể quan sát hiệu quả hoạt động kinh doanh thông qua biểu đồ doanh thu theo thời gian trong năm. Biểu đồ này được phân loại dựa trên tình trạng xe đã bán, cho phép phân tích và so sánh doanh thu từ xe mới và xe cũ, mang lại cái nhìn tổng quan về hiệu suất kinh doanh. Hệ thống cung cấp hai dạng biểu đồ chính: biểu đồ cột thể hiện số lượng xe bán ra theo từng tháng, phân chia rõ ràng giữa xe mới và xe đã qua sử dụng, và biểu đồ tròn thể hiện tỷ lệ phần trăm giữa hai loại xe này. Người dùng có thể tùy chọn xem thống kê theo từng năm để dễ dàng so sánh và đánh giá kết quả kinh doanh trong các giai đoạn khác nhau. Ngay bên dưới khu vực biểu đồ là phần hiển thị chi tiết danh sách các giao dịch đã diễn ra, trong đó mỗi giao dịch đều được trình bày rõ ràng với đầy đủ thông tin như xe máy cụ thể được giao dịch, thông tin người thực hiện giao dịch cùng với người dùng, thời điểm tiến hành, hình thức giao dịch là mua hay bán, giá trị giao dịch, và trạng thái hiện tại như đang xử lý (Pending) hoặc đã hoàn thành (Completed). Ngoài ra, mỗi mục giao dịch còn đi kèm nút thao tác (Action) tương ứng, mở ra các tùy chọn tương tác phù hợp với trạng thái và loại giao dịch. Người bán có thể xác nhận hoàn thành giao dịch khi việc trao đổi đã hoàn tất hoặc hủy các giao dịch còn đang xử lý khi cần thiết. Hệ thống cũng cho phép người dùng lọc danh sách giao dịch theo loại (mua hoặc bán), trạng thái, khoảng thời gian và giá trị để dễ dàng tìm kiếm và quản lý. Đồng thời, tính năng liên hệ trực tiếp giữa người mua và người bán được tích hợp nhằm tăng tính linh hoạt và minh bạch trong quá trình giao dịch, từ đó nâng cao trải nghiệm người dùng và hiệu quả sử dụng hệ thống.

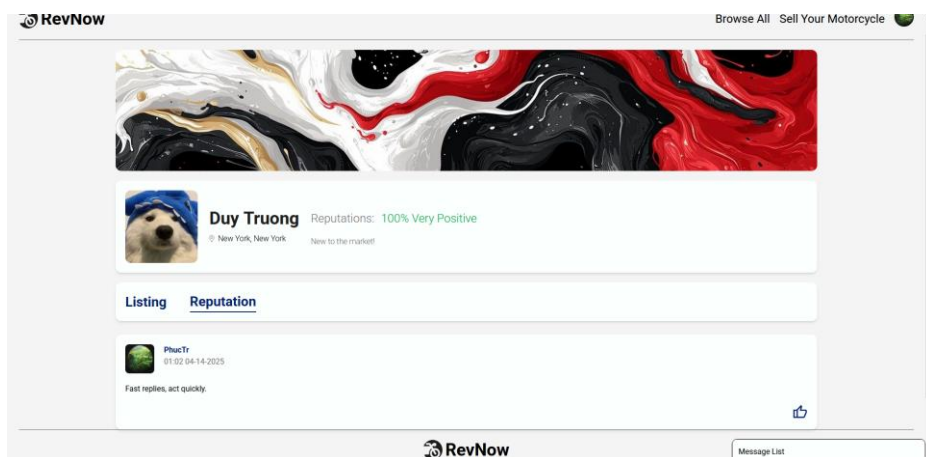


Hình 3. 30 Lưu đồ chức năng giao dịch

3.2.9. Đánh giá uy tín



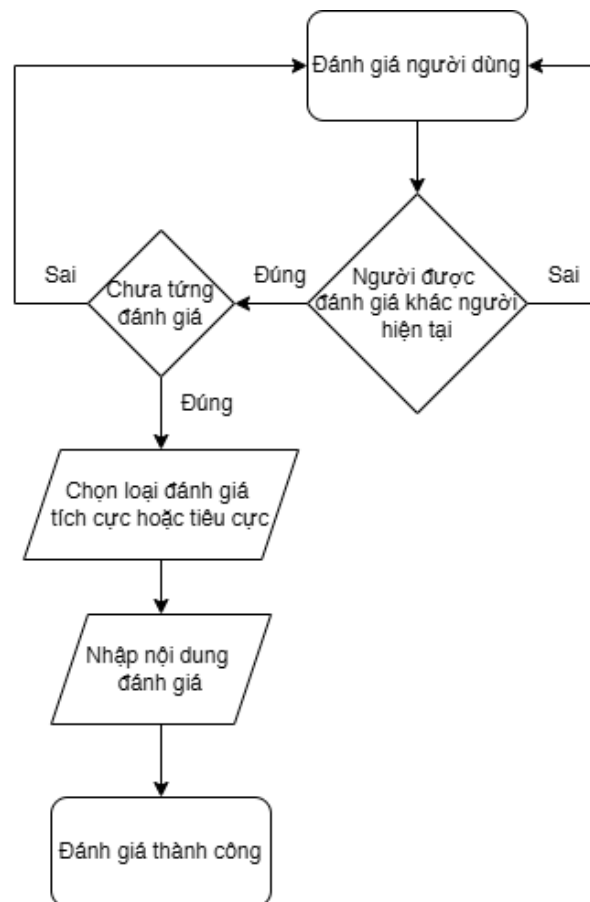
Hình 3. 31 Mục đánh giá uy tín của người dùng khác



Hình 3. 32 Hiện thị ra những đánh giá của người đó

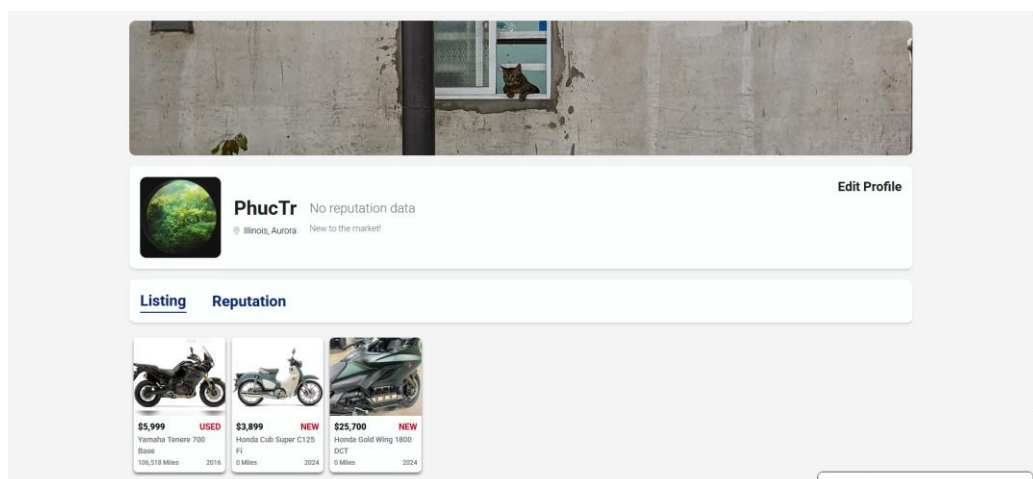
– Chức năng đánh giá uy tín người dùng trong hệ thống được thiết kế nhằm cung cấp một cái nhìn toàn diện và minh bạch về độ tin cậy của từng cá nhân, dựa trên lịch sử các phản hồi từ cộng đồng. Hệ thống sẽ tự động tính toán tỷ lệ đánh giá

tích cực so với tổng số đánh giá mà người dùng nhận được, từ đó phân loại uy tín thành sáu mức độ rõ ràng: Rất tiêu cực (0–40% đánh giá tốt), Chủ yếu tiêu cực (41–60%), Trung lập (61–70%), Tích cực (71–80%), và Rất tích cực (81–100%). Kết quả hiển thị dưới dạng phần trăm đánh giá tích cực, đi kèm với nhãn phân loại và màu sắc tương ứng để người dùng dễ dàng nhận biết.

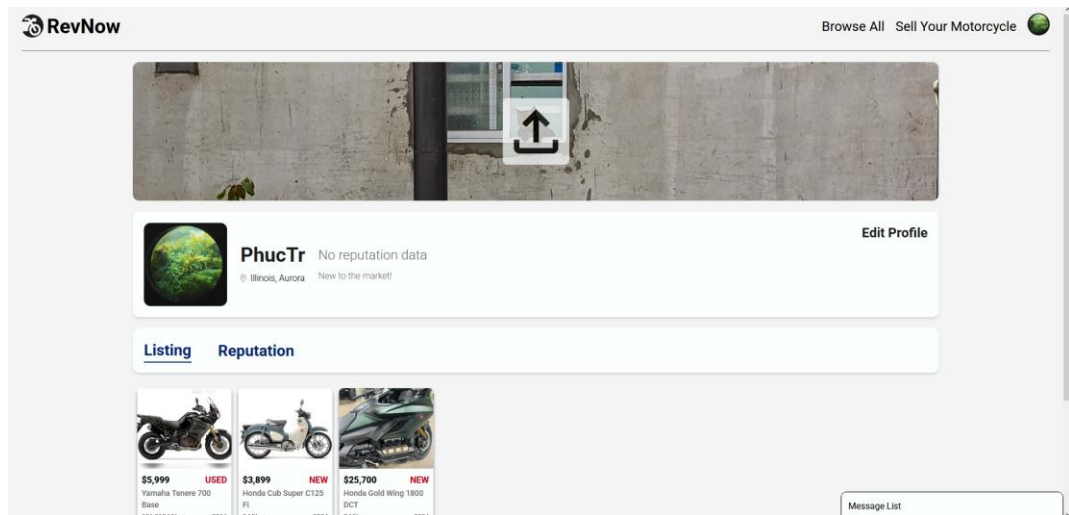


Hình 3. 33 Lưu đồ chức năng đánh giá người dùng

3.2.10. Cá nhân hoá người dùng



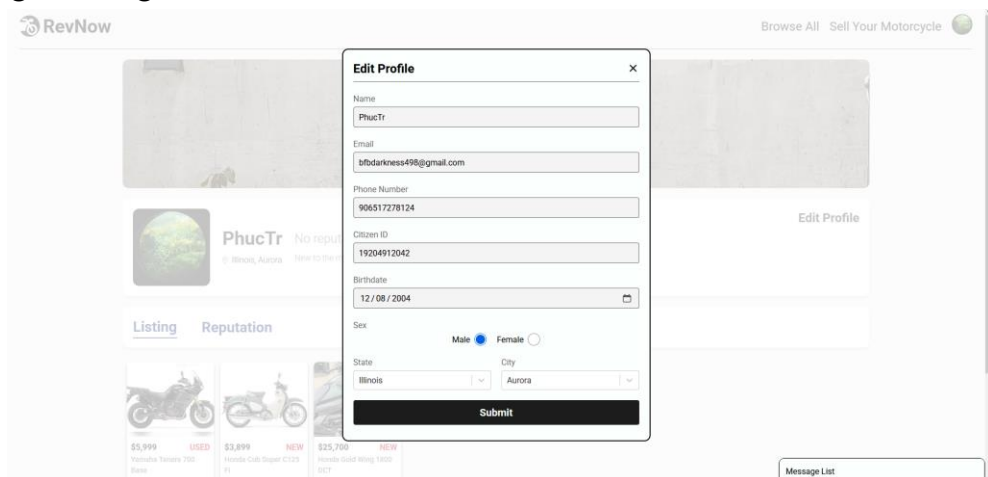
Hình 3. 34 Trang cá nhân



Hình 3. 35 Nút tải lên của ảnh bìa

– Trang hồ sơ cá nhân trong hệ thống cho phép người dùng xem và cập nhật các thông tin cá nhân của chính mình cũng như xem trang cá nhân của người dùng khác. Giao diện được thiết kế trực quan, thân thiện với người dùng, với điểm nhấn là ảnh bìa và ảnh đại diện có thể tùy chỉnh dễ dàng. Khi người dùng di chuột đến giữa ảnh bìa hoặc ảnh đại diện, hệ thống sẽ hiển thị nút tải lên, cho phép người dùng cập nhật hình ảnh mới một cách thuận tiện.

– Ngay bên dưới ảnh bìa là phần thông tin cơ bản của người dùng, bao gồm tên hiển thị, địa chỉ liên hệ và chỉ số thể hiện độ uy tín của tài khoản. Phía dưới là thanh điều hướng giúp người dùng chuyển đổi giữa các nội dung khác nhau trên trang hồ sơ, trong đó phần mặc định là mục Listing, nơi hiển thị danh sách các xe đang được người dùng rao bán.



Hình 3. 36 Chỉnh sửa thông tin cá nhân

– Khi người dùng chọn mục Edit Profile, một biểu mẫu sẽ xuất hiện, cho phép cập nhật các thông tin cá nhân bao gồm: họ tên, email, số điện thoại, CMND/CCCD, ngày sinh, giới tính, bang và thành phố cư trú.

– Khi truy cập vào giao diện chỉnh sửa, hệ thống sẽ tự động tải danh sách các bang từ một tập dữ liệu JSON lưu trữ sẵn. Nếu trước đó người dùng đã từng nhập địa

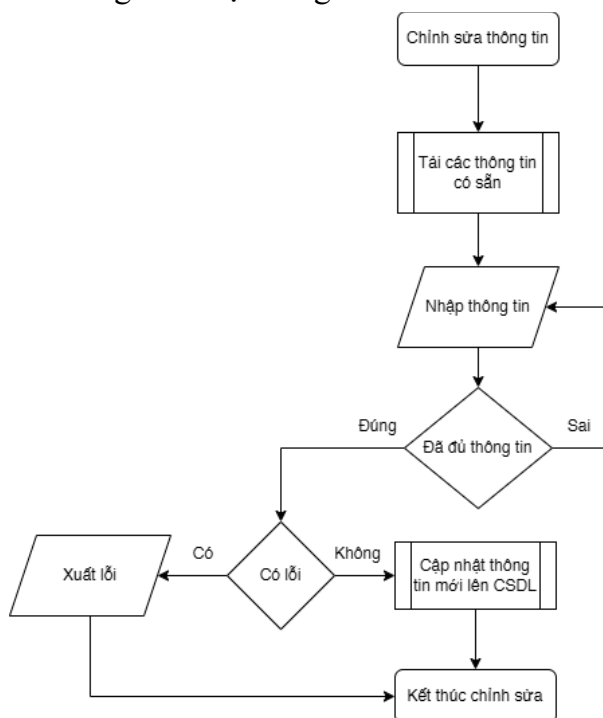
chỉ, hệ thống sẽ khôi phục lại thông tin bang và thành phố tương ứng, đồng thời tự động tải danh sách các thành phố thuộc bang đã chọn. Điều này giúp quá trình chỉnh sửa diễn ra nhanh chóng và chính xác hơn.

- Nếu người dùng thay đổi bang, danh sách thành phố sẽ được cập nhật tương ứng. Ngược lại, nếu không có bang nào được chọn, danh sách thành phố sẽ bị xóa để ngăn việc chọn nhầm. Hệ thống sẽ tạm thời lưu lại thành phố đã chọn để sử dụng khi người dùng nhấn nút Submit.

- Khi gửi biểu mẫu, hệ thống sẽ kiểm tra tính hợp lệ của tất cả các trường bắt buộc: họ tên, email, số điện thoại, CMND/CCCD, ngày sinh và giới tính. Các dữ liệu đầu vào như số điện thoại và CMND/CCCD sẽ được chuẩn hóa từ chuỗi sang số nguyên để đảm bảo đúng kiểu dữ liệu trong cơ sở dữ liệu. Trường giới tính (biến `is_man`) cũng được chuyển đổi từ chuỗi "true" hoặc "false" thành kiểu boolean để tránh lỗi định dạng.

- Nếu dữ liệu hợp lệ, hệ thống sẽ gửi truy vấn cập nhật đến bảng USER trong Supabase, sử dụng uid của người dùng để xác định bản ghi cần cập nhật. Nếu cập nhật thành công, hệ thống sẽ hiển thị thông báo “Profile updated successfully!” và tự động tải lại trang để áp dụng thay đổi mới. Trong trường hợp xảy ra lỗi kết nối hoặc truy vấn thất bại, hệ thống sẽ hiển thị thông báo “Update failed!” và giữ nguyên dữ liệu hiện tại để người dùng dễ dàng sửa và gửi lại. Các lỗi này cũng được ghi log để phục vụ việc kiểm tra và xử lý sau.

- Tính năng chỉnh sửa hồ sơ không chỉ giúp người dùng cập nhật thông tin cá nhân một cách thuận tiện và chính xác, mà còn đảm bảo dữ liệu luôn đồng bộ, đúng chuẩn và an toàn trong toàn hệ thống.



Hình 3. 37 Lưu đồ chức năng chỉnh sửa thông tin người dùng

KẾT LUẬN – ĐÁNH GIÁ

❖ Kết quả đạt được

Dự án đã hoàn thiện các chức năng cốt lõi, đảm bảo hệ thống có thể hoạt động như một nền tảng giao dịch xe máy trực tuyến. Qua đó, từng bước xây dựng được một môi trường giao thương dành riêng cho xe máy, tạo tiền đề phát triển cộng đồng người dùng có nhu cầu mua bán, trao đổi phương tiện. Hệ thống bước đầu đã đáp ứng được nhu cầu cơ bản của người dùng và là nền tảng để tiếp tục mở rộng và hoàn thiện trong tương lai.

❖ Thu hoạch chuyên môn – Kinh nghiệm

Trong quá trình thực hiện đồ án, đã tích lũy được nhiều kiến thức và kinh nghiệm thực tiễn về phát triển ứng dụng web hiện đại. Cụ thể là nắm vững cách xây dựng giao diện người dùng với React kết hợp cùng TailwindCSS và công cụ phát triển Vite, đồng thời sử dụng thành thạo Visual Studio Code, Git và GitHub để quản lý mã nguồn. Ngoài ra, đã tiếp cận và triển khai hiệu quả các dịch vụ Supabase, đặc biệt là việc quản lý cơ sở dữ liệu bằng PostgreSQL, tích hợp các cơ chế xác thực người dùng thông qua JWT và Google Auth. Quá trình phát triển còn giúp hiểu rõ hơn về cách sử dụng các thư viện hỗ trợ trong React như Chart.js để trực quan hóa dữ liệu, react-query để tối ưu quá trình gọi API, react-router để điều hướng ứng dụng.... Việc kết hợp các công nghệ và thư viện một cách hiệu quả đã góp phần nâng cao tư duy lập trình và khả năng giải quyết vấn đề trong quá trình xây dựng hệ thống thực tế.

❖ Ưu điểm, Hạn chế - Nguyên nhân

• Ưu điểm

Hệ thống mang lại nhiều điểm mạnh đáng ghi nhận trong quá trình triển khai. Các chức năng cốt lõi như thêm, sửa, xóa thông tin xe máy và người dùng được xây dựng hoàn chỉnh và hoạt động ổn định, không xuất hiện lỗi nghiêm trọng trong quá trình thử nghiệm. Tính năng được thiết kế đáp ứng đúng nhu cầu sử dụng cơ bản của người dùng, đảm bảo trải nghiệm tương đối mượt mà và chính xác.

• Hạn chế

Tuy nhiên, hệ thống vẫn tồn tại một số hạn chế. Giao diện chưa được đầu tư nhiều về hiệu ứng chuyển động nên trải nghiệm người dùng khi chuyển trang hoặc tải dữ liệu còn thiếu mượt mà. Ngoài ra, do phụ thuộc hoàn toàn vào hạ tầng Supabase để xử lý API và truy xuất dữ liệu từ cơ sở dữ liệu PostgreSQL, hệ thống yêu cầu có kết nối internet ổn định mới có thể hoạt động bình thường. Một số tính năng hiện tại chưa được tối ưu hiệu suất tốt, dẫn đến hiện tượng tải chậm trong một vài trường hợp, đặc biệt khi dữ liệu lớn hơn hoặc mạng yếu. Những hạn chế này xuất phát từ việc giới hạn về thời gian phát triển, tài nguyên và kinh nghiệm xử lý tối ưu hiệu năng cho các

ứng dụng quy mô lớn.

❖ **Hướng phát triển**

Trong tương lai, hệ thống sẽ tiếp tục được hoàn thiện và phát triển theo nhiều hướng nhằm nâng cao trải nghiệm người dùng và mở rộng tính ứng dụng thực tế. Trước hết, cần tập trung vào việc tối ưu hóa hiệu năng tổng thể của hệ thống, đặc biệt ở các thao tác xử lý dữ liệu lớn và cải thiện tốc độ tải trang nhằm đảm bảo khả năng mở rộng và hoạt động ổn định khi lượng người dùng tăng cao. Bên cạnh đó, việc tích hợp hệ thống thanh toán trực tuyến là một mục tiêu quan trọng, cho phép người dùng thực hiện giao dịch mua bán trực tiếp thông qua ngân hàng hoặc các cổng thanh toán trung gian như Momo. Điều này sẽ giúp nâng cao tính tiện lợi và chuyên nghiệp cho nền tảng. Về mặt bảo mật, hệ thống cần bổ sung các chức năng quan trọng như đổi mật khẩu, khôi phục mật khẩu qua email, xác thực số điện thoại, và cập nhật thông tin cá nhân. Những chức năng này yêu cầu tích hợp dịch vụ từ bên thứ ba để gửi mã xác thực qua SMS và email, giúp nâng cao tính an toàn cho tài khoản người dùng. Cuối cùng, để sản phẩm có thể vận hành ổn định trong thời gian dài và được đưa ra sử dụng thực tế, cần tiến hành mua tên miền, thuê máy chủ để triển khai hệ thống và duy trì hoạt động. Việc đưa sản phẩm ra thị trường sẽ là bước quan trọng để kiểm chứng hiệu quả và giá trị của nền tảng trong môi trường kinh doanh thực tế.

TÀI LIỆU THAM KHẢO

Tài liệu tiếng việt:

- [1]. Nguyễn Chí Cường. Lập trình căn bản. Khoa Kỹ thuật – Công nghệ, Đại Học Tây Đô, 2023.
- [2]. Nguyễn Chí Cường. Phân tích và thiết kế hệ thống thông tin. Khoa Kỹ thuật – Công Nghệ, Đại Học Tây Đô, 2024.
- [3]. Ngô Thị Lan. Nhập Môn Công Nghệ Phần Mềm. Khoa Kỹ thuật – Công Nghệ, Đại Học Tây Đô, 2024.
- [4]. Lâm Tấn Phương. Giáo trình Lý thuyết Thiết kế và lập trình Web, Khoa Kỹ thuật – Công nghệ, Đại Học Tây Đô, 2024.
- [5]. Lâm Tấn Phương. Chuyên đề ngôn ngữ lập trình, Khoa Kỹ thuật – Công nghệ, Đại Học Tây Đô, 2024.

Tài liệu từ Internet:

- [6]. Nick White. Git Tutorial For Dummies. <https://www.youtube.com/watch?v=mJ-qvsxPHpY> (Tháng 3/2025).
- [7]. IBM Technology. What is an API? <https://www.ibm.com/think/topics/api> (Tháng 3/2025).
- [8]. IBM Technology. What is a REST API?. <https://www.youtube.com/watch?v=lsMQRaeKNDk> (Tháng 3/2025).
- [9]. Kevin Stratvert. Figma Tutorial for Beginners. <https://www.youtube.com/watch?v=ezldKx-jPag> (Tháng 3/2025).
- [10]. React. React Reference. <https://react.dev/reference/react> (Tháng 3/2025).
- [11]. JavaScript Mastery. React JS 19 Full Course 2025 | Build an App and Master React in 2 Hours. <https://www.youtube.com/watch?v=dCLhUialKPQ>. (Tháng 3/2025).
- [12]. Tailwindcss. Tailwindcss Documents. <https://tailwindcss.com/docs> (Tháng 3/2025).
- [13]. JavaScript Mastery. Tailwind CSS v4 Full Course 2025 | Master Tailwind in One Hour. https://www.youtube.com/watch?v=6biMWgD6_JY (Tháng 3/2025).
- [14]. Supabase. Supabase Docs. <https://supabase.com/docs> (Tháng 3/2025).
- [15]. PedroTech. React Supabase CRUD Tutorial | Supabase Beginner Tutorial. <https://www.youtube.com/watch?v=tW1HO7i9EIM> (Tháng 3/2025)
- [16]. PedroTech. Build and Deploy a Fullstack Social Media App with Authentication | React, Supabase, PostgreSQL. <https://www.youtube.com/watch?v=sSTzz13tVY> (Tháng 3/2025).
- [17]. Mozilla. JavaScript Reference. <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference> (Tháng 3/2025).