

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:

XÂY DỰNG WEBSITE BÁN KÍNH THUỐC HỖ TRỢ VIỆC THỬ MẮT KÍNH TRỰC TUYẾN

Người hướng dẫn: **THS. ĐỖ THỊ TUYẾT HOA**

Sinh viên thực hiện: **NGUYỄN THÀNH ĐẠT**

Số thẻ sinh viên: **102200165**

Lớp: **20TCLC_DT4**

Đà Nẵng, 6/2024

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:

XÂY DỰNG WEBSITE BÁN KÍNH THUỐC
HỖ TRỢ VIỆC THỬ MẮT KÍNH
TRỰC TUYẾN

Người hướng dẫn: THS. ĐỖ THỊ TUYẾT HOA

Sinh viên thực hiện: NGUYỄN THÀNH ĐẠT

Số thẻ sinh viên: 102200165

Lớp: 20TCLC_DT4

Đà Nẵng, 6/2024

TÓM TẮT

Tên đề tài: Xây dựng website bán kính thuốc hỗ trợ việc thử mắt kính trực tuyến

Sinh viên thực hiện: Nguyễn Thành Đạt

Số thẻ SV: 102200165

Lớp: 20TCLC_DT4

Trong thời đại số hóa hiện nay, thương mại điện tử đã trở thành một phần không thể thiếu của cuộc sống, đặc biệt là trong lĩnh vực mua sắm trực tuyến. Tuy nhiên, việc mua kính mắt trực tuyến vẫn gặp thách thức lớn do người tiêu dùng không thể thử kính trực tiếp trước khi mua, dẫn đến khả năng mua kính không phù hợp. Để giải quyết vấn đề này, dự án đề xuất phát triển một nền tảng thương mại điện tử kết hợp các thuật toán xử lý ảnh nhằm cung cấp trải nghiệm thử kính mắt trực tuyến.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Nguyễn Thành Đạt

Số thẻ sinh viên: 102200165

Lớp: 20TCLC_DT4 Khoa: Công nghệ thông tin Ngành: Công nghệ phần mềm

1. Tên đề tài đồ án: *Xây dựng website bán kính thuốc hỗ trợ việc thử mắt kính trực tuyến*
2. Đề tài thuộc diện: ☐ Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện
3. Các số liệu và dữ liệu ban đầu:
4. Nội dung các phần thuyết minh và tính toán:
 - Chương 1: Tổng quan
 - Chương 2: Cơ sở lý thuyết
 - Chương 3: Phân tích và thiết kế hệ thống
 - Chương 4: Xây dựng chương trình
 - Chương 5: Kết luận
5. Các bản vẽ, đồ thị:
 - Biểu đồ Usecase
 - Biểu đồ trình tự
 - Biểu đồ lớp
6. Họ tên người hướng dẫn: ThS. Đỗ Thị Tuyết Hoa
7. Ngày giao nhiệm vụ đồ án: 04/04/2024
8. Ngày hoàn thành đồ án: 24/06/2024

Đà Nẵng, ngày tháng 06 năm 2024

Trưởng Bộ môn

Người hướng dẫn

LỜI NÓI ĐẦU

Em xin gửi lời cảm ơn chân thành và sâu sắc đến các thầy cô trong Khoa Công nghệ thông tin, cũng như tất cả các thầy cô trong trường Đại học Bách khoa – Đại học Đà Nẵng đã dìu dắt, dạy dỗ và truyền đạt kiến thức, kinh nghiệm quý báu của mình trong suốt quá trình em học tập và nghiên cứu tại trường.

Em xin bày tỏ tình cảm và lòng biết ơn chân thành của em tới cô giáo ThS. Đỗ Thị Tuyết Hoa, người đã từng bước hướng dẫn, giúp đỡ em tận tình trong quá trình thực hiện đồ án tốt nghiệp của mình. Nhờ đó em có thể hoàn thành đồ án đúng tiến độ và tích lũy cho mình nhiều kiến thức quý báu.

Con xin gửi lời cảm ơn to lớn nhất đến cha, mẹ và gia đình. Cha, mẹ và gia đình đã luôn ở bên con, là nguồn động lực không mệt mỏi và là chỗ dựa tinh thần vững chắc giúp con vượt qua những khó khăn để hoàn thành đồ án này.

Mặc dù đã cố gắng hoàn thành đồ án tốt nhất nhưng thời gian và kiến thức còn có hạn nên sẽ không tránh khỏi những thiếu sót nhất định, rất mong được sự cảm thông và tận tình chỉ bảo, góp ý của quý thầy cô giáo cũng như tất cả các bạn để kết quả của em được hoàn thiện hơn.

Một lần nữa em xin chân thành cảm ơn!

CAM ĐOAN

Em xin cam đoan:

1. Nội dung trong đề án này là do em thực hiện dưới sự hướng dẫn trực tiếp của ThS. Đỗ Thị Tuyết Hoa
2. Các tham khảo dùng trong đề án đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.
3. Nếu có những sao chép không hợp lệ, vi phạm quy chế đào tạo, em xin chịu hoàn toàn trách nhiệm.

Đà Nẵng, ngày tháng 6 năm 2024
Sinh viên thực hiện

Nguyễn Thành Đạt

MỤC LỤC

TÓM TẮT	
NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP	
LỜI NÓI ĐẦU	i
CAM ĐOAN.....	ii
MỤC LỤC	iii
DANH SÁCH CÁC BẢNG, HÌNH VẼ	vi
DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT	ix
MỞ ĐẦU	1
Chương 1: TỔNG QUAN	3
1.1. Giới thiệu đề tài	3
1.2. Mục đích, ý nghĩa, yêu cầu.....	3
1.2.1. Mục đích	3
1.2.2. Ý nghĩa	4
1.2.3. Yêu cầu	4
2. Trình tự triển khai.....	4
3. Công cụ sử dụng	4
4. Những kết quả dự kiến đạt được	5
Chương 2: CƠ SỞ LÝ THUYẾT	6
2.1. Ngôn ngữ lập trình.....	6
2.1.1. Typescript	6
2.1.2. Python.....	7
2.1.3. SQL.....	8
2.2. Công nghệ Frontend sử dụng.....	9
2.2.1. ReactJs	9
2.3. Công nghệ Backend sử dụng	11
2.3.1. NestJs.....	11
2.3.2. Cache memory	12
2.3.3. Flask.....	13
2.4. Hệ quản trị CSDL	14
2.4.1. Postgresql.....	14
2.4.2. Cloudinary	14
2.5. Giao thức truyền thông Client-Server	15

2.5.1.	Giao thức http	15
2.5.2.	Giao thức websocket	16
2.6.	Các khái niệm liên quan xử lý ảnh	17
2.6.1.	Haar Cascade Classifier.....	17
2.6.2.	OpenCV	18
2.6.3.	NumPy	18
Chương 3: PHÂN TÍCH, THIẾT KẾ DỮ LIỆU		19
3.1.	Phân tích đặc tả yêu cầu	19
3.1.1.	Chức năng.....	19
3.1.2.	Phi chức năng	20
3.2.	Biểu đồ Usecase.....	21
3.3.	Biểu đồ trình tự.....	38
3.4.	Biểu đồ lớp	43
Chương 4: XÂY DỰNG CHƯƠNG TRÌNH		51
4.1.	Kiến trúc hệ thống	51
4.1.1.	Mô tả chi tiết từng phần.....	51
4.1.2.	Luồng hoạt động.....	52
4.2.	Sơ đồ thuật toán ghép kính vào khuôn mặt	53
4.3.	Giao diện chương trình.....	55
4.3.1.	Trang đăng nhập	55
4.3.2.	Trang đăng ký.....	56
4.3.3.	Trang chủ.....	57
4.3.4.	Trang xem trước chi tiết sản phẩm	58
4.3.4.	Trang kết quả của việc thử kính	58
4.3.5.	Trang chi tiết sản phẩm	59
4.3.6.	Trang các sản phẩm liên quan	60
4.3.7.	Trang xem đánh giá sản phẩm.....	60
4.3.8.	Trang đánh giá sản phẩm.....	61
4.3.9.	Trang cập nhật thông tin cá nhân.....	61
4.3.10.	Trang danh sách yêu thích.....	62
4.3.11.	Trang giỏ hàng.....	62
4.3.12.	Trang xác nhận thông tin mua hàng	63
4.3.13.	Trang chọn phương thức thanh toán ngân hàng	63
4.3.14.	Trang thanh toán bằng mã QR.....	64
4.3.15.	Trang xác nhận thông tin mua hàng	64

4.3.16.	Trang cảm ơn mua hàng	65
4.3.17.	Trang lịch sử mua hàng	65
4.3.18.	Trang quản xem thống kê doanh thu	66
Chương 5: KẾT LUẬN		67
5.1.	Những kết quả đạt được.....	67
5.1.1.	Về mặt lý thuyết	67
5.1.2.	Về mặt ứng dụng	67
5.2.	Hạn chế.....	68
5.3.	Hướng phát triển.....	68

DANH SÁCH CÁC BẢNG, HÌNH VẼ

Bảng 3.1 Đặc tả Usecase đăng nhập.....	22
Bảng 3.2 Đặc tả Usecase đăng ký.....	24
Bảng 3.3 Đặc tả Usecase sản phẩm yêu thích	25
Bảng 3.4 Đặc tả Usecase thông tin cá nhân.....	26
Bảng 3.5 Đặc tả Usecase sản phẩm	28
Bảng 3.6 Đặc tả Usecase đánh giá sản phẩm	29
Bảng 3.7 Đặc tả Usecase quản lý giỏ hàng.....	30
Bảng 3.8 Đặc tả Usecase quản lý đơn hàng.....	31
Bảng 3.9 Đặc tả Usecase mua hàng.....	33
Bảng 3.10 Đặc tả Usecase thử kính online	34
Bảng 3.11 Đặc tả Usecase doanh thu	35
Bảng 3.12 Đặc tả Usecase quản lý người dùng	36
Bảng 3.13 Thuộc tính bảng User	44
Bảng 3.14 Thuộc tính bảng Glasses	45
Bảng 3.15 Thuộc tính bảng Favorite	46
Bảng 3.16 Thuộc tính bảng Review	46
Bảng 3.17 Thuộc tính bảng CartItem	47
Bảng 3.18 Thuộc tính bảng Order	48
Bảng 3.19 Thuộc tính bảng OrderItem.....	49
Bảng 3.20 Thuộc tính bảng Payment.....	49

DANH SÁCH CÁC BẢNG, HÌNH VẼ

Hình 2.1 Typescript logo	6
Hình 2.2 Typescript compiler	6
Hình 2.3 Python logo	7
Hình 2.4 Ứng dụng của Python	8
Hình 2.5 Câu lệnh ví dụ cho SQL	9
Hình 2.6 ReactJs logo	9
Hình 2.7 Vòng đời của ReactJs	10
Hình 2.8 NestJs logo	11
Hình 2.9 NestJs request lifecycle	12
Hình 2.10 Mô hình caching trong ứng dụng web	13
Hình 2.11 Flask logo	13
Hình 2.12 Postgresql logo	14
Hình 2.13 Mô hình lưu trữ Cloudinary	15
Hình 2.14 Mô hình http Client - Server	16
Hình 2.15 Kết nối websocket	17
Hình 3.1. Biểu đồ usecase tổng quát hệ thống	21
Hình 3.2 Biểu đồ Usecase đăng nhập	22
Hình 3.3 Biểu đồ Usecase đăng ký	23
Hình 3.4 Biểu đồ Usecase sản phẩm yêu thích	25
Hình 3.5 Biểu đồ Usecase thông tin cá nhân	26
Hình 3.6 Biểu đồ Usecase sản phẩm	28
Hình 3.7 Biểu đồ Usecase đánh giá sản phẩm	29
Hình 3.8 Biểu đồ Usecase quản lý giỏ hàng	30
Hình 3.9 Biểu đồ Usecase quản lý đơn hàng	31
Hình 3.10 Biểu đồ Usecase mua hàng	33
Hình 3.11 Biểu đồ Usecase thử kính online	34
Hình 3.12 Biểu đồ Usecase doanh thu	35
Hình 3.13 Biểu đồ Usecase quản lý người dùng	36
Hình 3.14 Biểu đồ trình tự phần Sign in	38
Hình 3.15 Biểu đồ trình tự phần Sign up	39
Hình 3.16 Biểu đồ trình tự phần Favorite	39
Hình 3.17 Biểu đồ trình tự phần Review	39
Hình 3.18 Biểu đồ trình tự phần Order	40

Hình 3.19 Biểu đồ trình tự phần Profile	40
Hình 3.20 Biểu đồ trình tự phần Glasses try on	41
Hình 3.21 Biểu đồ trình tự phần Purchase.....	42
Hình 3.22 Biểu đồ lớp hệ thống.....	43
Hình 4.1 Kiến trúc hệ thống	51
Hình 4.2 Sơ đồ thuật toán ghép kính vào khuôn mặt	53
Hình 4.3 Trang đăng nhập	55
Hình 4.4 Trang đăng ký	56
Hình 4.5 Trang chủ.....	57
Hình 4.6 Trang xem trước chi tiết sản phẩm	58
Hình 4.7 Trang kết quả của việc thử kính	58
Hình 4.8 Trang chi tiết sản phẩm	59
Hình 4.9 Trang các sản phẩm liên quan	60
Hình 4.10 Trang xem đánh giá sản phẩm.....	60
Hình 4.11 Trang đánh giá sản phẩm.....	61
Hình 4.12 Trang cập nhật thông tin cá nhân.....	61
Hình 4.13 Trang danh sách yêu thích	62
Hình 4.14 Trang giỏ hàng.....	62
Hình 4.15 Trang xác nhận thông tin mua hàng	63
Hình 4.16 Trang chọn phương thức thanh toán ngân hàng	63
Hình 4.17 Trang thanh toán bằng mã QR.....	64
Hình 4.19 Trang xác nhận thông tin mua hàng	64
Hình 4.20 Trang cảm ơn mua hàng	65
Hình 4.21 Trang lịch sử mua hàng	65
Hình 4.22 Trang quản xem thống kê doanh thu	66

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

KÝ HIỆU	CHỮ VIẾT TẮT
CSDL	Cơ sở dữ liệu
API	Application Programming Interface
RESTful	Representational State Transfer
HTTP	HyperText Markup Language
UI	User Interface
OOP	Object-Oriented Programming
FP	Functional Programming
DI	Dependency Injection
DDL	Data Definition Language
ACID	Atomicity, Consistency, Isolation, Durability
CDN	Content Delivery Network
SDK	Software Development Kit
TCP	Transmission Control Protocol
SSL	Secure Sockets Layer
TLS	Transport Layer Security
QR	Quick Response
LRU	Least Recently Used
FIFO	First In First Out
TTL	Time to Live
ORM	Object-Relational Mapping
NumPy	Numerical Python
SciPy	Scientific Python
URL	Uniform Resource Locator

MỞ ĐẦU

1. Mục đích thực hiện đề tài

Trong thời đại số hóa ngày nay, việc mua sắm trực tuyến đã trở thành một phần không thể thiếu của cuộc sống hàng ngày. Người tiêu dùng ngày càng chuyển sang mua hàng trực tuyến để tiết kiệm thời gian và tận hưởng sự thuận tiện. Tuy nhiên, việc mua kính mắt trực tuyến vẫn gặp thách thức do người mua không thể thử kính trước khi mua, dẫn đến khả năng mua kính không phù hợp với khuôn mặt hoặc phong cách của họ. Đề tài này được thực hiện một nền tảng thương mại điện tử kết hợp công nghệ xử lý ảnh, cho phép người trải nghiệm mua sắm của người tiêu dùng và hỗ trợ các nhà bán lẻ kính mắt trực tuyến.

Mục tiêu đề tài:

- Phát triển một hệ thống thử kính mắt trực tuyến sử dụng các thuật toán xử lý ảnh.
- Đảm bảo hệ thống có thể phát hiện và nhận diện khuôn mặt, mắt, từ đó gắn kính mắt vào hình ảnh của người dùng một cách tự động và chính xác.
- Tạo ra một trải nghiệm mua sắm kính mắt trực tuyến chân thực và thuận tiện cho người tiêu dùng.
- Hỗ trợ các nhà bán lẻ kính mắt trực tuyến trong việc cung cấp dịch vụ tốt hơn và mở rộng thị trường kinh doanh.

Phạm vi và đối tượng nghiên cứu:

- Nghiên cứu và ứng dụng các thuật toán xử lý ảnh trong việc phát hiện khuôn mặt và mắt.
- Phát triển và triển khai hệ thống thử kính mắt trực tuyến trên nền tảng thương mại điện tử.
- Đối tượng nghiên cứu là người tiêu dùng mua sắm kính mắt trực tuyến và các nhà bán lẻ kính mắt trực tuyến.

Phương pháp nghiên cứu:

- Phương pháp nghiên cứu tài liệu: Tìm hiểu và phân tích các thuật toán xử lý ảnh và các công nghệ liên quan.
- Phương pháp thực nghiệm: Xây dựng và thử nghiệm hệ thống thử kính mắt trực tuyến, đánh giá hiệu quả và tính khả thi của hệ thống.
- Phương pháp phân tích và đánh giá: Đánh giá kết quả thử nghiệm, phân tích các yếu tố ảnh hưởng đến hiệu quả của hệ thống, từ đó đưa ra các đề xuất cải tiến.

2. Cấu trúc của đề án tốt nghiệp

- Chương 1: Tổng quan - Trình bày giới thiệu, mục đích, ý nghĩa, yêu cầu, trình tự triển khai, công cụ sử dụng
- Chương 2: Cơ sở lý thuyết - Trình bày các khái niệm về các công nghệ sử dụng và các thư viện liên quan.
- Chương 3: Phân tích và thiết kế hệ thống - Phân tích đặc tả yêu cầu, thiết kế kiến trúc hệ thống và các sơ đồ hệ thống.
- Chương 4: Xây dựng chương trình - Trình bày sơ đồ chức năng hệ thống, một số sơ đồ thuật toán, và giao diện chương trình của hệ thống.
- Chương 5: Kết luận – Trình bày những kết quả đạt được, mặt hạn chế, và hướng phát triển của đề tài.

Chương 1: TỔNG QUAN

1.1. Giới thiệu đề tài

Trong thời đại số hóa ngày nay, thương mại điện tử đã trở thành một phần không thể thiếu của cuộc sống hàng ngày, đặc biệt là trong lĩnh vực mua sắm trực tuyến. Người tiêu dùng ngày càng chuyển sang việc mua hàng trực tuyến để tiết kiệm thời gian và tận hưởng sự thuận tiện mà nó mang lại. Trong tình hình này, việc mua kính mắt trực tuyến cũng không phải là một ngoại lệ.

Tuy nhiên, mặc dù việc mua sắm kính mắt trực tuyến mang lại sự thuận tiện về việc lựa chọn và đặt hàng từ xa, nhưng một thách thức lớn đối với người tiêu dùng là không thể thử kính trực tiếp trước khi mua. Việc này có thể dẫn đến việc mua kính không phù hợp với khuôn mặt hoặc phong cách của họ, gây ra sự không hài lòng và cảm giác bất tiện. Do đó, việc phát triển một nền tảng thương mại điện tử kết hợp việc thử kính mắt trực tuyến trở nên cần thiết hơn bao giờ hết.

Dự án này nhằm giải quyết vấn đề này bằng cách tạo ra một trải nghiệm mua sắm kính mắt trực tuyến mới mẻ, thú vị và chân thực hơn bằng việc sử dụng công nghệ để thử kính online. Bằng cách áp dụng các thuật toán và phương pháp trong lĩnh vực học máy và xử lý hình ảnh, dự án sẽ phát triển một hệ thống thử kính mắt trực tuyến chính xác và hiệu quả, giúp người tiêu dùng có thể thử kính mắt trực tuyến một cách dễ dàng và chính xác nhất có thể.

Việc phát triển một nền tảng thương mại điện tử về kính mắt kết hợp việc thử kính online không chỉ giúp nâng cao trải nghiệm mua sắm trực tuyến của người tiêu dùng mà còn tạo ra cơ hội kinh doanh mới cho các nhà sản xuất kính mắt.

1.2. Mục đích, ý nghĩa, yêu cầu

1.2.1. Mục đích

Dự án này được thực hiện với mục tiêu xây dựng một nền tảng thương mại điện tử độc đáo và tiện lợi, đặc biệt là cho việc mua sắm kính mắt trực tuyến. Dự án nhằm cung cấp cho người tiêu dùng một trải nghiệm mua sắm mới mẻ và hiệu quả, giúp họ có thể thử kính mắt trực tuyến một cách chân thực và thuận tiện từ nhà.

1.2.2. Ý nghĩa

Việc phát triển một nền tảng thương mại điện tử cho kính mắt kết hợp việc thử kính trực tuyến mang lại nhiều ý nghĩa quan trọng. Đối với người tiêu dùng, dự án sẽ giúp họ có thể lựa chọn và mua sắm kính mắt trực tuyến một cách tự tin và hiệu quả hơn, bằng cách thử kính mắt trực tuyến ngay tại nhà mà không cần phải đến cửa hàng. Điều này không chỉ tiết kiệm thời gian mà còn giúp họ tăng cường sự thoải mái và tự tin khi quyết định mua sắm kính mắt.

Đồng thời, dự án cũng mang lại lợi ích kinh doanh đối với các nhà sản xuất và nhà bán lẻ kính mắt. Bằng cách tạo ra một nền tảng thương mại điện tử tiên tiến và tích hợp tính năng thử kính mắt trực tuyến, họ có thể tiếp cận một thị trường tiềm năng lớn và mở rộng phạm vi kinh doanh của mình. Đồng thời, việc thu thập dữ liệu từ quá trình thử kính cũng giúp họ hiểu rõ hơn về nhu cầu và sở thích của người tiêu dùng, từ đó tối ưu hóa chiến lược kinh doanh và phát triển sản phẩm một cách hiệu quả.

1.2.3. Yêu cầu

- Phát triển một giao diện thương mại điện tử đẹp mắt, dễ sử dụng, và thân thiện với người dùng.
- Xây dựng hệ thống thử kính mắt trực tuyến bằng công nghệ, đảm bảo khớp kính với từng khuôn mặt một cách chính xác.
- Tích hợp khả năng tải ảnh khuôn mặt để người dùng có thể thử kính mắt trực tuyến một cách thuận tiện và chân thực.

2. Trình tự triển khai

- Nghiên cứu và phân tích yêu cầu người dùng
- Thiết kế giao diện người dùng và cơ sở dữ liệu
- Phát triển hệ thống thử kính mắt trực tuyến
- Kiểm thử và sửa lỗi
- Triển khai duy trì hệ thống

3. Công cụ sử dụng

- Ngôn ngữ lập trình: Python, NestJs (Typescript), ReactJs
- Công cụ phát triển: Visual Studio Code
- Cơ sở dữ liệu: Postgresql, Cloudinary

4. Những kết quả dự kiến đạt được

Về mặt lý thuyết:

- Nền tảng thương mại điện tử về kính mắt: Dự án sẽ nghiên cứu và áp dụng các lý thuyết về thiết kế giao diện người dùng, trải nghiệm người dùng, và
- quản lý dữ liệu để xây dựng một nền tảng thương mại điện tử về kính mắt độc đáo và hấp dẫn.
- Thuật toán xử lý ảnh trong thử kính mắt: Qua việc tìm hiểu về các thuật toán và phương pháp trong lĩnh vực học máy và xử lý hình ảnh, dự án sẽ áp dụng và phát triển hệ thống thử kính mắt trực tuyến, giúp đảm bảo khớp kính với từng khuôn mặt một cách chính xác.

Về mặt ứng dụng

- Xây dựng hệ thống thử kính mắt trực tuyến: Dự án sẽ triển khai các thuật toán và phương pháp học máy để xử lý hình ảnh từ người dùng tải lên và đối chiếu với cơ sở dữ liệu kính mắt để đảm bảo việc thử kính mắt trực tuyến một cách chính xác và chân thực.
- Tạo ra trải nghiệm mua sắm trực tuyến mới: Sử dụng kiến thức về trải nghiệm người dùng và phát triển giao diện người dùng thân thiện, dự án sẽ xây dựng một trải nghiệm mua sắm trực tuyến mới mẻ và tiện lợi, giúp người dùng cảm thấy thoải mái và tự tin khi mua sắm kính mắt trực tuyến.
- Tăng cường hiệu quả kinh doanh: Bằng cách cung cấp một nền tảng thương mại điện tử thú vị và tích hợp tính năng thử kính mắt trực tuyến, dự án sẽ giúp tăng cường hiệu quả kinh doanh cho các nhà sản xuất kính mắt bằng cách mở rộng thị trường và nắm bắt nhu cầu của người tiêu dùng một cách chính xác.

Chương 2: CƠ SỞ LÝ THUYẾT

2.1. Ngôn ngữ lập trình

2.1.1. Typescript

TypeScript là một ngôn ngữ lập trình mã nguồn mở do Microsoft phát triển, được giới thiệu vào năm 2012. Nó là một phiên bản mở rộng của JavaScript, cung cấp thêm tính năng kiểu tĩnh (static typing) và các công cụ hỗ trợ lập trình hướng đối tượng như lớp (class) và giao diện (interface). TypeScript được biên dịch thành JavaScript trước khi chạy, cho phép các lập trình viên phát hiện lỗi sớm trong quá trình phát triển và cải thiện độ tin cậy của mã nguồn. TypeScript hoàn toàn tương thích với các thư viện JavaScript, được áp dụng rộng rãi trong nhiều framework và thư viện phát triển web và server. Với sự hỗ trợ của kiểu tĩnh và các công cụ phát triển mạnh mẽ, TypeScript giúp cải thiện hiệu suất, độ tin cậy và khả năng bảo trì của mã nguồn, đồng thời vẫn giữ được sự linh hoạt và tương thích với các dự án JavaScript hiện có.



Hình 2.1 Typescript logo



Hình 2.2 Typescript compiler

2.1.2. Python

Python là một ngôn ngữ lập trình đa mục đích, mã nguồn mở, được tạo ra bởi Guido van Rossum và ra mắt lần đầu vào năm 1991. Python nổi bật với cú pháp rõ ràng, dễ đọc và dễ học, giúp giảm thiểu lỗi và làm cho mã nguồn dễ bảo trì. Ngôn ngữ này hỗ trợ nhiều phong cách lập trình như lập trình hướng đối tượng, lập trình hàm và lập trình thủ tục, mang lại sự linh hoạt cho lập trình viên. Python đi kèm với một thư viện tiêu chuẩn phong phú, cung cấp nhiều module và hàm tiện ích, hỗ trợ nhiều tác vụ từ xử lý chuỗi, quản lý tệp đến kết nối mạng. Đặc biệt, Python có một cộng đồng phát triển mạnh mẽ, đóng góp hàng nghìn thư viện và công cụ mở rộng, làm tăng thêm tính hiệu quả và khả năng ứng dụng của ngôn ngữ này. Python là ngôn ngữ đa nền tảng, có thể chạy trên nhiều hệ điều hành khác nhau như Windows, macOS và Linux. Với ứng dụng rộng rãi trong phát triển web (Django, Flask), khoa học dữ liệu và học máy (NumPy, pandas, TensorFlow), tự động hóa và DevOps, cũng như phát triển phần mềm, Python là lựa chọn lý tưởng cho cả người mới học lập trình và các chuyên gia phát triển phần mềm.



Hình 2.3 Python logo

Applications of Python



Hình 2.4 Ứng dụng của Python

2.1.3. SQL

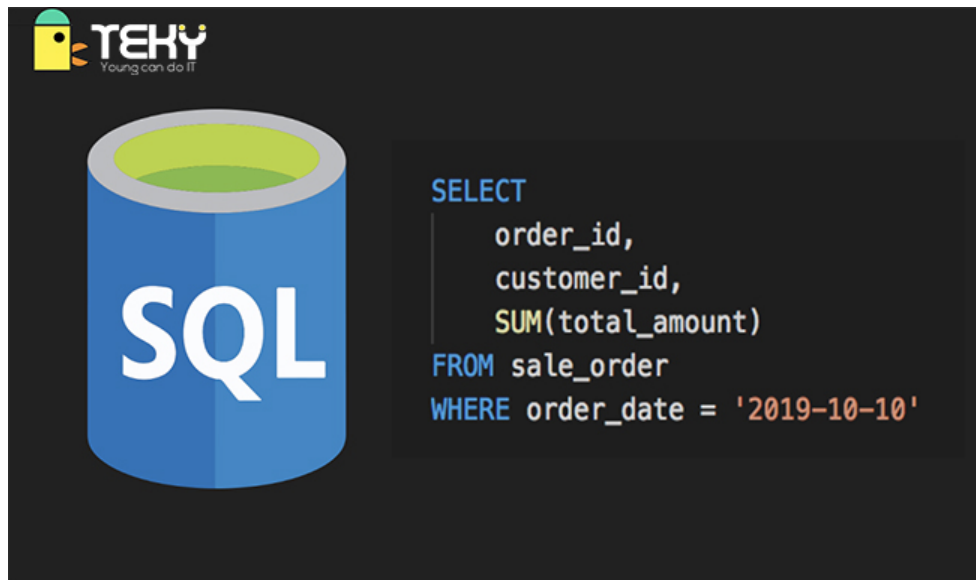
SQL (Structured Query Language) là ngôn ngữ tiêu chuẩn được sử dụng để quản lý và thao tác các cơ sở dữ liệu quan hệ. SQL được sử dụng để thực hiện nhiều chức năng khác nhau như truy vấn dữ liệu, chèn, cập nhật, và xóa dữ liệu, cũng như quản lý cấu trúc và quyền truy cập của cơ sở dữ liệu.

Khái niệm chính của SQL:

- **Truy vấn dữ liệu (Querying Data):** SQL cho phép truy vấn dữ liệu từ một hoặc nhiều bảng trong cơ sở dữ liệu. Câu lệnh SELECT là câu lệnh được sử dụng phổ biến nhất để truy vấn dữ liệu.
- **Thao tác dữ liệu (Manipulating Data):** SQL cung cấp các câu lệnh để thêm, cập nhật, và xóa dữ liệu trong cơ sở dữ liệu: INSERT INTO để thêm dữ liệu, UPDATE để cập nhật dữ liệu, DELETE để xóa dữ liệu.
- **Định nghĩa dữ liệu (Defining Data):** SQL cho phép định nghĩa cấu trúc của cơ sở dữ liệu và các bảng bằng các câu lệnh DDL (Data Definition Language) như: CREATE TABLE để tạo bảng mới, ALTER TABLE để sửa đổi cấu trúc của bảng, DROP TABLE để xóa bảng.
- **Kiểm soát truy cập (Controlling Access):** SQL cung cấp các câu lệnh để quản

lý quyền truy cập và bảo mật dữ liệu trong cơ sở dữ liệu, như GRANT và REVOKE.

- **Giao dịch (Transactions):** SQL hỗ trợ các giao dịch để đảm bảo tính nhất quán và toàn vẹn của dữ liệu. Các câu lệnh như BEGIN TRANSACTION, COMMIT, và ROLLBACK được sử dụng để quản lý giao dịch.

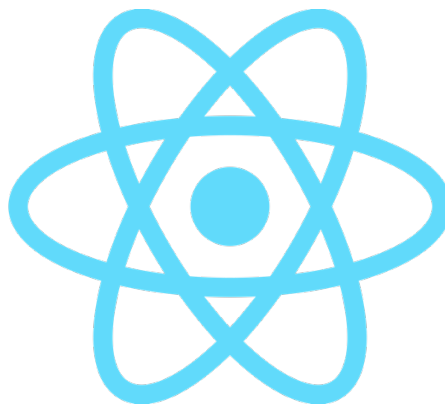


Hình 2.5 Câu lệnh ví dụ cho SQL

2.2. Công nghệ Frontend sử dụng

2.2.1. ReactJs

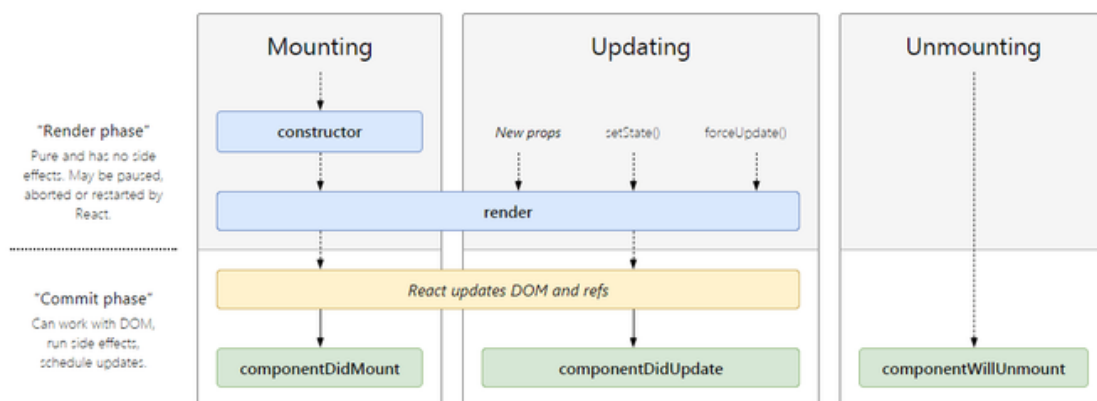
ReactJS là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, được sử dụng để xây dựng giao diện người dùng (UI) cho các ứng dụng web. ReactJS tập trung vào việc tạo ra các thành phần UI có thể tái sử dụng và quản lý trạng thái hiệu quả.



Hình 2.6 ReactJs logo

Khái niệm chính của ReactJs:

- **Kiến trúc dựa trên thành phần (Component-Based Architecture):** ReactJS xây dựng UI bằng cách chia nhỏ giao diện thành các thành phần độc lập, có thể tái sử dụng. Mỗi thành phần là một khối mã tự quản lý, có thể được kết hợp để tạo ra các giao diện phức tạp.
- **DOM ảo (Virtual DOM):** React sử dụng một phiên bản DOM ảo để tối ưu hóa hiệu suất. Khi trạng thái của một thành phần thay đổi, React sẽ tạo ra một DOM ảo mới và so sánh nó với DOM ảo trước đó để tìm ra sự khác biệt. Sau đó, React chỉ cập nhật những phần của DOM thực sự cần thay đổi, giảm thiểu số lượng thao tác với DOM thực tế và cải thiện hiệu suất.
- **JavaScript XML (JSX):** JSX là một cú pháp mở rộng cho JavaScript, cho phép viết HTML trong JavaScript. JSX giúp dễ dàng tạo ra các thành phần React bằng cách sử dụng cú pháp quen thuộc của HTML.
- **Ràng buộc dữ liệu một chiều (One-Way Data Binding):** React sử dụng ràng buộc dữ liệu một chiều, nghĩa là dữ liệu chỉ chảy theo một hướng, từ các thành phần cha xuống các thành phần con. Điều này giúp dễ dàng theo dõi và kiểm soát luồng dữ liệu trong ứng dụng.
- **Trạng thái và thuộc tính (State and Props):** Trạng thái là một đối tượng lưu trữ dữ liệu động của một thành phần, có thể thay đổi trong suốt vòng đời của thành phần đó. Thuộc tính là dữ liệu được truyền từ thành phần cha xuống thành phần con. Props không thể thay đổi bên trong thành phần con.
- **Phương thức vòng đời (Lifecycle Methods):** React cung cấp các phương thức vòng đời để quản lý các giai đoạn khác nhau trong vòng đời của một thành phần, như khi nó được gắn vào DOM, cập nhật, hoặc gỡ bỏ khỏi DOM.



Hình 2.7 Vòng đời của ReactJs

2.3. Công nghệ Backend sử dụng

2.3.1. NestJs

NestJs là một framework phát triển ứng dụng backend dựa trên Node.js, được xây dựng với TypeScript và lấy cảm hứng từ các kiến trúc lập trình hướng đối tượng (OOP), lập trình hàm (FP) và lập trình phản ứng (Reactive Programming). NestJS được thiết kế để giúp việc xây dựng các ứng dụng server-side có cấu trúc rõ ràng, dễ bảo trì và mở rộng, phù hợp với các ứng dụng quy mô từ nhỏ đến lớn.



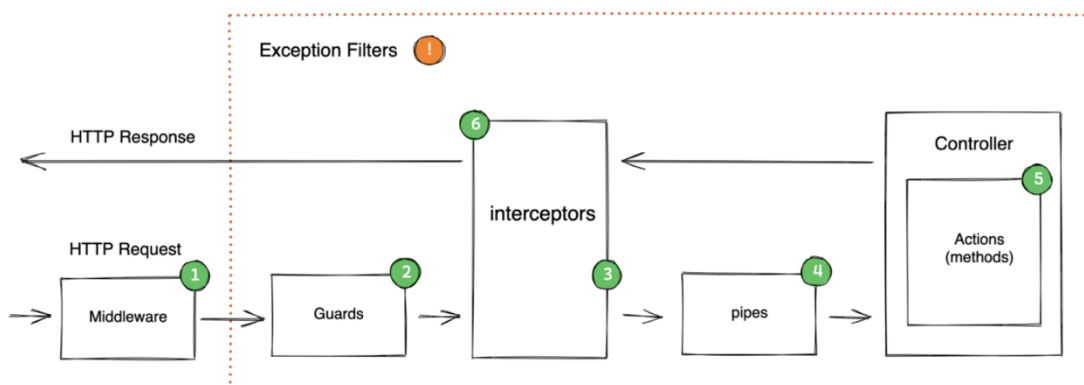
Hình 2.8 NestJs logo

Khái niệm chính của NestJs:

- **Typescript:** NestJS được xây dựng với TypeScript, mang lại các lợi ích của kiểu tĩnh và các công cụ phát triển hiện đại, giúp tăng cường độ tin cậy và khả năng bảo trì của mã nguồn.
- **Kiến trúc mô-đun:** NestJS sử dụng kiến trúc mô-đun, cho phép chia nhỏ ứng dụng thành các mô-đun độc lập, có thể tái sử dụng và quản lý dễ dàng. Mỗi mô-đun là một phần riêng biệt của ứng dụng, chứa các controller, service và các thành phần khác.
- **Decorator và Dependency Injection (DI):** NestJS sử dụng các decorator để xác định các thành phần của ứng dụng như controller, service, và các middleware. Dependency Injection được sử dụng để quản lý sự phụ thuộc giữa các thành phần, giúp mã nguồn rõ ràng và dễ kiểm tra.
- **Controller và Service:** Controller xử lý các yêu cầu HTTP và trả về phản hồi tương ứng và xác định các endpoint của API. Service chứa logic nghiệp vụ của ứng dụng, giúp tách biệt giữa logic và giao diện của API.
- **Middleware và Guards:** NestJS hỗ trợ middleware để xử lý các yêu cầu trước khi chúng đến controller, cũng như guards để kiểm tra quyền truy cập và xác thực.

người dùng.

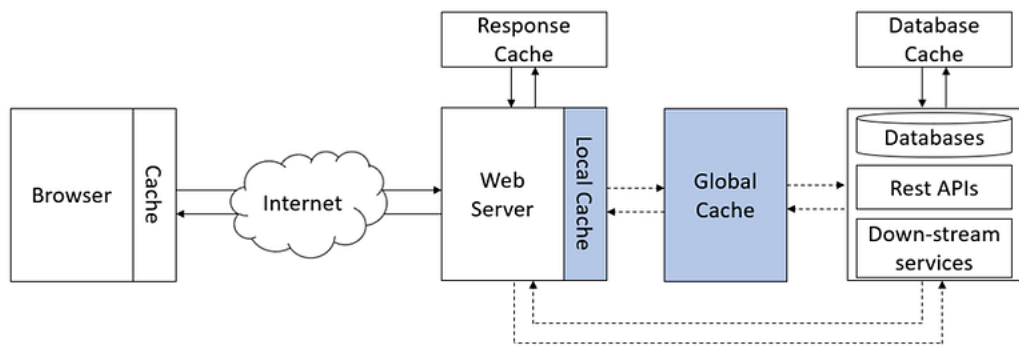
- Hỗ trợ cho nhiều cơ sở dữ liệu và ORM: NestJS tích hợp tốt với nhiều cơ sở dữ liệu và các công cụ ORM như TypeORM, Mongoose, và Sequelize, giúp dễ dàng kết nối và tương tác với cơ sở dữ liệu.



Hình 2.9 NestJs request lifecycle

2.3.2. Cache memory

Cache memory trong server backend là một bộ nhớ tạm thời được sử dụng để lưu trữ dữ liệu mà máy chủ có thể truy xuất nhanh chóng, nhằm cải thiện hiệu suất và tốc độ phản hồi của hệ thống. Bằng cách lưu trữ các dữ liệu thường xuyên được truy cập, cache giúp giảm tải cho cơ sở dữ liệu và các dịch vụ backend khác, từ đó cải thiện hiệu suất tổng thể và giảm độ trễ, mang lại trải nghiệm tốt hơn cho người dùng. Có nhiều loại cache khác nhau, như in-memory cache (ví dụ: Redis, Memcached) và distributed cache (ví dụ: Redis Cluster), giúp mở rộng khả năng lưu trữ và tăng độ tin cậy. Cache sử dụng các chiến lược lưu trữ và làm mới dữ liệu như write-through và write-back, cũng như các chính sách loại bỏ dữ liệu cũ khi cache đầy như Least Recently Used (LRU) hoặc First In First Out (FIFO). Quản lý tính nhất quán của dữ liệu trong cache là một thách thức quan trọng, đòi hỏi các kỹ thuật như TTL (Time to Live) và cache invalidation để đảm bảo dữ liệu trong cache luôn đồng bộ với dữ liệu gốc. Việc triển khai cache memory một cách hiệu quả đòi hỏi sự cân nhắc kỹ lưỡng để đảm bảo hệ thống hoạt động ổn định và đáng tin cậy.



Hình 2.10 Mô hình caching trong ứng dụng web

2.3.3. *Flask*

Flask là một micro web framework mã nguồn mở được viết bằng Python, phát triển bởi Armin Ronacher và ra mắt lần đầu vào năm 2010. Được coi là microframework do nó nhẹ và chỉ cung cấp các tính năng cơ bản như định tuyến và hệ thống mẫu, Flask mang lại sự linh hoạt tối đa cho lập trình viên trong việc lựa chọn các thành phần cho ứng dụng của mình. Với cú pháp đơn giản, dễ học và dễ sử dụng, Flask phù hợp cho cả người mới bắt đầu và các lập trình viên có kinh nghiệm. Thiết kế modular của Flask cho phép mở rộng ứng dụng thông qua các tiện ích mở rộng, mà không áp đặt bất kỳ quyết định kiến trúc nào, mang lại toàn quyền kiểm soát cho lập trình viên. Flask dễ dàng tích hợp với các thư viện và công cụ khác của Python, lý tưởng cho việc phát triển các ứng dụng web phức tạp, các RESTful API, các dự án web nhỏ và vừa, cũng như việc tạo nguyên mẫu nhanh. Sự kết hợp giữa tính đơn giản và khả năng tùy chỉnh cao khiến Flask trở thành một lựa chọn phổ biến và mạnh mẽ trong cộng đồng phát triển web.

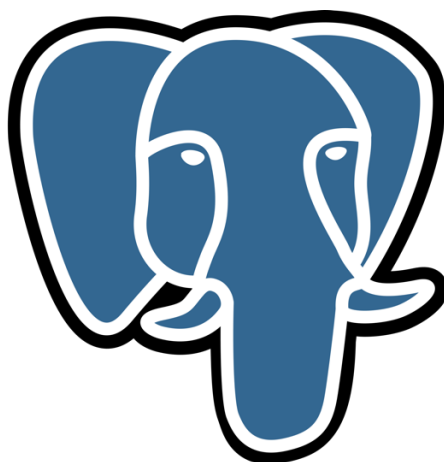


Hình 2.11 Flask logo

2.4. Hệ quản trị CSDL

2.4.1. PostgreSQL

PostgreSQL, thường được gọi tắt là Postgres, là một hệ quản trị cơ sở dữ liệu quan hệ đối tượng mã nguồn mở mạnh mẽ, được phát triển và duy trì bởi cộng đồng. Ra đời lần đầu vào năm 1986 tại Đại học California, Berkeley, và chính thức phát hành vào năm 1996, PostgreSQL đã trở thành một trong những hệ quản trị cơ sở dữ liệu tiên tiến và phổ biến nhất hiện nay. PostgreSQL hỗ trợ cả mô hình dữ liệu quan hệ truyền thống và các tính năng đối tượng, cho phép định nghĩa các kiểu dữ liệu và hàm phức tạp. Là một dự án mã nguồn mở, PostgreSQL có giấy phép sử dụng miễn phí và cho phép người dùng tự do sửa đổi, phân phối lại, thúc đẩy sự phát triển liên tục từ cộng đồng toàn cầu. Hệ quản trị này tuân thủ nghiêm ngặt các thuộc tính ACID (Atomicity, Consistency, Isolation, Durability), đảm bảo các giao dịch được thực hiện đáng tin cậy và bảo mật. PostgreSQL hỗ trợ đầy đủ các chuẩn SQL, xử lý tốt các truy vấn phức tạp và cung cấp nhiều tính năng bảo mật mạnh mẽ như kiểm soát truy cập dựa trên vai trò, xác thực qua nhiều phương thức và mã hóa dữ liệu. Với khả năng mở rộng và tích hợp tốt, PostgreSQL là một lựa chọn lý tưởng cho nhiều dự án cơ sở dữ liệu hiện đại, từ nhỏ đến lớn.



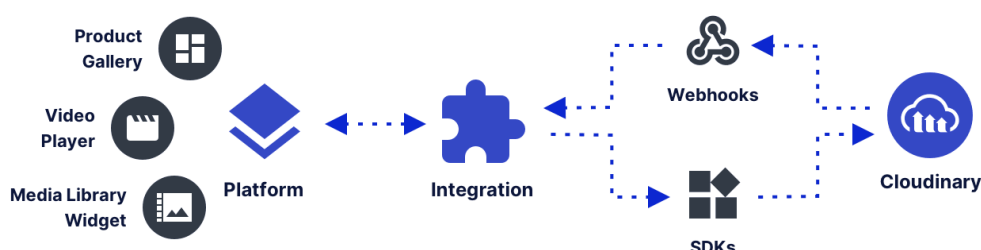
Hình 2.12 Postgresql logo

2.4.2. Cloudinary

Cloudinary là một dịch vụ quản lý và phân phối nội dung đa phương tiện trên nền tảng đám mây, giúp các nhà phát triển web và ứng dụng di động dễ dàng quản lý, tối ưu hóa và phân phối các tài sản đa phương tiện như hình ảnh, video và các tệp tin khác. Được thành lập vào năm 2012, Cloudinary cung cấp các công cụ mạnh mẽ để tự động tối ưu hóa hình ảnh và video nhằm cải thiện hiệu suất tải trang, bao gồm việc thay đổi kích thước, nén, cắt, thay đổi định dạng và điều chỉnh chất lượng. Ngoài ra, Cloudinary cho phép thực hiện các thao tác phức tạp trên hình ảnh và video như thêm watermark, tạo

thumbnail và áp dụng các hiệu ứng, tất cả thông qua URL hoặc API đơn giản. Sử dụng mạng phân phối nội dung (CDN), Cloudinary đảm bảo rằng các tệp đa phương tiện được phân phối nhanh chóng và hiệu quả đến người dùng toàn cầu, giúp giảm độ trễ và cải thiện trải nghiệm người dùng. Với các SDK và API cho nhiều ngôn ngữ lập trình và framework phổ biến, việc tích hợp Cloudinary vào các ứng dụng trở nên dễ dàng và linh hoạt. Hơn nữa, Cloudinary cung cấp các công cụ quản lý mạnh mẽ và các tính năng bảo mật như xác thực API, kiểm soát truy cập và mã hóa dữ liệu, giúp tổ chức, tìm kiếm và kiểm soát truy cập đối với các tài sản đa phương tiện một cách hiệu quả.

General System Overview



Hình 2.13 Mô hình lưu trữ Cloudinary

2.5. Giao thức truyền thông Client-Server

2.5.1. Giao thức http

HTTP (HyperText Transfer Protocol) là một giao thức truyền tải siêu văn bản được sử dụng rộng rãi trong các hệ thống thông tin siêu văn bản toàn cầu như World Wide Web. HTTP xác định cách các yêu cầu và phản hồi giữa máy khách (client) và máy chủ (server) được định dạng và truyền tải qua mạng Internet.

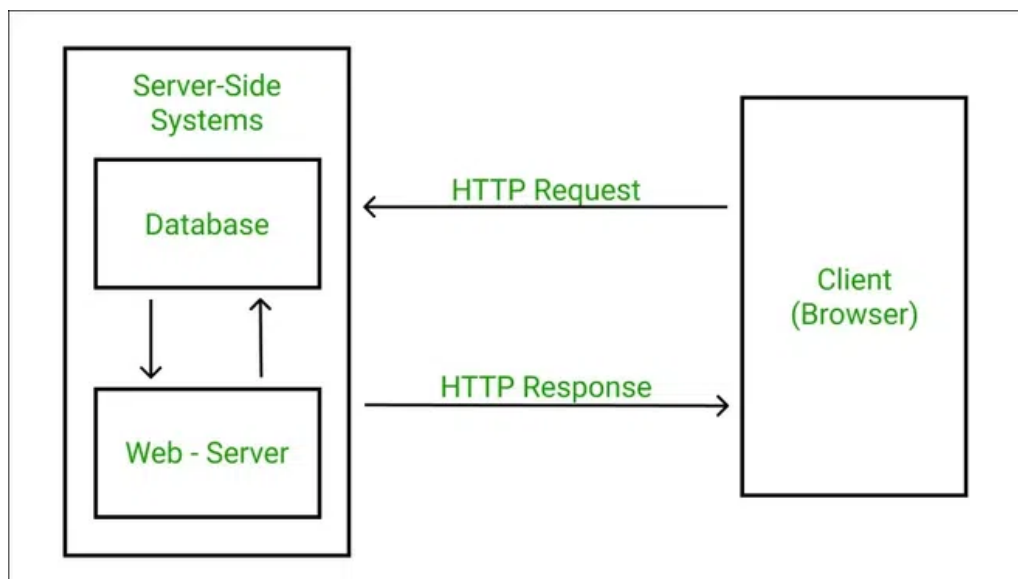
HTTP hoạt động theo mô hình yêu cầu-phản hồi (request-response). Máy khách gửi một yêu cầu HTTP đến máy chủ, và máy chủ phản hồi với một thông điệp HTTP chứa dữ liệu yêu cầu, trạng thái và các thông tin khác.

HTTP hỗ trợ nhiều phương thức (methods) khác nhau để chỉ định loại hành động yêu cầu, bao gồm:

- GET: Yêu cầu tài nguyên từ máy chủ.
- POST: Gửi dữ liệu đến máy chủ để xử lý.
- PUT: Cập nhật hoặc tạo mới tài nguyên trên máy chủ.
- DELETE: Xóa tài nguyên từ máy chủ.
- HEAD, OPTIONS, PATCH: Và một số phương thức khác phục vụ các mục đích cụ thể.

Phản hồi từ máy chủ bao gồm một mã trạng thái HTTP (HTTP status code) để cho biết kết quả của yêu cầu, chẳng hạn như:

- 200 OK: Yêu cầu thành công.
- 404 Not Found: Tài nguyên không được tìm thấy.
- 500 Internal Server Error: Lỗi máy chủ.



Hình 2.14 Mô hình http Client - Server

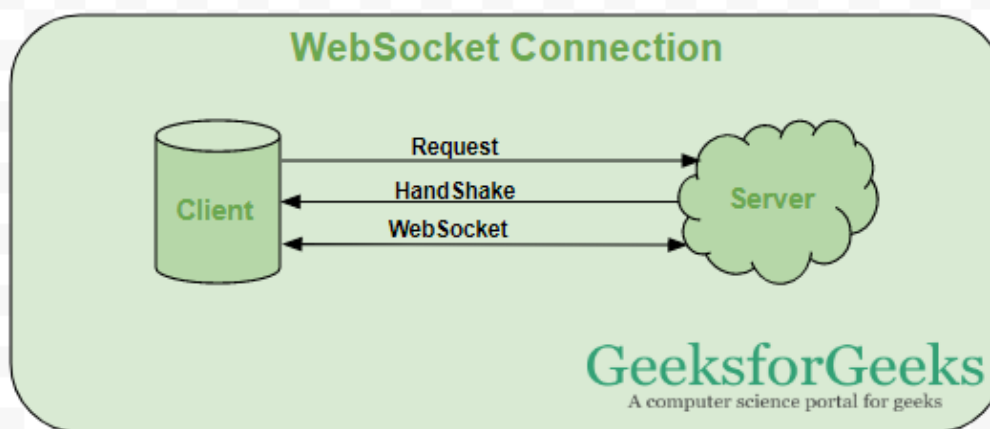
2.5.2. Giao thức websocket

WebSocket là một giao thức truyền thông hai chiều full-duplex được thiết kế để thiết lập kết nối liên tục giữa máy khách và máy chủ thông qua một kênh đơn. Khác với các giao thức HTTP thông thường, nơi mà mỗi yêu cầu từ máy khách phải được phản hồi riêng lẻ từ máy chủ, WebSocket cho phép giao tiếp liên tục và tức thời giữa các bên, làm cho nó trở thành lựa chọn lý tưởng cho các ứng dụng yêu cầu cập nhật thời gian thực như chat, thông báo, trò chơi trực tuyến và truyền phát dữ liệu.

WebSocket cho phép giao tiếp hai chiều, nghĩa là cả máy khách và máy chủ có thể gửi và nhận dữ liệu bất cứ lúc nào thông qua một kết nối duy nhất. Điều này khác với HTTP, nơi mà máy khách phải bắt đầu yêu cầu và chờ phản hồi từ máy chủ.

Sau khi kết nối WebSocket được thiết lập, nó vẫn duy trì mở cho đến khi bị đóng bởi một trong hai bên hoặc do sự cố mạng. Điều này giúp giảm thiểu độ trễ và tải mạng so với việc phải thiết lập nhiều kết nối HTTP riêng lẻ.

WebSocket hoạt động trên giao thức TCP, giúp đảm bảo độ tin cậy và khả năng truyền dữ liệu theo thứ tự.



Hình 2.15 Kết nối websocket

Với WebSocket, không cần phải gửi các tiêu đề HTTP trong mỗi thông điệp, giảm bớt lưu lượng mạng và tăng tốc độ truyền thông. Điều này rất hữu ích cho các ứng dụng yêu cầu độ trễ thấp và truyền tải dữ liệu thường xuyên.

Kết nối WebSocket bắt đầu bằng một quá trình handshake qua HTTP/HTTPS. Máy khách gửi một yêu cầu HTTP đặc biệt đến máy chủ để yêu cầu nâng cấp kết nối lên WebSocket. Nếu máy chủ đồng ý, kết nối WebSocket được thiết lập và chuyển sang giao thức WebSocket.

WebSocket có thể được sử dụng qua các kết nối bảo mật (wss://) bằng cách sử dụng SSL/TLS, đảm bảo rằng dữ liệu được truyền tải một cách an toàn và bảo mật.

2.6. Các khái niệm liên quan xử lý ảnh

2.6.1. Haar Cascade Classifier

Haar Cascade Classifier là một phương pháp học máy dựa trên các đặc trưng Haar để phát hiện đối tượng trong ảnh, được sử dụng rộng rãi trong thị giác máy tính, đặc biệt là phát hiện khuôn mặt. Đặc trưng Haar là những mẫu hình chữ nhật màu đen và trắng đơn giản giúp mô tả cấu trúc và hình dạng trong ảnh, bao gồm các đặc trưng như cạnh, đường thẳng và vùng có sự thay đổi phức tạp. Phương pháp này sử dụng một tập hợp lớn ảnh dương (chứa đối tượng) và ảnh âm (không chứa đối tượng) để huấn luyện mô hình bằng thuật toán AdaBoost, chọn ra các đặc trưng Haar tốt nhất. Khi sử dụng, ảnh đầu vào được chuyển đổi sang ảnh xám và quét toàn bộ bằng các cửa sổ có kích thước và tỉ lệ khác nhau để phát hiện đối tượng. Các giai đoạn phân loại sau đó xác định vùng chứa đối tượng. Haar Cascade Classifier không chỉ được ứng dụng trong phát hiện khuôn mặt và các bộ phận trên khuôn mặt như mắt, mũi, miệng mà còn trong nhiều bài toán khác như nhận diện biển số xe và nhận dạng văn bản. Công cụ này, được hỗ trợ mạnh mẽ bởi thư viện OpenCV, giúp việc phát hiện đối tượng trở nên dễ dàng và nhanh chóng, giải quyết hiệu quả nhiều bài toán trong lĩnh vực thị giác máy tính.

2.6.2. OpenCV

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở nổi tiếng, được phát triển nhằm mục đích cung cấp các công cụ và thuật toán để xử lý ảnh và thị giác máy tính. OpenCV hỗ trợ nhiều ngôn ngữ lập trình như C++, Python, Java, và MATLAB, và có khả năng chạy trên nhiều nền tảng khác nhau, bao gồm Windows, Linux, macOS, iOS, và Android. Các tính năng của OpenCV bao gồm phát hiện đối tượng, nhận diện khuôn mặt, theo dõi chuyển động, phân loại hình ảnh, và nhiều ứng dụng khác trong lĩnh vực thị giác máy tính. Thư viện này cung cấp các hàm tiện ích để xử lý các tác vụ như chuyển đổi không gian màu, thay đổi kích thước ảnh, áp dụng các bộ lọc ảnh, và thực hiện các phép toán hình học trên ảnh. Nhờ vào sự linh hoạt và hiệu quả của nó, OpenCV đã trở thành một công cụ quan trọng trong việc phát triển các ứng dụng xử lý ảnh và thị giác máy tính, phục vụ cho các nghiên cứu và phát triển trong nhiều lĩnh vực khác nhau như robot, y tế, an ninh và tự động hóa.

2.6.3. NumPy

NumPy là một thư viện mã nguồn mở cho ngôn ngữ lập trình Python, được thiết kế để hỗ trợ tính toán khoa học và số học. Thư viện này cung cấp một đối tượng mảng đa chiều mạnh mẽ, gọi là ndarray, cùng với các hàm toán học phong phú để thao tác và thực hiện các phép toán trên các mảng này một cách hiệu quả. NumPy được xây dựng với hiệu suất cao, cho phép xử lý và tính toán trên các dữ liệu lớn và phức tạp một cách nhanh chóng. Ngoài ra, NumPy còn hỗ trợ tích hợp với nhiều thư viện khoa học khác của Python như SciPy, Matplotlib và Pandas, tạo nên một hệ sinh thái mạnh mẽ cho khoa học dữ liệu, học máy và xử lý tín hiệu. Nhờ vào tính năng mạnh mẽ và sự linh hoạt của mình, NumPy đã trở thành công cụ quan trọng và phổ biến trong cộng đồng nghiên cứu và phát triển khoa học dữ liệu.

Chương 3: PHÂN TÍCH, THIẾT KẾ DỮ LIỆU

3.1. Phân tích đặc tả yêu cầu

3.1.1. Chức năng

User:

- Đăng nhập và đăng ký tài khoản: cho phép người dùng đăng ký một tài khoản mới và đăng nhập vào hệ thống. Người dùng là thành viên của hệ thống có những ưu điểm so với người dùng thông thường sau:
 - Ghi nhớ thông tin cá nhân khi mua hàng
 - Có thể xem, lưu sản phẩm yêu thích
 - Có thể xem lịch sử mua hàng
 - Có thể đánh giá một sản phẩm
 - Có thể lưu những sản phẩm dự định mua trong giỏ hàng
- Xem sản phẩm: hiển thị danh sách sản phẩm cho người dùng để duyệt và lựa chọn. Người dùng có thể tìm kiếm, lọc các tiêu chí của sản phẩm để quyết định mua hàng. Bất kể người dùng đã đăng nhập hay chưa đều có quyền xem được.
- Thử mắt kính trực tuyến: có 2 hình thức thử kính là tải hình ảnh cá nhân và thử kính trực tuyến bằng webcam. Cả 2 hình thức đều cho ra kết quả khuôn mặt của khách hàng mang kính mà đã được chọn trước đó.
- Thêm vào giỏ hàng: cho phép người dùng thêm sản phẩm vào giỏ hàng để mua sau.
- Mua hàng: người dùng phải trải qua các bước:
 - Xác nhận thông tin trước khi mua hàng (tên, địa chỉ, số điện thoại, ...)
 - Chọn phương thức thanh toán (chuyển khoản bằng mã QR hoặc thanh toán bằng thẻ ngân hàng)
 - Nếu thực hiện thanh toán thành công, người dùng bấm xác nhận và quay về trang chủ. Nếu thất bại, hủy bỏ toàn bộ quá trình giao dịch. Cả hai trường hợp đều lưu vào lịch sử mua hàng của người dùng.
- Quản lý tài khoản: cho phép người dùng cập nhật thông tin cá nhân, xem lịch sử đơn hàng, xem lịch sử đánh giá, đổi mật khẩu và xem các sản phẩm yêu thích.
- Đánh giá sản phẩm: người dùng có thể đánh giá và viết nhận xét về sản phẩm sau khi mua hàng, mọi bài đánh giá của người dùng được lưu lại và có thể xem lại các bài đánh giá đó. Mỗi người dùng chỉ có thể đánh giá một lần cho một sản phẩm, nếu người dùng muốn đánh giá sản phẩm lần hai cho đợt mua thứ hai của

sản phẩm đó, người dùng có thể cập nhật lại bài đánh giá của mình.

- Thêm sản phẩm vào yêu thích: lưu trữ các sản phẩm yêu thích của người dùng, có thể thêm, xoá sản phẩm bất kì trong danh sách

Admin:

- Quản lý doanh thu: cho phép admin xem thống kê doanh thu theo ngày, tuần, tháng. Có thể tra cứu đơn hàng đã được thanh toán.
- Quản lý người dùng: cho phép admin thao tác chỉnh sửa tài khoản người dùng như cập nhật thông tin, huỷ bỏ tài khoản.

3.1.2. Phi chức năng

Bảo mật: hệ thống cần đảm bảo an toàn thông tin cá nhân của người dùng và giao dịch thanh toán.

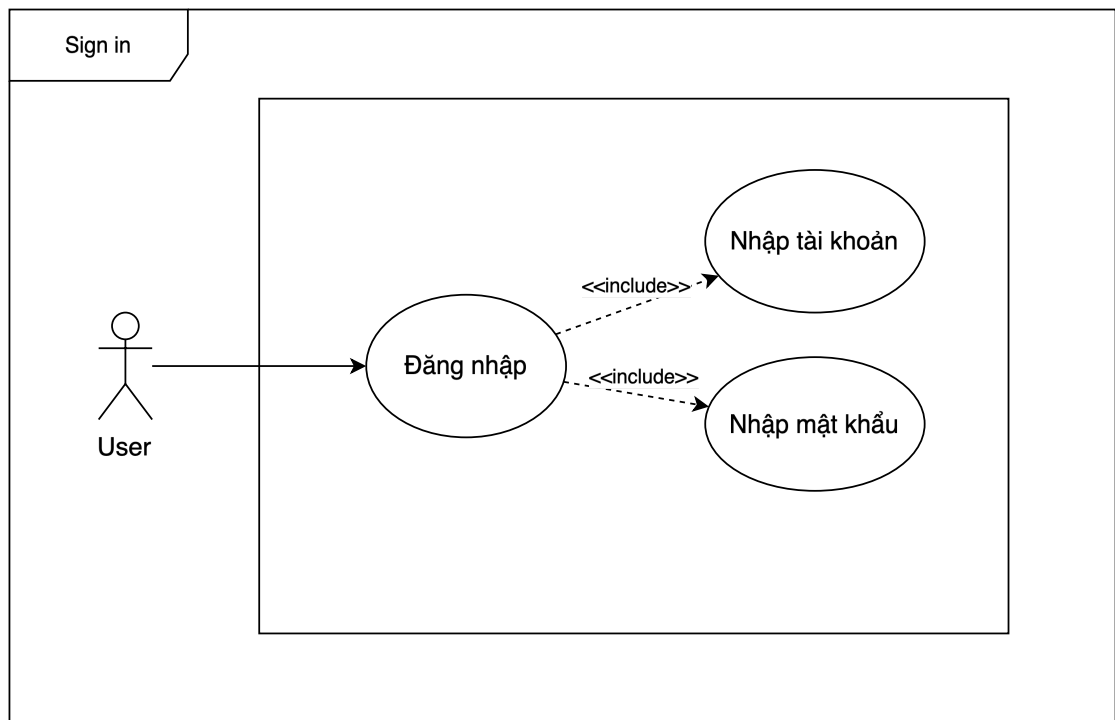
Hiệu suất: giao diện người dùng cần phải responsive và tải trang nhanh chóng để cung cấp trải nghiệm người dùng tốt nhất.

Hỗ trợ đa ngôn ngữ và đa quốc gia: Hệ thống cần hỗ trợ nhiều ngôn ngữ và đa quốc gia để phục vụ người dùng ở nhiều khu vực khác nhau trên thế giới

3.2. Biểu đồ Usecase



Hình 3.1. Biểu đồ usecase tổng quát hệ thống

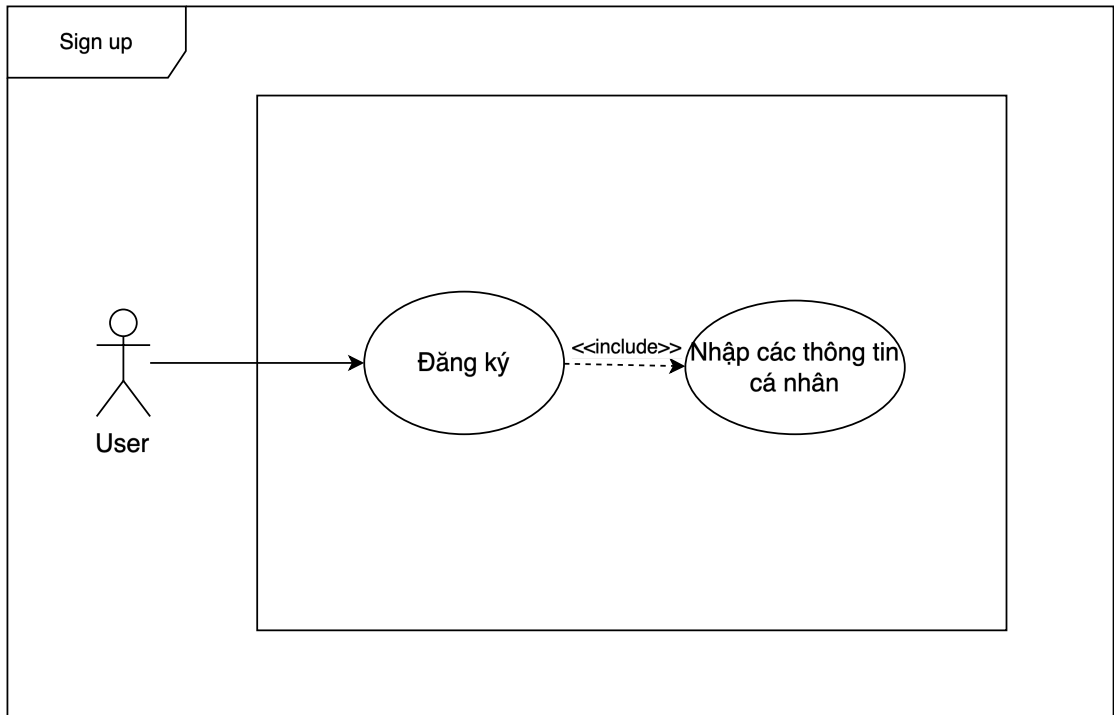


Hình 3.2 Biểu đồ Usecase đăng nhập

Bảng 3.1 Đặc tả Usecase đăng nhập

Tên Usecase	Đăng nhập
Mô tả	Người dùng nhập thông tin tài khoản và mật khẩu để truy cập vào hệ thống.
Tác nhân	User
Tiền điều kiện	Người dùng đã có tài khoản trong hệ thống.
Các bước thực hiện	1. Người dùng chọn vào màn hình đăng nhập 2. Hệ thống yêu cầu người dùng nhập tài khoản và mật khẩu 3. Người dùng nhập tài khoản và mật khẩu 4. Hệ thống kiểm tra thông tin tài khoản và mật khẩu 5. Nếu thông tin đúng, người dùng được truy cập vào hệ thống. Nếu sai, hiển thị thông báo lỗi

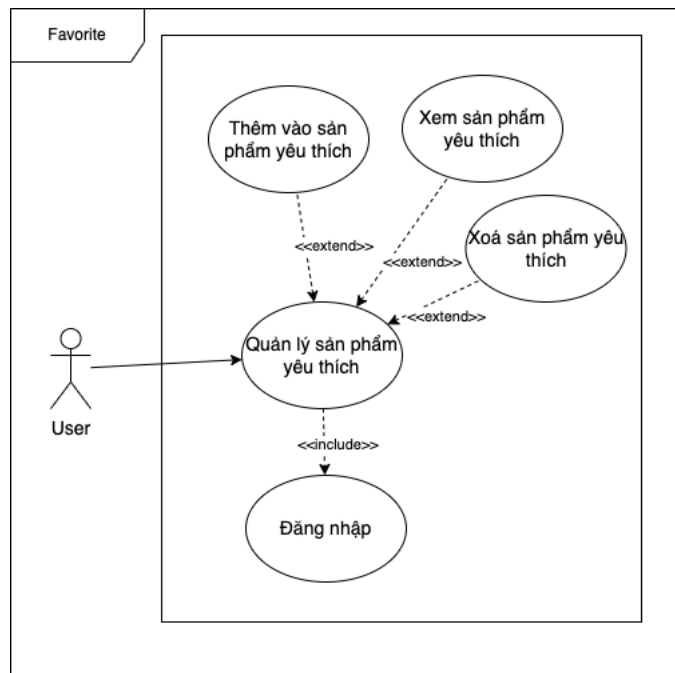
Kết quả	Người dùng được xác thực và truy cập vào hệ thống thành công.
Trường hợp lỗi	Lỗi do sai thông tin đăng nhập



Hình 3.3 Biểu đồ Usecase đăng ký

Bảng 3.2 Đặc tả Usecase đăng ký

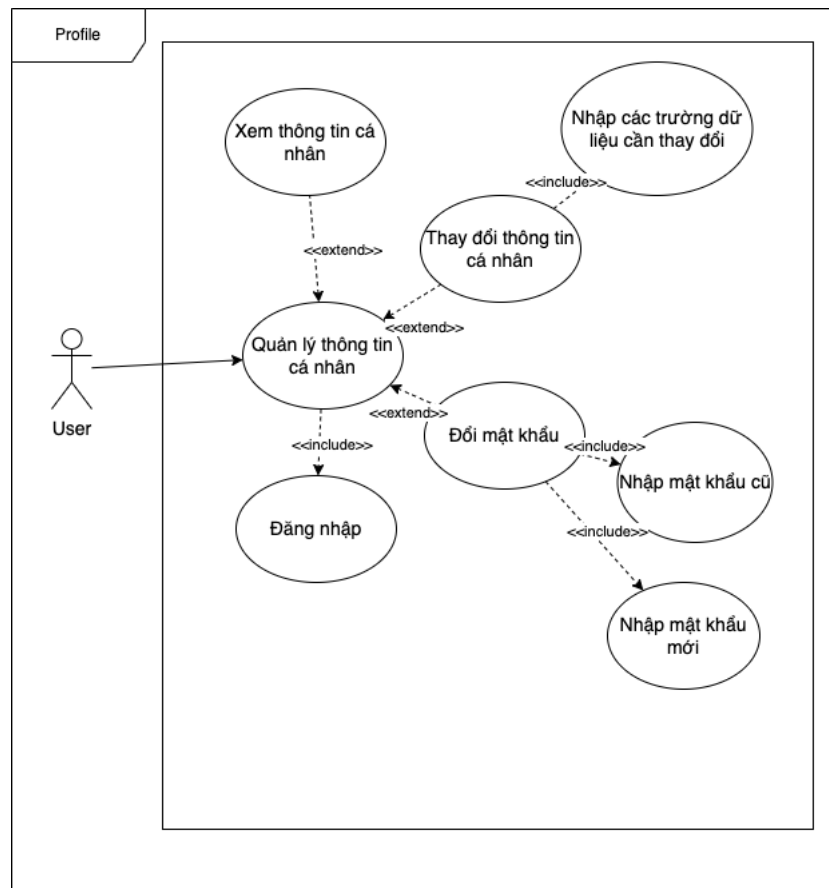
Tên Usecase	Đăng ký
Tác nhân	User
Mô tả	Người dùng nhập các thông tin cần thiết để tạo mới một tài khoản trong hệ thống
Tiền điều kiện	Không có
Các bước thực hiện	1. Người dùng vào màn hình đăng ký 2. Hệ thống yêu cầu người dùng nhập các thông tin cần thiết 3. Người dùng nhập thông tin cần thiết (tên, email, mật khẩu, v.v.) 4. Hệ thống kiểm tra tính hợp lệ của thông tin. Nếu thông tin hợp lệ, tài khoản mới được tạo thành công. Nếu không, hiển thị thông báo lỗi.
Kết quả	Người dùng tạo thành công một tài khoản mới trong hệ thống.
Trường hợp lỗi	Điền sai thông tin cá nhân



Hình 3.4 Biểu đồ Usecase sản phẩm yêu thích

Bảng 3.3 Đặc tả Usecase sản phẩm yêu thích

Tên Usecase	Quản lý sản phẩm yêu thích
Tác nhân	User
Mô tả	Người dùng quản lý danh sách các sản phẩm yêu thích của mình, bao gồm việc thêm, xem và xóa sản phẩm yêu thích.
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống.
Các bước thực hiện	1. Người dùng vào mục quản lý sản phẩm yêu thích 2. Sau đăng nhập, người dùng có thể thực hiện các chức năng: 3. Thêm vào sản phẩm yêu thích 4. Xem sản phẩm yêu thích 5. Xóa sản phẩm yêu thích
Kết quả	Hệ thống lưu trạng thái sản phẩm khi thực hiện thành công
Trường hợp lỗi	Người dùng chưa đăng nhập

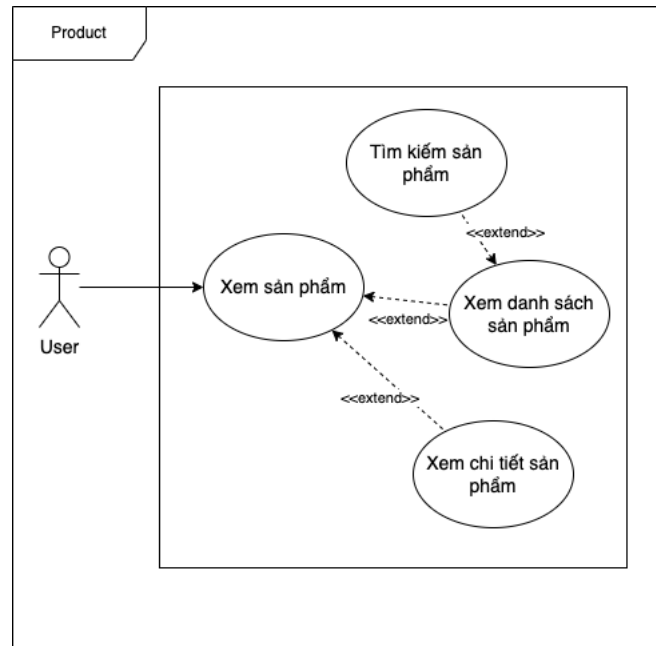


Hình 3.5 Biểu đồ Usecase thông tin cá nhân

Bảng 3.4 Đặc tả Usecase thông tin cá nhân

Tên Usecase	Quản lý thông tin cá nhân
Tác nhân	User
Mô tả	Người dùng có thể quản lý thông tin cá nhân của mình, bao gồm xem thông tin, thay đổi thông tin cá nhân và đổi mật khẩu.
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống.
Các bước thực hiện	1. Người dùng vào mục thông tin cá nhân 2. Người dùng có thể thực hiện các chức năng: 3. Xem thông tin cá nhân 4. Thay đổi thông tin cá nhân

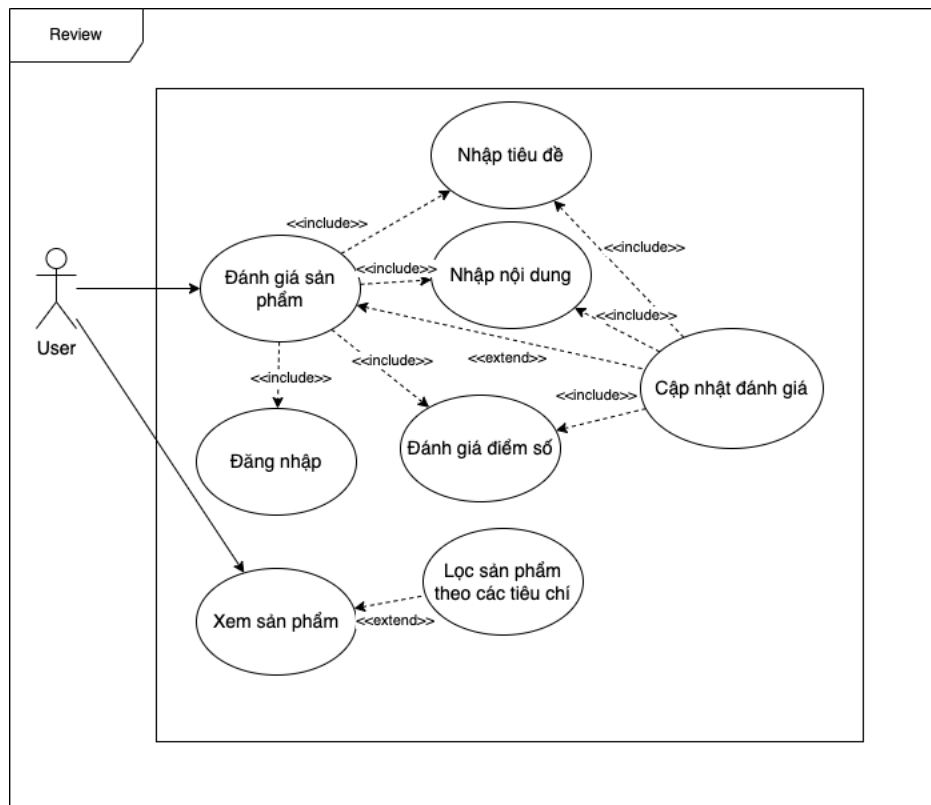
	5. Đổi mật khẩu
Kết quả	Người dùng chỉnh sửa thông tin cá nhân, thay đổi mật khẩu thành công
Trường hợp lỗi	1. Người dùng chưa đăng nhập 2. Người dùng nhập sai định dạng thông tin cá nhân



Hình 3.6 Biểu đồ Usecase sản phẩm

Bảng 3.5 Đặc tả Usecase sản phẩm

Tên Usecase	Xem sản phẩm
Tác nhân	User
Mô tả	Người dùng có thể xem thông tin các sản phẩm, bao gồm danh sách sản phẩm và chi tiết sản phẩm.
Tiền điều kiện	Không có
Các bước thực hiện	1. Người dùng vào trang chủ website 2. Người dùng có thể chọn một trong các chức năng: 3. Tìm kiếm sản phẩm 4. Xem danh sách sản phẩm theo bộ lọc 5. Xem chi tiết sản phẩm
Kết quả	Người dùng có thể xem danh sách sản phẩm và chi tiết sản phẩm.
Trường hợp lỗi	Truy suất sản phẩm không thành công

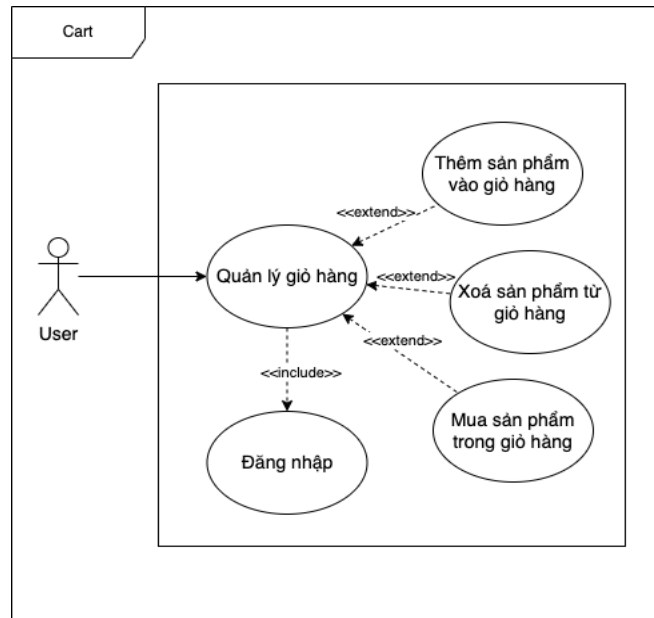


Hình 3.7 Biểu đồ Usecase đánh giá sản phẩm

Bảng 3.6 Đặc tả Usecase đánh giá sản phẩm

Tên Usecase	Quản lý đánh giá sản phẩm
Tác nhân	User
Mô tả	Người dùng có thể quản lý đánh giá sản phẩm thông qua xem, thêm, sửa, xoá một bài đánh giá
Tiền điều kiện	Không có
Các bước thực hiện	1. Người dùng chọn sản phẩm muốn đánh giá 2. Người dùng nhập tiêu đề cho đánh giá 3. Người dùng nhập nội dung cho đánh giá 4. Người dùng đánh giá điểm số cho sản phẩm 5. Người dùng gửi đánh giá
Kết quả	Đánh giá của người dùng được lưu trữ và cập nhật cho sản phẩm.

Trường hợp lỗi	1. Người dùng chưa đăng nhập 2. Người dùng không nhập đủ thông tin (tiêu đề, nội dung, đánh giá)
----------------	---

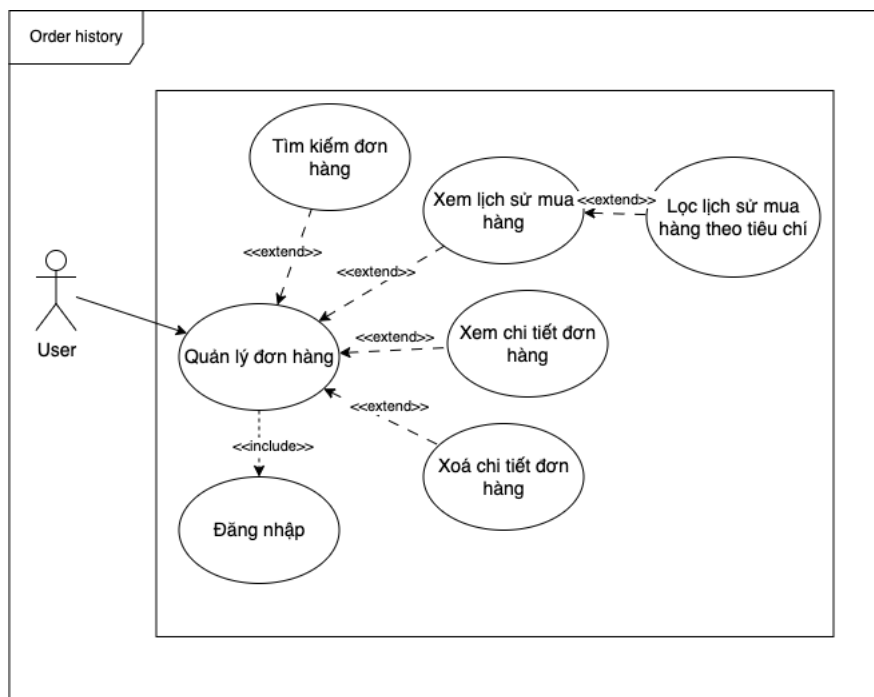


Hình 3.8 Biểu đồ Usecase quản lý giỏ hàng

Bảng 3.7 Đặc tả Usecase quản lý giỏ hàng

Tên Usecase	Quản lý giỏ hàng
Tác nhân	User
Mô tả	Cho phép người dùng quản lý giỏ hàng, bao gồm việc thêm sản phẩm vào giỏ hàng, xóa sản phẩm khỏi giỏ hàng, và mua sản phẩm trong giỏ hàng.
Tiền điều kiện	Người dùng phải đăng nhập vào hệ thống.
Các bước thực hiện	1. Người dùng chọn quản lý giỏ hàng. 2. Hệ thống hiển thị các sản phẩm hiện có trong giỏ hàng của người dùng. 3. Người dùng có thể chọn một trong các hành động sau: 4. Thêm sản phẩm vào giỏ hàng. 5. Xóa sản phẩm khỏi giỏ hàng.

	6. Mua sản phẩm trong giỏ hàng. 7. Hệ thống thực hiện hành động tương ứng và cập nhật giỏ hàng.
Kết quả	Giỏ hàng của người dùng được cập nhật dựa trên hành động đã chọn.
Trường hợp lỗi	1. Người dùng chưa đăng nhập 2. Sản phẩm thêm vào giỏ hàng không còn tồn kho

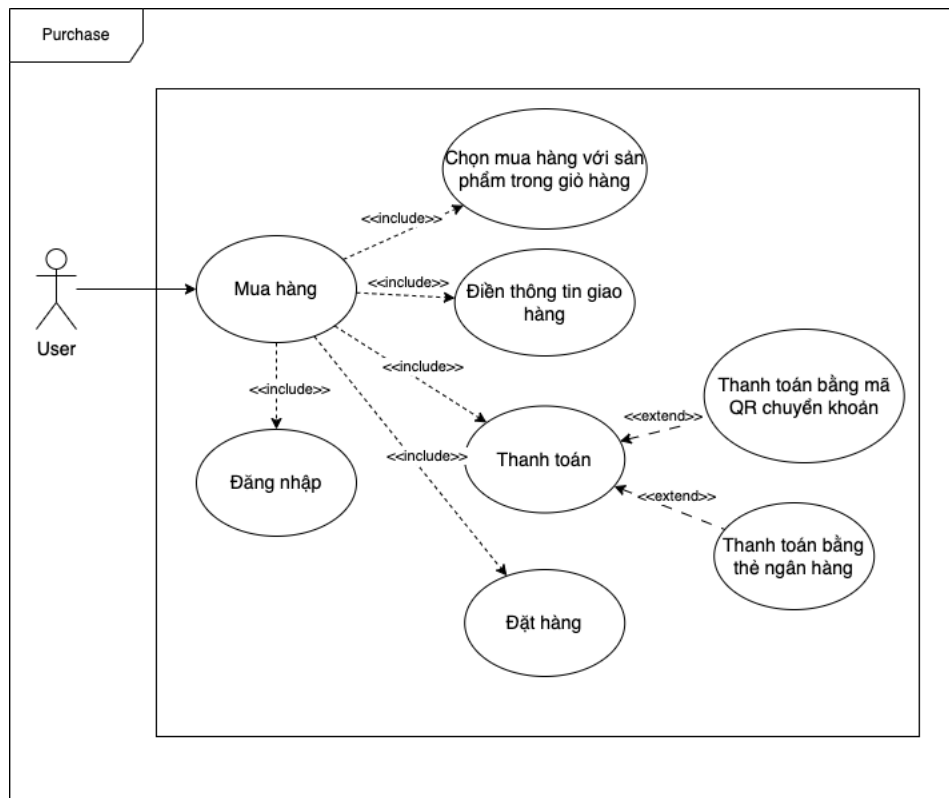


Hình 3.9 Biểu đồ Usecase quản lý đơn hàng

Bảng 3.8 Đặc tả Usecase quản lý đơn hàng

Tên Usecase	Quản lý đơn hàng
Tác nhân	User
Mô tả	Người dùng có thể quản lý giỏ hàng của họ, bao gồm tìm kiếm đơn hàng, xem lịch sử mua hàng, lọc lịch sử mua hàng theo tiêu chí, xem chi tiết đơn hàng và xóa chi tiết đơn hàng.

Tiền điều kiện	Người dùng phải đăng nhập vào hệ thống.
Các bước thực hiện	1. Người dùng chọn quản lý đơn. 2. Hệ thống hiển thị các đơn hàng hiện có của người dùng 3. Người dùng có thể chọn một trong các hành động sau: 4. Tìm kiếm đơn hàng 5. Xem lịch sử mua hàng 6. Lọc lịch sử mua hàng theo tiêu chí 7. Xem chi tiết đơn hàng 8. Xoá chi tiết đơn hàng 9. Hệ thống thực hiện hành động tương ứng và cập nhật đơn hàng
Kết quả	Đơn hàng của người dùng được cập nhật dựa trên hành động đã chọn.
Trường hợp lỗi	1. Người dùng chưa đăng nhập 2. Đơn hàng không tồn tại

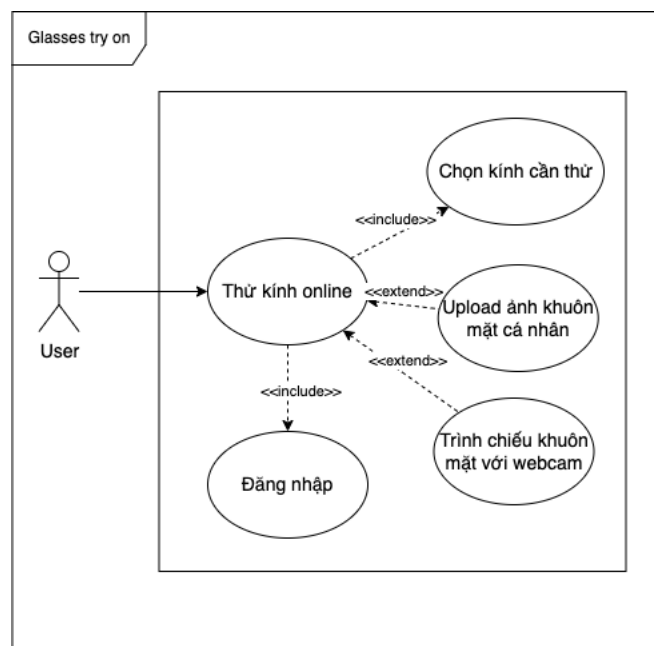


Hình 3.10 Biểu đồ Usecase mua hàng

Bảng 3.9 Đặc tả Usecase mua hàng

Tên Usecase	Quản lý đơn hàng
Tác nhân	User
Mô tả	Người dùng mua hàng từ giỏ hàng của họ. Quá trình này bao gồm chọn mua hàng với sản phẩm trong giỏ hàng, điền thông tin giao hàng, thanh toán, và đặt hàng.
Tiền điều kiện	Người dùng phải đăng nhập vào hệ thống.
Các bước thực hiện	1. Chọn mua hàng với sản phẩm trong giỏ hàng. 2. Điền thông tin giao hàng. 3. Chọn phương thức thanh toán và thanh toán. 4. Đặt hàng.

Kết quả	Đơn hàng được đặt thành công và người dùng nhận được thông báo xác nhận.
Trường hợp lỗi	1. Người dùng chưa đăng nhập 2. Thông tin giao hàng không hợp lệ 3. Hệ thống bảo trì phương thức thanh toán đó

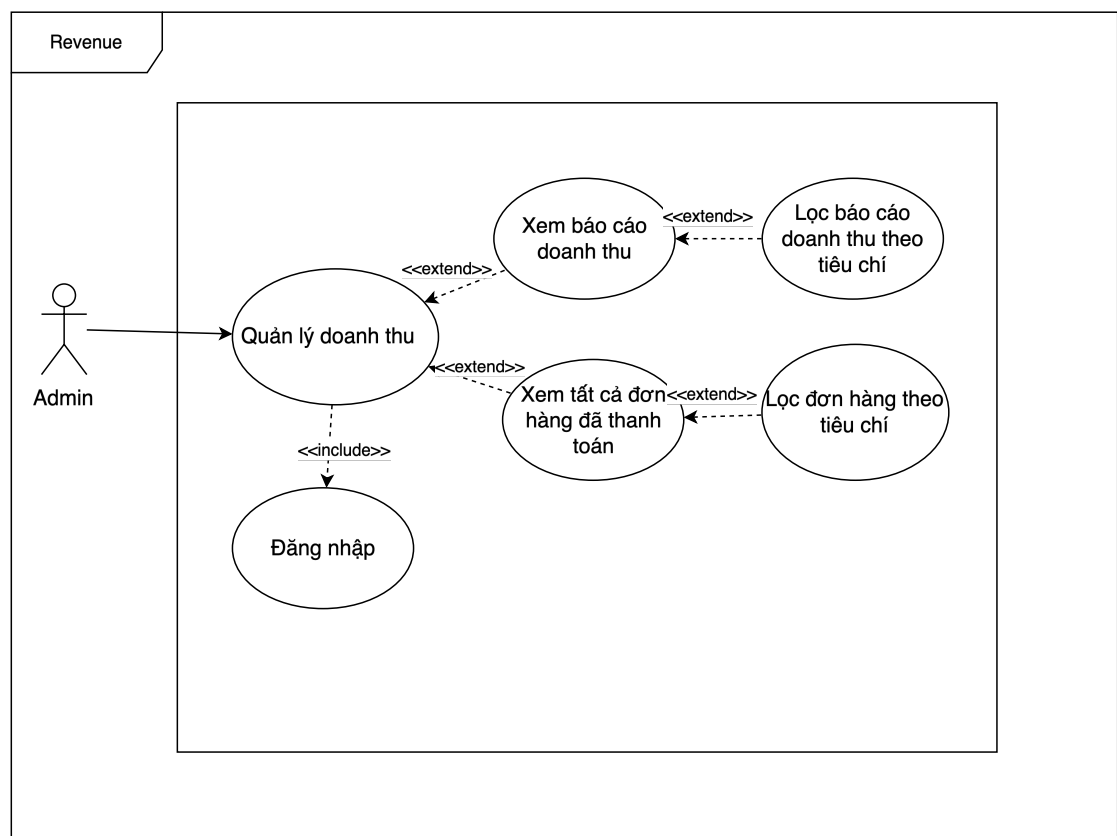


Hình 3.11 Biểu đồ Usecase thử kính online

Bảng 3.10 Đặc tả Usecase thử kính online

Tên Usecase	Thử kính online
Tác nhân	User
Mô tả	Người dùng thử kính trực tuyến thông qua việc chọn kính, upload ảnh khuôn mặt cá nhân, và trình chiếu khuôn mặt với webcam.
Tiền điều kiện	Người dùng phải đăng nhập vào hệ thống.
Các bước thực hiện	1. Chọn kính cần thử 2. Upload ảnh khuôn mặt cá nhân hoặc trình chiếu

	khuôn mặt với webcam 3. Xác nhận ảnh và thực hiện thao tác ghép ảnh
Kết quả	Hệ thống hiển thị kính trên khuôn mặt của người dùng.
Trường hợp lỗi	1. Người dùng chưa đăng nhập 2. Ảnh khuôn mặt không hợp lệ 3. Webcam không hoạt động 4. Kính chưa được quyền thử online

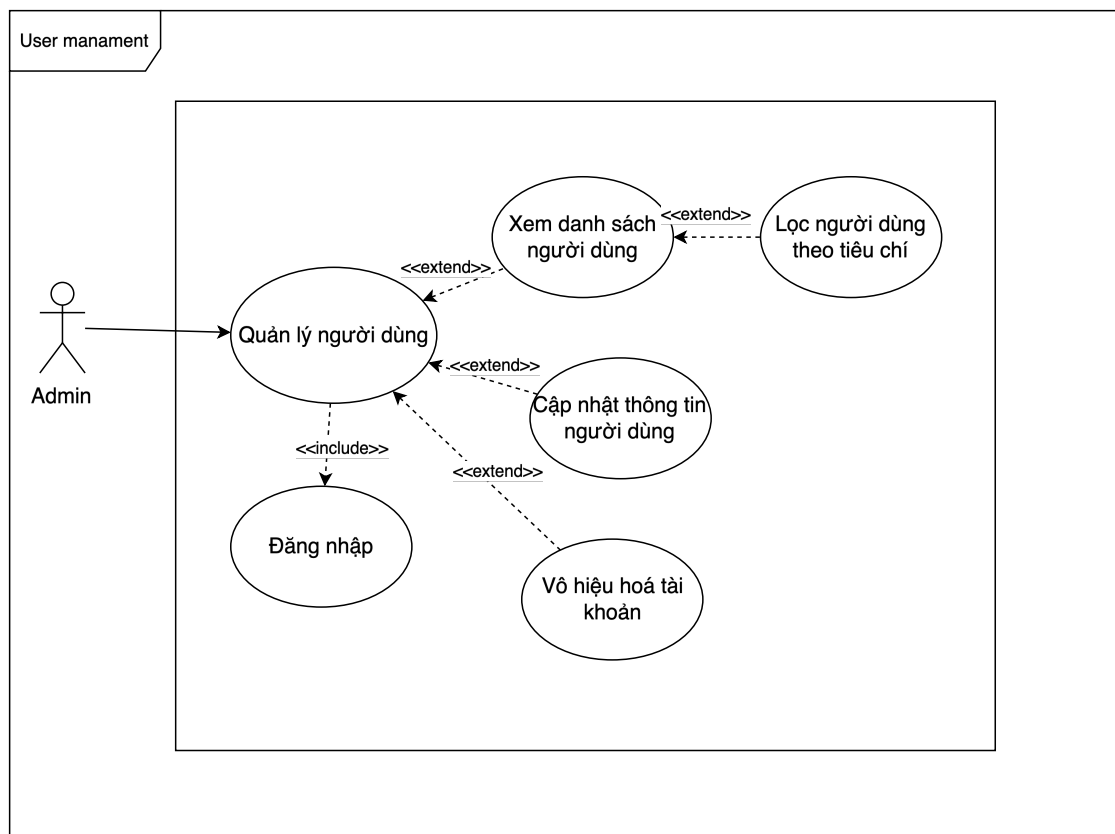


Hình 3.12 Biểu đồ Usecase doanh thu

Bảng 3.11 Đặc tả Usecase doanh thu

Tên Usecase	Quản lý doanh thu
Tác nhân	Admin
Mô tả	Admin quản lý doanh thu qua việc xem thống kê

	doanh thu và xem tất cả đơn hàng đã được thanh toán
Tiền điều kiện	Người dùng phải đăng nhập vào hệ thống.
Các bước thực hiện	- Admin vào trang chủ của website - Admin xem thống kê doanh thu và đơn hàng đã được thanh toán
Kết quả	Hệ thống hiển thị kết quả doanh thu
Trường hợp lỗi	Admin chưa đăng nhập



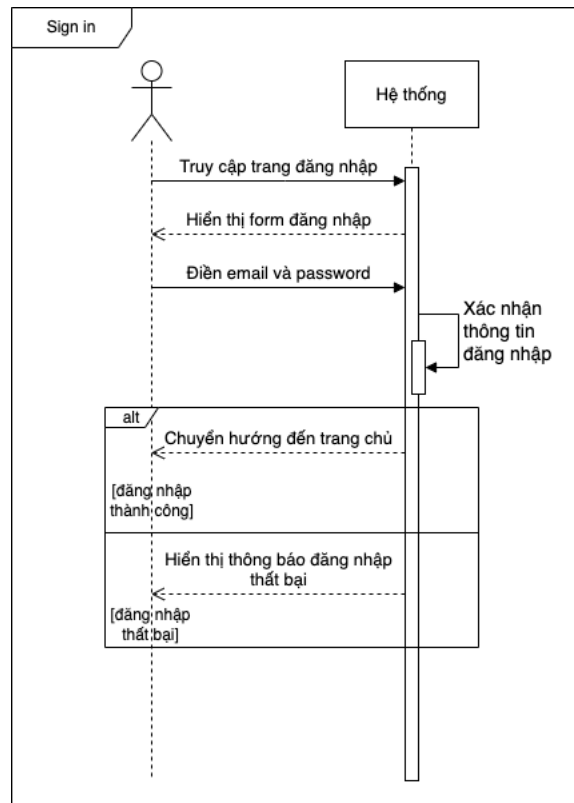
Hình 3.13 Biểu đồ Usecase quản lý người dùng

Bảng 3.12 Đặc tả Usecase quản lý người dùng

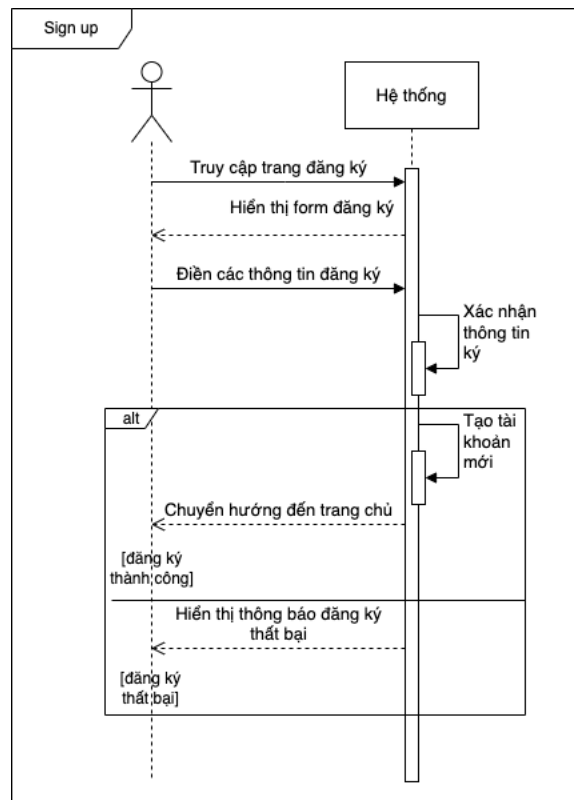
Tên Usecase	Quản lý người dùng
Tác nhân	Admin
Mô tả	Admin quản lý người dùng qua việc xem thông tin

	người dùng, và thao tác với tài khoản người dùng
Tiền điều kiện	Admin phải đăng nhập vào hệ thống.
Các bước thực hiện	1. Admin vào mục quản lý người dùng 2. Admin xem toàn bộ thông tin người dùng và thực hiện các thao tác với tài khoản người dùng mong muốn
Kết quả	Hệ thống phản hồi đúng kết quả Admin cần thao tác
Trường hợp lỗi	Admin chưa đăng nhập

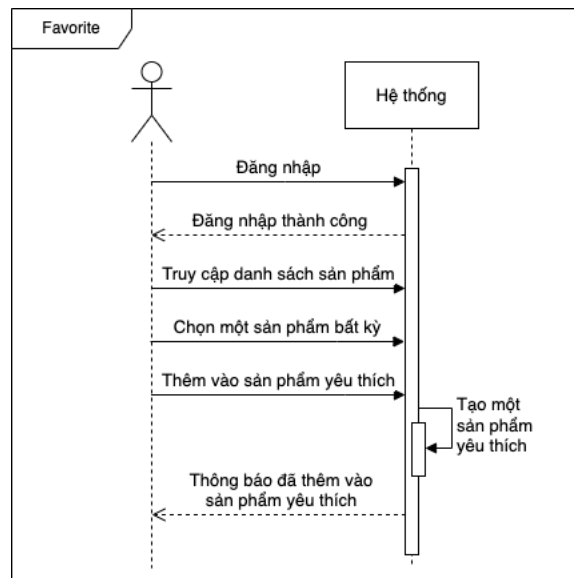
3.3. Biểu đồ trình tự



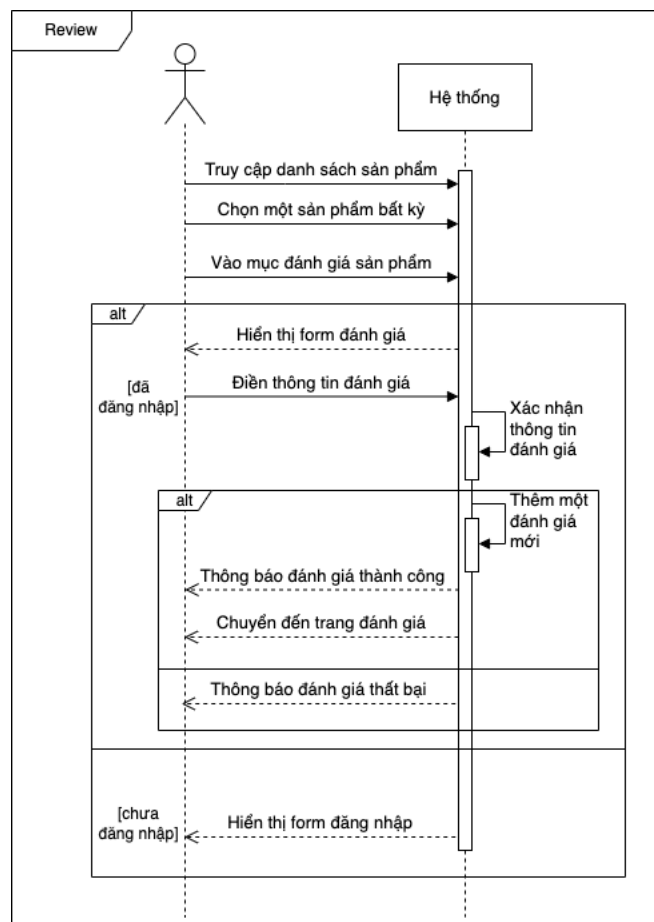
Hình 3.14 Biểu đồ trình tự phần Sign in



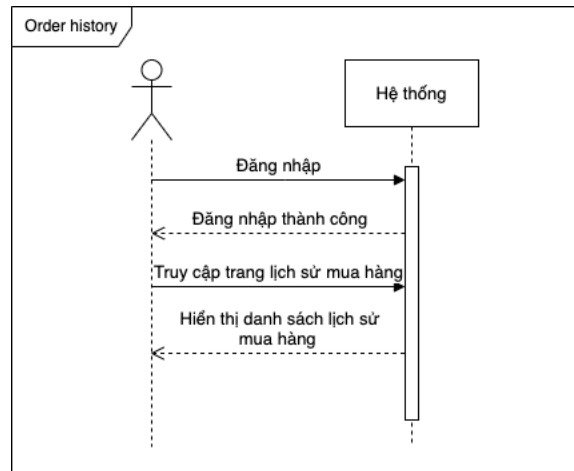
Hình 3.15 Biểu đồ trình tự phần Sign up



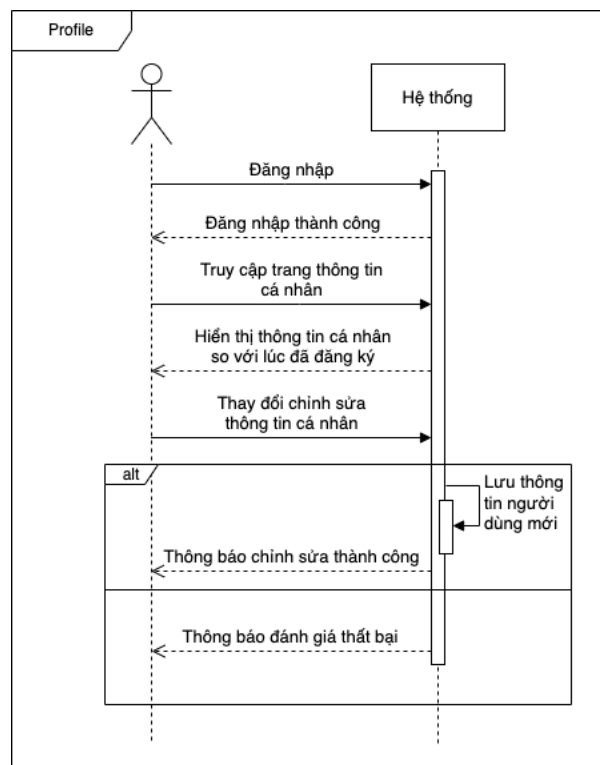
Hình 3.16 Biểu đồ trình tự phần Favorite



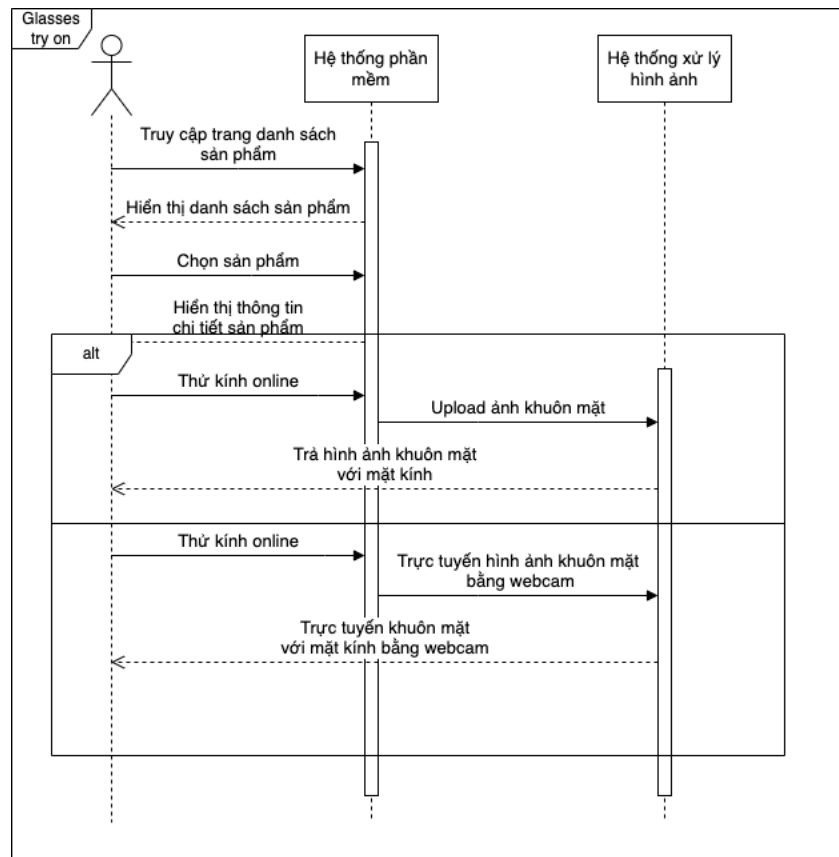
Hình 3.17 Biểu đồ trình tự phần Review



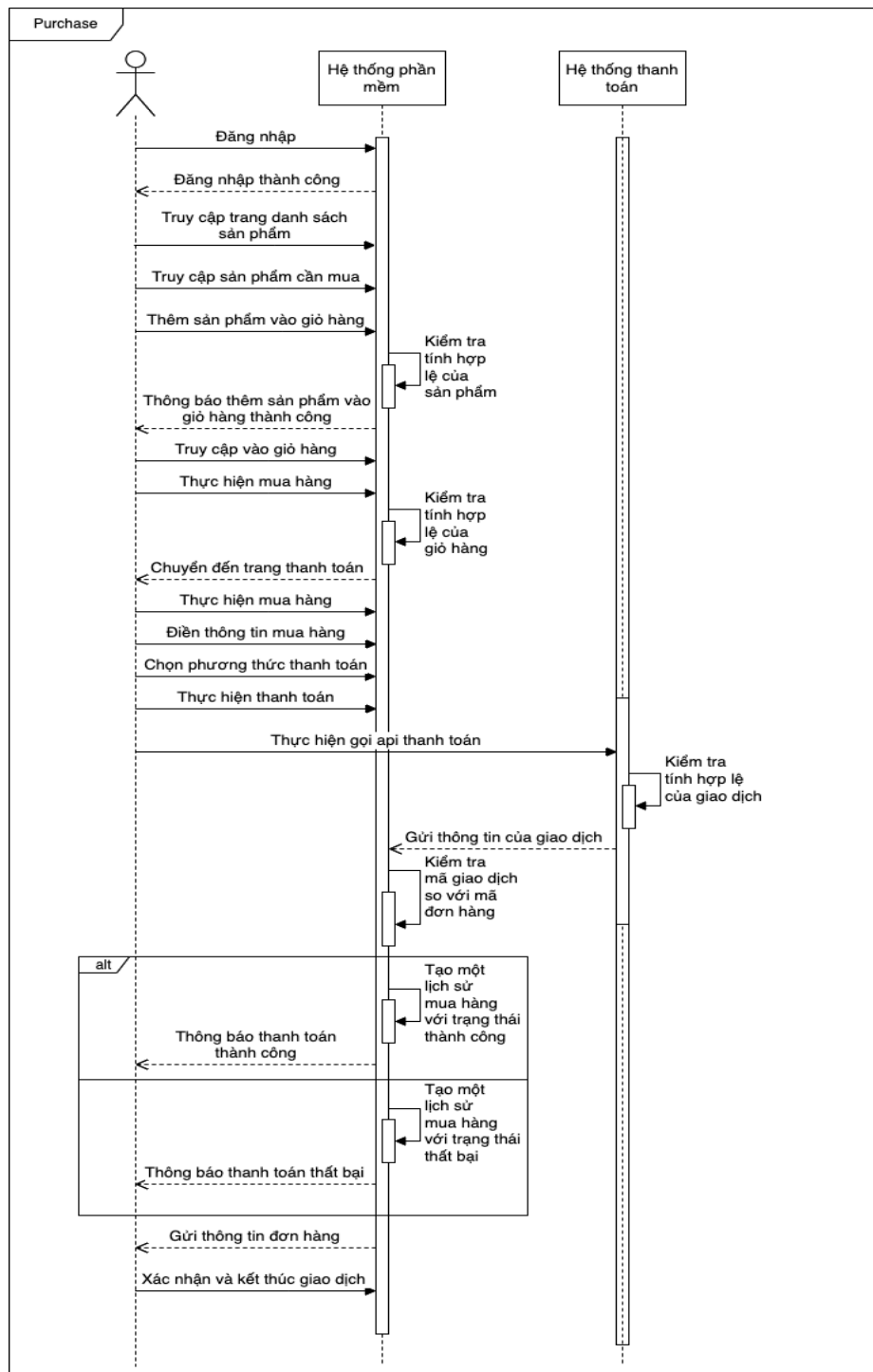
Hình 3.18 Biểu đồ trình tự phần Order



Hình 3.19 Biểu đồ trình tự phần Profile

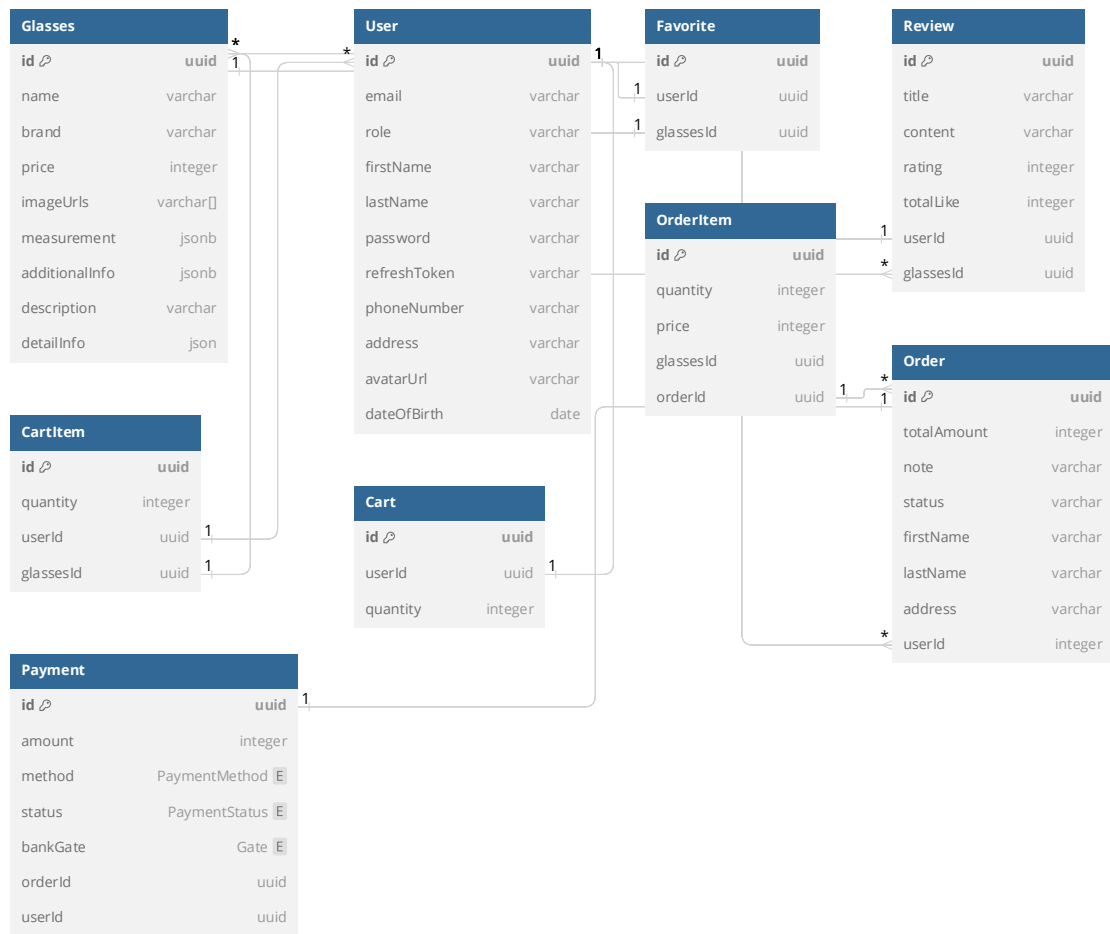


Hình 3.20 Biểu đồ trình tự phần Glasses try on



Hình 3.21 Biểu đồ trình tự phần Purchase

3.4. Biểu đồ lớp



Hình 3.22 Biểu đồ lớp hệ thống

Chi tiết các bảng

Loại database: Postgresql

- Bảng User
- Mô tả: Chứa thông tin về người dùng
- Thuộc tính:

Bảng 3.13 Thuộc tính bảng User

Tên trường	Kiểu dữ liệu	Nullable	Default	Mô tả
id	uuid			Khoá chính
email	varchar			Email người dùng
role	varchar		USER	Vai trò của người dùng
firstName	varchar			Tên riêng
lastName	varchar			Họ
password	varchar			Mật khẩu được mã hoá
refreshToken	varchar	True		Token để làm mới phiên đăng nhập
phoneNumber	varchar			Số điện thoại
address	varchar			Địa chỉ
avatarUrl	varchar	True		Hình đại diện dưới dạng url
dateOfBirth	date		now()	Ngày sinh

- Bảng Glasses
- Mô tả: Chứa thông tin về các loại kính
- Thuộc tính:

Bảng 3.14 Thuộc tính bảng Glasses

Tên trường	Kiểu dữ liệu	Nullable	Default	Mô tả
id	uuid			Khoá chính
name	varchar			Tên kính
brand	varchar			Tên thương hiệu
price	integer			Giá sản phẩm
imageUrls	varchar[]		[]	Danh sách URL hình ảnh của kính
quantity	integer		1	Số lượng hàng trong kho
measurement	jsonb		{}	Thông số đo lường (chiều dài gọng kính, khoảng cách giữa 2 tròng, ...)
additionalInfo	jsonb		{}	Thông tin bổ sung
description	varchar	True		Thông tin mô tả (về xuất xứ, về thương hiệu, ...)
detailInfo	jsonb		{}	Thông tin chi tiết (nặng, thông số kỹ thuật, ...)

- Bảng Favorite
- Mô tả: Chứa thông tin về kính yêu thích của người dùng
- Thuộc tính:

Bảng 3.15 Thuộc tính bảng Favorite

Tên trường	Kiểu dữ liệu	Nullable	Default	Mô tả
id	uuid			Khoá chính
userId	uuid			Khoá ngoại đến bảng User
glassesId	uuid			Khoá ngoại đến bảng Glasses

- Bảng Review
- Mô tả: Chứa thông tin đánh giá của người dùng về sản phẩm
- Thuộc tính:

Bảng 3.16 Thuộc tính bảng Review

Tên trường	Kiểu dữ liệu	Nullable	Default	Mô tả
id	uuid			Khoá chính
title	varchar			Tiêu đề đánh giá
content	varchar			Nội dung đánh giá
rating	integer		1	Đánh giá (số sao)
totalLike	integer		0	Tổng số lượt thích
userId	uuid			Khoá ngoại đến bảng User
glassesId	uuid			Khoá ngoại đến bảng Glasses

- Bảng Cart
- Mô tả: Chứa thông tin về giỏ hàng của người dùng
- Thuộc tính

Tên trường	Kiểu dữ liệu	Nullable	Default	Mô tả
id	uuid			Khoá chính
quantity	integer		1	Số lượng sản phẩm trong giỏ
userId	uuid			Khoá ngoại đến bảng User

- Bảng CartItem
- Mô tả Chứa thông tin chi tiết về các sản phẩm trong giỏ hàng
- Thuộc tính:

Bảng 3.17 Thuộc tính bảng CartItem

Tên trường	Kiểu dữ liệu	Nullable	Default	Mô tả
id	uuid			Khoá chính
quantity	integer		1	Số lượng của sản phẩm
userId	uuid			Khoá ngoại đến bảng User
glassesId	uuid			Khoá ngoại đến bảng Glasses

- Bảng Order
- Mô tả: Chứa thông tin về đơn hàng
- Thuộc tính:

Bảng 3.18 Thuộc tính bảng Order

Tên trường	Kiểu dữ liệu	Nullable	Default	Mô tả
id	uuid			Khoá chính
firstName				Tên người đặt đơn hàng
lastName				Họ người đặt đơn hàng
address				Địa chỉ người đặt hàng
totalAmount	integer		0	Tổng tiền của đơn hàng
note	varchar	True		Ghi chú của người dùng cho đơn hàng
status	varchar		PENDING	Trạng thái của đơn hàng, có các giá trị: PENDING, COMPLETED, CANCEL
userId	uuid			Khoá ngoại đến bảng User

- Bảng OrderItem
- Mô tả: Chứa thông tin chi tiết về hoá đơn từng sản phẩm
- Thuộc tính:

Bảng 3.19 Thuộc tính bảng OrderItem

Tên trường	Kiểu dữ liệu	Nullable	Default	Mô tả
id	uuid			Khoá chính
quantity	integer		1	Số lượng của sản phẩm
price	integer		0	Giá của số lượng sản phẩm
glassesId	uuid			Khoá ngoại đến bảng Glasses
orderId	uuid			Khoá ngoại đến bảng Order

- Bảng Payment
- Mô tả: Chứa thông tin về giao dịch thanh toán
- Thuộc tính:

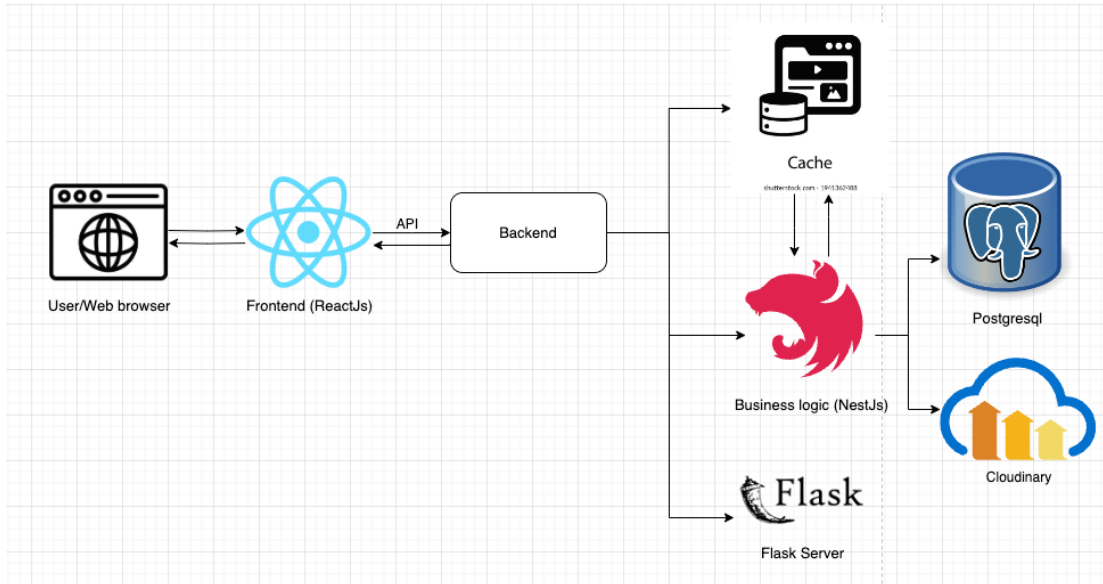
Bảng 3.20 Thuộc tính bảng Payment

Tên trường	Kiểu dữ liệu	Nullable	Default	Mô tả
id	uuid			Khoá chính
amount	integer		0	Số tiền thanh toán
method	varchar			Phương thức thanh toán, có các giá trị: BANK_ACCOUNT, CREDIT_NUMBER
status	varchar		PENDING	Trạng thái của đơn hàng, có các

				giá trị: PENDING, COMPLETED, CANCEL
bankGate	varchar	true	null	Loại cổng thanh toán người thụ hưởng
userId	uuid			Khoá ngoại đến bảng User
orderId	uuid			Khoá ngoại đến bảng Order

Chương 4: XÂY DỰNG CHƯƠNG TRÌNH

4.1. Kiến trúc hệ thống



Hình 4.1 Kiến trúc hệ thống

4.1.1. Mô tả chi tiết từng phần

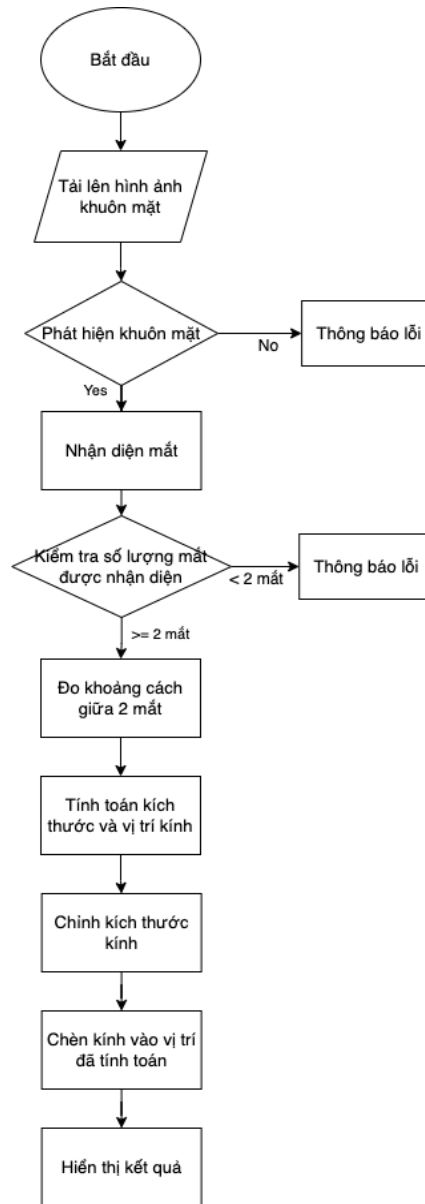
- **Người dùng/Trình duyệt:** Giao diện người dùng, nơi người dùng sử dụng trình duyệt để tương tác với website
- **Frontend (ReactJs):** Giao diện người dùng được xây dựng bằng ReactJS. Đây là nơi các trang web được hiển thị và người dùng có thể tương tác với các tính năng của hệ thống.
- **Backend (NestJS):** Backend trực tiếp nhận các yêu cầu từ frontend và xử lý chúng. Backend này có các thành phần:
 - **Cache Memory:** Bộ nhớ đệm được sử dụng để lưu trữ tạm thời các dữ liệu cần truy cập nhanh, giúp cải thiện hiệu suất hệ thống.
 - **Business Logic (NestJS):** Xử lý các nghiệp vụ logic và các yêu cầu từ frontend. Đây là nơi thực hiện các tác vụ chính của ứng dụng.
- **Server Python:** Máy chủ Python chuyên xử lý các tác vụ phức tạp như học máy và xử lý dữ liệu. Nó có thể nhận dữ liệu từ Business Logic để thực hiện các công việc cụ thể.
- **PostgreSQL:** Cơ sở dữ liệu quan hệ nơi lưu trữ tất cả các thông tin liên quan đến sản phẩm, khách hàng, đơn hàng, v.v.
- **Cloudinary:** Dịch vụ lưu trữ đám mây dùng để lưu trữ và quản lý các tài nguyên

đa phương tiện như hình ảnh và video

4.1.2. Luồng hoạt động

- Người dùng truy cập vào website thông qua trình duyệt và tương tác với giao diện được xây dựng bằng ReactJS.
- Các yêu cầu từ frontend được gửi trực tiếp đến backend (NestJS).
- Backend tiếp nhận các yêu cầu và xử lý
- Kết quả từ backend (NestJS hoặc Python) được trả về frontend để hiển thị cho người dùng.
- Dữ liệu đa phương tiện như hình ảnh và video được lưu trữ và quản lý bởi Cloudinary.

4.2. Sơ đồ thuật toán ghép kính vào khuôn mặt



Hình 4.2 Sơ đồ thuật toán ghép kính vào khuôn mặt

Mô tả chi tiết từng bước:

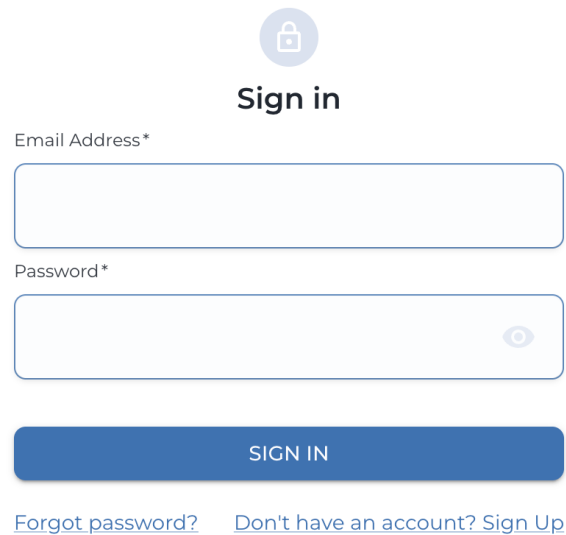
- **Nhận diện khuôn mặt:** Dùng mô hình haarcascade_frontalface_alt.xml để nhận diện khuôn mặt trong ảnh. Để tăng tốc độ và độ chính xác của việc nhận diện khuôn mặt, ảnh gốc được chuyển sang dạng grayscale.
- **Nhận diện mắt:** Sau khi nhận diện khuôn mặt, hệ thống xác định vùng chứa khuôn mặt trong ảnh gốc. Vùng này sẽ được dùng để nhận diện mắt nhằm giảm bớt khu vực tìm kiếm và tăng độ chính xác. Tiếp tục sử dụng mô hình Haar

Cascade nhưng được huấn luyện đặc biệt cho việc nhận diện mắt để phát hiện các mắt trong vùng khuôn mặt. Quá trình này tương tự như nhận diện khuôn mặt nhưng được tinh chỉnh cho mắt.

- **Kiểm tra số lượng mắt được nhận diện:** Nếu ít hơn hai mắt được nhận diện, hệ thống sẽ thông báo lỗi và dừng quá trình. Điều này đảm bảo rằng kính sẽ được đặt chính xác trên cả hai mắt. Nếu có hai hoặc nhiều hơn mắt được nhận diện, hệ thống sẽ chọn hai mắt có khoảng cách gần nhất, đảm bảo đó là mắt trái và mắt phải.
- **Đo khoảng cách giữa hai mắt:** Hệ thống sẽ đo khoảng cách giữa tâm của hai mắt được chọn. Khoảng cách này sẽ được sử dụng để tính toán kích thước của kính nhằm đảm bảo kính phù hợp với khuôn mặt.
- **Tính toán kích thước và vị trí kính:** Dựa trên khoảng cách giữa hai mắt, hệ thống sẽ tính toán kích thước mới cho kính. Hệ số điều chỉnh sẽ được sử dụng để đảm bảo kính có kích thước phù hợp với khoảng cách giữa hai mắt. Kính sẽ được điều chỉnh kích thước theo tỷ lệ đã tính toán. Việc này đảm bảo kính sẽ vừa vặn với khuôn mặt khi được chèn vào ảnh.
- **Chèn kính vào vị trí đã tính toán:** Dựa trên tọa độ tâm mắt và kích thước kính, hệ thống sẽ xác định vị trí đặt kính. Điều này bao gồm tính toán sao cho kính nằm đúng vị trí và không bị lệch. Hệ thống sẽ điều chỉnh vị trí nếu cần để đảm bảo kính nằm trong giới hạn của ảnh, tránh việc kính bị cắt bỏ hay không hiển thị đúng.
- **Xử lý mask để loại bỏ nền trắng:** Chuyển đổi ảnh kính sang grayscale và tạo mặt nạ để chỉ giữ lại phần kính, loại bỏ nền trắng. Mặt nạ này sẽ được sử dụng để chỉ chèn phần kính vào ảnh gốc. Tạo một mặt nạ nghịch đảo để loại bỏ phần nền của kính khi chèn vào ảnh gốc.
- **Chèn kính vào ảnh gốc:** Hệ thống sẽ kết hợp phần nền của ảnh gốc với phần kính đã được chèn, sử dụng mặt nạ và mặt nạ nghịch đảo để chỉ giữ lại phần kính và loại bỏ nền trắng. Sau khi chèn kính, ảnh kết quả sẽ được hiển thị để người dùng kiểm tra và điều chỉnh nếu cần.

4.3. Giao diện chương trình

4.3.1. Trang đăng nhập



The image shows a 'Sign in' form. At the top, there is a circular icon containing a padlock. Below the icon, the text 'Sign in' is centered. The form consists of two input fields: 'Email Address*' and 'Password*'. The 'Email Address*' field is a simple text box. The 'Password*' field is a text box with a small eye icon on the right side, indicating a toggle for password visibility. Below the input fields is a blue button with the text 'SIGN IN' in white. At the bottom of the form, there are two links: 'Forgot password?' and 'Don't have an account? Sign Up'.

Hình 4.3 Trang đăng nhập

4.3.2. Trang đăng ký



Sign up

First Name*

Last Name*

Email Address*

Phone Number*

Address*

Password*

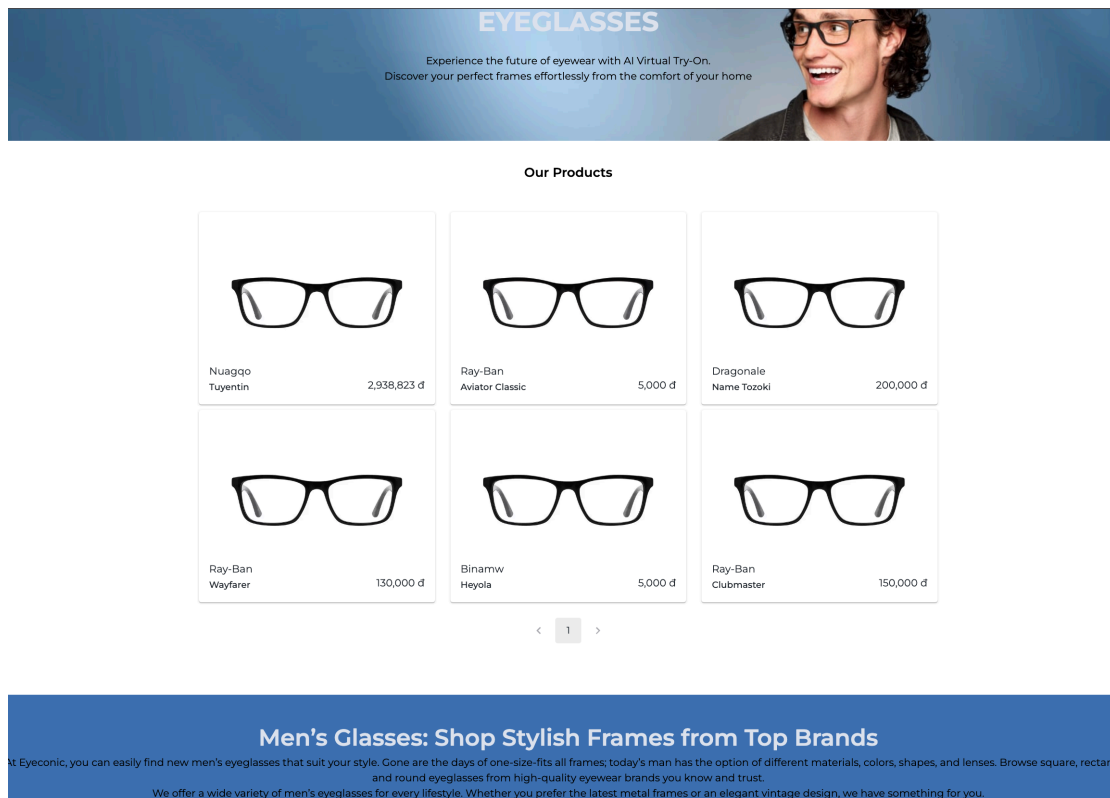
Confirm Password*

SIGN UP

[Already have an account? Sign in](#)

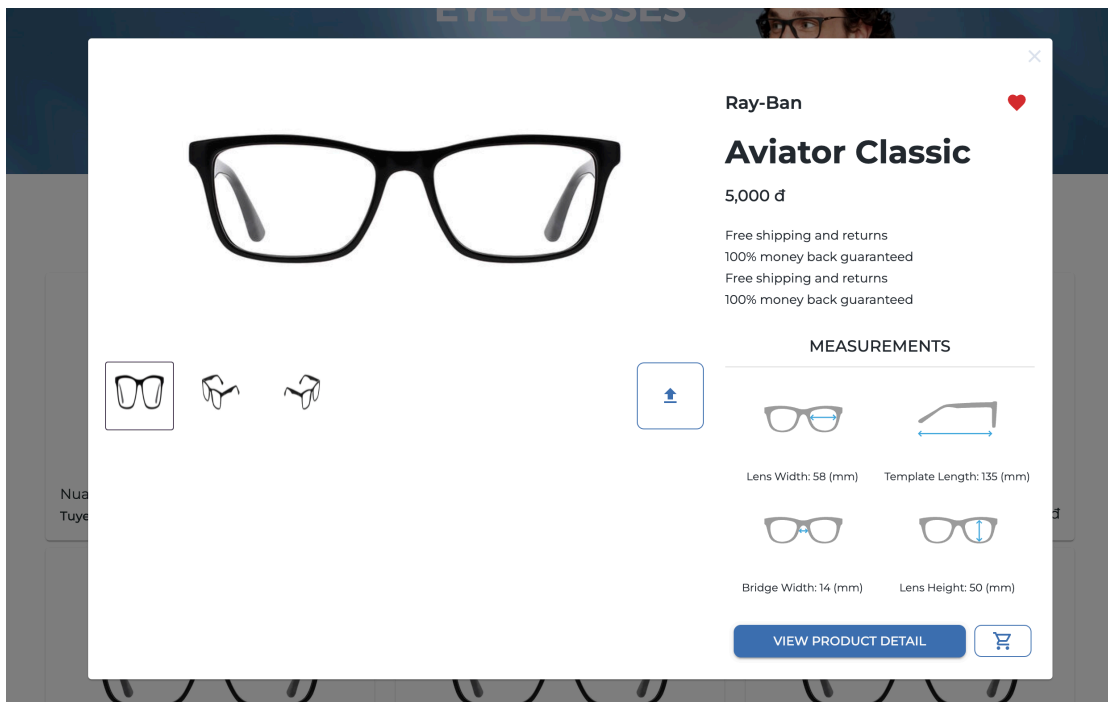
Hình 4.4 Trang đăng ký

4.3.3. Trang chủ



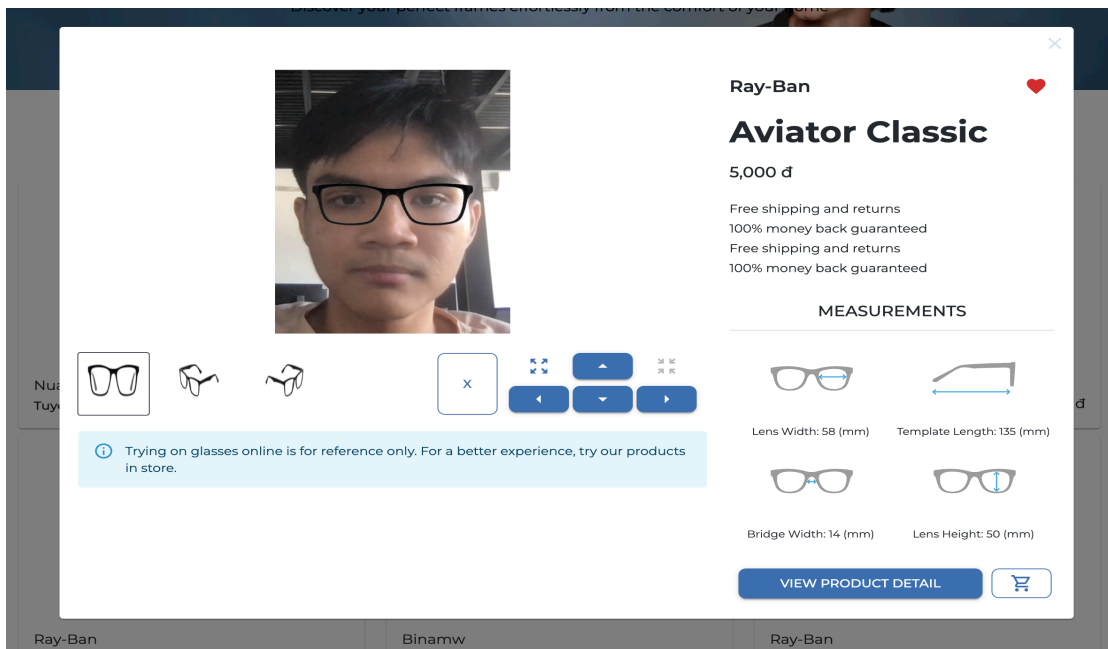
Hình 4.5 Trang chủ

4.3.4. Trang xem trước chi tiết sản phẩm



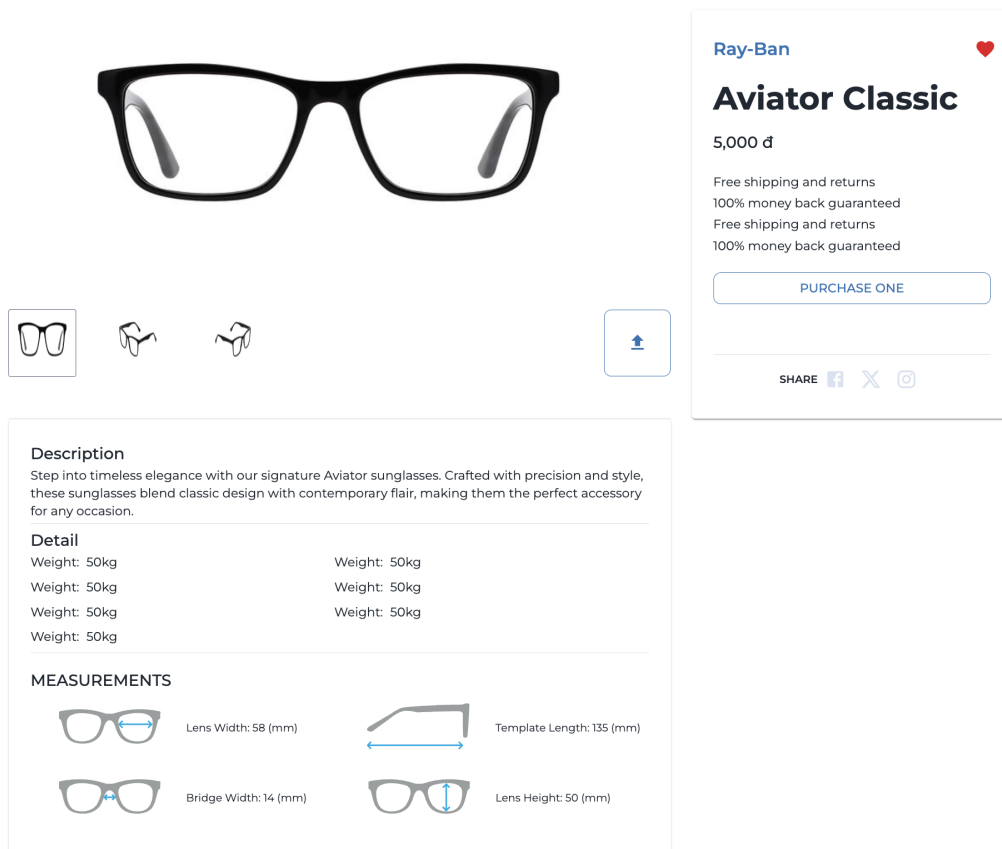
Hình 4.6 Trang xem trước chi tiết sản phẩm

4.3.4. Trang kết quả của việc thử kính



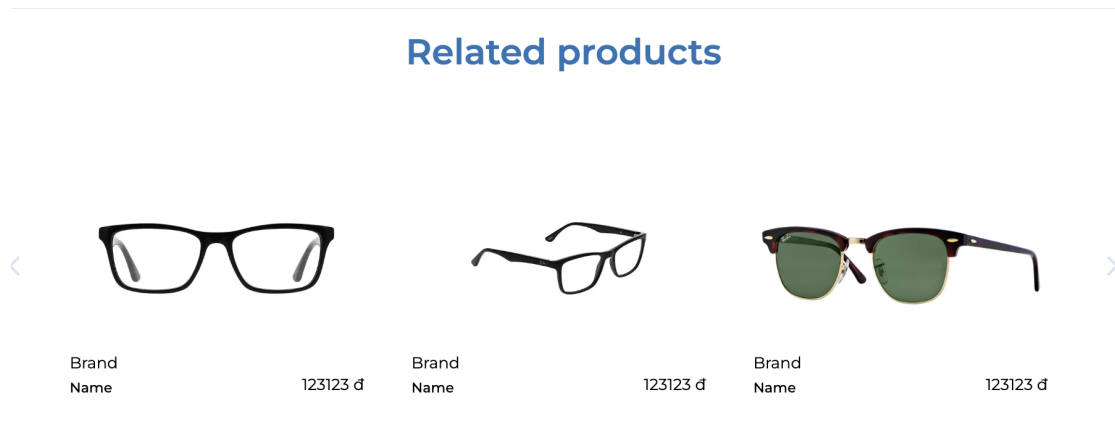
Hình 4.7 Trang kết quả của việc thử kính

4.3.5. Trang chi tiết sản phẩm



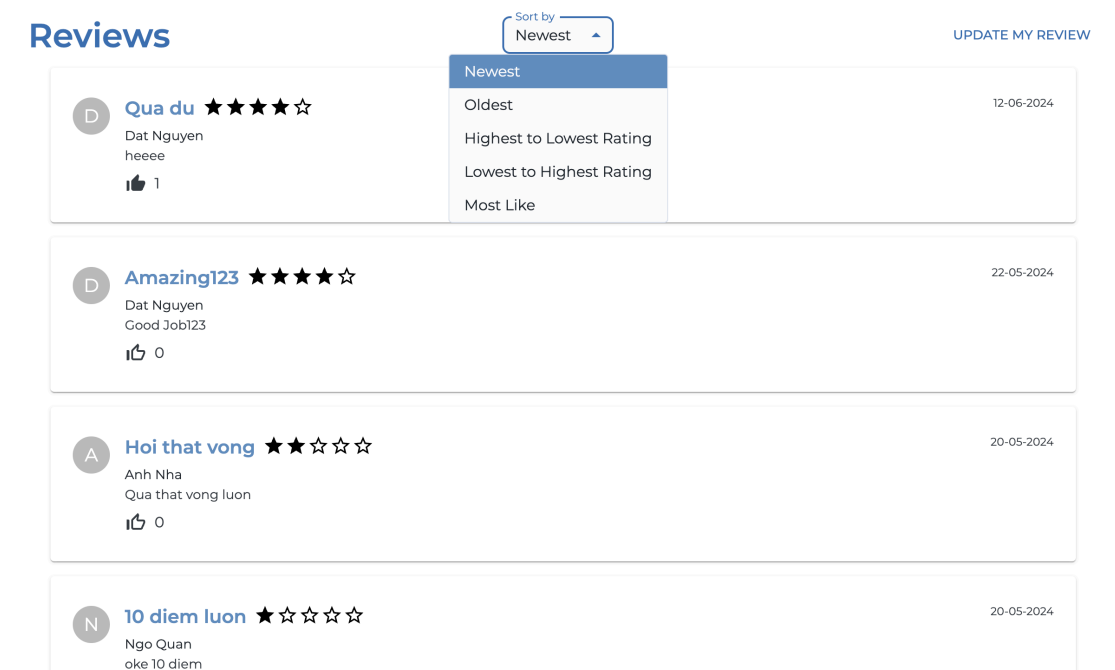
Hình 4.8 Trang chi tiết sản phẩm

4.3.6. Trang các sản phẩm liên quan



Hình 4.9 Trang các sản phẩm liên quan

4.3.7. Trang xem đánh giá sản phẩm



Hình 4.10 Trang xem đánh giá sản phẩm

4.3.8. Trang đánh giá sản phẩm

The screenshot shows a modal window titled "Write new review" for the product "Ray-Ban Aviator Classic". The user "Anh Nha" is logged in. The form includes a star rating section, a title input field, a text area for the review, and a "SUBMIT" button. The text area has a character count of 0/200.

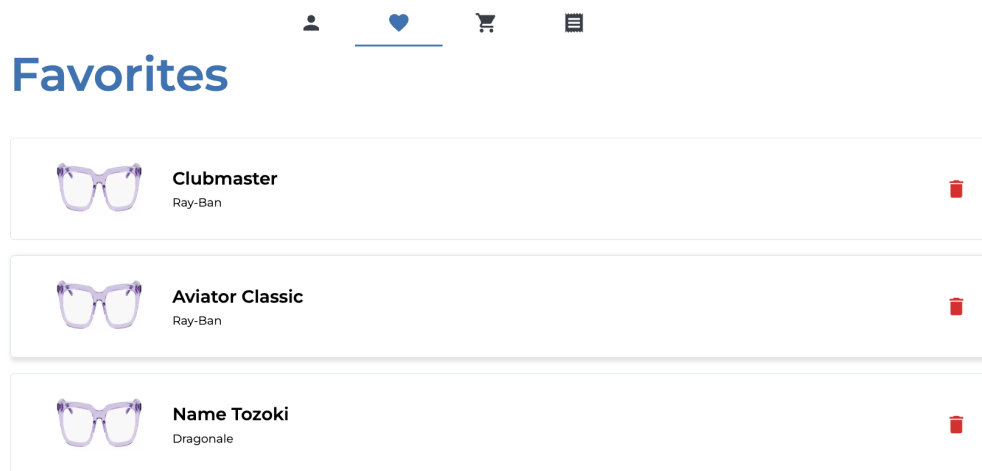
Hình 4.11 Trang đánh giá sản phẩm

4.3.9. Trang cập nhật thông tin cá nhân

The screenshot shows a "Profile" page with a navigation bar containing icons for profile, favorites, cart, and menu. The profile section on the left displays the user's name "Dat Nguyen", email "admin@gmail.com", date of birth "14-05-2024", and address "Da Nang", along with a "LOG OUT" button. The main section on the right is titled "Personal details" and contains form fields for "First name" (Dat), "Last name" (Nguyen), "Address" (Da Nang), "Phone number" (0932871387), and "Date of birth" (14-05-2024). Below this is a "Password change" section with fields for "Current password", "New password", and "Confirm new password". "CANCEL" and "SAVE" buttons are present at the bottom of each section.

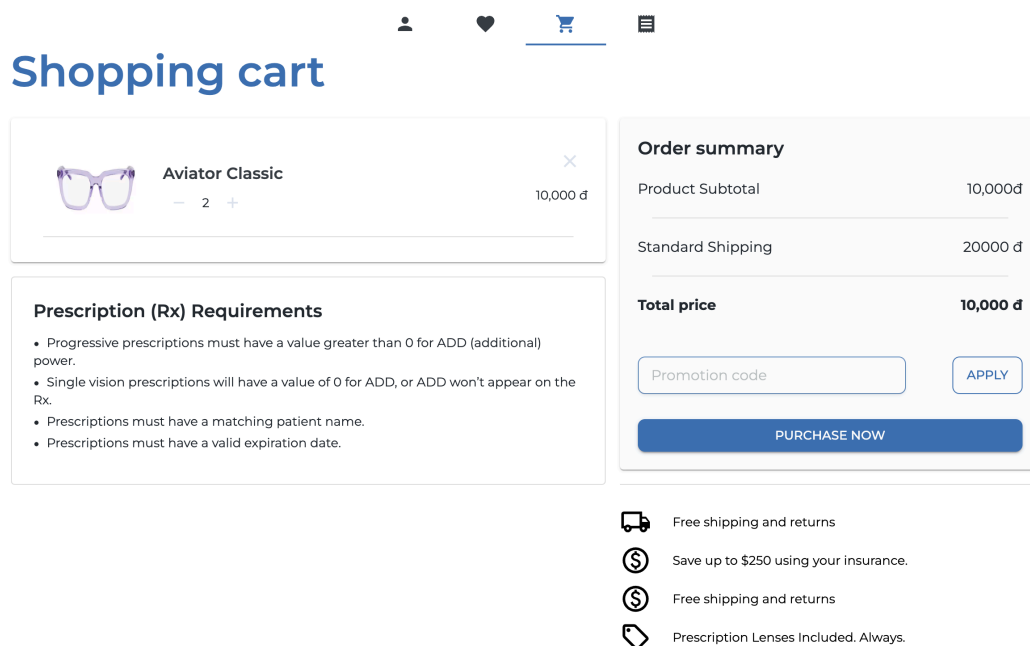
Hình 4.12 Trang cập nhật thông tin cá nhân

4.3.10. Trang danh sách yêu thích



Hình 4.13 Trang danh sách yêu thích

4.3.11. Trang giỏ hàng



Hình 4.14 Trang giỏ hàng

4.3.12. Trang xác nhận thông tin mua hàng

[← BACK](#)

Total

5,000 đ

 Aviator Classic
1 x 5000 đ

5,000 đ

1 Shipping address

2 Payment details

3 Review your order

Please confirm this information below to process the order.

First name*

Last name*

Anh

Nha

Address*

Ngu Hanh Son

Phone number*

0932981293

Order note

Note about the orders, for example: the delivery time, specific location.

0/300

NEXT >

Hình 4.15 Trang xác nhận thông tin mua hàng

4.3.13. Trang chọn phương thức thanh toán ngân hàng

[← BACK](#)

Total

5,000 đ

 Aviator Classic
1 x 5000 đ

5,000 đ

✓ Shipping address

2 Payment details

3 Review your order

Bank account

Card

Please choose one of the banks below to proceed with your money transfer:





[← PREVIOUS](#)

15:00

Hình 4.16 Trang chọn phương thức thanh toán ngân hàng

4.3.14. Trang thanh toán bằng mã QR

[← BACK](#)

Total
5,000 đ

 Aviator Classic
1 x 5000 đ 5,000 đ

1 Shipping address

2 Payment details

3 Review your order

Bank account

Card

Please choose one of the banks below to proceed with your money transfer:



You have 15 minutes to purchase the order. After that time, the order will be declined.

Your order will be processed once we receive the funds.

Bank account

Please scan the QR below to transfer the payment to the bank account.

ID Payment: **cfb960bb**

ID Order: **ca8a033e**

QR Code:



[← PREVIOUS](#)

14:41

Hình 4.17 Trang thanh toán bằng mã QR

4.3.15. Trang xác nhận thông tin mua hàng

[← BACK](#)

Total
5,000 đ

 Aviator Classic
1 x 5000 đ 5,000 đ

1 Shipping address

2 Payment details

3 Review your order

Products
1 items 5,000 đ

Shipping
Plus taxes 20,000 đ

Total 5,000 đ

Shipment details

Name: Anh Nha

Address: Ngu Hanh Son

Order note:

Payment details

Order ID: a60de237

Payment method: BANK_ACCOUNT

Payment person: Anh Nha

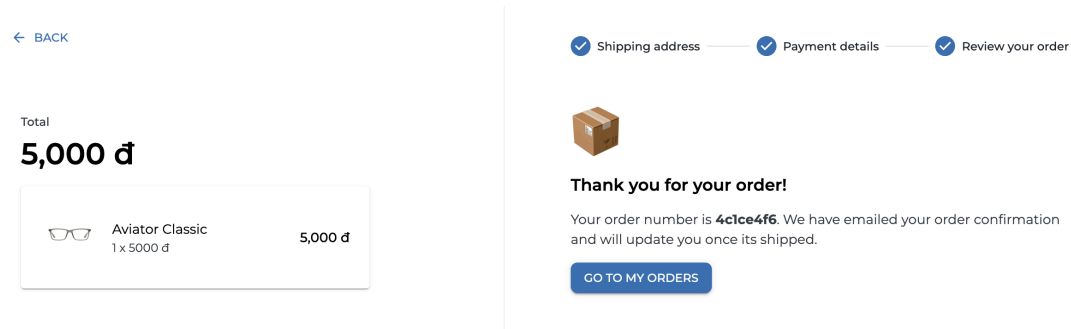
Date of payment: 15-06-2024

[← PREVIOUS](#)

DONE →

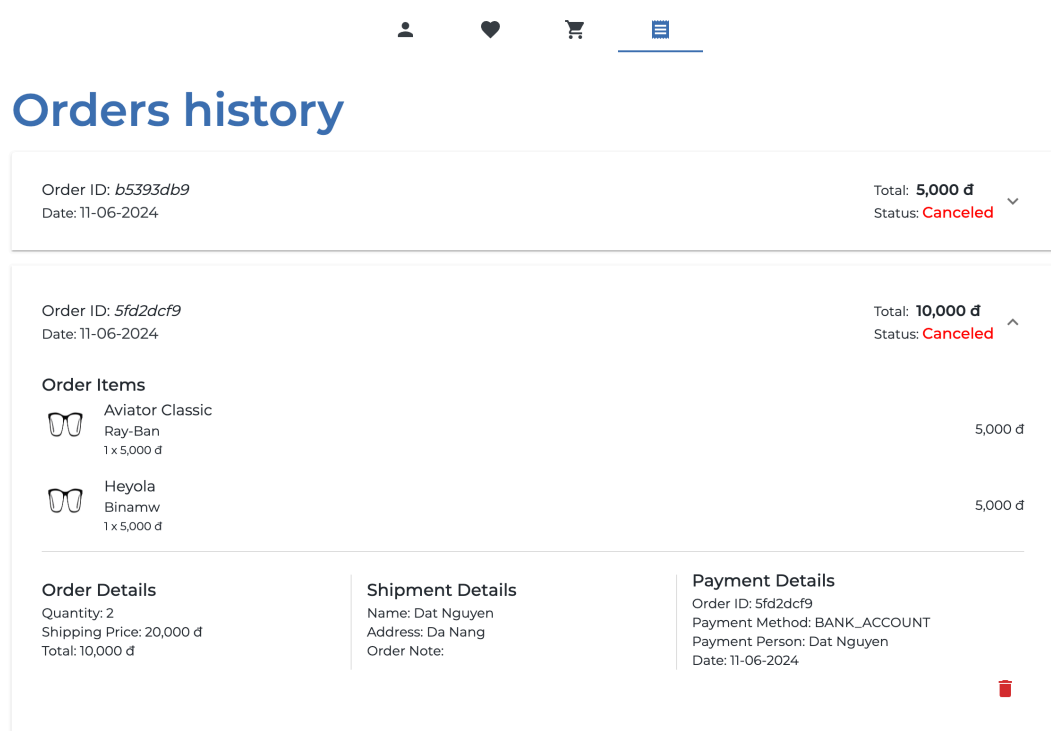
Hình 4.18 Trang xác nhận thông tin mua hàng

4.3.16. Trang cảm ơn mua hàng



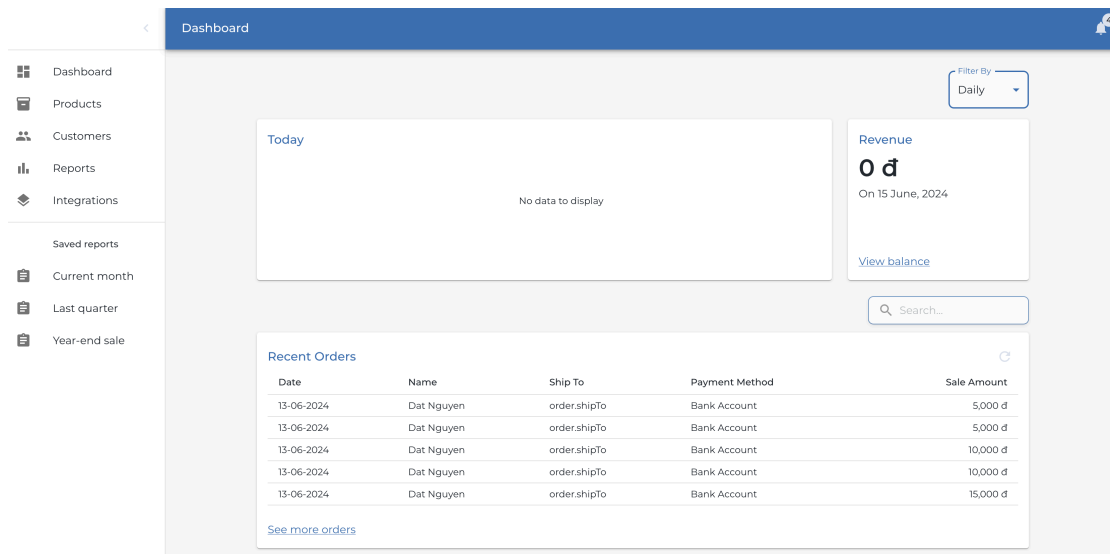
Hình 4.19 Trang cảm ơn mua hàng

4.3.17. Trang lịch sử mua hàng



Hình 4.20 Trang lịch sử mua hàng

4.3.18. Trang quản xem thống kê doanh thu



Hình 4.21 Trang quản xem thống kê doanh thu

Chương 5: KẾT LUẬN

5.1. Những kết quả đạt được

5.1.1. Về mặt lý thuyết

- Nắm vững hơn về quy trình phát triển một trang web thương mại điện tử, từ việc thiết kế giao diện người dùng, phát triển các chức năng cần thiết, đến việc triển khai và bảo trì hệ thống.
- Học được cách tích hợp nhiều công nghệ khác nhau vào một hệ thống hoàn chỉnh.
- Trong suốt quá trình nghiên cứu và phát triển, dự án đã gặp phải nhiều thách thức kỹ thuật và đã học được cách tìm ra giải pháp, cải thiện hệ thống và tối ưu hóa hiệu suất.
- Quá trình phát triển công cụ thử kính trực tuyến đã giúp nâng cao kiến thức về các thuật toán xử lý ảnh như nhận diện khuôn mặt, theo dõi đặc điểm khuôn mặt và gắn vật thể ảo lên hình ảnh.

5.1.2. Về mặt ứng dụng

- Website đã được thiết kế để quản lý sản phẩm một cách hiệu quả, bao gồm việc thêm, xóa, sửa và hiển thị thông tin chi tiết của các sản phẩm kính thuốc. Hệ thống quản lý đơn hàng cũng hoạt động ổn định, cho phép người dùng đặt hàng, mua hàng và thực hiện thanh toán trực tuyến một cách dễ dàng.
- Chức năng thử kính trực tuyến là điểm nổi bật của website. Người dùng có thể sử dụng tải lên hình ảnh của họ để thử kính. Thuật toán xử lý ảnh đã được tích hợp giúp nhận diện khuôn mặt và gắn kính ảo lên hình ảnh của người dùng một cách chính xác và tự nhiên.
- Giao diện của website được thiết kế thân thiện và dễ sử dụng. Tối ưu hóa các thao tác người dùng để giảm thiểu thời gian và công sức cần thiết để thực hiện các chức năng như tìm kiếm sản phẩm, thử kính, và đặt hàng. Hệ thống cũng hỗ trợ nhiều phương thức thanh toán, mang lại sự tiện lợi tối đa cho người dùng.
- Việc tích hợp công nghệ xử lý ảnh vào chức năng thử kính trực tuyến đã được thực hiện thành công. Đã sử dụng các thuật toán nhận diện khuôn mặt và xử lý hình ảnh để đảm bảo kính ảo được gắn chính xác và phù hợp với khuôn mặt của người dùng.

5.2. Hạn chế

- Mặc dù đã đề ra hai phương pháp thử kính trực tuyến: thử kính 2D với hình ảnh khuôn mặt, thử kính 3D với trình chiếu webcam. Nhưng với thời gian nghiên cứu còn hạn chế, dự án đã chỉ cho ra phương pháp thử kính 2D với hình ảnh khuôn mặt.
- Mặc dù thuật toán xử lý ảnh đã hoạt động tốt trong nhiều trường hợp, nhưng vẫn còn một số tình huống mà độ chính xác của việc nhận diện khuôn mặt và gắn kính ảo chưa được tối ưu, đặc biệt là khi người dùng có đặc điểm khuôn mặt không điển hình hoặc sử dụng hình ảnh chất lượng thấp.
- Việc xử lý ảnh và gắn kính ảo yêu cầu một lượng tài nguyên tính toán đáng kể, điều này có thể gây ra hiện tượng chậm trễ, đặc biệt là trên các thiết bị di động có cấu hình thấp. Cải thiện hiệu suất và tốc độ xử lý là một thách thức cần được giải quyết để đảm bảo trải nghiệm người dùng mượt mà hơn.
- Mặc dù giao diện người dùng đã được tối ưu hóa, nhưng vẫn có thể cần thêm các tùy chọn tùy chỉnh và cá nhân hóa để đáp ứng tốt hơn nhu cầu đa dạng của người dùng. Đặc biệt là việc hỗ trợ nhiều ngôn ngữ và văn hóa khác nhau.
- Dữ liệu sản phẩm kính thuốc có thể chưa đầy đủ để đáp ứng mọi nhu cầu kính mắt của đại đa số người dùng và chưa được cập nhật thường xuyên, điều này có thể ảnh hưởng đến việc lựa chọn của người dùng. Việc duy trì một cơ sở dữ liệu sản phẩm phong phú cùng dữ liệu đa dạng và cập nhật thường xuyên là điều cần thiết.

5.3. Hướng phát triển

- Phát triển thêm tính năng thử kính trực tuyến 3D với trình chiếu webcam, sử dụng các ngôn ngữ lập trình 3D kết hợp vẽ thực thể kính mắt trong môi trường ảo nhằm tăng trải nghiệm thực tế ảo của khách hàng khi trải nghiệm sản phẩm.
- Tăng cường độ chính xác của các thuật toán nhận diện khuôn mặt và gắn kính ảo bằng cách áp dụng các mô hình học sâu (deep learning) tiên tiến hơn. Điều này sẽ giúp hệ thống nhận diện được nhiều đặc điểm khuôn mặt phức tạp hơn và hoạt động tốt hơn trong các điều kiện ánh sáng và góc chụp khác nhau.
- Nghiên cứu và triển khai các giải pháp tối ưu hóa thuật toán để giảm thời gian xử lý và sử dụng ít tài nguyên hơn, đảm bảo hệ thống hoạt động mượt mà trên nhiều thiết bị, bao gồm cả các thiết bị di động có cấu hình thấp.
- Cho phép người dùng tùy chỉnh các chi tiết của kính thuốc như màu sắc, kiểu dáng, và chất liệu trước khi mua hàng. Điều này sẽ tăng tính cá nhân hóa và sự hài lòng của khách hàng.

- Phát triển phiên bản đa ngôn ngữ để mở rộng thị trường ra quốc tế, đảm bảo người dùng từ các quốc gia khác nhau có thể dễ dàng sử dụng và mua sắm trên website.
- Sử dụng trí tuệ nhân tạo để đề xuất các sản phẩm phù hợp dựa trên lịch sử mua sắm và sở thích của người dùng. Điều này sẽ giúp người dùng tìm thấy các sản phẩm họ thích nhanh hơn và dễ dàng hơn.

TÀI LIỆU THAM KHẢO

[1] Numpy introduction

https://www.w3schools.com/python/numpy/numpy_intro.asp

[2] Python tutorial

https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html

[3] NestJs documentation

<https://docs.nestjs.com/>

[4] Typescript tutorial

<https://www.w3schools.com/typescript/>

[5] ReactJs tutorial

<https://legacy.reactjs.org/tutorial/tutorial.html>

[6] Cloudinary

<https://viblo.asia/p/su-dung-cloudinary-de-quan-ly-anh-cho-ung-dung-cua-ban-E7bGoxggv5e2>

[7] Một số thuật toán xử lý ảnh

<https://vinbigdata.com/camera-ai/xu-ly-hinh-anh-trong-python-tu-thuat-toan-den-cong-cu.html>