

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**



NGUYỄN THỊ THU UYÊN

**XÂY DỰNG WEBSITE BÁN QUẦN ÁO NỮ ONLINE
TRACYSHOP**

**KHÓA LUẬN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ THÔNG TIN**

TP. HỒ CHÍ MINH, 2022

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH



NGUYỄN THỊ THU UYÊN

XÂY DỰNG WEBSITE BÁN QUẦN ÁO NỮ ONLINE
TRACYSHOP

Mã số sinh viên: 1851050182

KHÓA LUẬN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ THÔNG TIN

Giảng viên hướng dẫn: ThS. NGUYỄN THỊ PHƯƠNG TRANG

TP. HỒ CHÍ MINH, 2022

TRƯỜNG ĐẠI HỌC MỞ CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
THÀNH PHỐ HỒ CHÍ MINH **Độc lập – Tự do – Hạnh phúc**
KHOA CÔNG NGHỆ THÔNG TIN

GIẤY XÁC NHẬN

Tôi tên là: Nguyễn Thị Thu Uyên

Ngày sinh: 17/01/2000 Nơi sinh: Quảng Ngãi.....

Chuyên ngành: Công nghệ thông tin.....

Mã sinh viên: 1851050182

Tôi đồng ý cung cấp toàn văn thông tin đồ án/ khóa luận tốt nghiệp hợp lệ về bản quyền cho Thư viện Trường Đại học Mở Thành phố Hồ Chí Minh. Thư viện Trường Đại học Mở Thành phố Hồ Chí Minh sẽ kết nối toàn văn thông tin đồ án/ khóa luận tốt nghiệp vào hệ thống thông tin khoa học của Sở Khoa học và Công nghệ Thành phố Hồ Chí Minh.

Ký tên
(Ghi rõ họ và tên)

Nguyễn Thị Thu Uyên

Ý KIẾN CHO PHÉP BẢO VỆ ĐỒ ÁN/ KHÓA LUẬN TỐT NGHIỆP
CỦA GIẢNG VIÊN HƯỚNG DẪN

Giảng viên hướng dẫn: Nguyễn Thị Phương Trang

Sinh viên thực hiện: Nguyễn Thị Thu Uyên Lớp: DH18IT02.....

Ngày sinh: 17/01/2000 Nơi sinh: Quảng Ngãi

Tên đề tài: Xây dựng website bán quần áo nữ online TracyShop.....

.....
.....
.....
.....
.....
.....

Ý kiến của giảng viên hướng dẫn về việc cho phép sinh viên được bảo vệ đồ án/ khóa luận trước Hội đồng:

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Thành phố Hồ Chí Minh, ngày ... tháng ... năm

Người nhận xét

LỜI CẢM ƠN

Trong quá trình làm đồ án ngành, em đã nhận được nhiều sự quan tâm, đóng góp ý kiến, chỉ dạy nhiệt tình từ quý thầy cô cũng như sự động viên từ gia đình, bạn bè.

Em xin chân thành cảm ơn quý thầy cô của trường Đại học Mở thành phố Hồ Chí Minh nói chung và quý thầy cô của khoa Công nghệ thông tin nói riêng đã tận tình giảng dạy, chia sẻ nhiều kinh nghiệm quý báu, giúp em có được những kiến thức và nền tảng vững chắc để làm được đồ án ngành này.

Đặc biệt, em xin gửi lời cảm ơn chân thành đến **ThS. Nguyễn Thị Phương Trang**, giảng viên khoa Công nghệ thông tin, trường Đại học Mở thành phố Hồ Chí Minh đã tận tình hướng dẫn, giúp đỡ và chỉ dạy cho em trong suốt quá trình làm đồ án ngành, cô luôn đưa ra những ý kiến giúp em có thể hoàn thành đồ án ngành một cách tốt nhất.

Cuối cùng, em xin gửi lời cảm ơn đến gia đình, bạn bè, những người đã luôn động viên, quan tâm, giúp đỡ em trong quá trình làm đồ án ngành này.

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

TÓM TẮT KHÓA LUẬN TỐT NGHIỆP

Kinh doanh online hay bán hàng online không phải là một cái gì đó xa lạ với nhiều người, nhất là ở thời đại mạng Internet đang ngày càng phát triển. Nhiều doanh nghiệp, cửa hàng lớn và nhỏ luôn chọn hình thức kinh doanh này song song với việc bán hàng trực tiếp. Bên cạnh đó, người mua cũng rất ưa chuộng hình thức mua sắm online này vì sự tiện lợi của nó mang lại. Bắt kịp với xu hướng và nhu cầu của khách hàng hiện nay, trong đề tài này, em xây dựng một trang web bán quần áo nữ online cho phép người mua có thể đặt hàng, mua hàng và có thể thanh toán trực tiếp thông qua cổng thanh toán điện tử. Tuy nhiên, cùng với việc tiện lợi của mua bán online thì nó cũng vướng phải một số vấn đề như sau: bảo mật thông tin người dùng, hiệu suất ứng dụng, chi phí bộ nhớ, vấn đề đa nền tảng. Với ASP.NET Core thì những vấn đề trên đều được giải quyết. Do đó, em có ý tưởng sử dụng công nghệ ASP.NET Core MVC để xây dựng website bán quần áo nữ online. Ngoài ra, trong đề tài, em có tích hợp thêm chức năng gửi mail để xác thực tài khoản email của khách hàng nhằm tăng tính bảo mật thông tin người dùng.

MỤC LỤC

LỜI CẢM ƠN	3
NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN.....	4
TÓM TẮT KHÓA LUẬN TỐT NGHIỆP	5
DANH MỤC THUẬT NGỮ TIẾNG ANH VÀ TỪ VIẾT TẮT.....	8
DANH MỤC HÌNH VẼ	9
DANH MỤC BẢNG	14
MỞ ĐẦU.....	15
Chương 1. TỔNG QUAN ĐỀ TÀI	17
1.1. Lý do chọn đề tài.....	17
1.2. Bố cục đề tài	17
Chương 2. CƠ SỞ LÝ THUYẾT.....	18
2.1. Tổng quan về ASP.NET Core MVC.....	18
2.1.1. ASP.NET Core MVC là gì?	18
2.1.2. Routing	18
2.1.3. Model binding	19
2.1.4. Model validation.....	20
2.1.5. Dependency Injection (DI).....	22
2.1.6. Tag Helpers	25
2.1.7. Migration	27
2.2. Hệ quản trị cơ sở dữ liệu SQL Server	29
2.3. Đăng nhập qua Facebook và Google với ASP.NET Core Identity.....	29
2.3.1. ASP.NET Core Identity.....	29
2.3.2. Đăng nhập qua Google	35
2.3.3. Đăng nhập qua Facebook	38
2.4. Kiểm tra gửi mail bằng Mailtrap.....	43
2.4.1. Giới thiệu về Mailtrap	43

2.4.2.	Mailtrap là gì?	43
2.4.3.	Cấu hình dịch vụ gửi mail qua Mailtrap.....	45
2.4.4.	Thiết lập gửi mail qua Mailtrap.....	46
2.5.	Thanh toán qua dịch vụ thanh toán trực tuyến PayPal.....	50
Chương 3.	WEBSITE BÁN QUẦN ÁO NỮ ONLINE TRACYSHOP	53
3.1.	Giới thiệu đê tài.....	53
3.2.	Phân tích và thiết kế hệ thống	54
3.2.1.	Sơ đồ usecase	54
3.2.2.	Đặc tả usecase	54
3.2.3.	Mô hình cơ sở dữ liệu.....	65
3.3.	Thực nghiệm hệ thống.....	69
3.3.1.	Giao diện của Admin.....	69
3.3.2.	Giao diện của Nhân viên	85
3.3.3.	Giao diện Khách hàng	93
Chương 4.	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỀN ĐỀ TÀI	107
4.1.	Kết luận	107
4.2.	Hướng phát triển đề tài.....	107
TÀI LIỆU THAM KHẢO	108	
PHỤ LỤC	110	

DANH MỤC THUẬT NGỮ TIẾNG ANH VÀ TỪ VIẾT TẮT

Danh mục thuật ngữ tiếng Anh

STT	Thuật ngữ	Dịch nghĩa
1	Online	Trực tuyến
2	Request	Yêu cầu
3	Output	Đầu ra
4	Redirect	Chuyển hướng
5	File	Tập tin
6	Routing engine	Công cụ định tuyến
7	Server	Máy chủ
9	Module	Tiêu chuẩn
10	Service	Dịch vụ
11	Windows	Hệ điều hành Windows
12	Mail	Thư
13	Template	Bản mẫu
14	Internet	Mạng Internet
15	Website	Trang web hoặc trang mạng
16	Realtime	Thời gian thực

Danh mục từ viết tắt

STT	Từ viết tắt	Dịch nghĩa
1	URL	Uniform Resource Locator (định vị tài nguyên thống nhất, địa chỉ hay liên kết)
2	DI	Dependency Injection
3	SMTP	Simple Mail Transfer Protocol (giao thức truyền tải thư tín đơn giản)
4	HTML	Hyper Text Markup Language (ngôn ngữ đánh dấu siêu văn bản)
5	CSS	Cascading Style Sheet

DANH MỤC HÌNH VẼ

Hình 2.1: Hiển thị lỗi cho người dùng khi sử dụng asp-validation-summary	21
Hình 2.2: Hiển thị lỗi cho người dùng khi sử dụng asp-validation-for	22
Hình 2.3: Thư mục Migrations trong dự án	28
Hình 2.4: Các thuộc tính trong bảng _EFMigrationsHistory	28
Hình 2.5: Dữ liệu trong bảng _EFMigrationsHistory	29
Hình 2.6: Giao diện của Manage NuGet Packages trong Visual Studio 2019	30
Hình 2.7: Các Packages trong Dependencies	30
Hình 2.8: Các bảng trong IdentityDbContext	31
Hình 2.9: Trang trả về 404 khi Roles của người dùng không hợp lệ	33
Hình 2.10: Giao diện của trang Google Cloud Platform	36
Hình 2.11: Giao diện sau khi tạo xong ứng dụng trong Google Cloud Platform.....	36
Hình 2.12: Thiết lập URIs	37
Hình 2.13: ID Client và Secret Client của Google	37
Hình 2.14: Giao diện đăng nhập qua Google	38
Hình 2.15: Giao diện của trang Facebook for Developers	39
Hình 2.16: Giao diện chọn loại ứng dụng trên Facebook for Developers.....	39
Hình 2.17: Giao diện thêm chi tiết ứng dụng trong Facebook for Developers	40
Hình 2.18: Nhập URL chuyển hướng trong Facebook for Developers	40
Hình 2.19: Các thông tin cơ bản của ứng dụng trên Facebook for Developers	41
Hình 2.20: Nhập mật khẩu Facebook để xem khóa bí mật của ứng dụng.....	41
Hình 2.21: ID ứng dụng và khóa ứng dụng của Facebook.....	41
Hình 2.22: Màn hình đăng nhập qua Facebook.....	42
Hình 2.23: Trang chủ của Mailtrap	43
Hình 2.24: Giao diện đăng ký của Mailtrap	44
Hình 2.25: Giao diện hộp thư đến trên Mailtrap	44
Hình 2.26: Các bản trả phí trên Mailtrap.....	45
Hình 2.27: Chọn Integrations	45
Hình 2.28: Cấu hình SMTP trong Mailtrap.....	45
Hình 2.29: Giao diện nhận mail trong Mailtrap	49
Hình 2.30: Giao diện đăng nhập của PayPal Developer	50

Hình 2.31: Câu hình ứng dụng trong PayPal Developer.....	51
Hình 2.32: Các tài khoản được tạo để thử nghiệm trong PayPal Developer.....	52
Hình 2.33: Giao diện thanh toán của PayPal.....	52
Hình 3.1: Sơ đồ usecase	54
Hình 3.2: Mô hình cơ sở dữ liệu.....	65
Hình 3.3: Giao diện tổng quan của Admin.....	69
Hình 3.4: Giao diện đăng nhập của admin	70
Hình 3.5: Giao diện xem danh sách nhân viên của admin	70
Hình 3.6: Giao diện xác nhận xóa nhân viên của admin	71
Hình 3.7: Giao diện xem thông tin nhân viên của admin	71
Hình 3.8: Giao diện chỉnh sửa thông tin nhân viên của admin	71
Hình 3.9: Giao diện địa chỉ nhân viên của admin	72
Hình 3.10: Giao diện thêm nhân viên của admin	72
Hình 3.11: Giao diện danh mục sản phẩm của admin.....	73
Hình 3.12: Giao diện thêm danh mục sản phẩm của admin.....	73
Hình 3.13: Giao diện chỉnh sửa tên danh mục sản phẩm của admin	73
Hình 3.14: Giao diện xem danh sách sản phẩm của admin.....	74
Hình 3.15: Giao diện trả về danh sách sản phẩm sau khi tìm kiếm của admin.....	74
Hình 3.16: Giao diện thêm sản phẩm mới của admin	74
Hình 3.17: Giao diện chỉnh sửa thông tin sản phẩm của admin.....	75
Hình 3.18: Giao diện xác nhận xóa sản phẩm của admin	75
Hình 3.19: Giao diện xem thông tin sản phẩm của admin	75
Hình 3.20: Giao diện xem danh sách khuyến mãi của admin	76
Hình 3.21: Giao diện thêm khuyến mãi của admin	76
Hình 3.22: Giao diện chỉnh sửa khuyến mãi của admin.....	76
Hình 3.23: Giao diện xác nhận xóa khuyến mãi của admin	77
Hình 3.24: Giao diện xem danh sách khách hàng của admin.....	77
Hình 3.25: Giao diện xem lịch sử mua hàng của khách hàng trong admin.....	77
Hình 3.26: Giao diện xem thông tin khách hàng.....	78
Hình 3.27: Giao diện xem số lượng hàng tồn trong kho của admin	78
Hình 3.28: Giao diện xem chi tiết số lượng hàng tồn một sản phẩm của admin	78
Hình 3.29: Giao diện nhập kho của admin	79

Hình 3.30: Giao diện xem thông tin lịch sử nhập kho của admin	79
Hình 3.31: Giao diện xem chi tiết nhập kho của admin	79
Hình 3.32: Giao diện xem danh sách MessageBot của admin	80
Hình 3.33: Giao diện xem thông tin chi tiết của MessageBot của admin	80
Hình 3.34: Giao diện thêm MessageBot của admin	80
Hình 3.35: Giao diện chỉnh sửa MessageBot của admin	81
Hình 3.36: Giao diện xóa một MessageBot của admin	81
Hình 3.37: Giao diện chức năng chatbot của admin	82
Hình 3.38: Giao diện xem danh sách tất cả đơn hàng của admin	82
Hình 3.39: Giao diện xem thông tin chi tiết đơn hàng của admin.....	83
Hình 3.40: Giao diện xem danh sách các đơn hàng chưa duyệt của admin	83
Hình 3.41: Giao diện xem đánh giá sản phẩm từ khách hàng của admin	83
Hình 3.42: Giao diện thống kê theo tháng của admin	84
Hình 3.43: Giao diện thống kê theo năm của admin	84
Hình 3.44: Biểu đồ cột thống kê doanh thu các tháng trong năm của admin	84
Hình 3.45: Biểu đồ đường thống kê doanh thu các tháng trong năm của admin	85
Hình 3.46: Giao diện tổng quan của Nhân viên	85
Hình 3.47: Giao diện đăng nhập của nhân viên	86
Hình 3.48: Giao diện danh mục sản phẩm của nhân viên	86
Hình 3.49: Giao diện xem danh sách sản phẩm của nhân viên.....	87
Hình 3.50: Giao diện xem thông tin sản phẩm của nhân viên.....	87
Hình 3.51: Giao diện trả về danh sách sản phẩm sau khi tìm kiếm của nhân viên	87
Hình 3.52: Giao diện xem danh sách khuyến mãi của nhân viên	88
Hình 3.53: Giao diện xem danh sách khách hàng của nhân viên.....	88
Hình 3.54: Giao diện xem thông tin khách hàng của nhân viên	88
Hình 3.55: Giao diện xem lịch sử mua hàng của khách hàng trong nhân viên	89
Hình 3.56: Giao diện xem số lượng hàng tồn của nhân viên	89
Hình 3.57: Giao diện nhập kho của nhân viên	90
Hình 3.58: Giao diện xem lịch sử nhập kho của nhân viên.....	90
Hình 3.59: Giao diện xem thông tin chi tiết nhập kho của nhân viên	90
Hình 3.60: Giao diện chức năng chatbot của nhân viên.....	91
Hình 3.61: Giao diện xem danh sách đơn hàng của nhân viên	92

Hình 3.62: Giao diện xem danh sách các đơn hàng chưa duyệt của nhân viên	92
Hình 3.63: Giao diện xem thông tin chi tiết đơn hàng của nhân viên	92
Hình 3.64: Giao diện xem phản hồi đánh giá sản phẩm của nhân viên	93
Hình 3.65: Giao diện trang chủ của khách hàng	93
Hình 3.66: Giao diện chức năng đăng ký của khách hàng	94
Hình 3.67: Giao diện thông báo đã gửi email xác thực của khách hàng	94
Hình 3.68: Giao diện mail xác thực địa chỉ email đăng ký	94
Hình 3.69: Giao diện thông báo xác thực email thành công	95
Hình 3.70: Giao diện đăng nhập của khách hàng	95
Hình 3.71: Giao diện đăng nhập qua Google	96
Hình 3.72: Giao diện đăng nhập qua Facebook	96
Hình 3.73: Giao diện chức năng quên mật khẩu của khách hàng	97
Hình 3.74: Giao diện thông báo gửi mail đặt lại mật khẩu của khách hàng	97
Hình 3.75: Giao diện mail đặt lại mật khẩu của khách hàng	98
Hình 3.76: Giao diện đặt lại mật khẩu của khách hàng	98
Hình 3.77: Giao diện thông báo đặt lại mật khẩu thành công của khách hàng	98
Hình 3.78: Giao diện xem và chỉnh sửa thông tin của khách hàng	99
Hình 3.79: Giao diện xem và chỉnh sửa địa chỉ của khách hàng	99
Hình 3.80: Giao diện đổi mật khẩu của khách hàng	100
Hình 3.81: Giao diện đổi mật khẩu của khách hàng	100
Hình 3.82: Giao diện sau khi rê chuột vào một sản phẩm của khách hàng	101
Hình 3.83: Giao diện chi tiết sản phẩm của khách hàng	101
Hình 3.84: Giao diện thông báo thêm vào giỏ hàng thành công	102
Hình 3.85: Giao diện hiển thị sản phẩm trong giỏ của khách hàng	102
Hình 3.86: Giao diện cập nhật sản phẩm trong giỏ hàng	102
Hình 3.87: Giao diện đặt hàng của khách hàng	103
Hình 3.88: Giao diện thông báo đặt hàng thành công	103
Hình 3.89: Giao diện đăng nhập thanh toán PayPal	104
Hình 3.90: Giao diện thanh toán trên PayPal	104
Hình 3.91: Giao diện phản hồi và đánh giá sản phẩm của khách hàng	105
Hình 3.92: Giao diện xem đánh giá sản phẩm của khách hàng	105
Hình 3.93: Giao diện lịch sử mua hàng của khách hàng	105

Hình 3.94: Giao diện chức năng chatbot của khách hàng	106
Hình 4.1: Các thành phần trong mô hình MVC và sự tương tác giữa chúng (nguồn: [4])	111
Hình 4.2: Mô hình MVC trong dự án ASP.NET Core MVC.....	111

DANH MỤC BẢNG

Bảng 3.1: Đặc tả usecase đăng nhập	55
Bảng 3.2: Đặc tả usecase đăng ký	55
Bảng 3.3: Đặc tả usecase đặt hàng	56
Bảng 3.4: Đặc tả usecase tra cứu	57
Bảng 3.5: Đặc tả usecase phản hồi đánh giá	57
Bảng 3.6: Đặc tả usecase quản lý giỏ hàng	59
Bảng 3.7: Đặc tả usecase quản lý nhân viên	60
Bảng 3.8: Đặc tả usecase quản lý sản phẩm	61
Bảng 3.9: Đặc tả usecase thống kê	61
Bảng 3.10: Đặc tả usecase quản lý kho	62
Bảng 3.11: Đặc tả usecase quản lý khách hàng	63
Bảng 3.12: Đặc tả usecase xem thông tin đặt hàng	64
Bảng 3.13: Đặc tả usecase chatbot	65
Bảng 4.1: Đặc tả usecase xem thông tin sản phẩm của khách hàng	112
Bảng 4.2: Đặc tả usecase lịch sử mua hàng	112
Bảng 4.3: Đặc tả usecase xem thông tin phản hồi	113
Bảng 4.4: Đặc tả usecase tìm kiếm	113
Bảng 4.5: Đặc tả usecase quản lý tài khoản	114
Bảng 4.6: Đặc tả usecase quản lý khuyến mãi	115
Bảng 4.7: Đặc tả usecase quên mật khẩu	116

MỞ ĐẦU

Trong điều kiện nền kinh tế ngày càng phát triển, đời sống của con người ngày một nâng cao thì các nhu cầu, dịch vụ của con người cũng ngày một tăng. Cùng với đó, mạng Internet cũng đang ngày một phát triển và phổ biến với người người nhà nhà, dẫn đến các hoạt động giải trí, sinh hoạt, học tập ít nhiều cũng có sự ảnh hưởng và thay đổi. Điều đó với sự thay đổi đó, việc kinh doanh, mua bán của các doanh nghiệp, cửa hàng cũng có những thay đổi rõ rệt nhất định. Kể đến là sự đa dạng hóa về hình thức kinh doanh, mua bán khi có sự góp mặt của mạng Internet. Từ kinh doanh, mua bán trực tiếp theo kiểu truyền thống đến kinh doanh, mua bán online. Việc kinh doanh online giúp cho những doanh nghiệp, cửa hàng tiết kiệm được nhiều chi phí, phải kể đến là chi phí thuê mặt bằng. Đối với những cửa hàng kinh doanh vừa và nhỏ, kinh phí thấp thì đây là một lợi thế vô cùng lớn khi họ có thể tiết kiệm được một nguồn chi phí khá lớn cho việc thuê mặt bằng. Bên cạnh đó, việc kinh doanh online sẽ giúp nhiều khách hàng biết đến các doanh nghiệp, cửa hàng và việc tiếp cận với khách hàng ở nhiều nơi khác nhau sẽ trở nên dễ dàng hơn. Ngoài ra, kinh doanh online cũng không bó buộc về thời gian như kinh doanh trực tiếp theo kiểu truyền thống như trước đây. Thông thường, một cửa hàng sẽ mở cửa vào lúc 7 hoặc 8 giờ sáng và đóng cửa lúc 10 giờ tối. Nếu mua hàng trực tiếp tại cửa hàng thì khách hàng chỉ có thể đi đến cửa hàng trong khoảng thời gian này. Điều này khá bất tiện cho những ai bận rộn, không có thời gian mua sắm hoặc với những khách hàng ở xa. Nhưng với việc mua hàng online, khách hàng có thể mua hàng mọi lúc, mọi nơi mà không cần quan ngại về vấn đề thời gian hay khoảng cách địa lý, chỉ cần những thao tác đơn giản là bạn có thể đặt một hay nhiều món hàng mà bạn cần mua, đặc biệt với những ai có công việc bận rộn hoặc ở xa thì đây là một cách mua sắm vô cùng tiện lợi. Nhờ đó, các cửa hàng hoặc doanh nghiệp không những có thêm nguồn lợi nhuận ngoài giờ bán trực tiếp mà còn tăng thêm lượng khách hàng từ nhiều nơi khác nhau, giúp cho việc kinh doanh phát triển, doanh thu và lợi nhuận của các cửa hàng và doanh nghiệp được tăng cao. Đồng thời, việc kinh doanh online cũng bắt nhịp được tâm lý mua hàng của các khách hàng hiện nay khi nhiều khách hàng không muốn tốn quá nhiều thời gian cho việc mua sắm cũng như đi lại mà chỉ cần ngồi tại nhà lựa chọn những sản phẩm cần mua và với vài cú nhấp chuột hay nhấp chọn là có thể lên đơn hàng.

Ngày nay hình thức kinh doanh online đã không còn quá xa lạ, chúng ta có thể bắt gặp trên các nền tảng mạng xã hội như Facebook, Instagram, Zalo, TikTok,... Dù vậy, việc kinh doanh online qua website vẫn luôn được các cửa hàng và doanh nghiệp ưu ái hơn cả. Song song với sự tiện lợi cũng có một vài bất cập, mà đáng quan tâm nhất là tính bảo mật người dùng. Trước những nhu cầu đó, em đã thực hiện đề tài Xây dựng website bán quần áo nữ online của thương hiệu TracyShop với ngôn ngữ C#, công nghệ sử dụng là ASP.NET Core MVC, sử dụng hệ thống chứng thực phân quyền ASP.NET Core Identity, xác thực email người dùng đăng ký qua Mailtrap và sử dụng cơ sở dữ liệu SQL Server để lưu trữ.

Chương 1. TỔNG QUAN ĐỀ TÀI

Chương này nhằm để trình bày tóm tắt lý do chọn đề tài và bối cảnh của đề tài, giúp mọi người có cái nhìn tổng quan hơn về đề tài.

1.1. Lý do chọn đề tài

Ngày nay, chất lượng cuộc sống được nâng cao kéo theo đó là nhu cầu về cái đẹp của con người cũng ngày một thay đổi. Vẻ đẹp áy không chỉ dừng lại ở nội tâm bên trong mà còn là vẻ đẹp về hình thức bên ngoài. Do đó, con người cũng dần quan tâm đến phong cách ăn mặc và vấn đề làm đẹp hơn. Đặc biệt đối với phái nữ, việc sắm cho mình những bộ cánh phù hợp với túi tiền nhưng lại không kém phần thời thượng luôn được hội chị em săn đón và lựa chọn. Việc khoác lên mình những bộ trang phục đẹp cũng phần nào giúp cho các bạn nữ thêm phần tự tin và tỏa sáng. Hiểu được các nhu cầu đó, các thương hiệu thời trang nữ đã dần mọc lên để đáp ứng nhu cầu của các khách hàng nữ. Với tiêu chí rẻ - đẹp - chất lượng, các thương hiệu thời trang đã cho ra đời những mẫu mã đa dạng, giá thành phải chăng nhưng chất lượng vẫn được đảm bảo. Vì vậy, em chọn đề tài “Xây dựng website bán quần áo nữ online TracyShop” để đưa các sản phẩm của các thương hiệu thời trang đến gần hơn với khách hàng và giúp cho việc mua sắm trở nên một cách dễ dàng mà không cần lo ngại về thời gian và khoảng cách địa lý.

1.2. Bối cảnh đề tài

Nội dung đề tài được chia làm 4 chương:

- Chương 1: Tổng quan về đề tài: nêu lý do chọn đề tài và bối cảnh của đề tài.
- Chương 2: Cơ sở lý thuyết: trình bày các khái niệm cũng như các lý thuyết liên quan đến công nghệ được sử dụng trong đề tài.
- Chương 3: Website bán quần áo nữ online TracyShop: giới thiệu đề tài, trình bày phân tích thiết kế và thực nghiệm hệ thống.
- Chương 4: Kết luận và hướng phát triển của đề tài.

Chương 2. CƠ SỞ LÝ THUYẾT

Chương này nhằm trình bày các lý thuyết liên quan đến các công nghệ được sử dụng trong đề tài, giúp mọi người hiểu rõ hơn các kiến thức và nền tảng cơ bản để xây dựng nên đề tài.

2.1. Tổng quan về ASP.NET Core MVC

2.1.1. ASP.NET Core MVC là gì?

ASP.NET Core MVC là một framework mã nguồn mở, nhẹ, có khả năng tối ưu hóa hiệu năng ứng dụng với ASP.NET Core. Nó cung cấp những tính năng dựa vào mô hình xây dựng các trang website động cho phép tách biệt các khối lệnh một cách rõ ràng [4].

2.1.2. Routing

2.1.2.1. Routing là gì?

Routing là quá trình mà khi các URL request được gửi đến, ASP.NET Core sẽ tiến hành xem xét và “chỉ dẫn” cho các URL request đó đường đi đến Controller Action được chọn [5]. Ngoài ra, nó cũng được dùng để tạo một URL output.

2.1.2.2. Nhiệm vụ

Nhiệm vụ của Routing là ánh xạ request được gửi đến vào Controller Action và tạo một URL output tương ứng với Controller Action [5].

Một cách dễ hiểu, khi một request đến thông qua cơ chế Routing, nó sẽ chọn Controller và Action tương ứng để xử lý request đó. Nếu cần thiết, nó sẽ tạo ra một routing output. Nếu trường hợp trả về là một RedirectToAction, output sẽ là một redirect URL tương ứng để chuyển hướng đến URL đó.

2.1.2.3. Convention – based routing

Convention – based routing [4] là các routing cấu hình toàn cục các định dạng URL trong Configure của file cấu hình dự án Startup.cs. Mỗi URL được ánh xạ đến một Action cụ thể của một Controller nhất định. Khi request đến được nhận, routing engine sẽ phân tích URL và xem nó khớp với định dạng URL nào đã được xác định, sau đó gọi Action của Controller tương ứng.

```

app.UseEndpoints(endpoints =>
{
    endpoints.MapControllerRoute(
        name: "default",
        pattern: "{controller=Home}/{action=Index}/{id?}");
})

```

2.1.2.4. Attribute routing

Attribute routing [4] là các routing được tạo riêng cho từng Action và các routing này được đặt trước các Action tương ứng thông qua các attribute [Route].

```

[Route("/about", Name = "about")]
public IActionResult About()
{
    return View();
}

```

2.1.3. Model binding

2.1.3.1. Model binding là gì?

Model binding là quá trình truyền dữ liệu từ View sang Controller thông qua Http Request và đi vào các tham số của Action Method ở trong Controller. Các tham số này có thể có kiểu dữ liệu objects hoặc các kiểu dữ liệu phức tạp [6]. Đây là một ví dụ về việc chuyển dữ liệu của thuộc tính Id, Name từ View sang Controller bằng phương thức Post sử dụng [HttpPost] và [Bind("Id,Name")]:

```

[HttpPost]
public async Task<IActionResult> Create([Bind("Id,Name")] Category
category)
{
    _context.Add(category);
    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}

```

Http Request có thể chứa các dữ liệu ở rất nhiều các định dạng khác nhau như route value, query string. Các dữ liệu này có thể được chứa trong Html Form.

2.1.3.2. Lấy dữ liệu từ Form Data trong Controller

Để lấy dữ liệu từ Form Data trong Controller, trước hết, ta cần có một lớp chứa các giá trị của dữ liệu được lấy từ form sau khi đã tạo hoặc chỉnh sửa. Khi người dùng nhấn vào button để gửi dữ liệu này đi thì dữ liệu này sẽ tự động được truyền đến lớp chứa các giá trị của dữ liệu trong form đó. Form sẽ gửi dữ liệu này đến Action xử lý tương ứng trong Controller.

2.1.4. Model validation

2.1.4.1. Giới thiệu về Model validation

Đa số các dữ liệu được nhập từ người dùng đều tiềm ẩn nguy cơ không hợp lệ, do đó không thể thiết lập trực tiếp vào cơ sở dữ liệu. Các thông tin này có thể chứa những lỗi vô ý hoặc cố ý như sai lỗi chính tả hoặc lỗi cố ý làm sai lệch dữ liệu hay bất cứ lỗi gì của người dùng làm cho dữ liệu không được chuẩn mực. Để ngăn chặn việc này, ASP.NET Core cung cấp cơ chế Model validation và thành phần thực hiện cơ chế này được gọi là Model validator [7], nó giúp chúng ta kiểm tra sự hợp lệ của dữ liệu trước khi thiết lập xuống cơ sở dữ liệu.

Có hai mức kiểm tra trong Model validation:

- Client – side validation (mức client): Khi request được gửi lên server, Client – side validation sẽ chặn request ở mức trình duyệt, giúp dữ liệu được hợp lệ ngay trên trình duyệt mà không cần gửi đến server. Do đó, nó giúp tiết kiệm được băng thông và tăng sự trải nghiệm cho người dùng. Đồng thời, nó cũng giúp được server giảm được một phần công việc và giúp chúng ta tiết kiệm được thời gian.
- Server – side validation (mức server): Tuy Client – side validation giúp trải nghiệm người dùng tốt hơn nhưng nó không tin cậy. Trong một số trường hợp, có những lúc trình duyệt vô tình hay cố ý vô hiệu hóa các javascript validation, làm các dữ liệu đẩy lên server có thể không đúng. Để ngăn chặn việc dữ liệu đưa vào hệ thống không hợp lệ, ta cần kiểm tra tính hợp lệ của dữ liệu ở mức server. Server – side validation đảm bảo được dữ liệu không hợp lệ sẽ không được đưa vào server. Do đó, nó có độ tin cậy cao và an toàn hơn. Tuy nhiên,

việc kiểm tra dữ liệu ở Server – side validation sẽ làm giảm trải nghiệm của người dùng, tiêu tốn băng thông và thời gian vì nó phải gửi request lên server.

Trong thực tế, người ta thường kết hợp Client – side validation và Server – side validation, tức là những dữ liệu không hợp lệ ở phía client bỏ qua thì đến server sẽ được kiểm tra tính hợp lệ một lần nữa. Qua đó ta có thể tận dụng được các ưu điểm và khắc phục các nhược điểm của đôi bên, làm cho hệ thống tối ưu hơn, vừa đảm bảo tính trải nghiệm của người dùng vừa đảm bảo dữ liệu đưa lên hệ thống được an toàn.

2.1.4.2. Validation Attributes

Trong ASP.NET Core có các Validation Attributes viết sẵn, ta chỉ cần đặt nó trước các thuộc tính của Model hoặc ViewModel để thiết lập các ràng buộc cho chúng. Các quy tắc này cũng được gọi là **DataAnnotations**, nó nằm trong namespace **System.ComponentModel.DataAnnotations**.

```
[Required(ErrorMessage = "Vui lòng điền số điện thoại")]
public string PhoneNumber { get; set; }
```

2.1.4.3. Validation Tag Helpers

Dùng Validation Tag Helper [7] để thông báo các lỗi nhập liệu cho người dùng trên View. Với `asp-validation-summary="All"` sẽ hiển thị các thông báo lỗi của riêng Model validation. Các lỗi này sẽ được đặt trong thẻ div và hiển thị ở phía trên cùng của form.

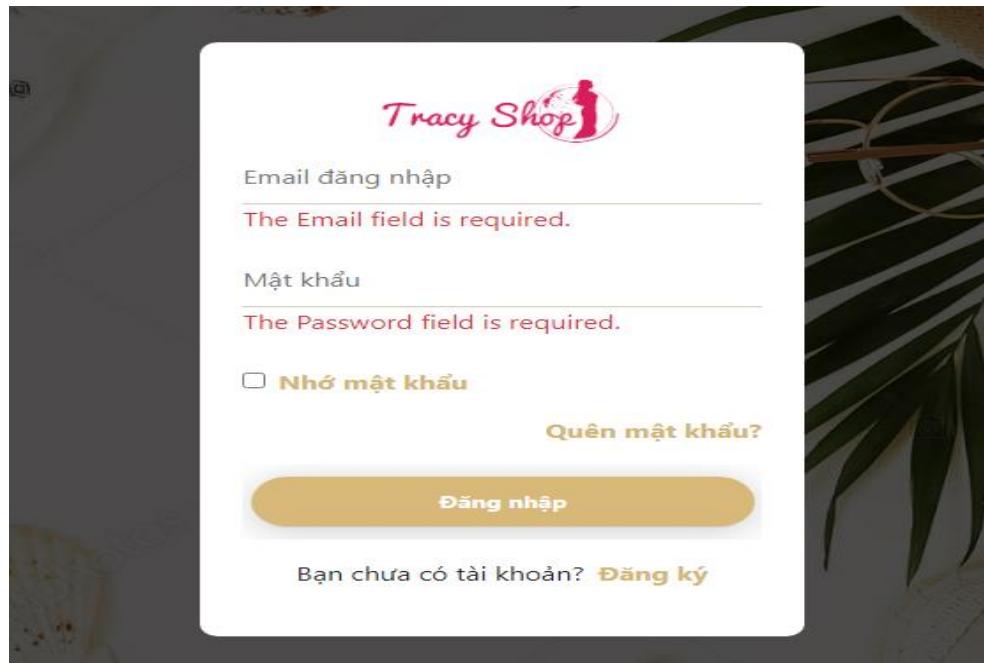
```
<div asp-validation-summary="All" class="text-danger"></div>
```



Hình 2.1: Hiển thị lỗi cho người dùng khi sử dụng `asp-validation-summary`

Với `asp-validation-for` sẽ hiển thị các lỗi của thuộc tính ở trên view. Các lỗi này được đặt trong thẻ `span` và hiển thị bên cạnh thuộc tính.

```
<div class="form-group form-input">
    <input asp-for="Email" type="text" placeholder="Email đăng nhập"/>
    <span asp-validation-for="Email" class="text-danger"></span>
</div>
```



Hình 2.2: Hiển thị lỗi cho người dùng khi sử dụng `asp-validation-for`

2.1.5. Dependency Injection (DI)

2.1.5.1. Giới thiệu về Dependency Injection (DI)

Trong thực tế, một hệ thống không chỉ dừng lại với những chức năng đã được thiết lập lúc ban đầu khi hệ thống được tạo ra. Các chức năng của hệ thống luôn phải được cập nhật và thay đổi tùy theo nhu cầu của con người. Điển hình là các ứng dụng như Facebook, Google,... luôn đưa ra những bản cập nhật mới để cập nhật các tính năng và khắc phục sự cố nhằm tăng tính trải nghiệm và đáp ứng nhu cầu của người dùng trên thế giới. Do đó, việc bảo trì và mở rộng hệ thống luôn được các nhà lập trình quan tâm và họ luôn tìm mọi cách để giúp việc bảo trì và mở rộng code được hiệu quả.

Với một hệ thống, tùy vào tư duy lập trình của các lập trình viên lúc viết code nên các hàm trong hệ thống luôn tiềm ẩn rủi ro làm giảm tính bảo trì và mở rộng. Ví dụ một chức năng Duyệt đơn hàng của khách hàng được viết như sau:

```

public class Cart
{
    public void CheckOrder(int orderId, int userId)
    {
        AppData db = new AppData();
        db.Save(orderId);
        Logger log = new Logger();
        log.LogInfo("Order has been checked!");
        EmailService es = new EmailService();
        es.SendEmail(userId);
    }
}

```

Nhìn chung, đoạn code trên có vẻ không có gì sai và việc viết code rất nhanh. Tuy nhiên, với cách làm này thì có thể sẽ dẫn đến một số vấn đề trong tương lai [8]:

- Hàm CheckOrder này rất khó trong việc kiểm tra vì nó liên quan đến hai module là AppData và EmailService.
- Nếu thay đổi AppData hoặc EmailService hoặc cả hai thì có thể phải sửa lại tất cả các code khởi tạo và gọi hai module này trong CheckOrder. Điều đó gây mất thời gian và rất dễ xảy ra lỗi.
- Về thời gian lâu dài, code sẽ có tính kết dính. Những module có tính kết dính cao, khi thay đổi một module bất kỳ sẽ dẫn đến hàng loạt các thay đổi khác. Điều này rất bất lợi cho việc bảo trì và mở rộng.

Do đó, việc Dependency Injection (DI) ra đời đã khắc phục được các vấn đề trên. Nó giúp linh hoạt hơn trong việc tổ chức code, dễ dàng kiểm tra và viết Unit Test và giúp cho việc bảo trì và mở rộng code dễ dàng hơn.

Hiện nay, Dependency Injection (DI) đã trở thành một phần trong ASP.NET Core.

2.1.5.2. Dependency Injection là gì?

Dependency Injection (viết tắt là DI) là một kĩ thuật trong đó một đối tượng có thể nhận được các đối tượng khác truyền vào nó khi nó cần đến. Đặc điểm chính của DI là làm giảm sự phụ thuộc giữa các đối tượng, giữa các module với nhau.

```

@inject TracyShop.Data.AppDbContext _context
<!DOCTYPE html>
<html lang= "en">
<head>
    <title>@_context.User.Name</title>
</head>
<body>
    <h1>@_context.User.Name</h1>
</body>
</html>

```

2.1.5.3. DI Container

DI Container là một đối tượng có trách nhiệm tạo mới các dependencies (các phụ thuộc) và gán nó cho các đối tượng yêu cầu nó [9].

Để DI Container biết được các đối tượng nào được tạo, ta cần đăng ký các đối tượng đó trong phương thức **ConfigureServices** của lớp **Startup**.

```
services.AddScoped<IEmailService, EmailService>();
```

2.1.5.4. Vòng đời trong DI (DI lifetime)

Việc hiểu rõ vòng đời của những services trong DI là rất cần thiết trước khi tiến hành sử dụng chúng. Nếu chúng ta tạo services mà không hiểu được sự khác biệt giữa các mức vòng đời của nó thì có thể dẫn đến việc hệ thống của chúng ta hoạt động thất thường. Ngoài ra, khi một service gửi yêu cầu đến một service khác thông qua DI, việc biết liệu nó nhận được một đối tượng mới hay là một đối tượng đã được tạo là rất quan trọng [9]. Do đó, việc biết được vòng đời của các services trong việc đăng ký chúng là vô cùng cần thiết.

Ta quản lý vòng đời của services trong DI bằng 3 mức độ:

- Transient: một đối tượng mới luôn luôn được tạo khi có yêu cầu.
- Scoped: với Scoped sẽ bị giới hạn so với Transient, một đối tượng mới được tạo sẽ được dùng chung cho tất cả các scope với mỗi scope là một yêu cầu. Trong scope, service sẽ được tái sử dụng lại.
- Singleton: service chỉ được tạo duy nhất một lần.

Vậy vấn đề đặt ra ở đây là nên dùng Transient, Scoped hay Singleton?

Transient service là một cách an toàn nhất dùng để tạo service vì nó luôn tạo một đối tượng mới khi có yêu cầu. Do đó, việc sử dụng Transient service sẽ tốn nhiều tài nguyên và bộ nhớ. Nếu có quá nhiều đối tượng được tạo (hay có quá nhiều yêu cầu) thì sẽ làm giảm hiệu năng của hệ thống. Transient service phù hợp với những service nhỏ và nhẹ, không chứa các trạng thái.

Scoped service là một tùy chọn tốt hơn nếu ta muốn duy trì các trạng thái trong một yêu cầu.

Singleton service chỉ được tạo một lần và nó không hủy cho đến khi hệ thống tắt. Do đó, nó sẽ chiếm một phần bộ nhớ của hệ thống và sau mỗi yêu cầu, bộ nhớ của nó sẽ tăng dần lên cho đến khi bị đầy và hệ thống không hoạt động. Tuy nhiên, Singleton cũng giúp tiết kiệm bộ nhớ nếu trong trường hợp xử lý tốt vì nó có cơ chế sử dụng lại và dùng được ở nhiều nơi.

Tùy vào mục đích sử dụng và kích thước của của từng service đưa vào hệ thống mà ta có những sự lựa chọn khác nhau để hệ thống được tối ưu, tăng trải nghiệm của người dùng và tiết kiệm tài nguyên bộ nhớ.

2.1.5.5. Ưu điểm

- Giảm sự phụ thuộc giữa các đối tượng với nhau.
- Code sạch và giúp dễ đọc hơn.
- Hệ thống dễ bảo trì và mở rộng.
- Dễ dàng cập nhật các triển khai mà không cần có nhiều sự thay đổi về code.

2.1.6. Tag Helpers

2.1.6.1. Tag Helper là gì?

Tag Helper nằm trong thư viện **Microsoft.AspNetCore.Mvc.TagHelpers** của ASP.NET Core [14], là một tính năng giúp thêm các đoạn code xử lý trên server dưới dạng giống với HTML, giúp nhinnie thân thiện với code HTML hơn. Các code này được xử lý bởi Razor engine, tạo ra các đoạn code HTML và gen lên trên trình duyệt. Ví dụ, ta dùng Form Tag Helper với asp-action:

```
<form asp-action="ForgotPassword">
```

Khi gen lên trình duyệt sẽ chuyển Form Tag Helper thành thẻ `<form>` và `asp-action`

thành action của <form> trên HTML:

```
<form action="ForgotPassword" method="post">
```

Nó cũng thao tác với các HTML elements để tạo ra các HTML elements mới [15]. Chúng ta cũng có thể sử dụng thêm, sửa, xóa để tạo ra.

Để thêm Tag Helper vào một dự án, ta sử dụng @addTagHelper. Mặc định, khi tạo một dự án ASP.NET Core MVC thì nó sẽ tự động thêm các Tag Helper vào trong layout _ViewImport.cshtml có sẵn của dự án.

```
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
```

2.1.6.2. Form Tag Helper

Form Tag Helper sử dụng thẻ <form> để bao bọc, và sử dụng một số thuộc tính như asp-controller, asp-action, asp-area giúp tạo ra các HTML trên trình duyệt.

```
<form asp-action="CheckedOrder" asp-controller="Order" asp-area="Admin" method="post" asp-route-id="@item.Id">
```

Đoạn code trên sau khi đưa lên trình duyệt:

```
<form method="post" action="/admin/order/checkedorder/2">
    <input name="__RequestVerificationToken" type="hidden" value="CfDJ8BfCWMX6ePhCgmcIn5vf_6pBUkL8utMuXX5QEfFpy58GtePHPJOI-4bz0aeQlrHf4ejYr58ivUtaBy-W5tXjVMea_8JGUy4sNtxwCi2M_ronFv6TDS2gNz8L1rcOc8aTyzL1RxTFeKILCCMbSGzaSgZ93MxJV3YjY6fx4M9pbASTwvVJVnqH5S4kI2xw0ncR9w">
</form>
```

2.1.6.3. Label Tag Helper

Label Tag Helper sử dụng thẻ <label> để bao bọc và nó sử dụng thuộc tính asp-for giúp tạo các HTML trên trình duyệt.

```
<label asp-for="Name" class="control-label">Họ và tên</label>
```

Đoạn code trên sau khi đưa lên trình duyệt:

```
<label class="control-label" for="Name">Họ và tên</label>
```

2.1.6.4. Input Tag Helper

Input Tag Helper sử dụng thẻ <input> và nó tạo ra một thẻ <input> tương ứng trong HTML, thường nó dùng để chuyên dữ liệu của một thuộc tính trong Model. Tương tự

label, nó cũng sử dụng asp-for để tạo các HTML trên trình duyệt.

```
<input asp-for="Name" class="form-control" type="text" />
```

Đoạn code trên sau khi đưa lên trình duyệt:

```
<input class="form-control" type="text" id="Name" name="Name" value="Uyên Nguyễn">
```

2.1.6.5. Lợi ích của Tag Helper

- Thân thiện với cú pháp HTML: Tag Helper có cú pháp giống với HTML giúp dễ dàng tương tác và sử dụng và cách sử dụng tương tự như HTML. Đồng thời, ta cũng có thể sử dụng thêm CSS để gán cho các Tag Helper như trong HTML để giao diện được đẹp hơn.
- Code sạch hơn: code dễ nhìn và sạch sẽ, không cần phải dùng ký tự “@” để chuyển đổi code C# nên giống như ta đang viết HTML.
- Dễ dàng tạo View hiển thị trên trình duyệt với những Tag Helper có sẵn. Ngoài ra ta có thể tự tạo một Tag Helper nếu nó không có sẵn trong thư viện.

2.1.7. Migration

Giống như những nền tảng khác như Django, Laravel, Ruby,... trong ASP.NET Core cũng hỗ trợ Migration trong mô hình Code First.

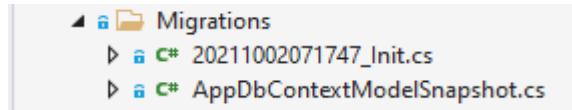
Migration là một kĩ thuật tương tác với cơ sở dữ liệu trong mô hình Code First. Nó giúp thay đổi cấu trúc cơ sở dữ liệu dưới cơ sở dữ liệu khi ta thay đổi cấu trúc cơ sở dữ liệu trong code ở Models, đảm bảo dữ liệu không bị mất và mỗi lần migrate nó sẽ lưu lại lịch sử cập nhật xuống cơ sở dữ liệu. Việc tạo và cập nhật các migrations được thực hiện trong Terminal.

Để migrate xuống cơ sở dữ liệu, ta thực hiện lệnh sau trong Terminal:

```
dotnet ef migrations Init
```

với Init là tên migrations, tên này ta tự đặt.

Sau khi migrations, trong cấu hình dự án sẽ xuất hiện một thư mục Migrations. Trong thư mục này sẽ sinh ra các file .cs.



Hình 2.3: Thư mục Migrations trong dự án

File AppDbContextModelSnapshot.cs sẽ chứa các cấu trúc dữ liệu của tại thời điểm ta tạo migrations đầu tiên. Mỗi lần migrations sẽ tự động tạo ra các file tương tự file 20211002071747_Init.cs. Dãy số đầu tiên trong tên của mỗi file được sinh ra bằng cách random ngẫu nhiên khi ta tạo migrations và nó sẽ gán vào đầu của mỗi tên file. Những file này sẽ lưu các cập nhật tại các thời điểm ta thay đổi code trong Models.

Để biết được trong dự án có những migrations nào, ta sử dụng lệnh:

```
dotnet ef migrations list
```

Khi chạy lệnh này nó sẽ kiểm tra và hiển thị các danh sách tất cả migrations đã được tạo của dự án trong Terminal.

- Một số lệnh trong Migrations
 - Lệnh xóa một migration cuối cùng được tạo:

```
dotnet ef migrations remove
```

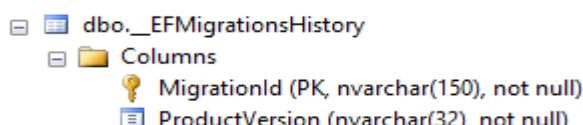
- Lệnh cập nhật tất cả các migrations xuống cơ sở dữ liệu:

```
dotnet ef database update
```

Nếu chỉ cập nhật một migration hoặc muốn quay về một migrations nào đó, ta chỉ cần chỉ tên của migration đó trong câu lệnh cập nhật, ví dụ với migration có tên Init:

```
dotnet ef database update Init
```

Sau khi cập nhật, nó sẽ lưu lại lịch sử cập nhật trong một bảng có tên là _EFMigrationsHistory dưới cơ sở dữ liệu.



Hình 2.4: Các thuộc tính trong bảng _EFMigrationsHistory

Bảng này gồm hai thuộc tính là MigrationId và ProductVersion với thuộc tính MigrationId lưu tên của các migrations và ProductVersion lưu thông tin version sử dụng của các migrations đó.

	MigrationId	ProductVersion
1	20211002071747_Init	5.0.10

Hình 2.5: Dữ liệu trong bảng _EFMigrationsHistory

- Lệnh hiển thị tất cả các câu truy vấn SQL mà nó thực hiện dưới cơ sở dữ liệu từ migration đầu đến migration cuối:

```
dotnet ef migrations script
```

- Lệnh lưu tất cả các truy vấn trong migrations ra file script SQL với tên file là MigrationScript:

```
dotnet ef migrations script -o MigrationScript.sql
```

- Lệnh xóa database:

```
dotnet ef database drop -f
```

2.2. Hệ quản trị cơ sở dữ liệu SQL Server

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ sử dụng các câu lệnh SQL (Transact – SQL) để tương tác với dữ liệu. Nhiều ứng dụng và phần mềm có thể kết nối với máy chủ cơ sở dữ liệu này để tiến hành lưu trữ, truy xuất và quản lý dữ liệu. Nó có thể hoạt động tốt trên môi trường dữ liệu lớn với dung lượng bộ nhớ lên đến tera – byte. SQL Server được xây dựng và phát triển bởi Microsoft [13].

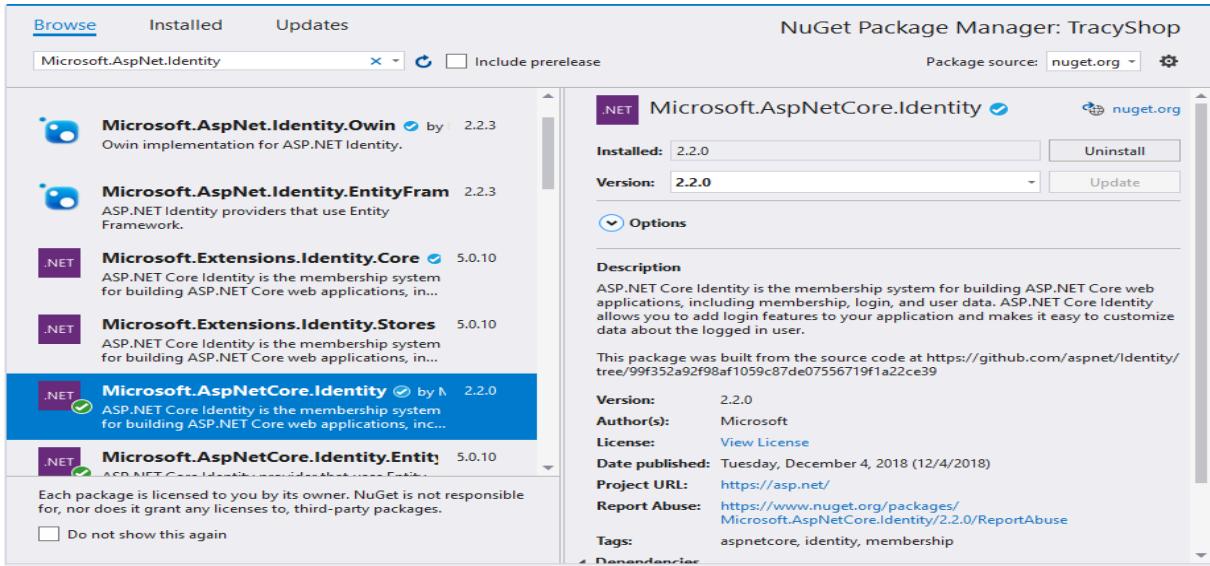
2.3. Đăng nhập qua Facebook và Google với ASP.NET Core Identity

2.3.1. ASP.NET Core Identity

ASP.NET Core Identity là một hệ thống chứng thực phân quyền mạnh mẽ trong ASP.NET Core. Nó giúp chứng thực và phân quyền người dùng và hỗ trợ các chức năng như Đăng ký, Đăng nhập, Đăng xuất trong Windows. Đồng thời, nó cũng hỗ trợ chức năng Đăng nhập từ bên thứ ba như Facebook, Google,...

Để sử dụng ASP.NET Core Identity, ta cần tích hợp các thư viện sau vào dự án bằng cách cài đặt nó trong Manage NuGet Packages:

- Microsoft.AspNetCore.Identity
- Microsoft.AspNetCore.Identity.UI
- Microsoft.AspNetCore.Identity.EntityFrameworkCore
- Microsoft.AspNetCore.Identity.Authentication

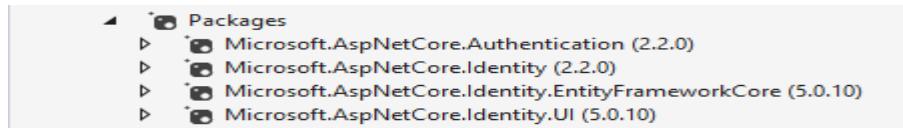


Hình 2.6: Giao diện của Manage NuGet Packages trong Visual Studio 2019

Hoặc ta có thể cài đặt các thư viện này trên Terminal bằng các lệnh sau:

```
dotnet add package Microsoft.AspNetCore.Identity
dotnet add package Microsoft.AspNetCore.Identity.EntityFrameworkCore
dotnet add package Microsoft.AspNetCore.Identity.UI
dotnet add package Microsoft.AspNetCore.Authentication
```

Sau khi cài đặt hai thư viện này, nó sẽ tích hợp vào dự án. Ta kiểm tra trong Dependencies Packages của dự án:



Hình 2.7: Các Packages trong Dependencies

Tiếp đến, ta tạo một lớp AppUser trong Models kế thừa lớp IdentityUser của IdentityDbContext trong ASP.NET Core Identity. Lớp này có sẵn trong thư viện **Microsoft.AspNetCore.Identity** mà ta đã cài đặt lúc nãy và nó đã có sẵn có các thuộc tính của User như Id, Username, Password, Email, PhoneNumber,... Ta chỉ cần thêm một số thuộc tính nếu cần như Name, Birthday,...

```
public class AppUser : IdentityUser
{
    public string Name { set; get; }
    public DateTime? Birthday { set; get; }
}
```

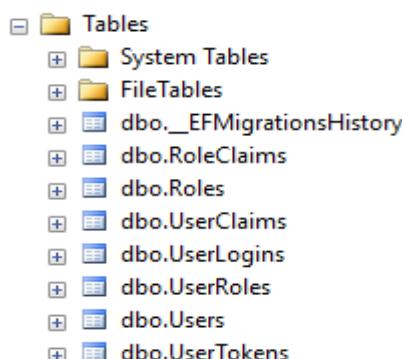
Ngoài lớp IdentityUser, trong IdentityDbContext [10] còn sử dụng các lớp như IdentityRole, IdentityUserRole, IdentityUserClaim, IdentityUserLogin, IdentityUserToken, IdentityRoleClaim. Các lớp này sẽ có trong DbContext của dự án.

DbContext là nơi trung gian, giúp kết nối và tương tác giữa các lớp Models và các bảng dưới cơ sở dữ liệu. Do đó, nó là một lớp vô cùng quan trọng. Khi dự án sử dụng ASP.NET Core Identity, mặc định **IdentityDbContext** là DbContext của dự án và nó đã định nghĩa sẵn các bảng trong IdentityDbContext. Để tạo một DbContext riêng, ta chỉ cần cho nó kế thừa IdentityDbContext.

Khi tạo các bảng dưới cơ sở dữ liệu, mặc định các bảng trong IdentityDbContext sẽ có tên với tiền tố là AspNet như AspNetUser, AspNetUserRoles,... Nếu muốn loại bỏ tiền tố đó khi tạo cơ sở dữ liệu, ta chỉ cần thêm đoạn code sau vào DbContext của dự án:

```
foreach (var entityType in builder.Model.GetEntityTypes())
{
    var tableName = entityType.GetTableName();
    if (tableName.StartsWith("AspNet"))
        entityType.SetTableName(tableName.Substring(6));
}
```

Các bảng sau khi được tạo dưới cơ sở dữ liệu:



Hình 2.8: Các bảng trong IdentityDbContext

Để sử dụng chức năng chứng thực và phân quyền trong ASP.NET Core Identity, ta thêm vào phương thức Configure trong lớp Startup hai dòng sau để sử dụng dịch vụ:

```
app.UseAuthentication();
app.UseAuthorization();
```

Với **UseAuthentication** dùng để xác thực người dùng và **UseAuthorization** dùng để phân quyền người dùng.

2.3.1.1. Xác thực người dùng

Chức năng xác thực người dùng được Identity xử lý rất tốt. Để xác thực thông tin người dùng trước khi truy cập một số chức năng, ta gán trước Action của các chức năng đó một thuộc tính [Authorize], thuộc tính này nằm trong thư viện **Microsoft.AspNetCore.Authorization** [14]. Trước khi truy cập các chức năng đó thì hệ thống sẽ kiểm tra thông tin người dùng đã được xác thực hay chưa. Nếu chưa, người dùng phải đăng nhập mới có thể truy cập được những chức năng này.

```
[Authorize]
public IActionResult Create()
{
    return View();
}
```

Nếu người dùng chưa có tài khoản thì phải tiến hành đăng ký tài khoản mới có thể truy cập. Với chức năng Đăng ký, khi người dùng gửi các thông tin cần đăng ký đến hệ thống, hệ thống sẽ tiến hành kiểm tra người dùng này đã có trong danh sách user chưa. Nếu tồn tại, nó sẽ trả về lỗi. Ngược lại, nó sẽ tiến hành băm mật khẩu của người dùng đăng ký và lưu xuống cơ sở dữ liệu cùng các thông tin khác mà người dùng cung cấp. Đồng thời nó cũng tạo cho người dùng một “tem an ninh” (Security Stamp) và “tem hợp lệ” (Concurrency Stamp) và hai giá trị này được lưu lần lượt trong thuộc tính SecurityStamp và ConcurrencyStamp của bảng User dưới cơ sở dữ liệu. Thuộc tính SecurityStamp được sử dụng cho mục đích bảo mật khi các thông tin quan trọng của người dùng thay đổi, chẳng hạn như thay đổi mật khẩu. Còn thuộc tính ConcurrencyStamp giúp ngăn chặn việc xảy ra xung đột trong việc cập nhật đồng thời thông tin của một người dùng. Qua đó, giúp cho thông tin người dùng được bảo mật hơn.

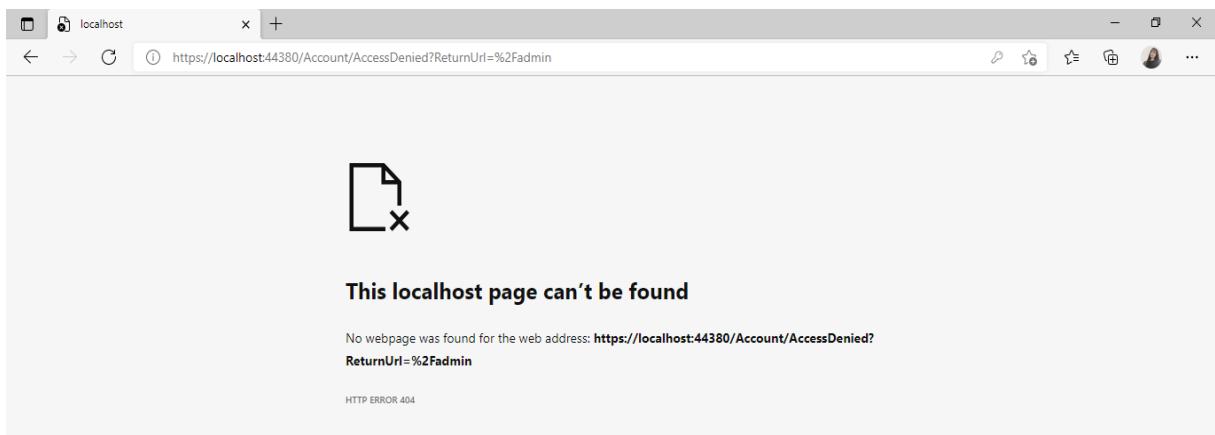
2.3.1.2. Phân quyền người dùng

Hệ thống phân quyền người dùng của Identity vô cùng mạnh mẽ và cách thiết lập nó cũng vô cùng đơn giản. Để phân quyền người dùng trên một Action của một

Controller, ta chỉ cần gán trước Action đó một thuộc tính [Authorize] và gắn thêm Roles cho nó.

```
[Authorize(Roles = "Admin, Employee")]
```

Nếu một người dùng có Roles khác Admin và Employee thì sẽ không cho truy cập và trả về trang HTTP ERROR 404.



Hình 2.9: Trang trả về 404 khi Roles của người dùng không hợp lệ

2.3.1.3. Các cấu hình dịch vụ trong Identity

Identity được biết đến với những tính năng mạnh mẽ trong việc xác thực và phân quyền người dùng. Bên cạnh đó, Identity cũng hỗ trợ những dịch vụ nhằm đảm bảo việc bảo mật thông tin người dùng được chặt chẽ. Những dịch vụ này được cung cấp bởi **IdentityOptions**. Nó nằm trong thư viện **Microsoft.AspNetCore.Identity** [14].

Identity cung cấp các cấu hình dịch vụ như các thiết lập về mật khẩu, lockout user (khóa người dùng), đăng nhập người dùng,... Những cấu hình dịch vụ này được cấu hình trong phương thức **ConfigureServices** trong lớp **Startup** của dự án.

Về mật khẩu, **IdentityOptions** có những thiết lập cụ thể giúp cho mật khẩu của người dùng mạnh mẽ hơn và có tính bảo mật cao, cụ thể là mật khẩu bắt buộc phải có số, chữ thường, ký tự đặc biệt hoặc chữ in hoa,... hay quy định số ký tự tối thiểu của mật khẩu, số ký tự đặc biệt,...

```

services.Configure<IdentityOptions>(options => {
    // Bắt buộc phải có số
    options.Password.RequireDigit = true;
    // Bắt buộc phải có chữ thường
    options.Password.RequireLowercase = true;
    // Bắt buộc có ký tự đặc biệt
    options.Password.RequireNonAlphanumeric = true;
    // Không bắt buộc chữ in
    options.Password.RequireUppercase = false;
    // Số ký tự tối thiểu của password
    options.Password.RequiredLength = 6;
    // Số ký tự riêng biệt
    options.Password.RequiredUniqueChars = 1;
}) ;

```

Trong quá trình đăng nhập sẽ không thể tránh khỏi việc người dùng đăng nhập thất bại nhiều lần do quên mật khẩu hay một người lạ cố tình tấn công tài khoản của một người dùng bằng cách đăng nhập thất bại nhiều lần để dò tìm tên đăng nhập hoặc mật khẩu. Dù nguyên nhân là gì thì nó luôn đe dọa đến việc thông tin người dùng bị đánh cắp. Do đó, Identity đã cung cấp tính năng lockout user (khóa người dùng), cho phép người dùng đăng nhập thất bại trong số lần cho phép.

```

services.Configure<IdentityOptions>(options => {
    // Khóa 5 phút
    options.Lockout.DefaultLockout TimeSpan =
        TimeSpan.FromMinutes(5);
    // Khóa khi đăng nhập thất bại 5 lần
    options.Lockout.MaxFailedAccessAttempts = 5;
    options.Lockout.AllowedForNewUsers = true;
}) ;

```

Identity cũng cung cấp những thiết lập cấu hình cho người dùng như quy định các ký tự dùng để đặt tên người dùng, email của người dùng phải duy nhất,...

```

services.Configure<IdentityOptions>(options => {
    // quy định các ký tự đặt tên user
    options.User.AllowedUserNameCharacters =
        "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ" +
        "-._@+";
    // Email là duy nhất
    options.User.RequireUniqueEmail = true;
}) ;

```

Ngoài ra, để tăng tính bảo mật, Identity cũng cung cấp tính năng xác thực email và số điện thoại để kiểm tra xem thông tin email và số điện thoại người dùng cung cấp có tồn tại hay không.

```

services.Configure<IdentityOptions>(options => {
    // Cấu hình xác thực địa chỉ email (email phải tồn tại)
    options.SignIn.RequireConfirmedEmail = true;
    // Xác thực số điện thoại
    options.SignIn.RequireConfirmedPhoneNumber = false;
}) ;

```

Với những dịch vụ bảo mật người dùng mà Identity cung cấp không những giúp cho việc tăng tính bảo mật thông tin người dùng mà còn giúp cho các lập trình viên không cần phải code quá nhiều, từ đó giúp tiết kiệm thời gian lập trình và kiểm tra code. Việc sử dụng các dịch vụ của Identity cũng vô cùng dễ dàng, chỉ cần đăng ký là có thể sử dụng các dịch vụ của nó. Do đó, nó giúp cho code được sạch sẽ và dễ nhìn hơn, đồng thời cũng giúp cho quá trình kiểm tra và chỉnh sửa được dễ dàng.

2.3.2. Đăng nhập qua Google

Ngoài chức năng xác thực và phân quyền trên Windows, ASP.NET Core Identity còn hỗ trợ đăng nhập qua nhiều bên thứ ba trong đó có Google. Hiện nay, hình thức xác thực qua Google được nhiều trang web hỗ trợ.

2.3.2.1. Cài đặt gói hỗ trợ

Để sử dụng chức năng đăng nhập qua Google, ta cần cài đặt gói hỗ trợ chứng thực qua Facebook bằng cách chạy lệnh sau trong Terminal để tiến hành cài đặt:

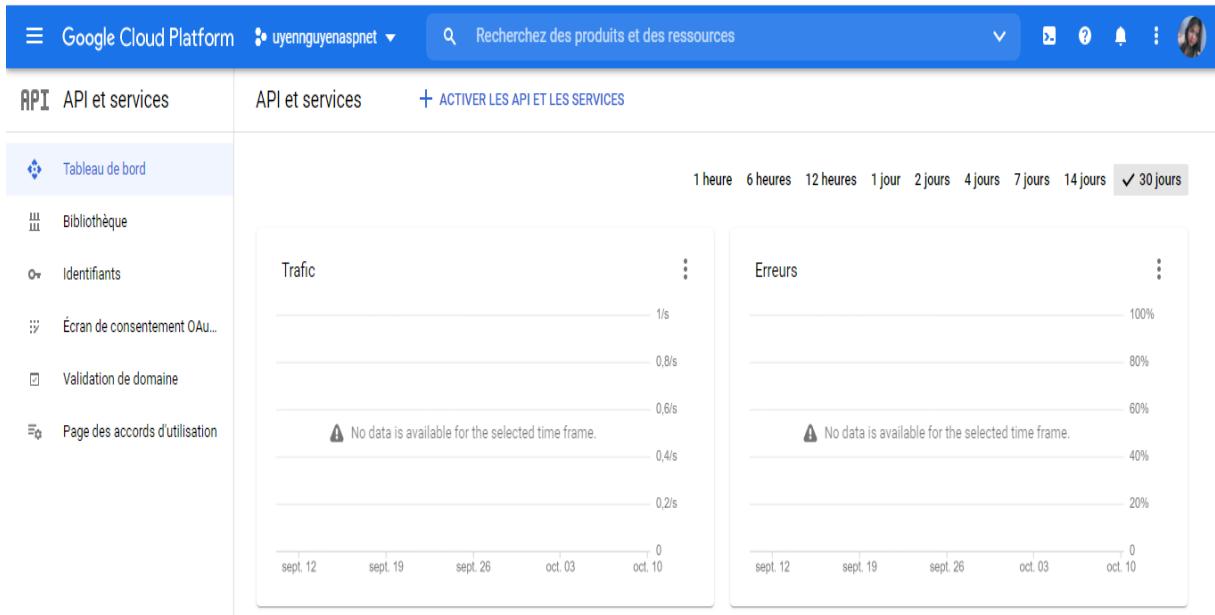
```
dotnet add package Microsoft.AspNetCore.Authentication.Google
```

Ngoài ra, ta cũng có thể cài đặt gói này trong Manage NuGet Packages.

2.3.2.2. Đăng ký dịch vụ từ Google

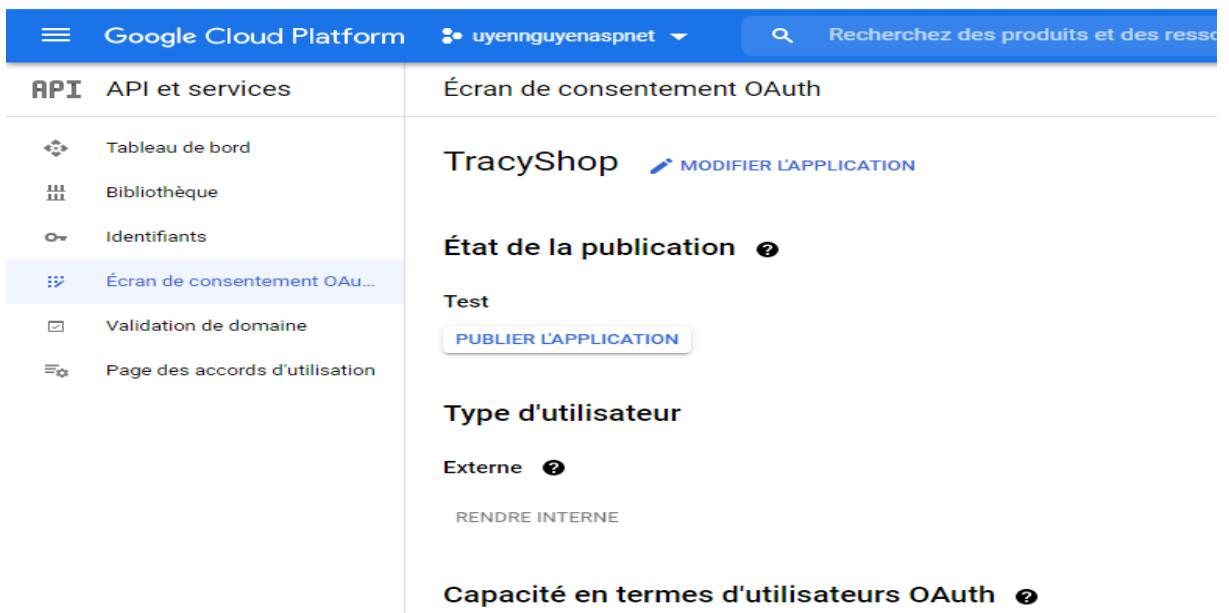
Sau khi cài đặt gói hỗ trợ cho dự án, ta sẽ tiến hành đăng ký dịch vụ từ Google:

- Truy cập vào trang <https://console.developers.google.com/> và bấm vào Create Project để tạo một dự án mới nếu bạn chưa có, giả sử ta tạo một dự án uyennuyenaspnet.



Hình 2.10: Giao diện của trang Google Cloud Platform

- Sau đó, ta tạo một ứng dụng mới, giả sử có tên là TracyShop.



Hình 2.11: Giao diện sau khi tạo xong ứng dụng trong Google Cloud Platform

- Qua tab Identifiants, ta tạo một Oauth Client ID với loại ứng dụng là Web application và tên là TracyShop. Tên này ta có thể tự đặt tùy ý. Tại đây ta điền một URI để nó chuyển đến khi chứng thực.

The screenshot shows the 'Identifiants' tab selected in the sidebar. At the top right, there are buttons for 'ID client pour Application Web' (with a back arrow), 'TÉLÉCHARGER LE FICHIER JSON' (with a download icon), and 'AJOUTER UN URI' (with a plus icon). Below these are sections for 'URI de redirection autorisés' (with a question mark icon) and 'À utiliser avec les requêtes provenant d'un serveur Web'. Two URIs are listed in input fields: 'https://localhost:44380/dang-nhap-tu-google' and 'https://localhost:5001/dang-nhap-tu-google'. There is also a 'URI *' input field with a placeholder 'https://localhost:44380/dang-nhap-tu-google'. At the bottom are 'ENREGISTRER' and 'ANNULER' buttons.

Hình 2.12: Thiết lập URIs

Sau khi hoàn thành xong, Google Cloud Platform sẽ trả về một ID Client và một Secret Client.

ID client	394988863481- ngihq258adtvuo8cf90r96epa8i7dm so.apps.googleusercontent.com
Code secret du client	9bqJqj1sJQ3Q78vF3bVAp-9O
Date de création	29 août 2021 à 08:59:02 GMT+7

Hình 2.13: ID Client và Secret Client của Google

Ta đặt hai giá trị này trong appsettings.json:

```
"Authentication": {
    "Google": {
        "ClientId": "394988863481-  
ngihq258adtvuo8cf90r96epa8i7dmso.apps.googleusercontent.com",
        "ClientSecret": "9bqJqj1sJQ3Q78vF3bVAp-9O"
    }
},
```

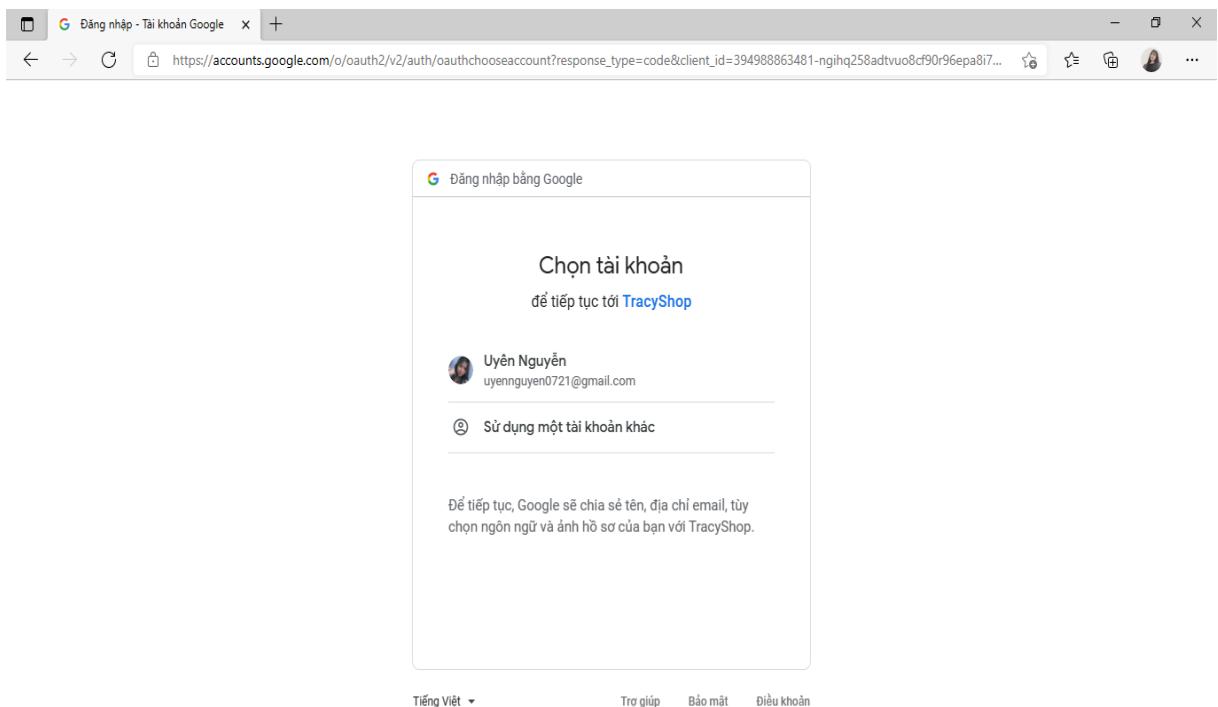
Tiếp đến, ta đăng ký dịch vụ xác thực bằng Google trong phương thức ConfigureServices của lớp Startup:

```

services.AddAuthentication().AddGoogle(googleOptions =>
{
    IConfigurationSection googleAuthNSection =
        Configuration.GetSection("Authentication:Google");
    googleOptions.ClientId = googleAuthNSection["ClientId"];
    googleOptions.ClientSecret =
        googleAuthNSection["ClientSecret"];
    googleOptions.ClaimActions.MapJsonKey("image", "picture");
    googleOptions.CallbackPath = "/dang-nhap-tu-google";
}) ;

```

Nếu các cấu hình cài đặt không gặp vấn đề sai sót gì thì Identity trong dự án đã có thể cho phép người dùng đăng nhập qua Google.



Hình 2.14: Giao diện đăng nhập qua Google

2.3.3. Đăng nhập qua Facebook

Ngoài việc hỗ trợ đăng nhập qua Google, ASP.NET Core còn hỗ trợ đăng nhập qua Facebook. Tương tự như Google, việc xác thực qua Facebook cũng được nhiều trang web hỗ trợ.

2.3.3.1. Cài đặt gói hỗ trợ

Để sử dụng chức năng đăng nhập qua Facebook, ta cần cài đặt gói hỗ trợ chúng thực qua Facebook bằng cách chạy lệnh sau trong Terminal để tiến hành cài đặt:

```
dotnet add package Microsoft.AspNetCore.Authentication.Facebook
```

Ngoài ra, ta cũng có thể cài đặt gói này trong Manage NuGet Packages.

2.3.3.2. Đăng ký dịch vụ từ Facebook

Sau khi cài đặt gói hỗ trợ cho dự án, ta sẽ tiến hành đăng ký dịch vụ từ Facebook. Để đăng ký dịch vụ, ta phải tạo tài khoản phát triển trên Facebook bằng cách truy cập vào địa chỉ <https://developers.facebook.com/apps/> và tạo một ứng dụng, ví dụ một ứng dụng có tên là TracyShop. Tên ứng dụng ta tự đặt tùy ý.

The screenshot shows the 'FACEBOOK for Developers' interface. In the top navigation bar, there are links for Tài liệu, Công cụ, Hỗ trợ, and Ứng dụng của tôi. A search bar is also present. Below the navigation, there is a section titled 'Ứng dụng' (Applications) with a search bar and a 'Tạo ứng dụng' (Create application) button. On the left, there is a filter sidebar labeled 'Lọc theo' (Filter by) with options 'Tất cả ứng dụng (3)' (All applications (3)) and 'Đã lưu trữ' (Saved). The main area displays three application cards: 'TracyShop' (ID: 214999683932773, Type: Người tiêu dùng), 'UyenNTT' (ID: 3015380622016615, Type: Người tiêu dùng), and 'Glorian Restaurant' (ID: 534483551121734, Type: Người tiêu dùng). Each card has a 'Quản trị viên' (Administrator) link and a more options button. At the bottom of the page, there are sections for FACEBOOK, Sản phẩm, Chương trình, and Tin tức.

Hình 2.15: Giao diện của trang Facebook for Developers

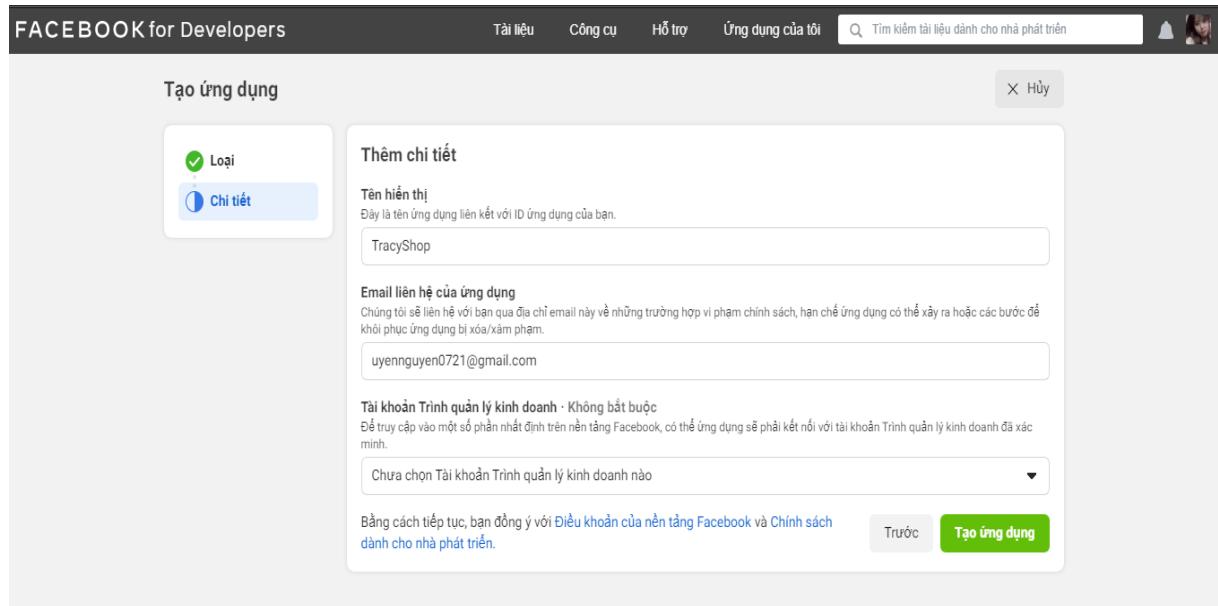
Khi tạo ứng dụng, đầu tiên nó sẽ hiển thị trang để người dùng chọn loại ứng dụng.

The screenshot shows the 'Tạo Ứng dụng' (Create Application) wizard. The first step, 'Chọn loại ứng dụng' (Select application type), is displayed. It shows two options: 'Loại' (Type) and 'Chi tiết' (Details), with 'Loại' selected. Below this, there are five categories with descriptions and icons: 'Kinh doanh' (Business), 'Người tiêu dùng' (Consumer), 'Trò chơi trực thi' (Game), 'Chơi game' (Game), and 'Workplace' (Workplace). Each category has a detailed description below it. A 'Hủy' (Cancel) button is at the top right, and a 'Đóng góp ý kiến' (Share feedback) button is at the bottom right.

Hình 2.16: Giao diện chọn loại ứng dụng trên Facebook for Developers

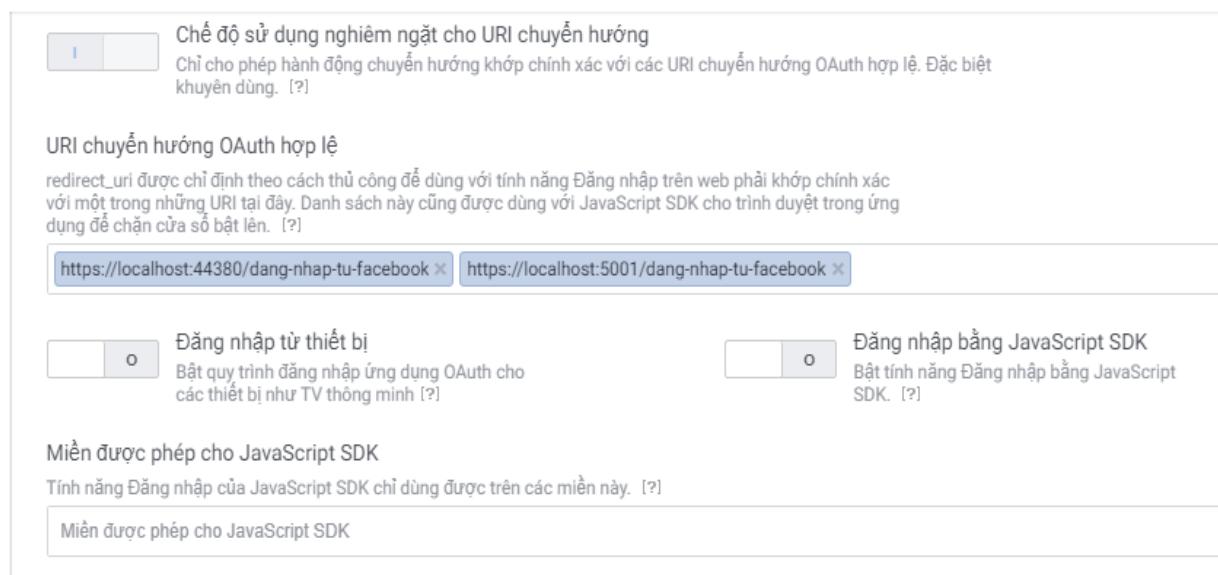
Tùy vào nhu cầu của người dùng mà ta chọn loại ứng dụng phù hợp. Với nhu cầu chỉ dùng để đăng nhập qua Facebook nên ta chọn loại ứng dụng Người tiêu dùng.

Sau khi chọn loại ứng dụng, nó sẽ yêu cầu người dùng điền thông tin của ứng dụng như tên ứng dụng, email liên hệ.



Hình 2.17: Giao diện thêm chi tiết ứng dụng trong Facebook for Developers

Tạo ứng dụng xong, ta thiết lập tính năng đăng nhập bằng Facebook và chọn nền tảng ứng dụng là web. Tiếp theo, ta vào Cài đặt trong mục Đăng nhập bằng Facebook và nhập địa chỉ URL chuyển hướng OAuth hợp lệ với tên miền là tên miền địa chỉ website của dự án đang chạy trên trình duyệt.



Hình 2.18: Nhập URL chuyển hướng trong Facebook for Developers

Cuối cùng, ta vào Cài đặt và chọn mục Thông tin cơ bản để xem giá trị ID của ứng dụng và khóa bí mật.

The screenshot shows the 'Basic' tab of an app's settings. It includes fields for:

- ID ứng dụng: 214999683932773
- Khóa bí mật của ứng dụng: (redacted)
- Tên hiển thị: TracyShop
- Vùng tên: (empty)
- Miền ứng dụng: (empty)
- Email liên hệ: uyennghien0721@gmail.com
- URL chính sách quyền riêng tư: https://facebook.com/nttu.dqt/
- URL Điều khoản dịch vụ: (empty)
- Xóa dữ liệu người dùng: URL gọi lại xóa dữ liệu: https://localhost:44380/delete
- Biểu tượng ứng dụng (1024x1024): An image placeholder.
- BUTTONS: Bỏ (cancel), Lưu thay đổi (Save changes).

Hình 2.19: Các thông tin cơ bản của ứng dụng trên Facebook for Developers
Mặc định, khóa bí mật sẽ không hiển thị. Để xem ta cần nhấp vào hiển thị bên phải khóa bí mật, nó sẽ yêu cầu bạn nhập mật khẩu Facebook của bạn để xác nhận:



Hình 2.20: Nhập mật khẩu Facebook để xem khóa bí mật của ứng dụng
Sau khi xác nhận xong, nó sẽ hiển thị khóa bí mật lên màn hình và khóa bí mật này ta có thể đặt lại nếu cảm thấy không an toàn.

The screenshot shows the 'Basic' tab again. The secret key is now visible in the 'Khóa bí mật của ứng dụng' field: 9d4fe2c81eed7605f1c764611dc13366. There is also a 'Đặt lại' (Reset) button.

Hình 2.21: ID ứng dụng và khóa ứng dụng của Facebook

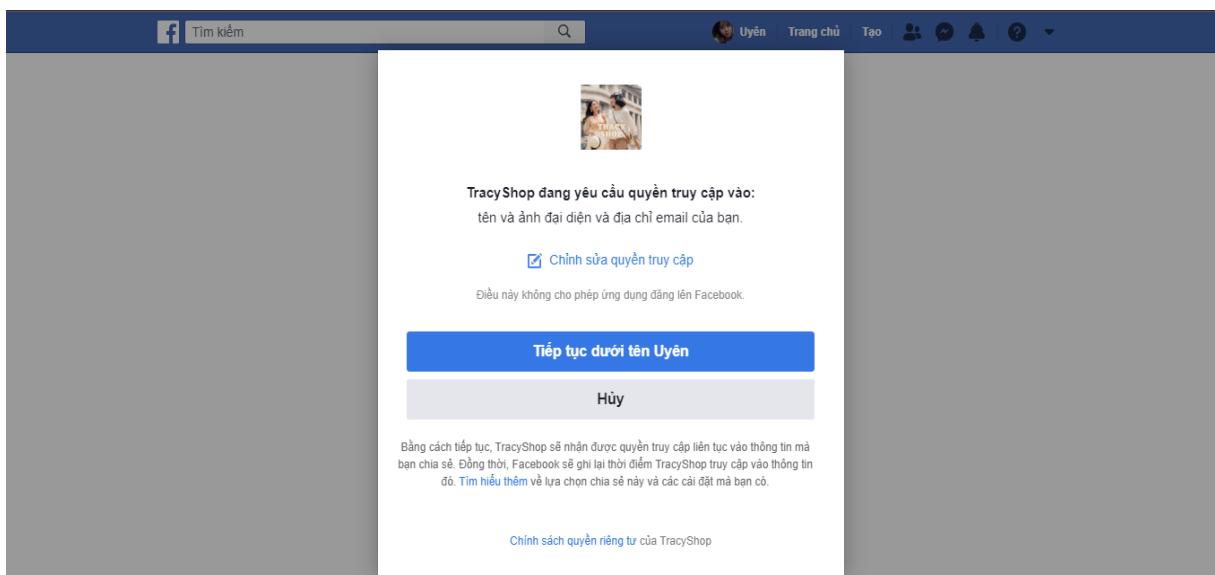
Ta đặt hai giá trị này trong appsettings.json của dự án:

```
"Authentication": {  
    "Facebook": {  
        "AppId": "214999683932773",  
        "AppSecret": "9d4fe2c81eed7605f1c764611dc13366"  
    }  
},
```

Tiếp đến, ta đăng ký dịch vụ xác thực bằng Facebook trong phương thức ConfigureServices của lớp Startup:

```
services.AddAuthentication()  
    .AddFacebook(facebookOptions =>  
    {  
        IConfigurationSection facebookAuthNSection =  
            Configuration.GetSection("Authentication:Facebook");  
        facebookOptions.AppId = facebookAuthNSection["AppId"];  
        facebookOptions.AppSecret =  
            facebookAuthNSection["AppSecret"];  
        facebookOptions.ClaimActions.MapJsonKey("image",  
            "picture");  
        facebookOptions.CallbackPath = "/dang-nhap-tu-facebook";  
    });
```

Nếu các cấu hình cài đặt không gặp vấn đề sai sót gì thì Identity trong dự án đã có thể cho phép người dùng đăng nhập qua Facebook:

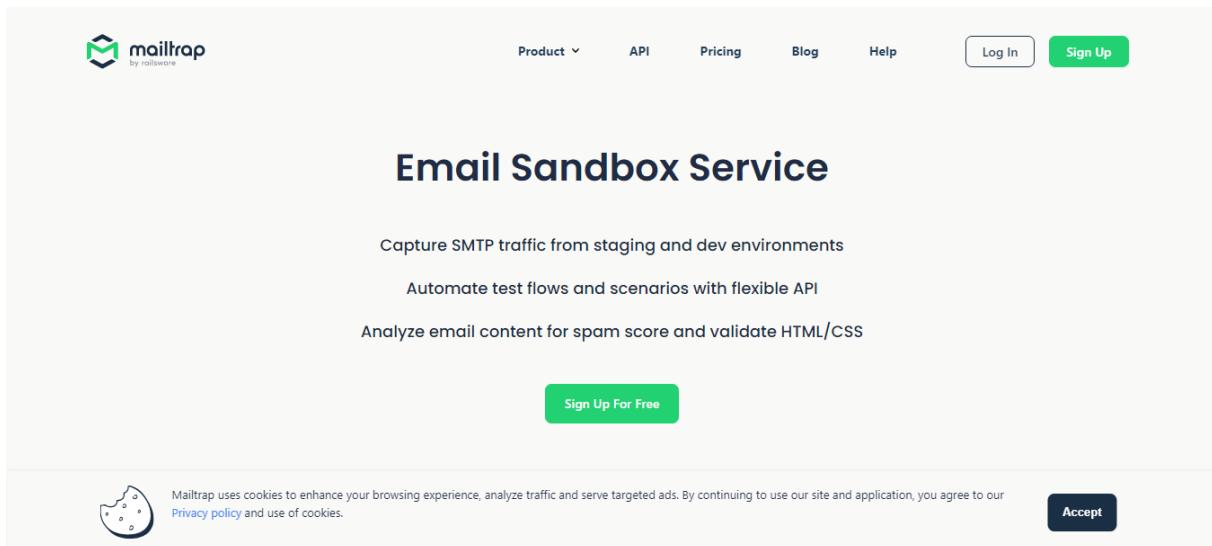


Hình 2.22: Màn hình đăng nhập qua Facebook

2.4. Kiểm tra gửi mail bằng Mailtrap

2.4.1. Giới thiệu về Mailtrap

Với một dự án công nghệ thông tin thì việc kiểm thử các chức năng luôn được tiến hành xuyên suốt từ giai đoạn viết code đến giai đoạn hoàn thành nhằm nhanh chóng phát hiện những lỗi sai và khắc phục chúng. Trong thực tế, để kiểm thử các chức năng của ứng dụng khi một dự án hoàn thành, các nhà phát triển website hay phần mềm thường đưa ra các phiên bản beta đến người dùng để người dùng trải nghiệm thử các tính năng và đưa ra các phản hồi. Từ các phản hồi của người dùng, các nhà lập trình sẽ tiến hành sửa lỗi và cập nhật lại các chức năng của ứng dụng. Nhưng với một số chức năng, việc kiểm thử trên người dùng sẽ gặp một số khó khăn. điển hình là chức năng gửi mail đến người dùng đã đăng ký, việc nhận hàng loạt các mail rác sẽ làm cho người dùng cảm thấy khó chịu và bị làm phiền. Đồng thời, nó cũng vi phạm tính bảo mật người dùng. Hoặc khi gửi mail đến danh sách người dùng đã đăng ký, ta không thể xác nhận được mail này đã gửi đến tất cả người dùng hay chưa. Và rồi, nhiều công cụ kiểm tra gửi mail đã ra đời để giải quyết các vấn đề trên, trong đó Mailtrap là một công cụ kiểm tra gửi mail được nhiều người sử dụng.



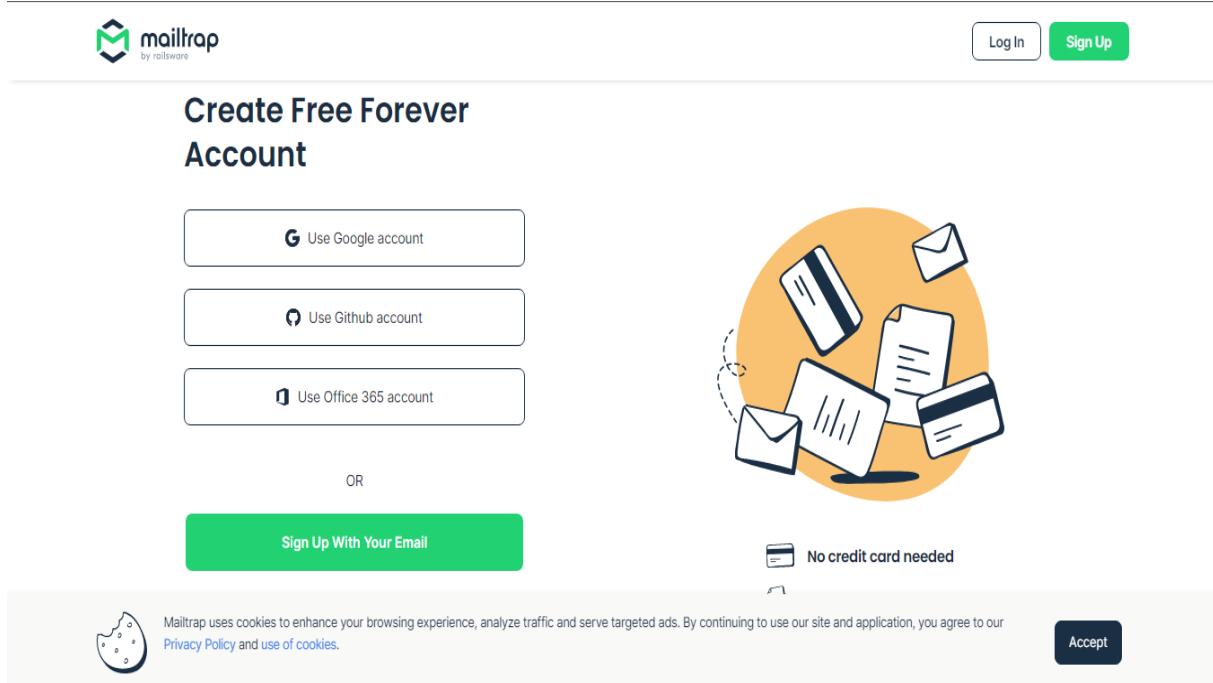
Hình 2.23: Trang chủ của Mailtrap

2.4.2. Mailtrap là gì?

Mailtrap là một công cụ cung cấp máy chủ SMTP giả dùng để kiểm tra các mail được gửi từ môi trường pre-production và môi trường kiểm thử thực tế, giúp tránh được tình trạng gửi thư rác đến người dùng. Đồng thời, nó tự động đưa các mail quan trọng

đến hộp thư thường xuyên và có hỗ trợ chia sẻ với nhóm [12].

Mailtrap có giao diện thân thiện với người dùng. Để sử dụng, ta cần đăng ký một tài khoản. Ta có thể đăng ký bằng email hay sử dụng tài khoản Google, Github hay Office 365.



Hình 2.24: Giao diện đăng ký của Mailtrap

Sau khi tạo tài khoản và đăng nhập, ta đã có thể tiến hành tích hợp Mailtrap vào dự án để tiến hành kiểm thử gửi mail.

The screenshot shows the Mailtrap inbox interface. On the left is a dark sidebar with the Mailtrap logo, 'Inboxes', 'API', and 'Billing' options. The main area has a header with 'Inboxes' and a user profile for 'Uyên Nguyễn'. It includes an 'Upgrade' button and a 'Add Project' button. Below this is a 'Projects' section for 'TracyShop' with an 'Add Inbox' button. The main content is an 'Inboxes' table with columns: 'Inboxes', 'Total Sent', 'Messages', 'Max size', 'Last message', and 'Action'. One row shows 'My Inbox' with values 3, 0/0, 50, and 'Empty' in the last column. Action icons are shown in the 'Action' column.

Hình 2.25: Giao diện hộp thư đến trên Mailtrap

Với dự án nhỏ chỉ kiểm tra gửi mail trên số lượng ít không quá 50 mail thì ta có thể sử dụng bản miễn phí của Mailtrap [12]. Ngược lại nếu nhu cầu của hệ thống cần kiểm tra gửi mail với số lượng lớn thì ta chọn bản trả phí để tăng giới hạn số lượng email. Dưới đây là các bản trả phí trên Mailtrap với các số lượng email giới hạn tương ứng:

Individual	Team	Business	Premium	Enterprise
For freelancers & individuals working in a one-person team, who want to benefit from higher email sending rate limits.	For an established team where users may invite others to join their project, which has a moderate volume of emails to test.	For companies with multiple tech teams working on several of their own or client projects that require higher rate limits, more emails, extra inboxes and a dedicated email address.	For large organizations looking for a reliable system to handle large-scale email testing backed up by prime limits and rich collaboration possibilities.	For clients who are capable of challenging Mailtrap by loading it with email testing from both development and staging environments with virtually no limits.
\$ 9.99 per month	\$ 24.99 per month	\$ 49.99 per month	\$ 99.99 per month	\$ 299.99 per month
Select	Select	Select	Select	Select
5,000 emails	15,000 emails	50,000 emails	100,000 emails	10,000,000 emails
<ul style="list-style-type: none"> ✓ 10 emails / 10 sec ✓ 5 Inboxes ✓ 0 Team Members ✓ 200 Emails / inbox ✓ API 	<ul style="list-style-type: none"> ✓ 25 emails / 10 sec ✓ 15 Inboxes ✓ 20 Team Members ✓ 400 Emails / inbox ✓ API 	<ul style="list-style-type: none"> ✓ 50 emails / 10 sec ✓ 40 Inboxes ✓ 40 Team Members ✓ 600 Emails / inbox ✓ API 	<ul style="list-style-type: none"> ✓ 100 emails / 10 sec ✓ 80 Inboxes ✓ 80 Team Members ✓ 800 Emails / inbox ✓ API 	<ul style="list-style-type: none"> ✓ 150 emails / 10 sec ✓ 300 Inboxes ✓ 999 Team Members ✓ 1,000 Emails / inbox ✓ API

Hình 2.26: Các bản trả phí trên Mailtrap

2.4.3. Cấu hình dịch vụ gửi mail qua Mailtrap

Mailtrap sử dụng phương thức SMTP để gửi mail nên nó có thể hoạt động trên các port 25, 2525, 465, 587 và một số port khác của SMTP. Để xem cấu hình của Mailtrap, ta vào My Inbox chọn mục SMTP Settings, ở phần Integrations chọn ngôn ngữ được sử dụng trong dự án. Trong dự án này sử dụng ngôn ngữ C# nên ta chọn Plain C#.

Integrations (?)

Plain C#

Hình 2.27: Chọn Integrations

Sau khi chọn Plain C#, nó hiện ra một đoạn code C#, trong đó có chứa các giá trị cấu hình của SMTP trong Mailtrap như host, port, username, password và EnableSSL.

My Inbox Total messages sent: 3

SMTP Settings	Email Address	Auto Forward	Manual Forward	Team Members
---------------	---------------	--------------	----------------	--------------

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net.Mail;
using System.Net;

namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {
            var client = new SmtpClient("smtp.mailtrap.io", 2525)
            {
                Credentials = new NetworkCredential("8bbb96d6d84a0d", "d0444096b5f246"),
                EnableSsl = true
            };
            client.Send("from@example.com", "to@example.com", "Hello world", "testbody");
            Console.WriteLine("Sent");
            Console.ReadLine();
        }
    }
}

```

Copy

Hình 2.28: Cấu hình SMTP trong Mailtrap

Ta đưa cấu hình này vào tập tin appsettings.json của dự án:

```
"SMTPConfig": {  
    "SenderAddress": "no-reply@tracyshop.com",  
    "SenderDisplayName": "TracyShop Team",  
    "UserName": "8bbb96d6d84a0d",  
    "Password": "d0444096b5f246",  
    "Host": "smtp.mailtrap.io",  
    "Port": 587,  
    "EnableSSL": true,  
    "UseDefaultCredentials": true,  
    "IsBodyHTML": true  
},
```

Giá trị SenderAddress trong cấu hình là địa chỉ email của người gửi. Giá trị này ta tự đặt sao cho đúng với cấu hình của một địa chỉ email. Tương tự với giá trị SenderDisplayName. Giá trị này là tên hiển thị của người gửi và giá trị này cũng được đặt tùy ý.

2.4.4. Thiết lập gửi mail qua Mailtrap

Trong ASP.NET Core, chức năng gửi mail được hỗ trợ bởi thư viện **System.Net**. Thư viện này cung cấp lớp **SmtpClient** hỗ trợ gửi mail thông qua giao thức SMTP [11]. Các thông tin của mail như tiêu đề, nội dung, địa chỉ email và tên người gửi được tạo trong đối tượng lớp **MailMessage**. Lớp này nằm trong thư viện **System.Net.Mail**, là một thư viện con của thư viện **System.Net**. Sau khi có các nội dung của mail, ta gọi phương thức **SendMailAsync** trong SmtpClient để tiến hành gửi mail cho người dùng.

Để thiết lập gửi mail bằng Mailtrap, đầu tiên ta xây dựng một giao diện có tên **IEmailService**. Giao diện này dùng để triển khai các dịch vụ đến người dùng, giả sử dịch vụ trong giao diện cần triển khai là kiểm tra gửi mail đến các người dùng đăng ký:

```
public interface IEmailService  
{  
    Task SendEmailForTestMail(UserEmailOptions userEmailOptions);  
}
```

Tiếp theo, ta xây dựng một lớp **EmailService**. Lớp này kế thừa và triển khai các dịch vụ trong giao diện **IEmailService**. Để lấy các giá trị cấu hình SMTP trong

appsettings.json, ta tạo một lớp có tên là SMTPConfigModel chứa các thuộc tính cấu hình của SMTP trong appsettings.json như SenderAddress (địa chỉ người gửi), SenderDisplayName (tên hiển thị của người gửi), Username (tên đăng nhập mail), Password (mật khẩu đăng nhập mail), Host (địa chỉ tên miền của host), Port,...

Trong EmailService, ta tạo một đối tượng lớp này và khởi tạo nó để sử dụng:

```
private readonly SMTPConfigModel _smtpConfig;
public EmailService(IOptions<SMTPConfigModel> smtpConfig)
{
    _smtpConfig = smtpConfig.Value;
}
```

Sau khi được khởi tạo, các giá trị cấu hình SMTPConfig trong appsettings.json sẽ được lưu vào đối tượng _smtpConfig.

Ta tiếp tục tạo phương thức SendEmail trong EmailService để xử lý gửi mail với tham số truyền vào là một đối tượng lớp UserEmailOptions, gồm các thuộc tính sau:

- ToEmail: danh sách địa chỉ email của người nhận
- Subject: chủ đề mail
- Body: nội dung mail
- PlaceHolders: đây là một thuộc tính có kiểu danh sách (List), mỗi phần tử trong danh sách là một KeyValuePair<string, string> chứa một thông tin của người dùng để gắn các key vào template của mail được tạo bằng HTML và nó sẽ trả về value của từng key để hiển thị lên giao diện mail.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title></title>
</head>
<body>
    <p>Hello {{UserName}}, <br />This is a test email</p>
</body>
</html>
```

Đoạn HTML trên là một template mail dùng để kiểm tra gửi mail đến người dùng đăng ký với {{Username}} là key của thuộc tính PlaceHolders được gán vào HTML và nó sẽ trả về value của key để hiển thị tên người dùng trong mail.

Trong phương thức SendEmail, ta tạo một đối tượng lớp MailMessage chứa thông tin của một mail như chủ đề, nội dung, thông tin người gửi với các thuộc tính như Subject, Body, From, IsBodyHtml,... Thuộc tính From là một đối tượng lớp MailAddress lưu thông tin của người gửi như địa chỉ email, tên hiển thị.

```
MailMessage mail = new MailMessage
{
    Subject = userEmailOptions.Subject,
    Body = userEmailOptions.Body,
    From = new MailAddress(_smtpConfig.SenderAddress,
                           _smtpConfig.SenderDisplayName),
    IsBodyHtml = _smtpConfig.IsBodyHTML
};
```

Để biết được mail này sẽ gửi đến những người nhận nào, ta cần thêm những địa chỉ email của người nhận chúa trong thuộc tính ToEmail của đối tượng UserEmailOptions vào thuộc tính To trong đối tượng mail.

```
foreach (var toEmail in userEmailOptions.ToEmails)
{
    mail.To.Add(toEmail);
}
```

Cũng trong phương thức này, ta tạo một đối tượng lớp SmtpClient chứa thông tin cấu hình gửi mail thông qua giao thức SMTP. Trong thông tin cấu hình này có chứa thông tin xác thực để khi lên server, nó sẽ tiến hành xác thực trước khi gửi mail. Ta tạo một đối tượng lớp NetworkCredential chứa thông tin xác thực SMTP gồm tên đăng nhập và mật khẩu được lấy từ SMTPConfig trong appsettings.json:

```
NetworkCredential networkCredential = new
    NetworkCredential(_smtpConfig.UserName, _smtpConfig.Password);
```

Sau đó gắn nó vào trong thuộc tính Credentials của đối tượng lớp SmtpClient. Thuộc tính này chứa thông tin xác thực SMTP:

```

SmtpClient smtpClient = new SmtpClient
{
    UseDefaultCredentials = false,
    Host = _smtpConfig.Host,
    Port = _smtpConfig.Port,
    EnableSsl = _smtpConfig.EnableSSL,
    Credentials = networkCredential
};

```

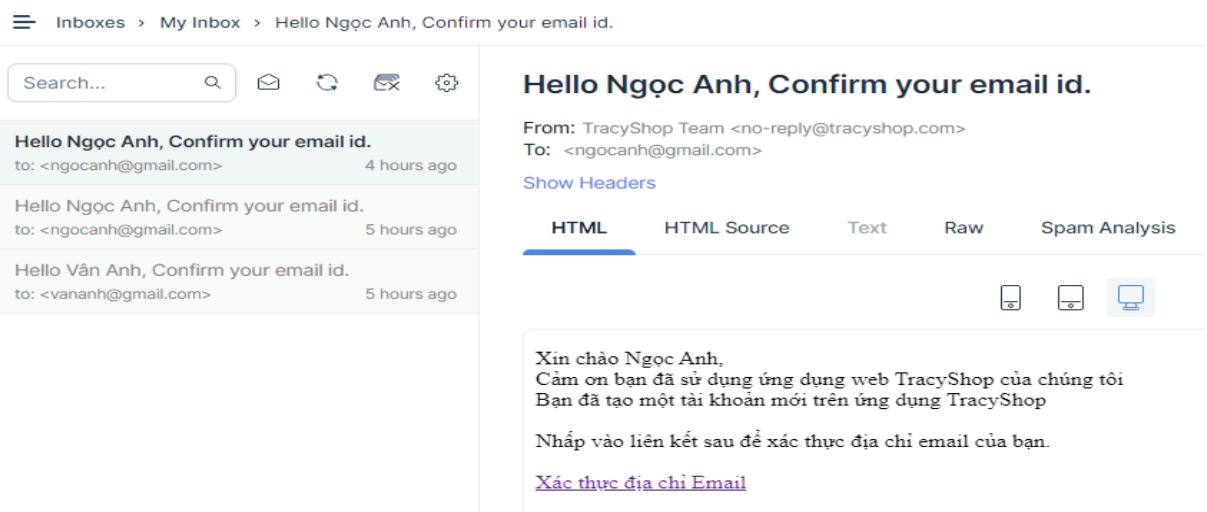
Để quy định hiển thị nội dung mail trong Mailtrap, ta sử dụng đối tượng lớp **Encoding** trong thư viện **System.Text** và gán giá trị cho thuộc tính **BodyEncoding** trong đối tượng mail:

```
mail.BodyEncoding = Encoding.Default;
```

Sau khi thiết lập cấu hình cũng như có các thông tin, nội dung của mail, ta gọi phương thức **SendMailAsync** trong lớp **SmtpClient** để gửi mail này đến địa chỉ email của người nhận:

```
await smtpClient.SendMailAsync(mail);
```

Trong lớp **EmailService**, ta tiếp tục tạo các phương thức gửi mail riêng cho từng chức năng như phương thức gửi mail cho chức năng xác nhận địa chỉ email, phương thức gửi mail cho chức năng quên mật khẩu,... Các phương thức này đều sử dụng phương thức **SendEmail** để gửi mail và với mỗi chức năng sẽ có những chủ đề và nội dung mail được tạo khác nhau. Nội dung mail của từng chức năng được tạo bằng HTML.

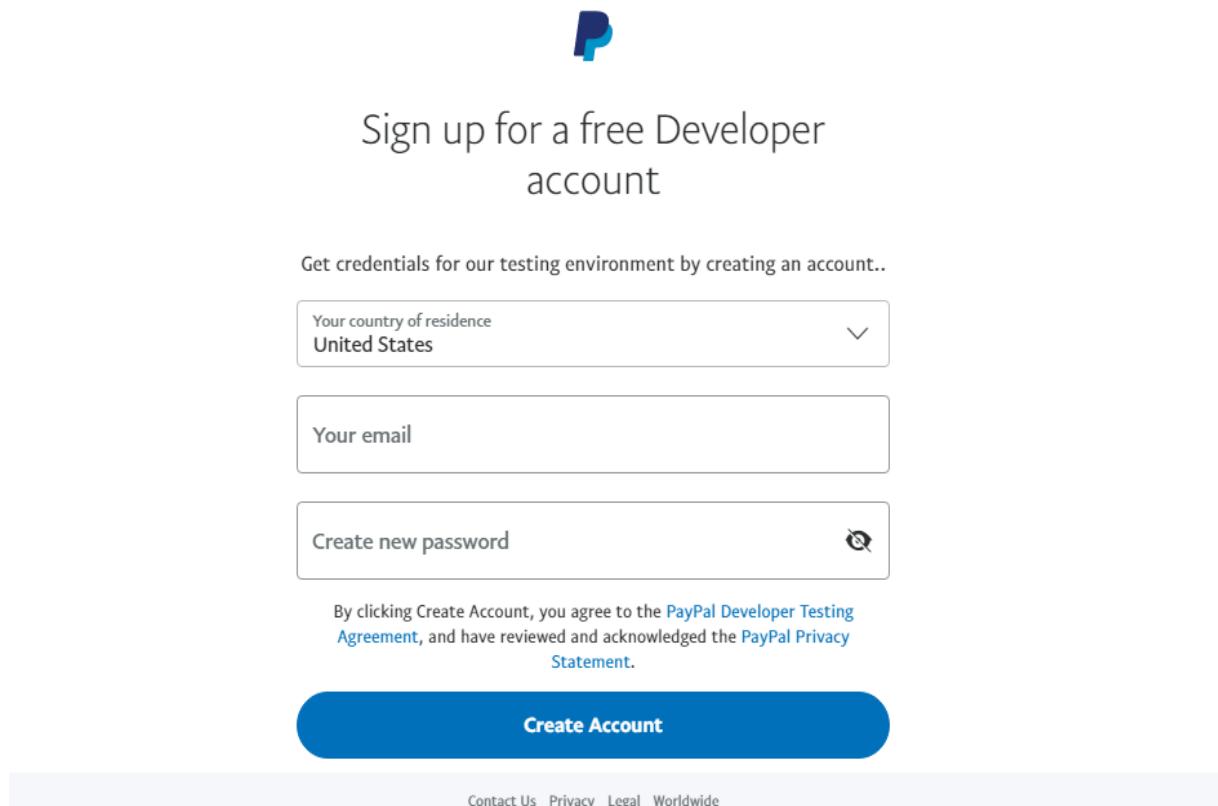


Hình 2.29: Giao diện nhận mail trong Mailtrap

2.5. Thanh toán qua dịch vụ thanh toán trực tuyến PayPal

PayPal là một dịch vụ thanh toán điện tử thông qua mạng internet cho phép các doanh nghiệp hoặc cá nhân thanh toán và chuyển tiền một cách an toàn và tin cậy với từ nhiều nguồn khác nhau như ví PayPal, ngân hàng, thẻ VISA.

Để tích hợp thanh toán qua PayPal, đầu tiên ta cần đăng ký tài khoản trên PayPal Developer qua đường dẫn <https://developer.paypal.com/>:



Hình 2.30: Giao diện đăng nhập của PayPal Developer

Sau khi đăng ký xong, ta tiến hành đăng nhập vào PayPal Developer. Mặc định, PayPal Developer sẽ tự động tạo một ứng dụng có tên Default Application, ta có thể sử dụng ứng dụng này hoặc tạo một ứng dụng mới. Mỗi ứng dụng được tạo đều có một Client ID và Secret riêng:

TracyShop

App display name: TracyShop 

SANDBOX API CREDENTIALS

Sandbox account
Join@business.example.com

Client ID
AceUo4hOCF0wnM4prXw2rGgKXBgvap5ffYG6RQ4IleTuTJRrZ6x29MkTh5Mh4cTuVVw6a4ekNkmIx3Mz

Secret
[Hide](#)

Note: When you generate a new secret, you still maintain the original secret. The maximum number of client secrets is two. A client secret is either in enabled or disabled state.

Created	Secret	Status	Action
Oct 27, 2021	ELRqwn8CaEJSbZL9MV7-RV-LGclCXC879MdECTuCO5X3WM9S2ngvn9aw0U h58l7yo9Dfoq-EfQwwcHlx	Enabled	...

[Generate new secret](#)

Hình 2.31: Cấu hình ứng dụng trong PayPal Developer

Để tiến hành tích hợp, ta đưa hai giá trị Client ID và Secret vào tập tin appsettings.json của dự án:

```
"PaypalSettings": {  
    "ClientId":  
        "AceUo4hOCF0wnM4prXw2rGgKXBgvap5ffYG6RQ4IleTuTJRrZ6x29MkTh5Mh4c  
        TuVVw6a4ekNkmIx3Mz",  
    "SecretKey": "ELRqwn8CaEJSbZL9MV7-RV-  
        LGclCXC879MdECTuCO5X3WM9S2ngvn9aw0Uh58l7yo9Dfoq-EfQwwcHlx"  
}
```

PayPal ngoài tự động tạo ứng dụng còn tạo hai tài khoản ứng với hai loại người dùng trong PayPal là Business (tài khoản người bán) và Personal (tài khoản người mua). Hai tài khoản này được dùng để thử nghiệm thanh toán qua PayPal. Ta có thể tạo thêm nhiều tài khoản khác ngoài hai tài khoản đã được tạo sẵn.

Sandbox Accounts:

Total Accounts: 4

<input type="checkbox"/>	Account name	Type	Country	Date created	Manage accounts
<input type="checkbox"/>	tracy-seller@gmail.com	Business	US	27 Oct 2021	...
<input type="checkbox"/>	tracy-buyer@gmail.com <small>DEFAULT</small>	Personal	US	27 Oct 2021	...
<input type="checkbox"/>	join@business.example.com <small>DEFAULT</small>	Business	US	27 Oct 2021	...
<input type="checkbox"/>	sb-qfjka8317573@personal.example.c... <small>DEFAULT</small>	Personal	US	27 Oct 2021	...

Hình 2.32: Các tài khoản được tạo để thử nghiệm trong PayPal Developer

Trong PayPal không hỗ trợ thanh toán bằng đơn vị tiền Việt Nam nên trước khi gửi thông tin thanh toán lên cho PayPal, ta cần phải chuyển đơn vị tiền VND sang các đơn vị tiền khác mà PayPal hỗ trợ, ví dụ như chuyển sang USD khi doanh nghiệp hay cá nhân đang sử dụng đơn vị tiền VND.

Test Store

\$17.58 USD [Change](#)

Hi, Linh!

Ship to
Linh Đảm
1 Main St, San Jose, CA 95131
[Change](#)

Make this my preferred shipping address

Pay with

Balance \$17.58 USD

Make this my preferred way to pay

Visa Credit ****2309

CREDIT UNION 1 Checking ****2190

[+ Add debit or credit card](#)

Pay later [NEW](#)

PayPal Credit
Apply for PayPal Credit
Pay over time for your purchase of \$17.58 with PayPal Credit.
Subject to credit approval. [See terms](#)

[View PayPal Policies](#) and your payment method rights.

[Continue](#)

You'll be able to review your order before you complete your purchase.

Hình 2.33: Giao diện thanh toán của PayPal

Chương 3. WEBSITE BÁN QUẦN ÁO NỮ ONLINE TRACYSHOP

Chương này nhằm giới thiệu về đề tài “Xây dựng website bán quần áo nữ online TracyShop”, trình bày về phân tích thiết kế của hệ thống gồm sơ đồ usecase, đặc tả các usecase chức năng, mô hình cơ sở dữ liệu và thực nghiệm hệ thống.

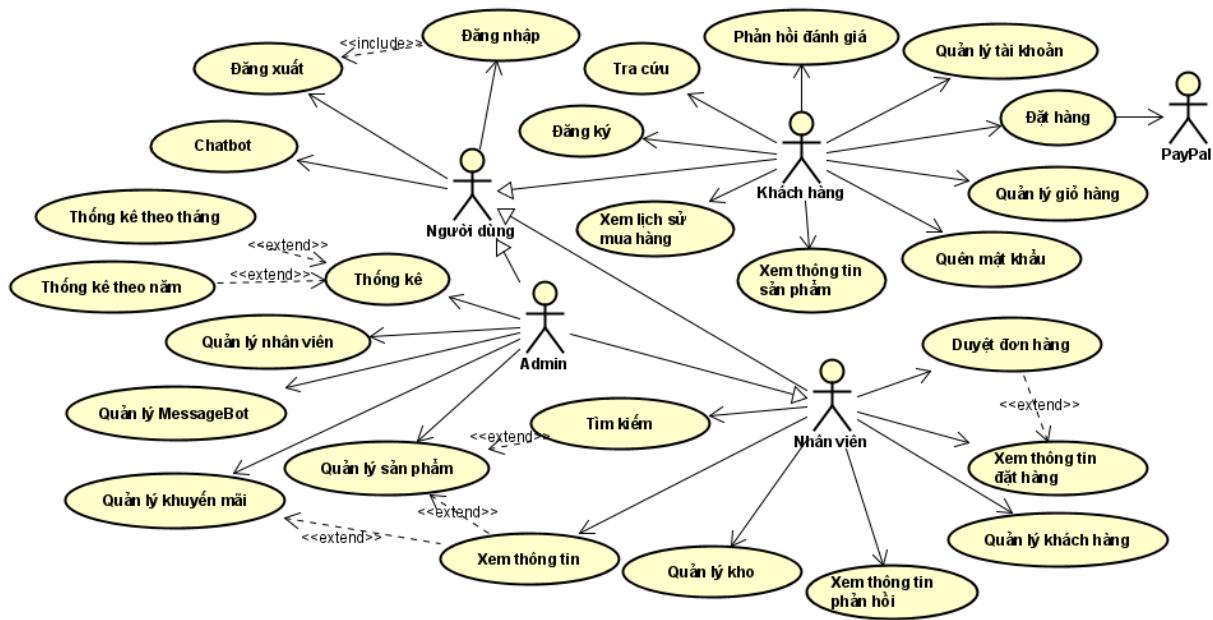
3.1. Giới thiệu đề tài

Hiện nay, tình hình dịch Covid-19 đang diễn biến phức tạp, việc đi lại của mọi người cũng trở nên khó khăn. Nhiều người đã lựa chọn việc mua hàng qua các trang website, ứng dụng thay cho việc mua hàng tại các cửa hàng, siêu thị. Do đó, mua hàng online đã trở nên phổ biến và là lựa chọn hàng đầu của nhiều người cũng một phần bởi tính tiện lợi mà nó mang lại. Cùng với đó, các cửa hàng thời trang cũng đẩy mạnh việc mua bán online để bắt kịp xu hướng và đáp ứng nhu cầu của người dùng.

Website bán quần áo nữ online TracyShop cung cấp nhiều tính năng cho người dùng mua sắm tại nhà. Người dùng có thể truy cập để xem danh sách và chi tiết các sản phẩm của shop, thực hiện các chức năng quản lý tài khoản. Đồng thời, hệ thống cũng cho phép người dùng đánh giá sản phẩm, bình luận và xem các đánh giá, bình luận của các người dùng khác và nhờ admin tư vấn thông qua chức năng chat. Ngoài ra, người dùng có thể đặt hàng, thêm sản phẩm vào giỏ hàng và được hỗ trợ thanh toán đơn hàng với nhiều hình thức như thanh toán khi nhận hàng và thanh toán qua dịch vụ thanh toán trực tuyến PayPal. Bên cạnh đó, hệ thống cũng có các chức năng quản lý dành cho admin như quản lý sản phẩm, quản lý đơn hàng, quản lý khách hàng,...

3.2. Phân tích và thiết kế hệ thống

3.2.1. Sơ đồ usecase



Hình 3.1: Sođồ usecase

3.2.2. Đặc tả usecase

3.2.2.1. Usecase đăng nhập

Mô tả	Use case cho phép người dùng đăng nhập vào hệ thống để thực hiện các chức năng
Actor chính	Người dùng
Actor phụ	Không có
Tiền điều kiện	Người dùng phải có tài khoản trên hệ thống
Hậu điều kiện	Nếu người dùng đăng nhập thành công thì được thực hiện chức năng của hệ thống, ngược lại sẽ tiến hành xử lý lỗi đăng nhập
Luồng hoạt động	<ol style="list-style-type: none"> 1. Hệ thống hiển thị màn hình đăng nhập 2. Người dùng nhập tên đăng nhập và mật khẩu 3. Hệ thống kiểm tra tên đăng nhập và mật khẩu. 4. Nếu thành công thì hiển thị trang mà người dùng chọn trước đó, ngược lại hiển thị thông báo lỗi và quay lại bước 2. 5. Kết thúc use case.

Luồng ngoại lệ	I - Mật khẩu không hợp lệ Khi người dùng nhập sai tên đăng nhập và mật khẩu: 1. Hệ thống hiển thị lại màn hình đăng nhập để người dùng nhập lại thông tin kèm theo thông báo lỗi. 2. Quay lại bước 2 trong luồng sự kiện chính.
----------------	---

Bảng 3.1: Đặc tả usecase đăng nhập

3.2.2.2. Usecase đăng ký

Mô tả	Use case cho phép người dùng đăng ký tài khoản
Actor chính	Khách hàng
Actor phụ	Không có
Tiền điều kiện	Tài khoản khách hàng đăng ký phải chưa được đăng ký trước đó.
Hậu điều kiện	Nếu khách hàng đăng ký thành công thì sẽ tiến hành đăng nhập vào hệ thống, ngược lại sẽ tiến hành xử lý lỗi đăng ký.
Luồng hoạt động	1. Hệ thống hiển thị màn hình đăng ký 2. Khách hàng nhập các thông tin yêu cầu 3. Hệ thống kiểm tra thông tin khách hàng cung cấp 4. Nếu thông tin tài khoản hợp lệ thì hệ thống sẽ gửi mail đến địa chỉ email của khách hàng dùng để đăng ký để xác nhận địa chỉ email này có tồn tại hay không, ngược lại sẽ hiển thị thông báo lỗi và quay lại bước 2. 5. Khách hàng kiểm tra mail, nếu mail đã được gửi đến thì khách hàng truy cập liên kết được gửi trong mail để xác thực, ngược lại khách hàng yêu cầu hệ thống gửi lại mail xác thực và tiếp tục thực hiện lại bước 5. 5. Kết thúc use case.
Luồng ngoại lệ	I – Tài khoản đã tồn tại: Khi hệ thống kiểm tra tài khoản này đã tồn tại: 1. Hệ thống hiển thị lại màn hình đăng ký để khách hàng nhập lại thông tin kèm theo thông báo lỗi. 2. Quay lại bước 2 trong luồng sự kiện chính.

Bảng 3.2: Đặc tả usecase đăng ký

3.2.2.3. Usecase đặt hàng

Mô tả	Use case cho phép khách hàng đặt hàng trên hệ thống
Actor chính	Khách hàng
Actor phụ	PayPal
Tiền điều kiện	Khách hàng phải đăng nhập thành công vào hệ thống
Luồng hoạt động	<p>1. Khách hàng nhấn vào mua hàng</p> <p>2. Hệ thống thêm sản phẩm vào giỏ hàng và chuyển đến trang giỏ hàng.</p> <p>3. Khách hàng chỉnh sửa số lượng và kích thước sản phẩm cần mua nếu cần và chọn phương thức thanh toán. Khách hàng cũng có thể thay đổi số điện thoại và địa chỉ nhận hàng.</p> <p>4. Nếu khách hàng chọn “Thanh toán khi nhận hàng” thì luồng phụ Thanh toán khi nhận hàng sẽ được thực hiện. Nếu khách hàng chọn “Thanh toán PayPal” thì luồng phụ Thanh toán PayPal sẽ được thực hiện.</p> <p>* Các luồng phụ:</p> <p>I – Thanh toán khi nhận hàng</p> <p>Usecase được thực hiện khi khách hàng chọn đặt hàng.</p> <p>1. Nếu thành công, hệ thống sẽ hiển thị thông báo đặt hàng thành công. Ngược lại, sẽ thông báo đặt hàng thất bại.</p> <p>2. Kết thúc usecase.</p> <p>II – Thanh toán PayPal</p> <p>Use case này được thực hiện khi khách hàng chọn đặt hàng.</p> <p>1. Hệ thống sẽ liên lạc với Hệ thống PayPal và chuyển đến trang đăng nhập tài khoản PayPal.</p> <p>2. Khách hàng đăng nhập, hệ thống sẽ hiển thị thông tin đơn hàng kèm số tiền thanh toán. Khách hàng chọn nguồn tiền thanh toán và nhập vào “Continue”. Hệ thống PayPal sẽ xử lý thanh toán cho khách hàng.</p> <p>3. Nếu thành công, hệ thống sẽ chuyển đến trang thông báo thanh toán thành công, ngược lại sẽ chuyển đến trang thông báo thanh toán thất bại. Kết thúc use case.</p>

Bảng 3.3: Đặc tả usecase đặt hàng

3.2.2.4. Usecase tra cứu

Mô tả	Use case cho phép khách hàng tra cứu các sản phẩm theo tên sản phẩm, loại sản phẩm, nhãn hàng, xuất xứ.
Actor chính	Khách hàng
Actor phụ	Không có
Luồng hoạt động	<ol style="list-style-type: none"> 1. Khách hàng nhập từ khóa trên thanh tìm kiếm và nhấn tìm kiếm. 2. Hệ thống sẽ tìm những sản phẩm có tên sản phẩm, loại sản phẩm, nhãn hiệu hoặc xuất xứ trùng với từ khóa khách hàng nhập vào và hiển thị ở trang sản phẩm. Nếu không tìm thấy thì sẽ hiển thị thông báo không tìm thấy sản phẩm. 3. Kết thúc use case.

Bảng 3.4: Đặc tả usecase tra cứu

3.2.2.5. Usecase phản hồi đánh giá

Mô tả	Use case cho phép khách hàng bình luận bằng cách viết và có thể thêm hình ảnh nếu có và đánh giá sản phẩm theo sao.
Actor chính	Khách hàng
Actor phụ	Không có
Tiền điều kiện	Khách hàng phải đăng nhập thành công vào hệ thống.

Luồng hoạt động	<ol style="list-style-type: none"> 1. Hệ thống hiển thị màn hình phản hồi và đánh giá sản phẩm. 2. Khách hàng viết bình luận và đánh giá sản phẩm bằng cách chọn sao. Khách hàng có thể thêm ảnh nếu có. 3. Hệ thống hiển thị trang chi tiết sản phẩm với những đánh giá sản phẩm của khách hàng. 5. Kết thúc use case.
-----------------	---

Bảng 3.5: Đặc tả usecase phản hồi đánh giá

3.2.2.6. Usecase quản lý giỏ hàng

Mô tả	Use case này cho phép khách hàng thực hiện các chức năng quản lý giỏ hàng, bao gồm các chức năng: thêm vào giỏ hàng, xem thông tin giỏ hàng, cập nhật, xóa khỏi giỏ hàng.
Actor chính	Khách hàng

Actor phụ	Không có
Tiền điều kiện	Khách hàng phải đăng nhập thành công vào hệ thống
Luồng hoạt động	<p>1. Khách hàng truy cập vào trang Giỏ hàng bằng cách nhấp chọn vào biểu tượng giỏ hàng.</p> <p>2. Hệ thống sẽ hiển thị danh sách sản phẩm có trong giỏ, khi đó luồng phụ Xem thông tin giỏ hàng được thực hiện.</p> <ul style="list-style-type: none"> - Nếu giỏ hàng trống thì hệ thống sẽ hiển thị thông báo không có sản phẩm trong giỏ. Nếu khách hàng chọn “Tiếp tục mua sắm” thì hệ thống sẽ chuyển đến trang Sản phẩm, ngược lại hệ thống sẽ không thay đổi, use case kết thúc. - Nếu có sản phẩm trong giỏ hàng, hệ thống sẽ hiển thị danh sách các sản phẩm trong giỏ hàng mà khách hàng đã thêm trước đó. <p>3. Nếu khách hàng đang ở trang Sản phẩm, khách hàng nhấp chọn “Thêm vào giỏ hàng” thì luồng phụ Thêm vào giỏ hàng sẽ được thực hiện. Nếu khách hàng đang xem danh sách sản phẩm trong giỏ, khách hàng chọn xóa sản phẩm khỏi giỏ thì luồng phụ Xóa khỏi giỏ hàng sẽ được thực hiện hoặc khách hàng chọn cập nhật số lượng sản phẩm trong giỏ thì luồng phụ Cập nhật giỏ hàng được thực hiện.</p> <p>* Các luồng phụ:</p> <p>I – Xem thông tin giỏ hàng</p> <p>Use case bắt đầu khi khách hàng chọn chức năng Quản lý giỏ hàng, hệ thống sẽ hiển thị danh sách các sản phẩm mà khách hàng đã thêm vào giỏ hàng trước đó.</p> <p>II – Thêm vào giỏ hàng</p> <p>Use case bắt đầu khi khách hàng nhấp chọn “Thêm vào giỏ hàng” của một sản phẩm cụ thể, hệ thống sẽ thêm sản phẩm vào giỏ hàng và thông báo cho khách hàng đã thêm vào giỏ hàng thành công.</p> <p>III – Cập nhật giỏ hàng</p> <p>1. Khách hàng nhấp chọn vào biểu tượng cập nhật của một sản phẩm trong giỏ hàng.</p> <p>2. Hệ thống hiển thị giao diện cập nhật để khách hàng chỉnh sửa các thông tin.</p>

	<p>3. Khách hàng chỉnh sửa size và số lượng sản phẩm nếu cần và chọn cập nhật.</p> <p>4. Hệ thống cập nhật lại số lượng và size của sản phẩm đó trong giỏ nếu có chỉnh sửa và chuyển đến trang giỏ hàng.</p>
--	--

Bảng 3.6: Đặc tả usecase quản lý giỏ hàng

3.2.2.7. Usecase quản lý nhân viên

Mô tả	Use case cho phép admin thực hiện các chức năng quản lý nhân viên.
Actor chính	Admin
Actor phụ	Không có
Tiền điều kiện	Admin đã đăng nhập thành công vào hệ thống
Luồng hoạt động	<p>1. Admin chọn chức năng quản lý nhân viên</p> <p>2. Hệ thống sẽ hiển thị các tùy chọn chức năng gồm danh sách nhân viên, thêm nhân viên.</p> <p>3. Nếu admin chọn “Thêm nhân viên” thì luồng phụ Thêm nhân viên sẽ được thực hiện. Nếu admin chọn “Danh sách nhân viên” thì hệ thống chuyển đến trang nhân viên và hiển thị danh sách nhân viên. Với mỗi nhân viên, ta có thể xem thông tin, sửa, thêm địa chỉ, xóa nhân viên.</p> <ul style="list-style-type: none"> - Nếu admin chọn biểu tượng sửa thì luồng phụ Sửa thông tin nhân viên sẽ được thực hiện. - Nếu admin chọn biểu tượng xem thông tin thì luồng phụ Xem thông tin nhân viên sẽ được thực hiện. - Nếu admin chọn thêm địa chỉ thì luồng phụ Thêm địa chỉ sẽ được thực hiện. - Nếu admin chọn biểu tượng xóa thì luồng phụ Xóa nhân viên sẽ được thực hiện. <p>* Các luồng phụ:</p> <p>I – Sửa thông tin nhân viên</p> <ol style="list-style-type: none"> 1. Hệ thống hiển thị các thông tin nhân viên cần chỉnh sửa. 2. Admin chỉnh sửa các thông tin của nhân viên nếu có và chọn lưu thay đổi. 3. Hệ thống cập nhật lại thông tin nhân viên và trả về trang

	<p>danh sách nhân viên.</p> <p>II – Xem thông tin nhân viên</p> <p>Usecase này được thực hiện khi admin nhấp vào biểu tượng xem thông tin. Hệ thống sẽ hiển thị thông tin chi tiết của nhân viên cần xem.</p> <p>III – Thêm địa chỉ</p> <ol style="list-style-type: none"> 1. Hệ thống hiển thị giao diện để admin thêm địa chỉ cho nhân viên. 2. Admin chọn tên tỉnh thành, quận huyện, và điền địa chỉ cụ thể của nhân viên. 3. Hệ thống cập nhật địa chỉ của nhân viên. <p>IV – Xóa nhân viên</p> <p>Usecase này được thực hiện khi admin chọn biểu tượng xóa. Hệ thống sẽ hỏi admin có chắc chắn xóa hay không. Nếu admin nhấp chọn xóa thì thuộc tính Is_active của sản phẩm sẽ chuyển thành false. Nếu admin chọn quay lại thì sẽ chuyển đến trang danh sách sản phẩm.</p>
--	--

Bảng 3.7: Đặc tả usecase quản lý nhân viên

3.2.2.8. Usecase quản lý sản phẩm

Mô tả	Use case cho phép admin thực hiện các chức năng thêm, xóa và cập nhật thông tin sản phẩm của quản lý sản phẩm
Actor chính	Admin
Actor phụ	Không có
Tiền điều kiện	Admin đã đăng nhập thành công vào hệ thống
Luồng hoạt động	<ol style="list-style-type: none"> 1. Admin chọn chức năng quản lý sản phẩm 2. Hệ thống sẽ hiển thị các tùy chọn chức năng gồm danh mục sản phẩm, sản phẩm. 3. Nếu admin chọn “Danh mục sản phẩm” thì hệ thống sẽ chuyển đến trang danh mục sản phẩm và trang này hiển thị tất cả các loại sản phẩm. Tại đây, admin có thể tiến hành thêm danh mục, với mỗi danh mục admin có thể xóa, sửa hoặc xem danh sách sản phẩm theo từng danh mục sản phẩm. Nếu admin chọn “Sản phẩm” thì hệ thống sẽ chuyển đến

	trang sản phẩm và trang này sẽ hiển thị tất cả các sản phẩm. Tại đây, admin có thể tiến hành thêm sản phẩm, với mỗi sản phẩm admin có thể tiến hành sửa, xóa, thêm hình ảnh, kích thước hoặc xem chi tiết sản phẩm.
--	---

Bảng 3.8: *Đặc tả usecase quản lý sản phẩm*

3.2.2.9. Usecase thống kê

Mô tả	Use case cho phép admin thống kê doanh thu theo năm và theo tháng.
Actor chính	Admin
Actor phụ	Không có
Tiền điều kiện	Admin đã đăng nhập thành công vào hệ thống.
Luồng hoạt động	<p>1. Khách hàng nhấp chọn lịch sử mua hàng</p> <p>2. Hệ thống sẽ hiển thị các tùy chọn chức năng gồm thống kê theo tháng, thống kê theo năm.</p> <p>3. Nếu admin chọn “Thống kê theo tháng” thì luồng phụ Thống kê theo tháng được thực hiện. Nếu admin chọn “Thống kê theo năm” thì luồng phụ Thống kê theo năm được thực hiện.</p> <p>* Các luồng phụ:</p> <p>I – Thống kê theo tháng</p> <p>1. Hệ thống hiển thị trang màn hình thống kê theo tháng, mặc định là tháng hiện tại.</p> <p>2. Admin chọn tháng cần thống kê.</p> <p>3. Hệ thống hiển thị thống kê doanh thu, lợi nhuận theo tháng mà admin yêu cầu.</p> <p>II – Thống kê theo năm</p> <p>1. Hệ thống hiển thị trang màn hình thống kê theo năm, mặc định là năm hiện tại.</p> <p>2. Admin chọn năm cần thống kê.</p> <p>3. Hệ thống hiển thị thống kê doanh thu, lợi nhuận theo năm mà admin yêu cầu</p>

Bảng 3.9: *Đặc tả usecase thống kê*

3.2.2.10. Usecase quản lý kho

Mô tả	Use case cho phép admin và nhân viên lượng hàng trong kho và nhập hàng.
Actor chính	Admin, nhân viên
Actor phụ	Không có
Tiền điều kiện	Admin hoặc nhân viên đã đăng nhập thành công vào hệ thống.
Luồng hoạt động	<p>1. Admin hoặc nhân viên chọn chức năng quản lý kho.</p> <p>2. Hệ thống sẽ hiển thị các tùy chọn chức năng gồm số lượng hàng tồn và nhập kho.</p> <p>3. Nếu admin hoặc nhân viên chọn “Số lượng hàng tồn” thì luồng phụ Số lượng hàng tồn sẽ được thực hiện. Nếu admin hoặc nhân viên chọn “Nhập kho” thì luồng phụ Nhập kho sẽ được thực hiện.</p> <p>* Các luồng phụ:</p> <p>I – Số lượng hàng tồn</p> <p>Usecase được thực hiện khi admin hoặc nhân viên nhấp chọn số lượng hàng tồn. Hệ thống sẽ chuyển đến trang số lượng hàng tồn và trang này hiển thị số lượng hàng tồn của từng sản phẩm.</p> <p>II – Nhập kho</p> <p>1. Hệ thống hiển thị giao diện nhập kho.</p> <p>2. Admin hoặc nhân viên cung cấp các thông tin yêu cầu như tên sản phẩm, số lượng nhập, đơn giá.</p> <p>3. Hệ thống sẽ cộng dồn số lượng nhập vào số lượng có trong kho của sản phẩm đó.</p>

Bảng 3.10: Đặc tả usecase quản lý kho

3.2.2.11. Usecase quản lý khách hàng

Mô tả	Use case cho phép admin và nhân viên xem danh sách khách hàng, với mỗi khách hàng admin và nhân viên có thể xem thông tin và lịch sử mua hàng của khách hàng.
Actor chính	Admin, nhân viên
Actor phụ	Không có

Tiền điều kiện	Admin hoặc nhân viên đã đăng nhập thành công vào hệ thống.
Luồng hoạt động	<p>1. Admin hoặc nhân viên chọn chức năng quản lý khách hàng</p> <p>2. Hệ thống sẽ hiển thị tùy chọn chức năng là danh sách khách hàng.</p> <p>3. Admin hoặc nhân viên chọn danh sách khách hàng, hệ thống chuyển đến trang danh sách khách hàng và trang này hiển thị danh sách tất cả các khách hàng của shop. Tại trang này, admin và nhân viên có thể xem thông tin và lịch sử mua hàng của từng khách hàng.</p> <ul style="list-style-type: none"> - Nếu admin hoặc nhân viên chọn biểu tượng xem thông tin thì luồng phụ Xem thông tin khách hàng được thực hiện. - Nếu admin hoặc nhân viên chọn “Lịch sử mua hàng” thì luồng phụ Lịch sử mua hàng được thực hiện. <p>* Các luồng phụ:</p> <p>I – Xem thông tin khách hàng</p> <p>Usecase này được thực hiện khi admin hoặc nhân viên chọn vào biểu tượng xem thông tin của một khách hàng. Hệ thống sẽ hiển thị các thông tin của khách hàng cần xem.</p> <p>II – Lịch sử mua hàng</p> <p>Usecase này được thực hiện khi admin hoặc nhân viên chọn “Lịch sử mua hàng” của một khách hàng. Hệ thống sẽ hiển thị các thông tin các đơn hàng mà khách hàng đó đã mua nếu có, nếu không thì sẽ hiển thị chưa có đơn hàng nào được tạo.</p>

Bảng 3.11: Đặc tả usecase quản lý khách hàng

3.2.2.12. Usecase xem thông tin đặt hàng

Mô tả	Use case cho phép admin và nhân viên xem thông tin đặt hàng của khách hàng.
Actor chính	Admin, nhân viên
Actor phụ	Không có
Tiền điều kiện	Admin hoặc nhân viên đã đăng nhập thành công vào hệ thống.
Luồng hoạt động	1. Admin hoặc nhân viên chọn chức năng thông tin đặt hàng.

	<p>2. Hệ thống sẽ hiển thị các tùy chọn chức năng gồm danh sách đơn hàng và đơn hàng chưa duyệt.</p> <p>3. Nếu admin hoặc nhân viên chọn “Danh sách đơn hàng” thì luồng phụ Danh sách đơn hàng được thực hiện. Nếu admin hoặc nhân viên chọn “Đơn hàng chưa duyệt” thì luồng phụ Duyệt đơn hàng sẽ được thực hiện.</p> <p>* Các luồng phụ:</p> <p>I – Danh sách đơn hàng</p> <p>Use case này được thực hiện khi admin hoặc nhân viên chọn “Danh sách đơn hàng”. Hệ thống sẽ chuyển đến trang danh sách đơn hàng và trang này hiển thị danh sách các đơn hàng của các khách hàng nếu có. Ngược lại sẽ hiển thị chưa có đơn hàng nào được tạo.</p> <p>II – Duyệt đơn hàng</p> <p>1. Hệ thống hiển thị danh sách các đơn hàng chưa được duyệt của các khách hàng. Tại đây, admin và nhân viên có thể xem thông tin đơn hàng và duyệt đơn hàng.</p> <p>2. Nếu admin hoặc nhân viên chọn vào biểu tượng xem thông tin thì hệ thống sẽ chuyển đến trang thông tin đơn hàng và hiển thị thông tin của đơn hàng ở trang này. Nếu admin hoặc nhân viên chọn vào biểu tượng duyệt đơn hàng thì thuộc tính Is_check của đơn hàng sẽ chuyển thành true và đơn hàng đã được duyệt.</p>
--	---

Bảng 3.12: Đặc tả usecase xem thông tin đặt hàng

3.2.2.13. Use case Chatbot

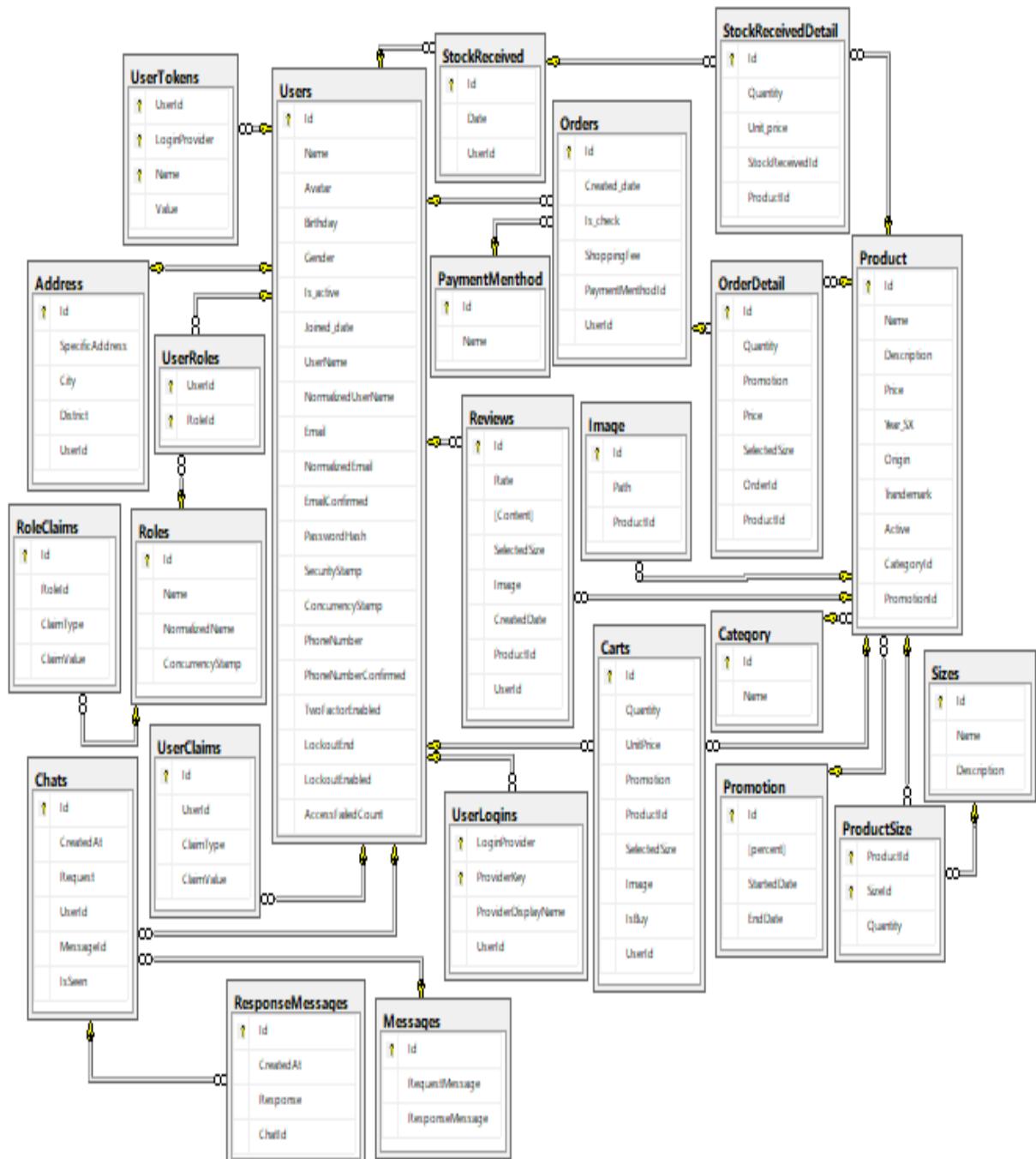
Mô tả	Use case cho phép người dùng gửi và trả lời tin nhắn hoặc trả lời bằng tin nhắn tự động khi tin nhắn gửi từ khách hàng.
Actor chính	Người dùng
Actor phụ	Không có
Luồng hoạt động	1. Người dùng nhập tin nhắn và nhấp nút gửi.

2.Nếu người dùng là admin hoặc nhân viên, hệ thống sẽ chuyển tin nhắn đến khách hàng tương ứng.

Nếu người dùng là khách hàng, hệ thống sẽ gửi một tin nhắn tự động tương ứng với yêu cầu của khách hàng hoặc tin nhắn thông báo sẽ trả lời sau. Kết thúc usecase.

Bảng 3.13: Đặc tả usecase chatbot

3.2.3. Mô hình cơ sở dữ liệu



Hình 3.2: Mô hình cơ sở dữ liệu

- Các bảng trong mô hình cơ sở dữ liệu:
- Bảng Users: dùng để lưu thông tin của người dùng. Bảng này kế thừa bảng Users trong ASP.NET Core Identity. Hai bảng Users và Roles có mối quan hệ nhiều – nhiều và bảng trung gian được tạo ra từ mối quan hệ nhiều – nhiều này là bảng UserRoles.
- Bảng Roles: dùng để lưu thông tin các vai trò của người dùng. Bảng này kế thừa bảng Roles trong ASP.NET Core Identity. Hai bảng Roles và Users có mối quan hệ nhiều – nhiều và bảng trung gian được tạo ra từ mối quan hệ nhiều – nhiều này là bảng UserRoles.
- Bảng UserRoles: dùng để lưu người dùng ứng với vai trò nào. Bảng này kế thừa bảng UserRoles trong ASP.NET Core Identity. Đây là bảng trung gian trong mối quan hệ nhiều – nhiều giữa bảng Users và bảng Roles. Trong bảng có hai khóa ngoại là UserId tham chiếu đến thuộc tính Id trong bảng Users và RoleId tham chiếu đến thuộc tính Id trong bảng Roles.
- Bảng RoleClaims: dùng để lưu các thiết lập yêu cầu của từng vai trò người dùng. Bảng này kế thừa bảng RoleClaims trong ASP.NET Core Identity. Hai bảng RoleClaims và Roles có mối quan hệ một – nhiều. Thuộc tính RoleId đóng vai trò là khóa ngoại tham chiếu đến thuộc tính khóa chính Id trong bảng Roles.
- Bảng UserClaims: dùng để lưu các thiết lập yêu cầu của từng người dùng. Bảng này kế thừa bảng UserClaims trong ASP.NET Core Identity. Hai bảng UserClaims và Users có mối quan hệ một – nhiều. Thuộc tính UserId đóng vai trò là khóa ngoại tham chiếu đến thuộc tính khóa chính Id trong bảng Users.
- Bảng UserTokens: dùng để lưu thông tin token của người dùng. Bảng này kế thừa bảng UserTokens trong ASP.NET Core Identity. Hai bảng UserTokens và Users có mối quan hệ một – nhiều. Thuộc tính UserId đóng vai trò là khóa ngoại tham chiếu đến thuộc tính khóa chính Id trong bảng Users.
- Bảng UserLogins: dùng để lưu thông tin đăng nhập bên thứ ba của người dùng. Bảng này kế thừa bảng UserLogins trong ASP.NET Core Identity. Hai bảng UserLogins và Users có mối quan hệ một – nhiều. Thuộc tính UserId đóng vai trò là khóa ngoại tham chiếu đến thuộc tính khóa chính Id trong bảng Users.
- Bảng Address: dùng để lưu thông tin địa chỉ của người dùng. Bảng này và bảng Users có mối quan hệ một – nhiều. Thuộc tính UserId đóng vai trò là khóa ngoại

tham chiếu đến thuộc tính khóa chính Id trong bảng Users và thuộc tính này được thiết lập ràng buộc cấm trùng trong bảng Address.

- Bảng Reviews: dùng để lưu các phản hồi đánh giá của khách hàng. Bảng này và bảng Users có mối quan hệ một – nhiều. Thuộc tính UserId đóng vai trò là khóa ngoại tham chiếu đến thuộc tính khóa chính Id trong bảng Users.
- Bảng PaymentMenthod: dùng để lưu các phương thức thanh toán. Bảng này và bảng Orders có mối quan hệ một – nhiều.
- Bảng StockReceived: dùng để lưu thông tin nhập kho. Bảng này và bảng Users có mối quan hệ một – nhiều. Thuộc tính UserId đóng vai trò là khóa ngoại tham chiếu đến thuộc tính khóa chính Id trong bảng Users. Ngoài ra, bảng StockReceived cũng có mối quan hệ nhiều – nhiều với bảng Product và bảng trung gian được tạo ra từ mối quan hệ này là StockReceivedDetail.
- Bảng Product: dùng để lưu thông tin của sản phẩm. Bảng này và bảng Category có mối quan hệ một – nhiều. Thuộc tính CategoryId đóng vai trò là khóa ngoại tham chiếu đến thuộc tính khóa chính Id trong bảng Category. Ngoài ra, bảng Product và bảng Promotion cũng có mối quan hệ một – nhiều. Thuộc tính PromotionId đóng vai trò là khóa ngoại tham chiếu đến thuộc tính khóa chính Id trong bảng Promotion. Bên cạnh đó, bảng Product còn có mối quan hệ nhiều – nhiều với bảng StockReceived, bảng Orders và bảng Sizes với bảng trung gian của hai mối quan hệ nhiều – nhiều này lần lượt là StockReceivedDetail, OrderDetail và ProductSize.
- Bảng StockReceivedDetail: được dùng để lưu thông tin chi tiết nhập hàng. Đây là bảng trung gian trong mối quan hệ nhiều – nhiều giữa bảng Product và bảng StockReceived. Trong bảng có hai khóa ngoại là ProductId tham chiếu đến thuộc tính Id trong bảng Product và StockReceivedId tham chiếu đến thuộc tính Id trong bảng StockReceived.
- Bảng Orders: dùng để lưu thông tin đặt hàng của khách hàng. Bảng này và bảng PaymentMenthod có mối quan hệ một – nhiều. Thuộc tính PaymentMenthodId đóng vai trò là khóa ngoại tham chiếu đến thuộc tính khóa chính Id trong bảng PaymentMenthod. Ngoài ra, bảng Orders và bảng Users cũng có mối quan hệ một – nhiều. Thuộc tính UserId đóng vai trò là khóa ngoại tham chiếu đến thuộc tính khóa chính Id trong bảng User. Bên cạnh đó, bảng Orders còn có mối quan

hệ nhiều – nhiều với bảng Product với bảng trung gian của mối quan hệ nhiều – nhiều này là bảng OrderDetail.

- Bảng OrderDetail: dùng để lưu thông tin chi tiết đặt hàng của khách hàng. Đây là bảng trung gian trong mối quan hệ nhiều – nhiều giữa bảng Product và bảng Orders. Trong bảng có hai khóa ngoại là ProductId tham chiếu đến thuộc tính Id trong bảng Product và OrderId tham chiếu đến thuộc tính Id trong bảng Orders.
- Bảng Promotion: dùng để lưu thông tin các chương trình khuyến mãi. Bảng này và bảng Product có mối quan hệ một – nhiều.
- Bảng Category: dùng để lưu danh mục sản phẩm. Bảng này và bảng Product có mối quan hệ một – nhiều.
- Bảng Image: dùng để lưu hình ảnh của sản phẩm. Bảng này và bảng Product có mối quan hệ một – nhiều. Thuộc tính ProductId đóng vai trò là khóa ngoại tham chiếu đến thuộc tính khóa chính Id trong bảng Product.
- Bảng Carts: dùng để lưu thông tin các sản phẩm mà khách hàng cho vào giỏ hàng. Bảng này và bảng Product có mối quan hệ một – nhiều. Thuộc tính ProductId đóng vai trò là khóa ngoại tham chiếu đến thuộc tính khóa chính Id trong bảng Product. Ngoài ra, bảng Carts và bảng Users cũng có mối quan hệ một – nhiều. Thuộc tính UserId đóng vai trò là khóa ngoại tham chiếu đến thuộc tính khóa chính Id trong bảng Users.
- Bảng Sizes: dùng để lưu thông tin kích thước sản phẩm. Bảng này và bảng Product có mối quan hệ nhiều – nhiều và bảng trung gian của mối quan hệ nhiều – nhiều này là bảng ProductSize.
- Bảng ProductSize: dùng để lưu thông tin sản phẩm với những kích thước tương ứng của nó. Đây là bảng trung gian trong mối quan hệ nhiều – nhiều giữa bảng Product và bảng Sizes. Trong bảng có hai khóa ngoại là ProductId tham chiếu đến thuộc tính Id trong bảng Product và SizeId tham chiếu đến thuộc tính Id trong bảng Sizes.
- Bảng Messages: dùng để lưu các tin nhắn tự động. Bảng này và bảng Users có mối quan hệ nhiều – nhiều và bảng trung gian của mối quan hệ nhiều – nhiều này là bảng Chats.
- Bảng Chats: dùng để lưu thông tin của cuộc trò chuyện với những tin nhắn gửi và trả lời ứng với từng người dùng. Đây là bảng trung gian trong mối quan hệ

nhiều – nhiều giữa bảng Messages và bảng Users. Trong bảng có hai khóa ngoại là MessageId tham chiếu đến thuộc tính Id trong bảng Messages và UserId tham chiếu đến thuộc tính Id trong bảng Users. Ngoài ra, bảng này và bảng ResponseMessages cũng có mối quan hệ một – nhiều.

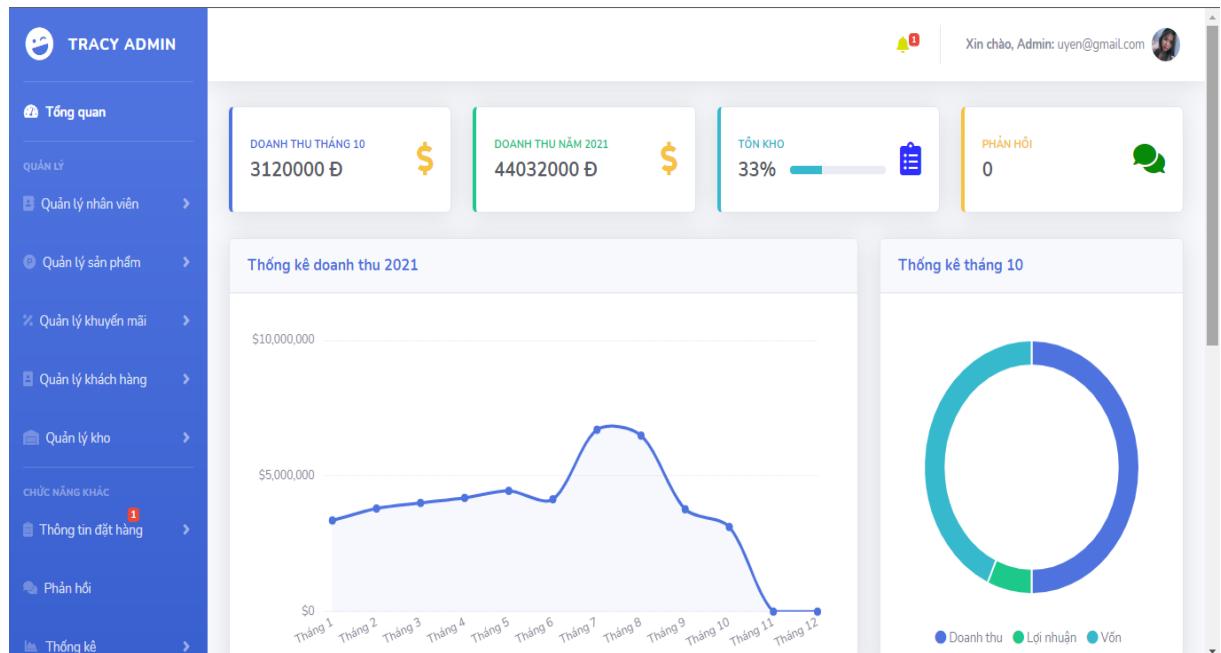
- Bảng ResponseMessages: dùng để lưu các tin nhắn trả lời từ admin hoặc nhân viên đến khách hàng. Bảng này và bảng Chats có mối quan hệ một – nhiều. Thuộc tính ChatId đóng vai trò là khóa ngoại tham chiếu đến thuộc tính khóa chính Id trong bảng Chats.

3.3. Thực nghiệm hệ thống

Đề tài “Xây dựng website bán quần áo nữ online Tracy” được xây dựng với ba loại người dùng là admin, nhân viên và khách hàng. Trong đó, admin và nhân viên đóng vai trò quản trị.

3.3.1. Giao diện của Admin

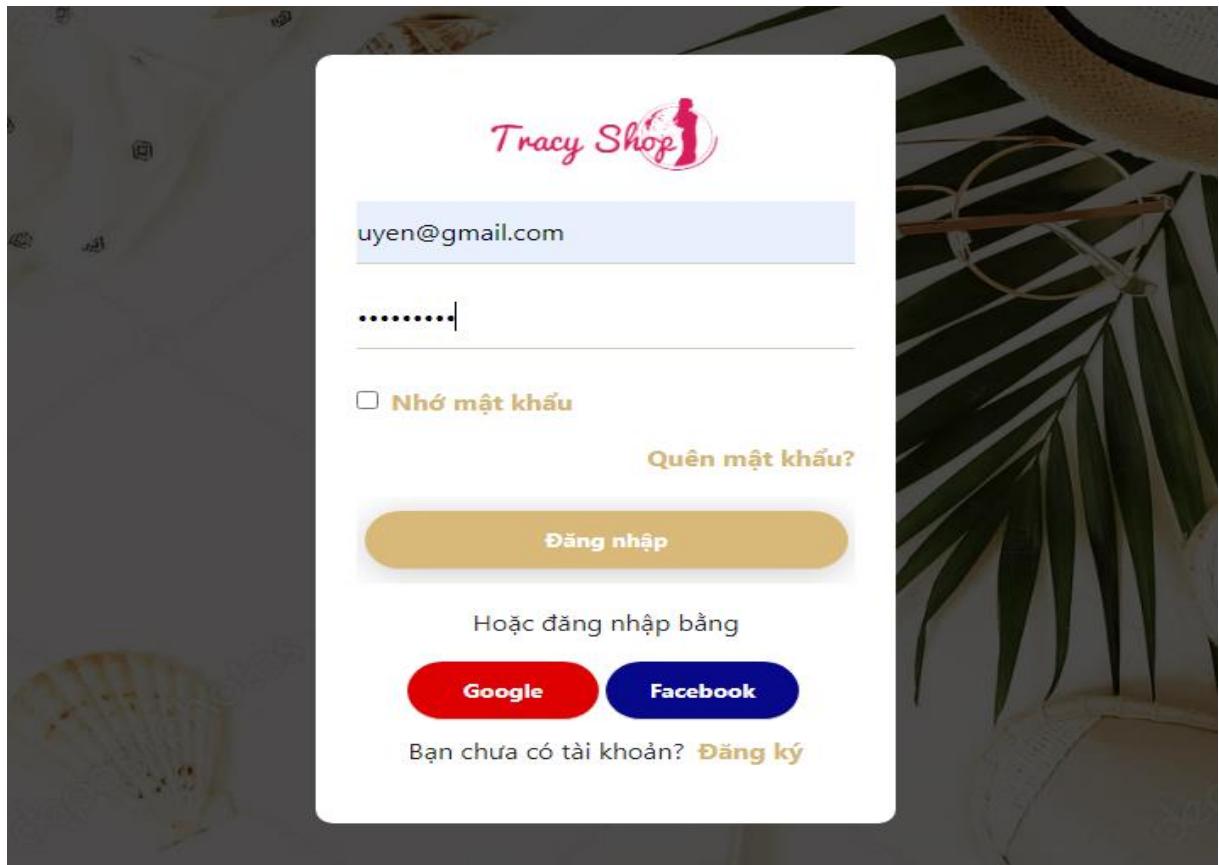
Với vai trò quản trị, admin được toàn quyền thực hiện các chức năng quản trị của hệ thống từ các chức năng quản lý như quản lý nhân viên, quản lý sản phẩm, quản lý khuyến mãi, quản lý khách hàng, quản lý kho đến các chức năng khác như đăng nhập, đăng xuất, theo dõi thông tin đặt hàng, các phản hồi đánh giá từ khách hàng và thống kê doanh thu, lợi nhuận của cửa hàng.



Hình 3.3: Giao diện tổng quan của Admin

3.3.1.1. Chức năng đăng nhập

Chức năng này cho phép admin đăng nhập với tài khoản đã được tạo sẵn khi triển khai hệ thống.



Hình 3.4: Giao diện đăng nhập của admin

3.3.1.2. Chức năng Quản lý nhân viên

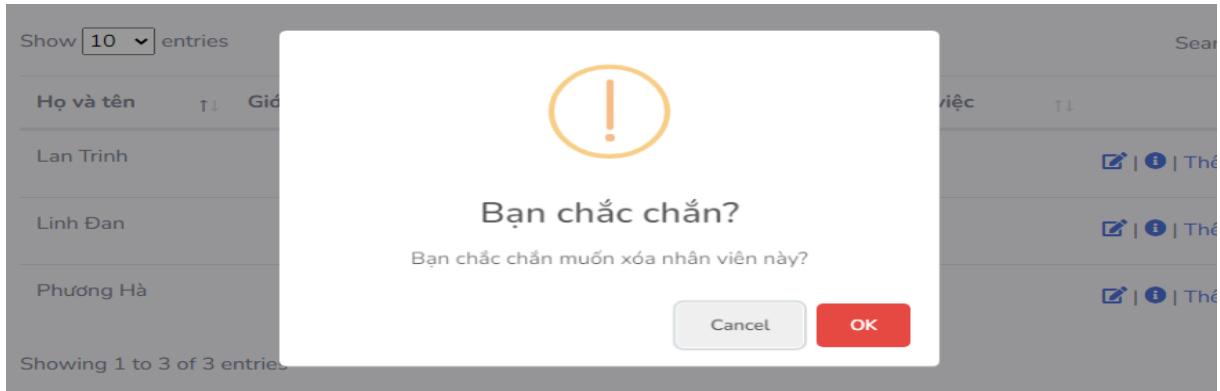
Chức năng này cho phép admin thực hiện các chức năng thêm, sửa, xóa, thêm địa chỉ và xem thông tin của nhân viên.

- Xem danh sách nhân viên:

A screenshot of the 'TRACY ADMIN' dashboard. On the left, there is a sidebar with a user icon and the text 'TRACY ADMIN'. Below this are several menu items under 'QUẢN LÝ': 'Quản lý nhân viên', 'Quản lý sản phẩm', 'Quản lý khuyến mãi', 'Quản lý khách hàng', 'Quản lý kho', and 'Quản lý MessageBot'. The main content area is titled 'Quản lý nhân viên'. It shows a table with three rows of employee data. The columns are: Họ và tên (Name), Giới tính (Gender), Ngày sinh (Date of Birth), Địa chỉ (Address), and Đang làm việc (Current Work). The first row has data: Lan Trinh, Nữ, 6/12/2001 10:39:02 PM, and two icons for edit and delete. The second row has data: Linh Đan, Nữ, 10/25/2000 10:39:02 PM, and two icons for edit and delete. The third row has data: Phương Hà, Nữ, 8/8/2000 10:39:02 PM, and two icons for edit and delete. At the top right of the main area, there is a message 'Xin chào, Admin: uyen@gmail.com' and a 'Thêm nhân viên' (Add employee) button. At the bottom right, there are buttons for 'Previous', '1', and 'Next'.

Hình 3.5: Giao diện xem danh sách nhân viên của admin

- Xóa nhân viên:



Hình 3.6: Giao diện xác nhận xóa nhân viên của admin

Nếu admin nhập vào “Xóa” thì nó sẽ xóa nhân viên đó và chuyển đến trang hiển thị danh sách nhân viên của cửa hàng.

- Xem thông tin nhân viên:

The screenshot shows a page titled "Thông tin nhân viên". It displays the following information for an employee:
 Họ và tên: Nguyễn Thị Vân Anh
 Giới tính: Nữ
 Ngày sinh: 08/08/2000 12:00:00 AM
 Email: vananh@gmail.com
 Số điện thoại: 1234567890
 At the top right, there is a blue button labeled "Chỉnh sửa thông tin". On the left, there is a sidebar with navigation links: Tổng quan, Quản lý, Quản lý nhân viên, Quản lý sản phẩm, Quản lý khuyến mãi, Quản lý khách hàng, Quản lý kho, Thông tin đặt hàng, and Phản hồi.

Hình 3.7: Giao diện xem thông tin nhân viên của admin

- Chính sửa thông tin nhân viên:

The screenshot shows a page titled "Chỉnh sửa thông tin nhân viên". It contains the following form fields:
 Họ và tên:
 Giới tính: Nam Nữ Khác
 Ngày sinh:
 Số điện thoại:
 Đang hoạt động:
 At the bottom right is a blue button labeled "Lưu thông tin". On the left, there is a sidebar with navigation links: Tổng quan, Quản lý, Quản lý nhân viên, Quản lý sản phẩm, Quản lý khuyến mãi, Quản lý khách hàng, Quản lý kho, Thông tin đặt hàng, and Phản hồi.

Hình 3.8: Giao diện chỉnh sửa thông tin nhân viên của admin

- Thêm địa chỉ cho nhân viên:

Địa chỉ nhân viên

Tỉnh / Thành phố

Quận / Huyện

Địa chỉ cụ thể

Hình 3.9: Giao diện địa chỉ nhân viên của admin

- Thêm nhân viên:

Thêm nhân viên mới

Tên nhân viên

Ngày sinh

Địa chỉ Email

Mật khẩu

Giới tính Nam Nữ Khác

Số điện thoại

Hình 3.10: Giao diện thêm nhân viên của admin

3.3.1.3. Chức năng Quản lý sản phẩm

Chức năng này cho phép admin thêm, sửa, xóa, xem thông tin và danh sách sản phẩm. Đồng thời, ở chức năng này, admin có thể xem, thêm, chỉnh sửa danh mục và tìm kiếm sản phẩm.

- Xem danh mục sản phẩm:

ID	Tên danh mục	
1	Áo	<input checked="" type="checkbox"/> Xem sản phẩm
2	Áo khoác	<input checked="" type="checkbox"/> Xem sản phẩm
3	Chân váy	<input checked="" type="checkbox"/> Xem sản phẩm
4	Đầm	<input checked="" type="checkbox"/> Xem sản phẩm
5	Quần	<input checked="" type="checkbox"/> Xem sản phẩm
6	Thời trang mặc nhà	<input checked="" type="checkbox"/> Xem sản phẩm

Hình 3.11: Giao diện danh mục sản phẩm của admin

- Thêm danh mục sản phẩm:

Hình 3.12: Giao diện thêm danh mục sản phẩm của admin

- Chính sửa danh mục sản phẩm:

Hình 3.13: Giao diện chỉnh sửa tên danh mục sản phẩm của admin

- Xem danh sách sản phẩm:

Hình 3.14: Giao diện xem danh sách sản phẩm của admin

- Tìm kiếm sản phẩm:

Hình 3.15: Giao diện trả về danh sách sản phẩm sau khi tìm kiếm của admin

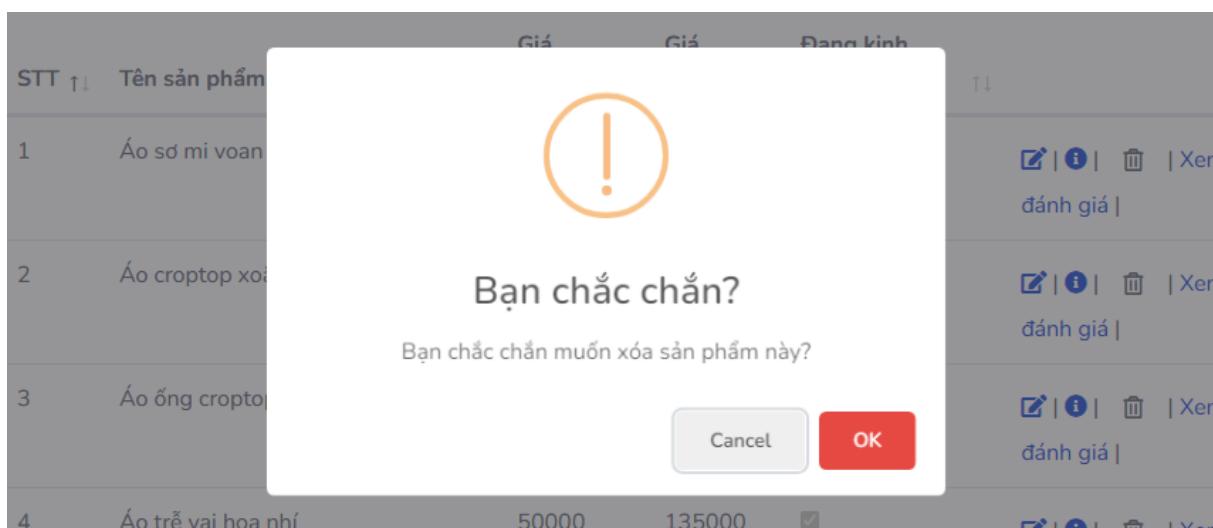
- Thêm sản phẩm:

Hình 3.16: Giao diện thêm sản phẩm mới của admin

- Chính sửa thông tin sản phẩm:

Hình 3.17: Giao diện chỉnh sửa thông tin sản phẩm của admin

- Xóa sản phẩm:



Hình 3.18: Giao diện xác nhận xóa sản phẩm của admin

- Xem chi tiết thông tin sản phẩm:

Hình 3.19: Giao diện xem thông tin sản phẩm của admin

3.3.1.4. Chức năng Quản lý khuyến mãi

Chức năng này cho phép admin thêm, sửa, xóa và xem danh sách khuyến mãi.

- Xem danh sách khuyến mãi:

ID	Mức giảm giá	Ngày bắt đầu	Ngày kết thúc
1	0	5/11/2022 12:00:00 AM	11/30/2022 12:00:00 AM
2	0.25	5/11/2022 12:00:00 AM	5/15/2022 12:00:00 AM
3	0.3	5/13/2022 12:00:00 AM	5/20/2022 12:00:00 AM

Hình 3.20: Giao diện xem danh sách khuyến mãi của admin

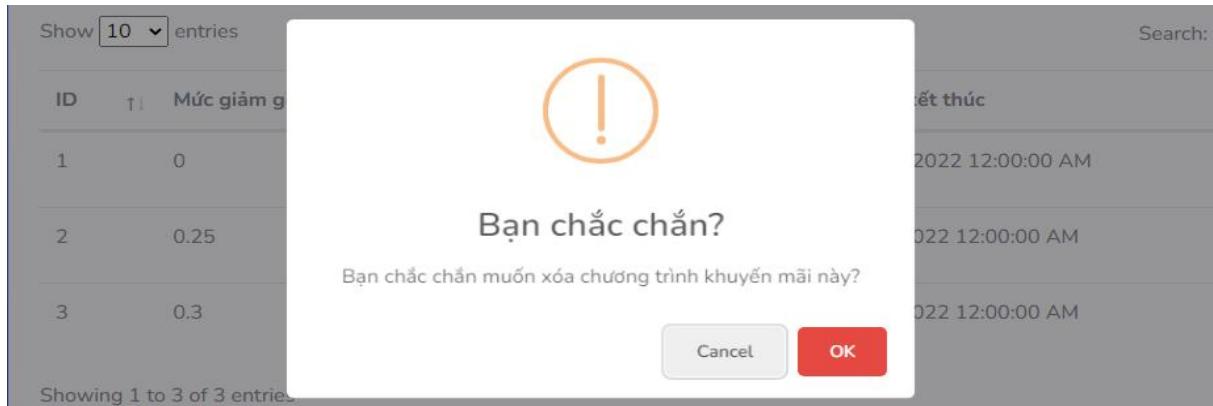
- Thêm khuyến mãi:

Hình 3.21: Giao diện thêm khuyến mãi của admin

- Chính sửa khuyến mãi:

Hình 3.22: Giao diện chỉnh sửa khuyến mãi của admin

- Xóa khuyến mãi:



Hình 3.23: Giao diện xác nhận xóa khuyến mãi của admin

3.3.1.5. Chức năng Quản lý khách hàng

Chức năng này cho phép admin xem thông tin và danh sách khách hàng, đồng thời xem lịch sử mua hàng của khách hàng.

- Xem danh sách khách hàng:

Họ và tên	Giới tính	Ngày sinh	Ngày đăng ký	Đang hoạt động
Bảo Linh			5/8/2022 10:39:02 PM	<input checked="" type="checkbox"/>
Dương Dương			5/8/2022 10:39:02 PM	<input checked="" type="checkbox"/>
Gia Hân			5/8/2022 10:39:02 PM	<input checked="" type="checkbox"/>
Hướng Ly			5/8/2022 10:39:02 PM	<input checked="" type="checkbox"/>
Khánh Chi			5/8/2022 10:39:02 PM	<input checked="" type="checkbox"/>

Hình 3.24: Giao diện xem danh sách khách hàng của admin

- Xem lịch sử mua hàng của khách hàng:

Id	Ngày mua	Phương thức thanh toán	Giá trị đơn hàng	Đã duyệt
6	5/9/2022 1:23:06 PM	Thanh toán PayPal	1330000 VND	<input checked="" type="checkbox"/>

Hình 3.25: Giao diện xem lịch sử mua hàng của khách hàng trong admin

- Xem thông tin khách hàng:

Họ và tên	Bảo Linh
Giới tính	Nữ
Ngày sinh	8/8/2000 10:39:02 PM
Email	baolinh@gmail.com
Số điện thoại	0123456789
Địa chỉ	35 Võ Văn Ngân, Quận Thủ Đức, Thành phố Hồ Chí Minh

Hình 3.26: Giao diện xem thông tin khách hàng

3.3.1.6. Chức năng Quản lý kho

Chức năng này cho phép admin xem số lượng hàng tồn trong kho, nhập kho và xem lịch sử nhập hàng.

- Xem số lượng hàng tồn trong kho:

ID	Tên sản phẩm	Số lượng tồn
1	Áo sơ mi voan họa tiết loan màu	4
4	Áo croptop xoắn eo body	9
5	Áo ống croptop đan dây eo	8
6	Áo trẻ vai hoa nhí	14
7	Áo thun tay lõi loan màu	3

Hình 3.27: Giao diện xem số lượng hàng tồn trong kho của admin

- Xem chi tiết thông tin số lượng hàng tồn của từng sản phẩm:

ID	Tên sản phẩm	Kích thước	Số lượng
1	Áo sơ mi voan họa tiết loan màu	M	0
2	Áo sơ mi voan họa tiết loan màu	L	4

Hình 3.28: Giao diện xem chi tiết số lượng hàng tồn một sản phẩm của admin

- Nhập kho:

Hình 3.29: Giao diện nhập kho của admin

- Xem lịch sử nhập hàng:

Mã nhập kho	Người nhập kho	Ngày nhập	
1	Uyên Nguyễn	5/1/2022 4:16:55 AM	<i>(edit)</i>
2	Uyên Nguyễn	5/1/2022 4:16:55 AM	<i>(edit)</i>
3	Uyên Nguyễn	5/1/2022 4:16:55 AM	<i>(edit)</i>
4	Uyên Nguyễn	5/1/2022 4:16:55 AM	<i>(edit)</i>
5	Uyên Nguyễn	5/1/2022 4:16:55 AM	<i>(edit)</i>

Hình 3.30: Giao diện xem thông tin lịch sử nhập kho của admin

- Chi tiết nhập kho:

ID	Tên sản phẩm	Số lượng	Giá nhập
1	Áo sơ mi voan họa tiết loan màu	15	150000

Showing 1 to 1 of 1 entries

Hình 3.31: Giao diện xem chi tiết nhập kho của admin

3.3.1.7. Chức năng Quản lý MessageBot

Chức năng này cho phép admin quản lý xem danh sách, xem chi tiết, tạo mới, chỉnh sửa và xóa các tin nhắn tự động.

- Xem danh sách MessageBot

Hình 3.32: Giao diện xem danh sách MessageBot của admin

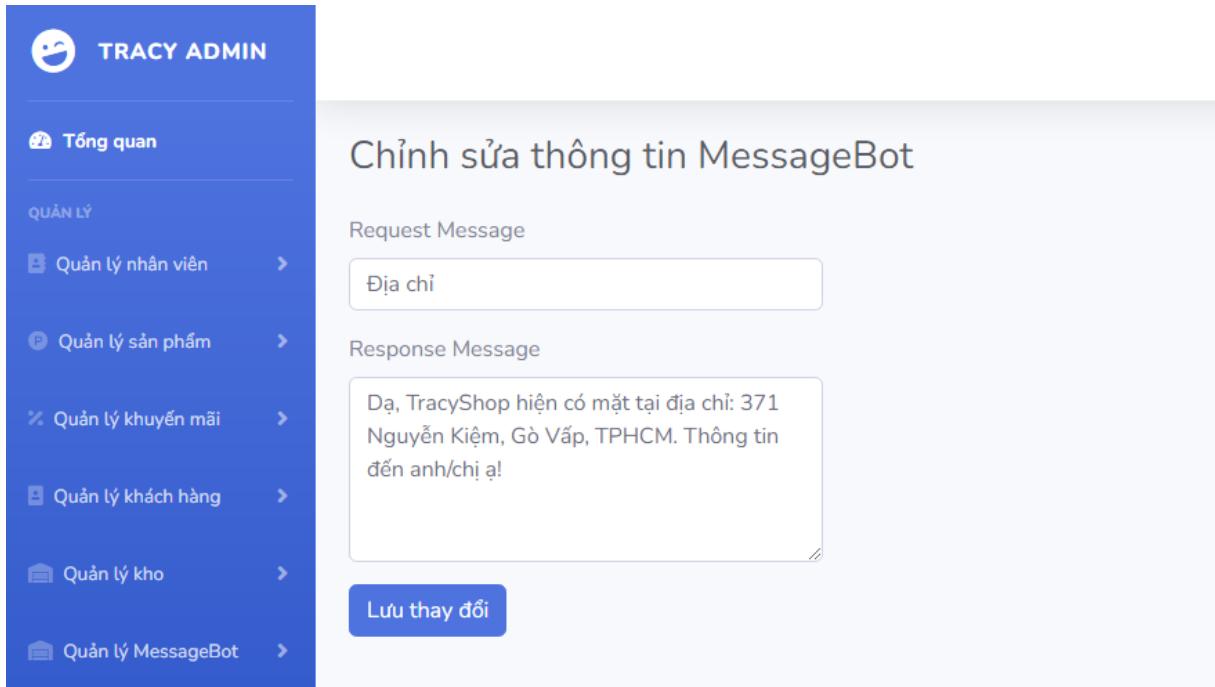
- Xem thông tin chi tiết của MessageBot

Hình 3.33: Giao diện xem thông tin chi tiết của MessageBot của admin

- Tạo mới một MessageBot

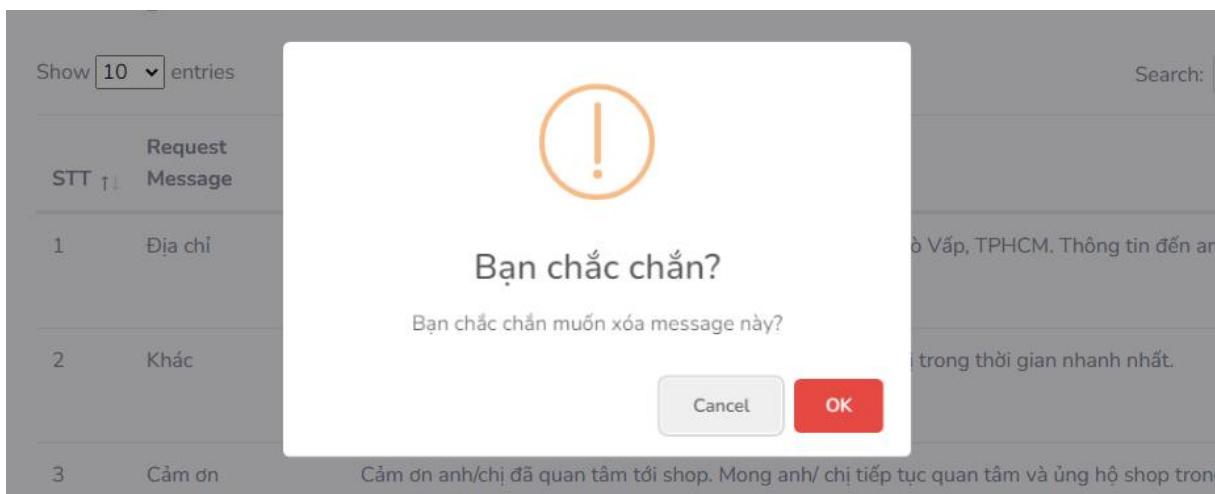
Hình 3.34: Giao diện thêm MessageBot của admin

- Chính sửa MessageBot



Hình 3.35: Giao diện chỉnh sửa MessageBot của admin

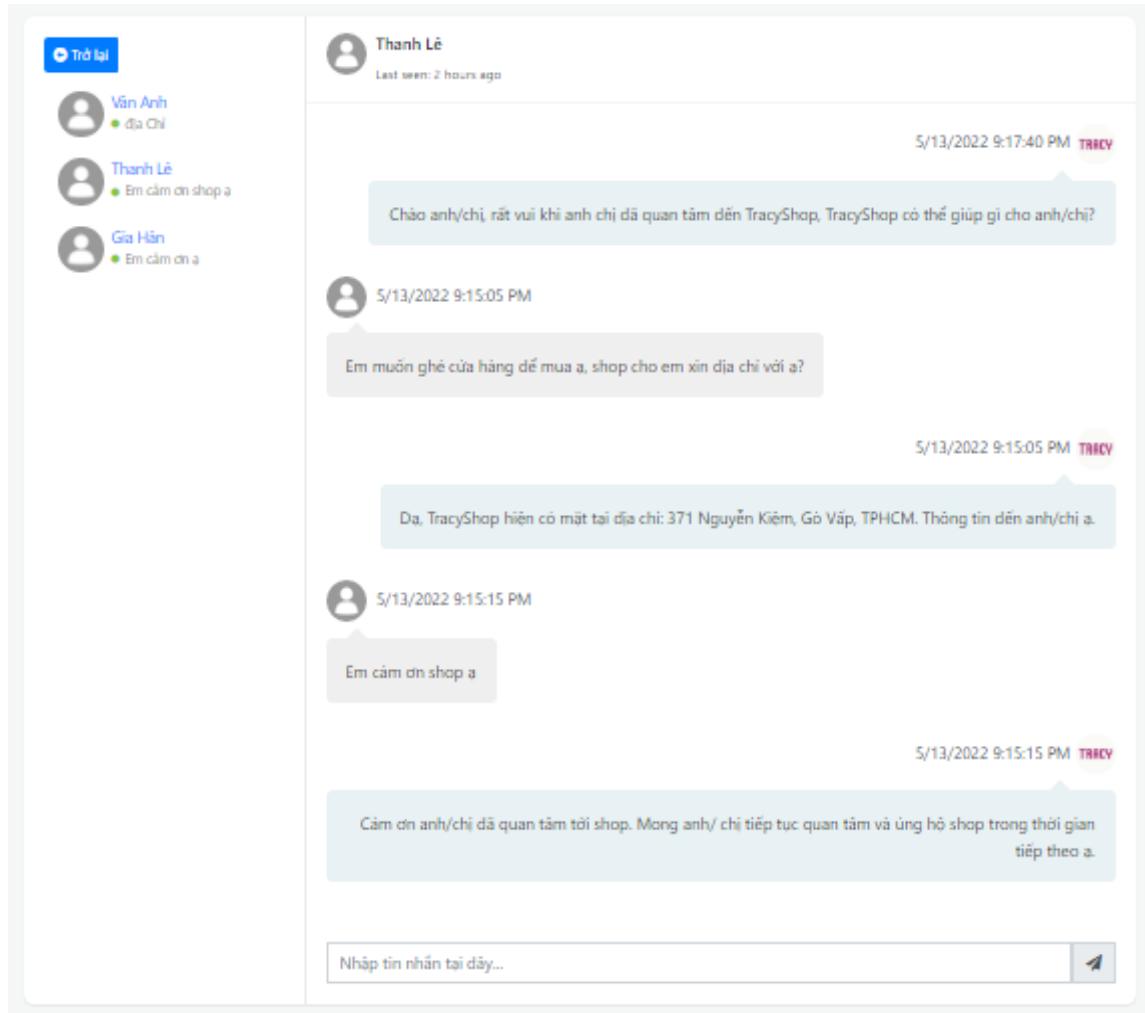
- Xóa một MessageBot



Hình 3.36: Giao diện xóa một MessageBot của admin

3.3.1.8. Chức năng Chatbot

Chức năng này cho phép admin xem và trả lời tin nhắn thắc mắc của từng khách hàng gửi đến shop để được nhờ tư vấn.



Hình 3.37: Giao diện chức năng chatbot của admin

3.3.1.9. Chức năng Xem thông tin đặt hàng

Chức năng này cho phép admin xem danh sách tất cả đơn hàng, danh sách các đơn hàng chưa duyệt và duyệt đơn hàng của khách hàng.

- Xem danh sách tất cả đơn hàng:

TRACY ADMIN		Danh sách tất cả đơn hàng													
		Xin chào, Admin: uyenv@gmail.com													
Tổng quan															
QUẢN LÝ															
Quản lý nhân viên		ID	Ngày mua	Tên khách hàng	Phương thức thanh toán	Giá trị đơn hàng	Đã duyệt								
Quản lý sản phẩm		1	5/8/2022 1:23:06 PM	Hương Ly	Thanh toán khi nhận hàng	645000 VND	<input checked="" type="checkbox"/>								
Quản lý khuyến mãi		2	5/8/2022 1:23:06 PM	Vân Anh	Thanh toán PayPal	990000 VND	<input checked="" type="checkbox"/>								
Quản lý khách hàng		3	5/8/2022 1:23:06 PM	Nhu Trúc	Thanh toán khi nhận hàng	900000 VND	<input checked="" type="checkbox"/>								
Quản lý kho		4	5/8/2022 1:23:06 PM	Thanh Tâm	Thanh toán PayPal	790000 VND	<input checked="" type="checkbox"/>								
Quản lý MessageBot		5	5/9/2022 1:23:06 PM	Tú Quyên	Thanh toán khi nhận hàng	730000 VND	<input checked="" type="checkbox"/>								

Hình 3.38: Giao diện xem danh sách tất cả đơn hàng của admin

- Xem thông tin chi tiết đơn hàng:

Hình 3.39: Giao diện xem thông tin chi tiết đơn hàng của admin

- Xem danh sách đơn hàng chưa duyệt:

Hình 3.40: Giao diện xem danh sách các đơn hàng chưa duyệt của admin

Để duyệt đơn hàng, admin nhấp vào biểu tượng có dấu tích nền xanh bên phải đơn hàng ở trang danh sách đơn hàng chưa duyệt.

3.3.1.10. Chức năng Xem phản hồi đánh giá

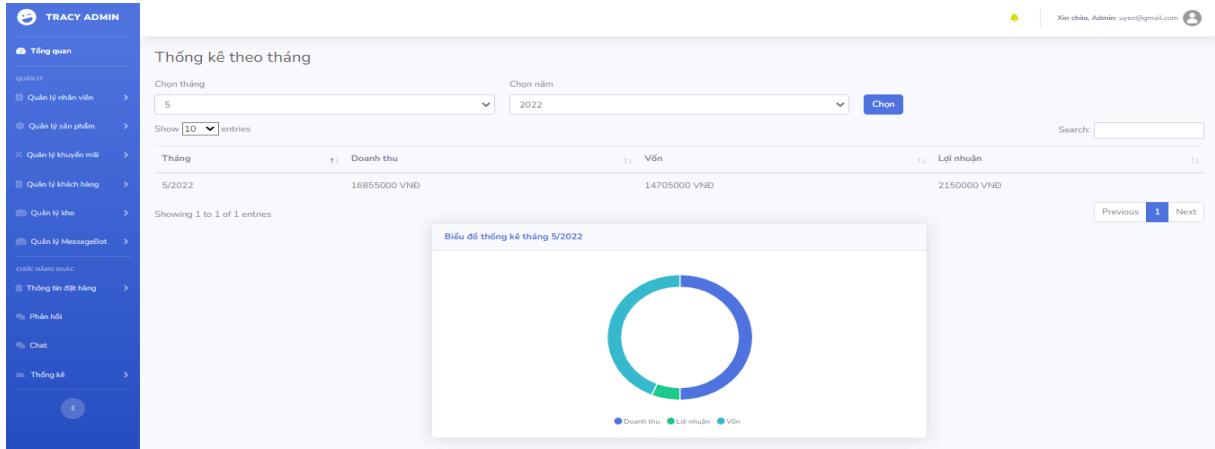
Chức năng này cho phép admin xem thông tin đánh giá phản hồi từ khách hàng.

Hình 3.41: Giao diện xem đánh giá sản phẩm từ khách hàng của admin

3.3.1.11. Chức năng Thống kê

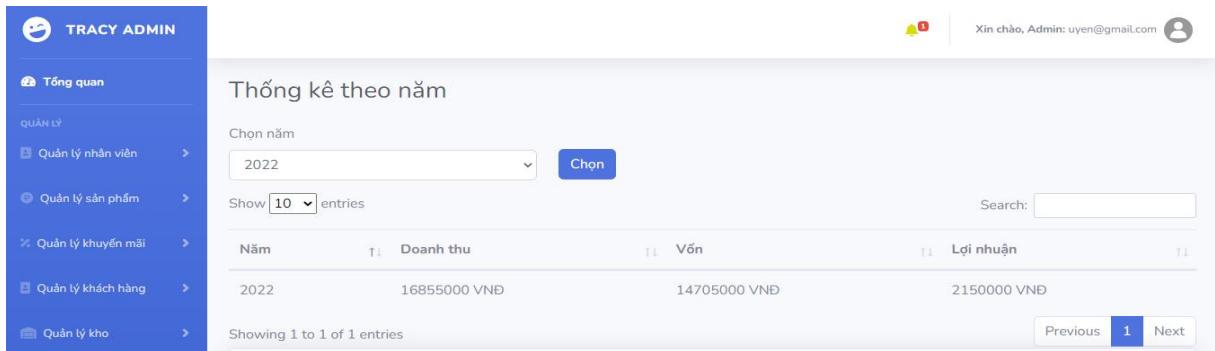
Chức năng này dùng để thống kê theo tháng và thống kê theo năm. Những thứ cần thống kê gồm: doanh thu, lợi nhuận, vốn.

- Thống kê theo tháng:



Hình 3.42: Giao diện thống kê theo tháng của admin

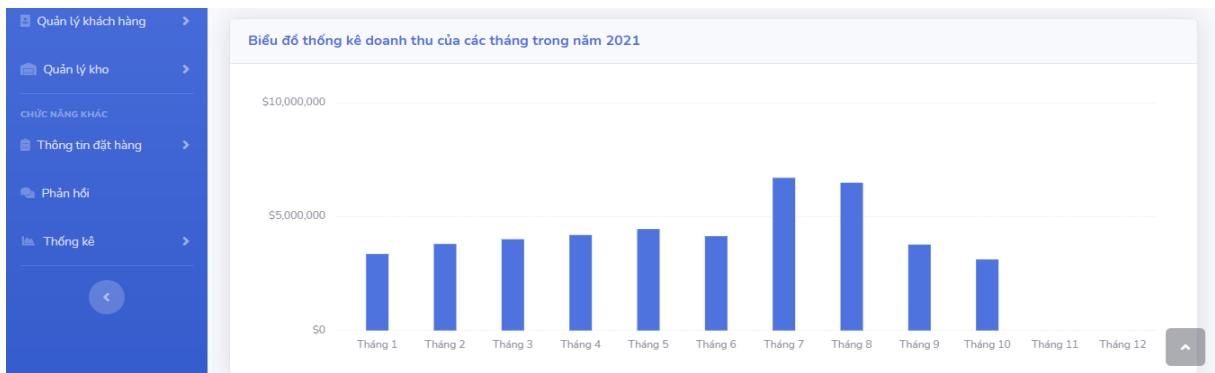
- Thống kê theo năm:



Hình 3.43: Giao diện thống kê theo năm của admin

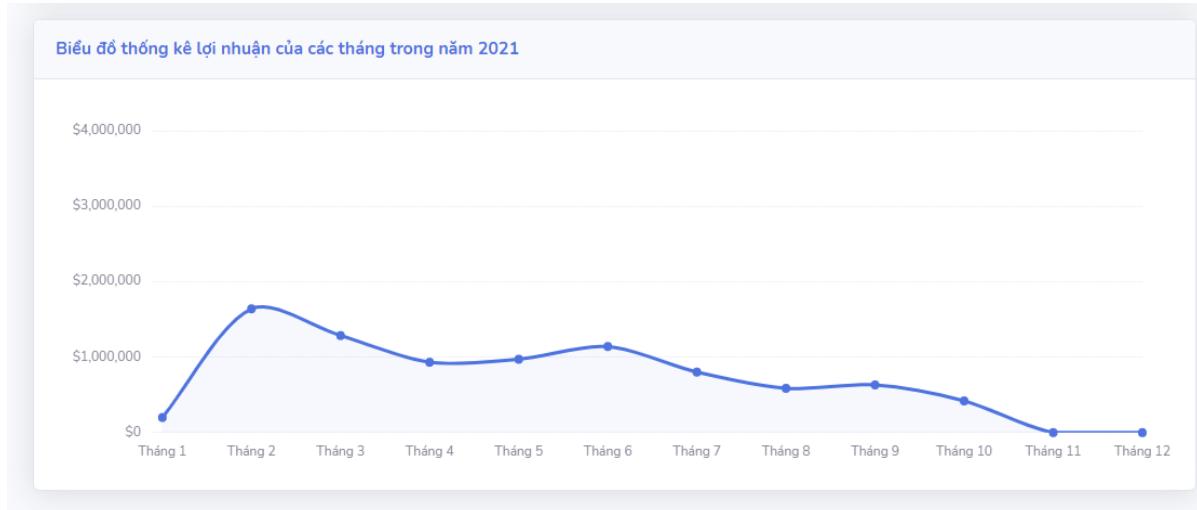
Các biểu đồ được hiển thị trong “Thống kê theo năm”:

- Biểu đồ cột thống kê doanh thu của các tháng trong năm:



Hình 3.44: Biểu đồ cột thống kê doanh thu các tháng trong năm của admin

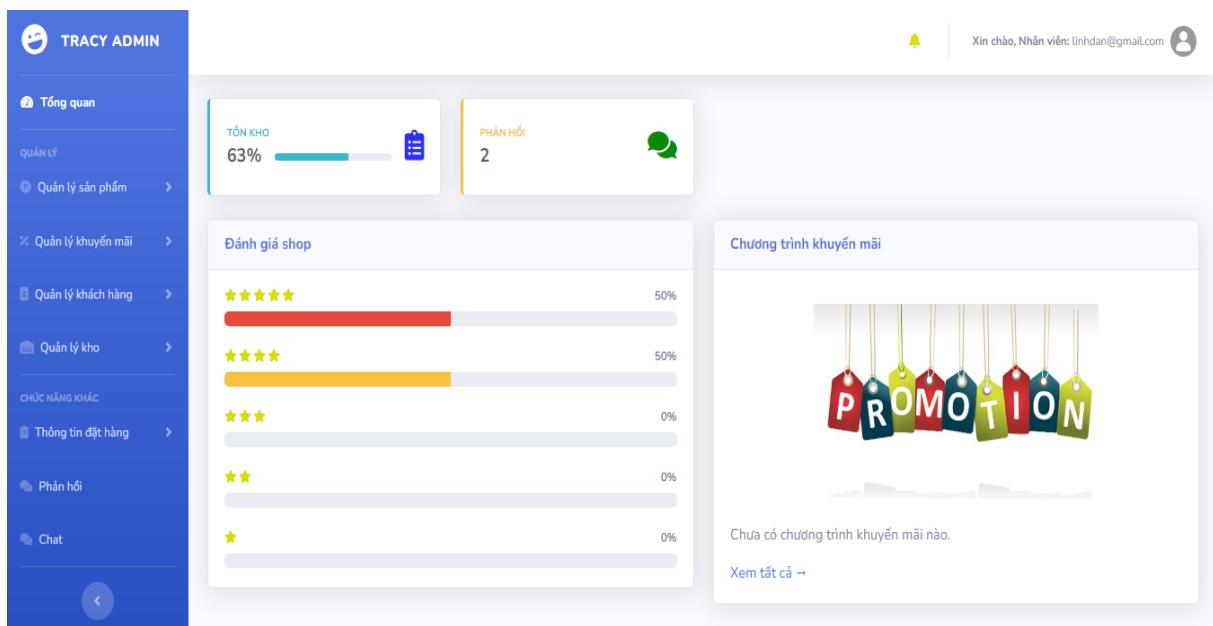
- Biểu đồ đường thống kê lợi nhuận của các tháng trong năm:



Hình 3.45: Biểu đồ đường thống kê doanh thu các tháng trong năm của admin

3.3.2. Giao diện của Nhân viên

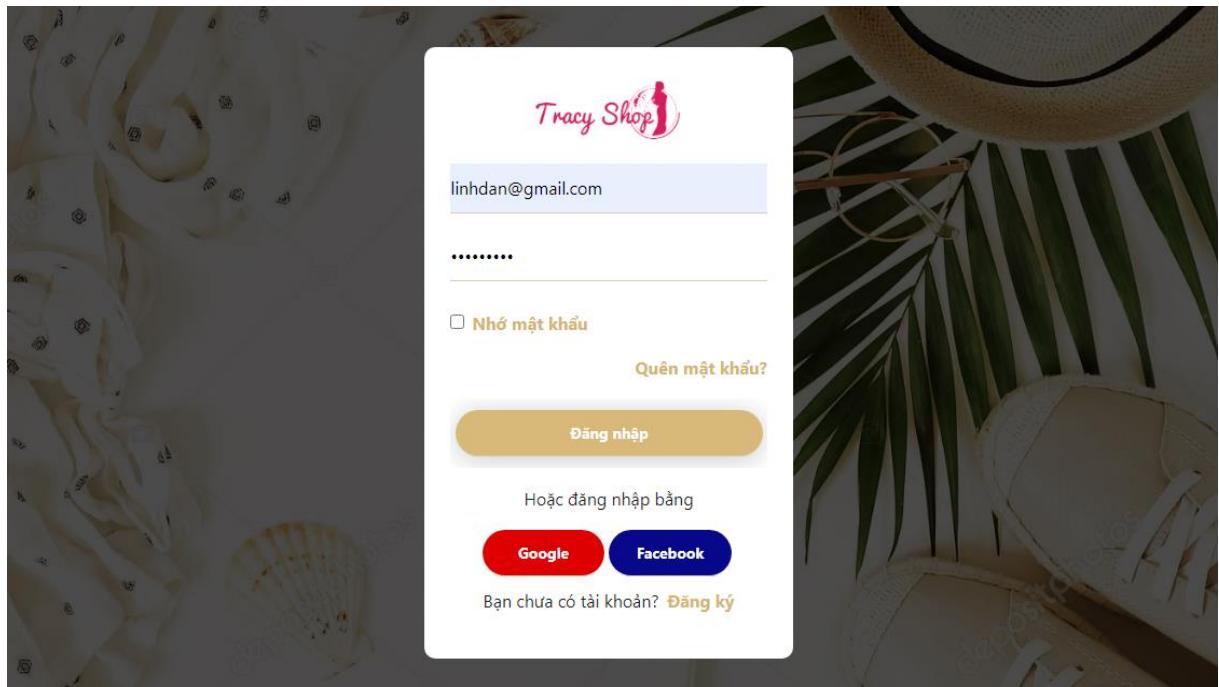
Tương tự như admin, nhân viên được quyền thực hiện các chức năng quản trị của hệ thống nhưng bị giới hạn ở một số chức năng. Các chức năng mà nhân viên có quyền được thực hiện như quản lý sản phẩm, quản lý khuyến mãi, quản lý khách hàng, quản lý kho và các chức năng khác như đăng nhập, đăng xuất, theo dõi thông tin đặt hàng, xem các phản hồi đánh giá từ khách hàng. Ở chức năng quản lý sản phẩm, nhân viên chỉ được quyền xem danh mục sản phẩm, thông tin và chi tiết sản phẩm. Và chức năng quản lý khuyến mãi, nhân viên chỉ được quyền xem danh sách các chương trình khuyến mãi.



Hình 3.46: Giao diện tổng quan của Nhân viên

3.3.2.1. Chức năng Đăng nhập

Chức năng này cho phép nhân viên đăng nhập với tài khoản đã được tạo bởi admin khi thêm nhân viên.



Hình 3.47: Giao diện đăng nhập của nhân viên

3.3.2.2. Chức năng Quản lý sản phẩm

Chức năng này cho phép nhân viên xem thông tin về danh mục và sản phẩm.

- Xem danh mục sản phẩm:

A screenshot of the "TRACY ADMIN" dashboard. On the left, a sidebar lists navigation options: Tổng quan, QUẢN LÝ (Quản lý sản phẩm, Quản lý khuyến mãi, Quản lý khách hàng, Quản lý kho, Quản lý MessageBot), CHỨC NĂNG KHÁC (Thông tin đặt hàng, Phản hồi). The main content area shows a table titled "Danh mục sản phẩm" with columns: ID, Tên danh mục, and actions (Xem sản phẩm). The table contains 6 entries: 1. Áo, 2. Áo khoác, 3. Chân váy, 4. Đầm, 5. Quần, 6. Thời trang mặc nhà. At the bottom, it says "Showing 1 to 6 of 6 entries". The top right shows a greeting "Xin chào, Nhân viên: linhdan@gmail.com" and user icons.

Hình 3.48: Giao diện danh mục sản phẩm của nhân viên

- Xem danh sách sản phẩm:

STT	Tên sản phẩm	Giá nhập	Giá bán	Đang kinh doanh	
1	Áo sơ mi voan họa tiết loan màu	150000	350000	<input checked="" type="checkbox"/>	Xem số lượng Xem ảnh Xem đánh giá
2	Áo croptop xoắn eo body	50000	150000	<input checked="" type="checkbox"/>	Xem số lượng Xem ảnh Xem đánh giá
3	Áo ống croptop đan dây eo	50000	130000	<input checked="" type="checkbox"/>	Xem số lượng Xem ảnh Xem đánh giá

Hình 3.49: Giao diện xem danh sách sản phẩm của nhân viên

- Xem thông tin chi tiết sản phẩm:

Tên sản phẩm	Áo sơ mi voan họa tiết loan màu
Mô tả	Giá thấp nhất, chất lượng tốt nhất, chất liệu voan, độ đàn hồi cao, đẹp, không có xù vải, màu sắc đẹp, thoải mái, thoáng khí
Giá	350000
Năm sản xuất	12/1/2019 7:00:00 PM
Thương hiệu	MOXY
Xuất xứ	Hàn Quốc
Kinh doanh	<input checked="" type="checkbox"/>

Hình 3.50: Giao diện xem thông tin sản phẩm của nhân viên

- Tìm kiếm sản phẩm:

STT	Tên sản phẩm	Giá nhập	Giá bán	Đang kinh doanh
23	Đầm mixi trễ vai	0	375000	<input checked="" type="checkbox"/>

Hình 3.51: Giao diện trả về danh sách sản phẩm sau khi tìm kiếm của nhân viên

3.3.2.3. Chức năng Quản lý khuyến mãi

Ở chức năng này, nhân viên chỉ được quyền xem danh sách khuyến mãi.

Hình 3.52: Giao diện xem danh sách khuyến mãi của nhân viên

3.3.2.4. Chức năng Quản lý khách hàng

Giống với admin, ở chức năng này nhân viên cũng được quyền xem thông tin và danh sách khách hàng, đồng thời xem lịch sử mua hàng của khách hàng.

- Xem danh sách khách hàng:

Hình 3.53: Giao diện xem danh sách khách hàng của nhân viên

- Xem thông tin khách hàng:

Hình 3.54: Giao diện xem thông tin khách hàng của nhân viên

- Xem lịch sử mua hàng của khách hàng:

ID	Ngày mua	Phương thức thanh toán	Giá trị đơn hàng	Đã duyệt
6	5/9/2022 1:23:06 PM	Thanh toán PayPal	1330000 VND	<input checked="" type="checkbox"/>

Hình 3.55: Giao diện xem lịch sử mua hàng của khách hàng trong nhân viên

3.3.2.5. Chức năng Quản lý kho

Ở chức năng này, nhân viên được toàn quyền như với admin, nhân viên cũng được quyền thực hiện các chức năng xem số lượng hàng tồn trong kho, nhập kho và xem lịch sử nhập kho. Trong lịch sử nhập kho, nhân viên cũng được xem thông tin chi tiết của những lần nhập kho.

- Xem số lượng hàng tồn:

ID	Tên sản phẩm	Số lượng tồn
1	Áo sơ mi voan họa tiết loan màu	4
4	Áo croptop xoắn eo body	9
5	Áo ống croptop đan dây eo	8

Hình 3.56: Giao diện xem số lượng hàng tồn của nhân viên

- Nhập kho:

Hình 3.57: Giao diện nhập kho của nhân viên

- Xem lịch sử nhập kho:

Mã nhập kho	Người nhập kho	Ngày nhập	
1	Uyên Nguyễn	5/1/2022 4:16:55 AM	<i>(info)</i>
2	Uyên Nguyễn	5/1/2022 4:16:55 AM	<i>(info)</i>
3	Uyên Nguyễn	5/1/2022 4:16:55 AM	<i>(info)</i>

Hình 3.58: Giao diện xem lịch sử nhập kho của nhân viên

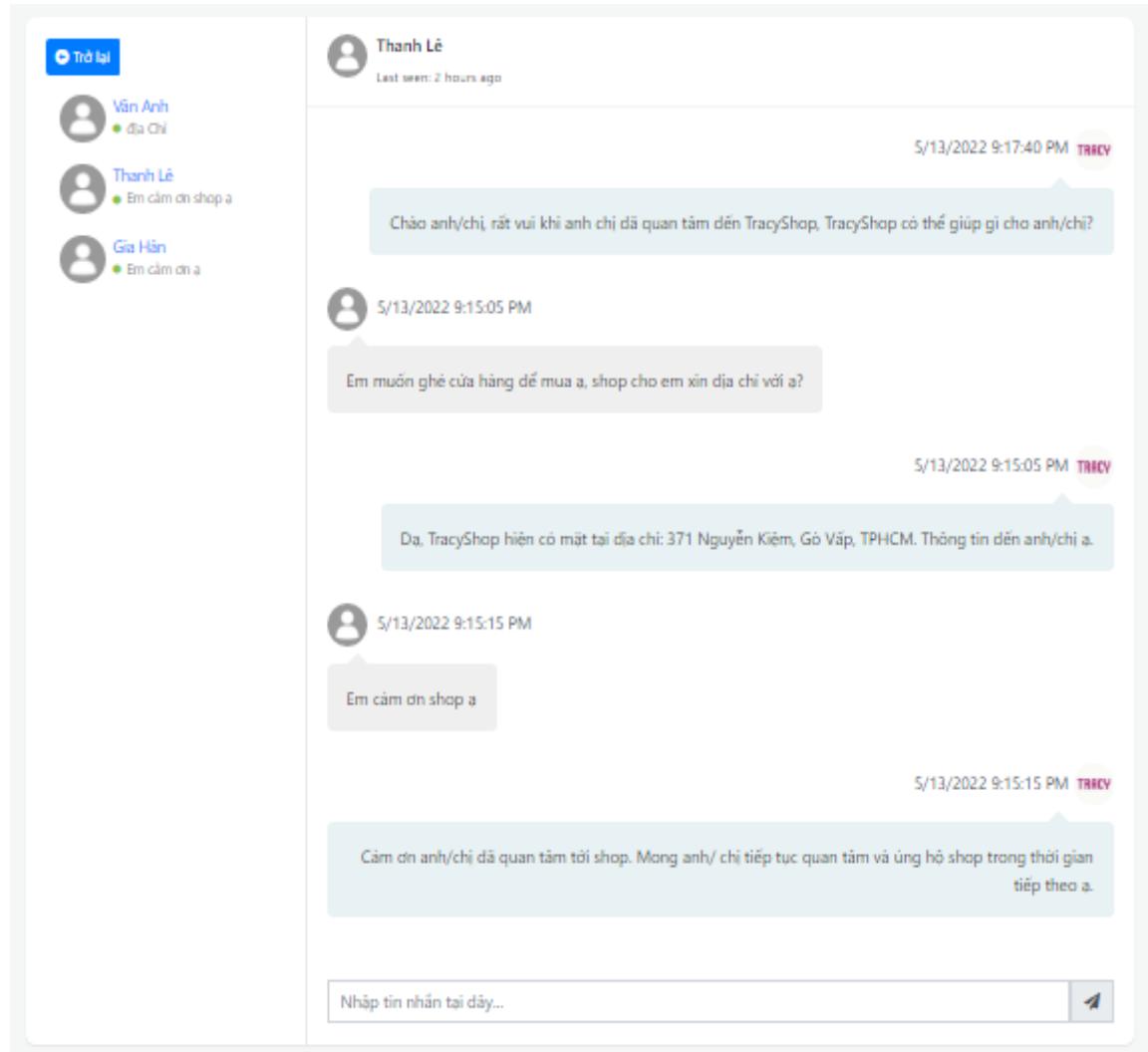
- Xem chi tiết nhập kho:

ID	Tên sản phẩm	Số lượng	Giá nhập
1	Áo sơ mi voan họa tiết loạn màu	15	150000

Hình 3.59: Giao diện xem thông tin chi tiết nhập kho của nhân viên

3.3.2.6. Chức năng Chatbot

Chức năng này cho phép nhân viên xem và trả lời tin nhắn thắc mắc của từng khách hàng gửi đến shop để được nhờ tư vấn.



Hình 3.60: Giao diện chức năng chatbot của nhân viên

3.3.2.7. Chức năng Xem thông tin đặt hàng

Chức năng Xem thông tin đặt hàng ở nhân viên cũng giống như ở admin, đều được xem danh sách tất cả đơn hàng, danh sách các đơn hàng chưa duyệt và duyệt đơn hàng của khách hàng. Trong danh sách các đơn hàng chưa duyệt hay danh sách tất cả đơn hàng, nhân viên đều được xem thông tin chi tiết của các đơn hàng đó.

- Xem danh sách tất cả đơn hàng:

Hình 3.61: Giao diện xem danh sách đơn hàng của nhân viên

- Xem danh sách các đơn hàng chưa duyệt và duyệt đơn hàng:

Hình 3.62: Giao diện xem danh sách các đơn hàng chưa duyệt của nhân viên

- Xem thông tin chi tiết đơn hàng:

Hình 3.63: Giao diện xem thông tin chi tiết đơn hàng của nhân viên

3.3.2.8. Chức năng Xem phản hồi đánh giá

Tương tự với admin, ở chức năng này admin được quyền xem những phản hồi đánh giá sản phẩm từ khách hàng.

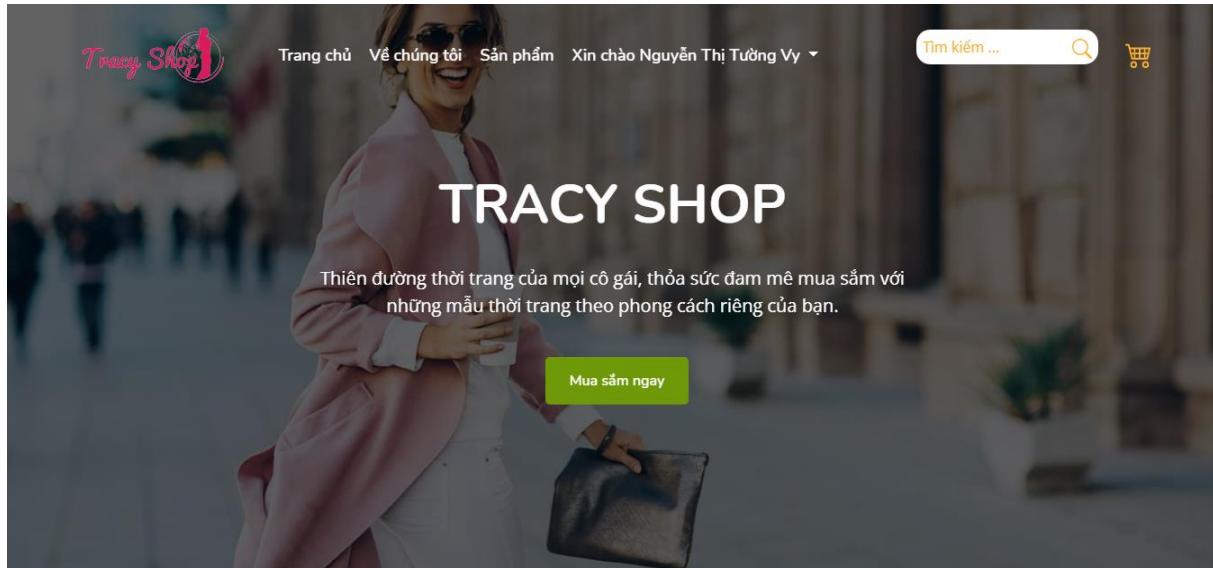
The screenshot shows a table of product reviews. The columns are: Khách hàng (Customer), ID, sản phẩm (Product), Bình luận (Comment), Đánh giá (Rating), Hình ảnh (Image), and Ngày tạo (Created Date). The first review is for a 'Áo sơ mi voan họa tiết loan màu' with a rating of 4/5. The second review is for a 'Áo khoác Bomber Jacket Dù Unisex' with a rating of 5/5. A small image of the white jacket is shown next to the second review.

Khách hàng	ID	sản phẩm	Bình luận	Đánh giá	Hình ảnh	Ngày tạo
Gia Hân	1	Áo sơ mi voan họa tiết loan màu	Đẹp xuất sắc luôn à ><	4 / 5		5/12/2022 3:38:24 PM
Gia Hân	2	Áo khoác Bomber Jacket Dù Unisex	Áo đẹp lắm à ><	5 / 5		5/12/2022 3:43:10 PM

Hình 3.64: Giao diện xem phản hồi đánh giá sản phẩm của nhân viên

3.3.3. Giao diện Khách hàng

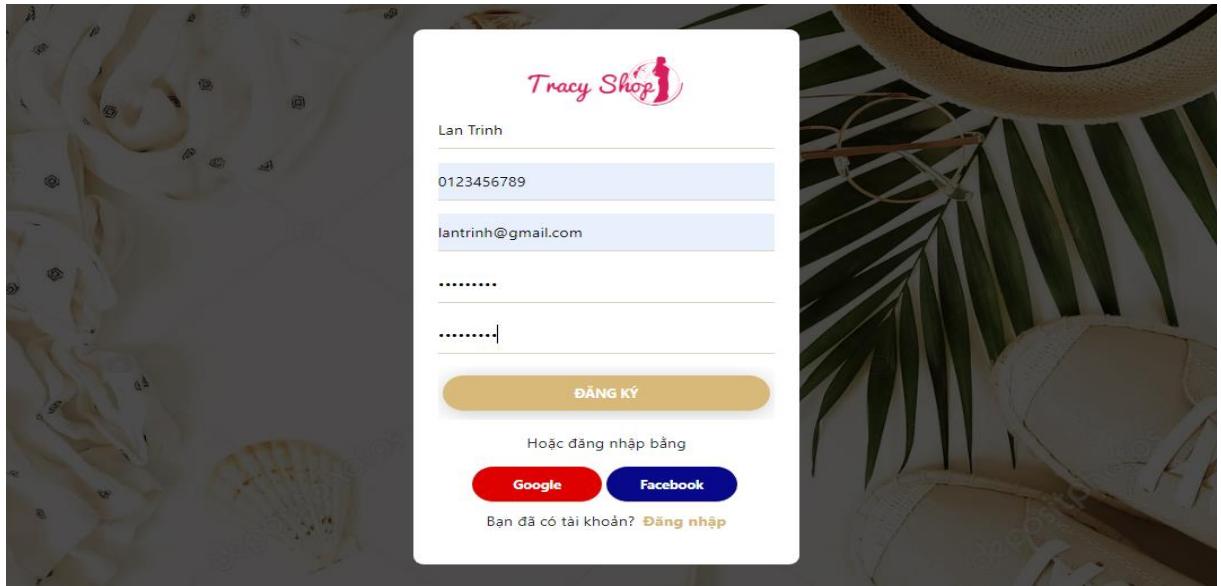
Khác với admin và nhân viên, khách hàng không được quyền thực hiện các chức năng quản trị. Các chức năng của khách hàng gồm đăng ký, đăng nhập, đăng xuất, đặt hàng, tra cứu, phản hồi đánh giá, quản lý tài khoản, quản lý giỏ hàng, xem lịch sử mua hàng, xem thông tin sản phẩm, thanh toán và chức năng quên mật khẩu.



Hình 3.65: Giao diện trang chủ của khách hàng

3.3.3.1. Chức năng Đăng ký

Chức năng này cho phép khách hàng đăng ký tài khoản người dùng để sử dụng một số chức năng cần được chứng thực.



Hình 3.66: Giao diện chức năng đăng ký của khách hàng

Sau khi nhấp vào nút đăng ký, hệ thống sẽ gửi một email xác thực đến email đăng ký của khách hàng và thông báo cho khách hàng biết để kiểm tra và xác thực.



Hình 3.67: Giao diện thông báo đã gửi email xác thực của khách hàng

Khách hàng đọc mail và nhấp vào liên kết để tiến hành xác thực email đăng ký.

Search...

Hello Lan Trinh, Confirm your email id.

From: TracyShop Team <no-reply@tracyshop.com>
To: <lantrinh@gmail.com>

2021-10-25 21:40, 1.4 KB

Show Headers

HTML [HTML Source](#) [Text](#) [Raw](#) [Spam Analysis](#) [HTML Check](#) [Tech Info](#)

Xin chào Lan Trinh,
Cảm ơn bạn đã sử dụng ứng dụng web TracyShop của chúng tôi
Bạn đã tạo một tài khoản mới trên ứng dụng TracyShop

Nhấp vào liên kết sau để xác thực địa chỉ email của bạn.

[Xác thực địa chỉ Email](#)

Hình 3.68: Giao diện mail xác thực địa chỉ email đăng ký

Sau khi khách hàng nhấp vào liên kết trong mail, nó sẽ chuyển đến trang thông báo xác thực email thành công nếu thành công, ngược lại sẽ hiển thị thông báo xác thực email thất bại.

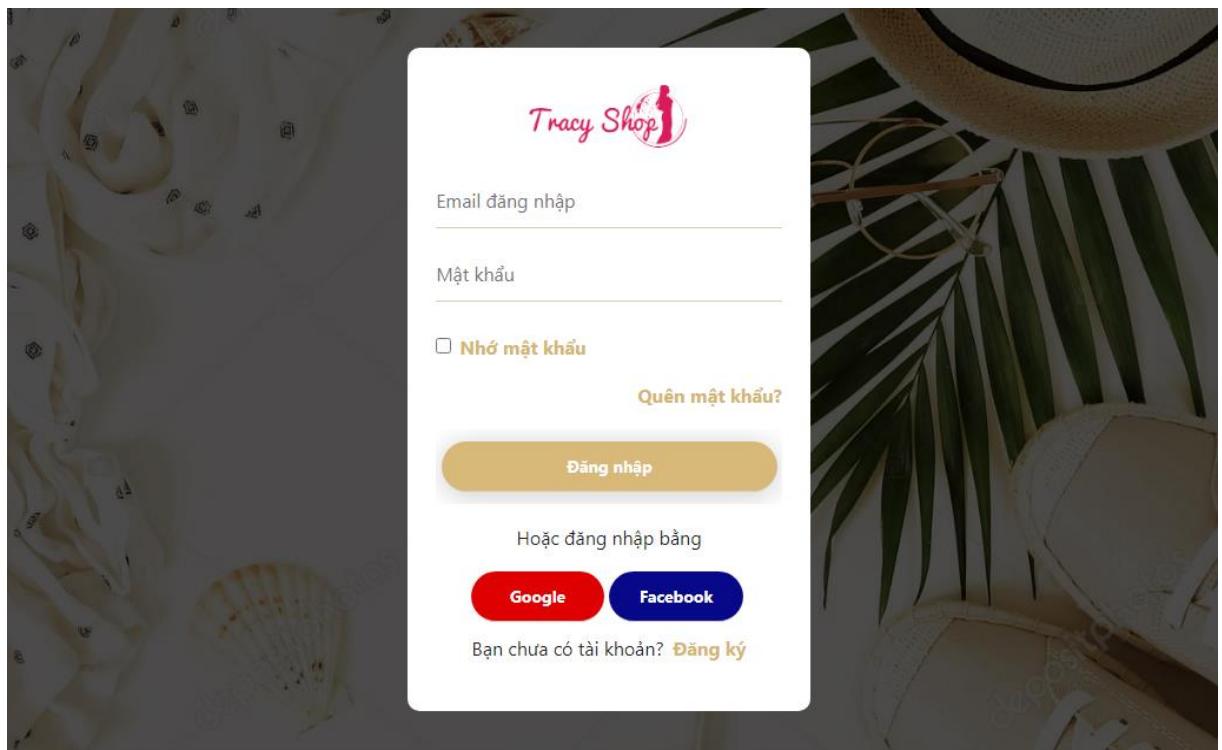


Hình 3.69: Giao diện thông báo xác thực email thành công

Để đăng nhập, khách hàng nhấp vào “Tiến hành đăng nhập” trong thông báo để đến trang đăng nhập.

3.3.3.2. Chức năng Đăng nhập

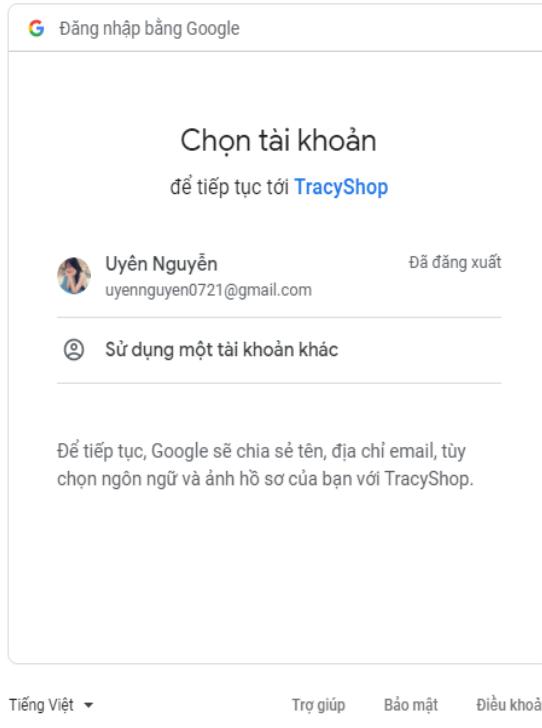
Chức năng này cho phép khách hàng đăng nhập với tài khoản đã đăng ký trước đó để sử dụng các chức năng yêu cầu chứng thực người dùng.



Hình 3.70: Giao diện đăng nhập của khách hàng

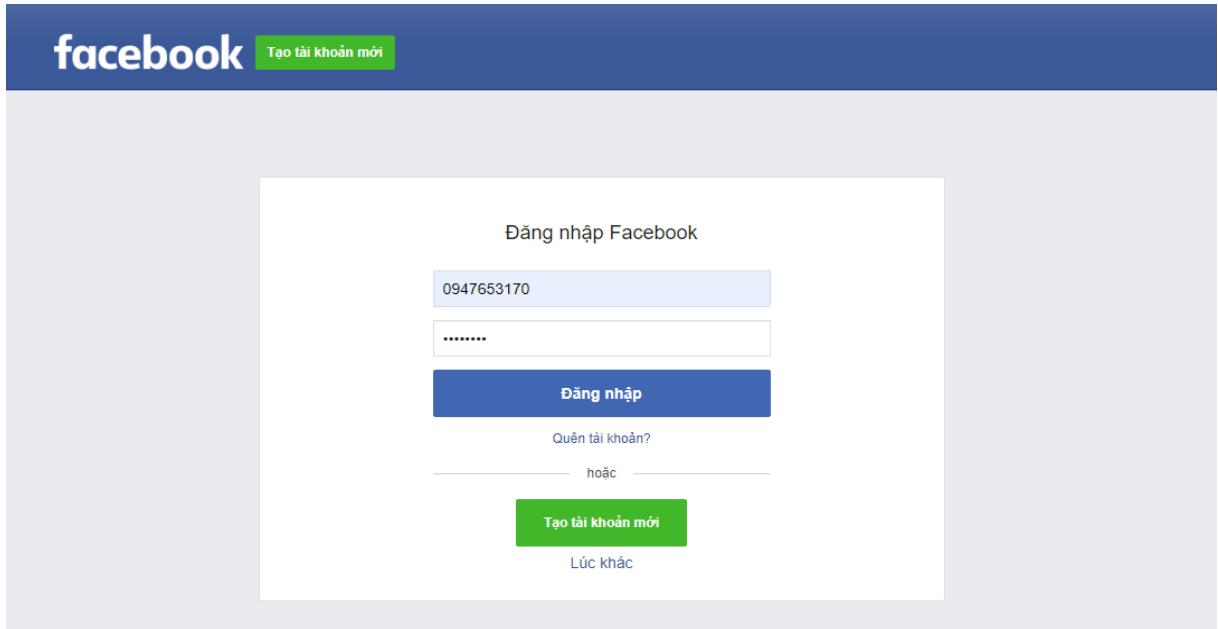
Ngoài chức năng đăng nhập theo kiểu truyền thống, hệ thống cũng hỗ trợ đăng nhập qua Google và Facebook.

- Đăng nhập qua Google:



Hình 3.71: Giao diện đăng nhập qua Google

- Đăng nhập qua Facebook:



Hình 3.72: Giao diện đăng nhập qua Facebook

3.3.3.3. Chức năng Quên mật khẩu

Chức năng này cho phép khách hàng lấy lại mật khẩu khi quên mật khẩu đăng nhập.



Hình 3.73: Giao diện chức năng quên mật khẩu của khách hàng

Khách hàng nhập email đăng ký vào nhấp vào “Tiếp tục”, hệ thống sẽ gửi một email đến email đăng ký của khách hàng và thông báo cho khách hàng biết có một email gửi đến hộp thư của khách hàng.



Hình 3.74: Giao diện thông báo gửi mail đặt lại mật khẩu của khách hàng

Khách hàng kiểm tra hộp thư và đọc mail mà hệ thống gửi đến để đặt lại mật khẩu.

The screenshot shows an email inbox with two messages. The top message is from 'TracyShop Team' to 'lantrinh@gmail.com' with the subject 'Hello Lan Trinh, reset your password.' It was sent 'a minute ago'. The bottom message is from 'Hello Lan Trinh, Confirm your email id.' to 'lantrinh@gmail.com' with the subject 'Hello Lan Trinh, Confirm your email id.' It was sent '6 minutes ago'. The interface includes standard email controls like search, reply, forward, and delete.

Hello Lan Trinh, reset your password.

From: TracyShop Team <no-reply@tracyshop.com>
To: <lantrinh@gmail.com>

Show Headers

HTML HTML Source Text Raw Spam Analysis HTML Check Tech Info

Xin chào Lan Trinh,
Nhấp vào liên kết sau để đặt lại mật khẩu của bạn.

[Đặt lại mật khẩu](#)

Hình 3.75: Giao diện mail đặt lại mật khẩu của khách hàng

Trong mail có một liên kết, khách hàng nhấp vào liên kết đó sẽ chuyển đến trang đặt lại mật khẩu yêu cầu khách hàng nhập mật khẩu mới.



Hình 3.76: Giao diện đặt lại mật khẩu của khách hàng

Sau khi nhập mật khẩu mới và nhập xác nhận lại mật khẩu, khách hàng nhấp vào nút “Đặt lại mật khẩu” để tiến hành đặt lại mật khẩu. Nếu thành công, hệ thống sẽ hiển thị trang thông báo cập nhật mật khẩu thành công, ngược lại sẽ hiển thị trang thông báo cập nhật mật khẩu thất bại.



Hình 3.77: Giao diện thông báo đặt lại mật khẩu thành công của khách hàng

Để đăng nhập, khách hàng nhấp vào “Tiến hành đăng nhập” trong thông báo để đến trang đăng nhập.

3.3.3.4. Chức năng Quản lý tài khoản

Chức năng này cho phép khách hàng đổi mật khẩu, xem và chỉnh sửa thông tin cá nhân, xem và thay đổi địa chỉ:

- Xem và chỉnh sửa thông tin cá nhân:

The screenshot shows a user profile on the left with a placeholder image and the name 'Linh Đan' and email 'linhdan@gmail.com'. On the right, a modal window titled 'Hồ sơ của tôi' (My Profile) is displayed under 'Quản lý hồ sơ'. It contains the following fields:

- Tên đăng nhập: linhdan@gmail.com
- Họ và tên: Linh Đan
- Email: linhdan@gmail.com
- Số điện thoại: 0123456789
- Giới tính: Nam Nữ Khác
- Ngày sinh: 25/10/2000
- Chọn ảnh đại diện: [Browse] Không có tệp nào được chọn
- Lưu thay đổi: button

Hình 3.78: Giao diện xem và chỉnh sửa thông tin của khách hàng

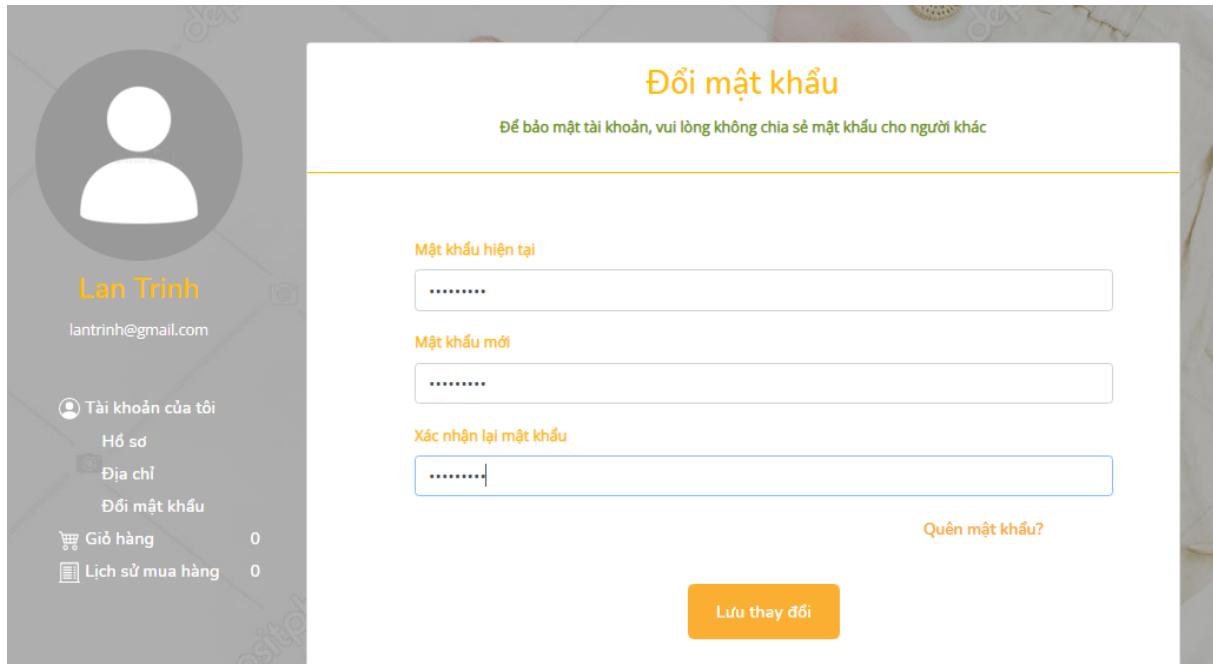
- Xem và thay đổi địa chỉ:

The screenshot shows a user profile on the left with a placeholder image and the name 'Lan Trinh' and email 'lantrinh@gmail.com'. On the right, a modal window titled 'Địa chỉ của tôi' (My Address) is displayed under 'Địa chỉ giao hàng'. It contains the following fields:

- Tỉnh / Thành phố: Thành phố Hồ Chí Minh
- Quận / Huyện: Quận Gò Vấp
- Địa chỉ cụ thể: 58/4 đường số 7, phường 7
- Lưu địa chỉ: button

Hình 3.79: Giao diện xem và chỉnh sửa địa chỉ của khách hàng

- Đổi mật khẩu:



Hình 3.80: Giao diện đổi mật khẩu của khách hàng

3.3.3.5. Chức năng Tra cứu

Chức năng này cho phép khách hàng tra cứu sản phẩm bằng cách khách hàng nhập từ khóa vào ô tìm kiếm và nhấp vào biểu tượng tìm kiếm, nó sẽ hiển thị giao diện danh sách sản phẩm liên quan đến từ khóa mà khách hàng nhập vào.

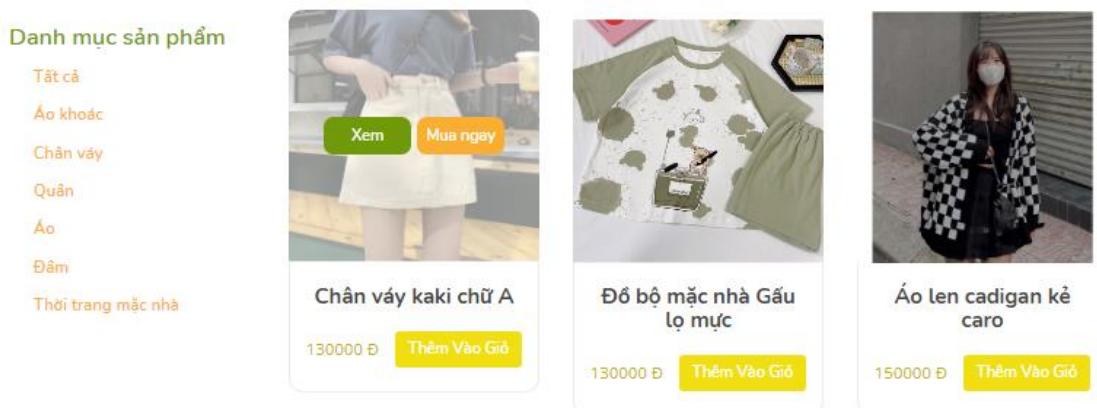


Hình 3.81: Giao diện đổi mật khẩu của khách hàng

Ngoài ra, khách hàng có thể nhấp vào tên của danh mục sản phẩm cần tìm bên trái màn hình, nó sẽ hiển thị danh sách sản phẩm theo danh mục tương ứng.

3.3.3.6. Chức năng Xem thông tin sản phẩm

Chức năng này cho phép khách hàng xem thông tin chi tiết của sản phẩm bằng cách rê chuột vào một sản phẩm bất kỳ, nó sẽ xuất hiện hai nút “Xem” và “Mua ngay”:



Hình 3.82: Giao diện sau khi rê chuột vào một sản phẩm của khách hàng
Khách hàng nhấp vào “Xem” nó sẽ hiển thị giao diện với thông tin chi tiết của sản phẩm.

Chân váy kaki chữ A
130000 VNĐ
Size S (Chiều cao: 150 - 155 cm, Cân nặng: 40 - 45 kg)
M (Chiều cao: 156 - 160 cm, Cân nặng: 43 - 46 kg)
L (Chiều cao: 160 - 164 cm, Cân nặng: 46 - 53 kg)
Số lượng sẵn có 5 sản phẩm
Thêm vào giỏ hàng **Mua ngay**

Chi tiết sản phẩm

- Tên sản phẩm: Chân váy kaki chữ A
- Xuất xứ: Việt Nam
- Nhãn hiệu: LYLY
- Năm sản xuất: 10-12-2020 00:00
- Mô tả: Chân váy Kaki chữ A ngắn màu trắng cạp cao phù hợp cho các nàng mặc đi làm công sở, đi học, đi chơi giúp tôn dáng.

Hình 3.83: Giao diện chi tiết sản phẩm của khách hàng

3.3.3.7. Chức năng Quản lý giỏ hàng

Chức năng này cho phép khách hàng thực hiện các chức năng Quản lý giỏ hàng như thêm sản phẩm vào giỏ hàng, xem giỏ hàng, cập nhật số lượng và kích thước của sản phẩm trong giỏ và xóa sản phẩm khỏi giỏ hàng.

- Thêm sản phẩm vào giỏ hàng:



Hình 3.84: Giao diện thông báo thêm vào giỏ hàng thành công

- Xem giỏ hàng:

Sản phẩm	Phân loại hàng	Đơn giá	Số lượng	Số tiền	Chỉnh sửa	Xoá
Chân váy kaki chữ A	S	130000	1	130000	<button>Chỉnh sửa</button>	<button>Xoá</button>

Tổng số sản phẩm: 1 sản phẩm Tổng thanh toán: 130000 VNĐ Mua hàng

Hình 3.85: Giao diện hiển thị sản phẩm trong giỏ của khách hàng

- Cập nhật số lượng và kích thước sản phẩm trong giỏ:

Chọn size

S

Số lượng

1

Cập nhật

Hình 3.86: Giao diện cập nhật sản phẩm trong giỏ hàng

3.3.3.8. Chức năng Đặt hàng

Chức năng này cho phép khách hàng đặt hàng. Khách hàng nhấp vào “Mua hàng” ở trang giỏ hàng, hệ thống sẽ chuyển đến trang hiển thị thông tin đặt hàng.

📍 Địa chỉ đặt hàng

0123456789

58/4 đường số 7, phường 7, Quận Gò Vấp, Thành phố Hồ Chí Minh

Thay đổi

Sản phẩm

Sản phẩm	Phân loại hàng	Đơn giá	Số lượng	Thành tiền
Chân váy kaki chữ A	S	130000	1	130000

Phí vận chuyển: 20000
Tổng số tiền (1 sản phẩm): 150000 VNĐ

Phương thức thanh toán

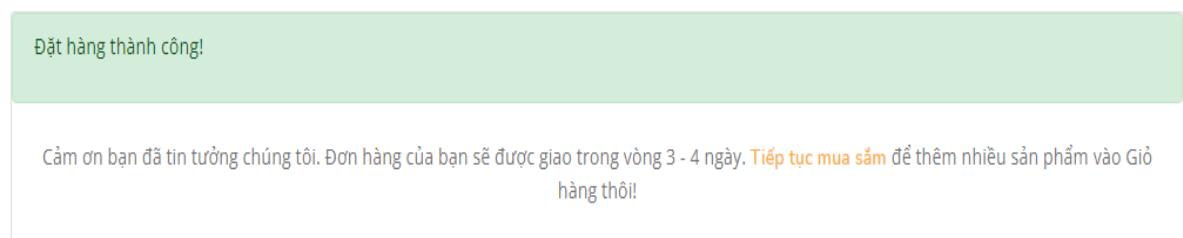
Thanh toán khi nhận hàng
 Thanh toán PayPal

Tổng tiền hàng 130000 VNĐ
Phí vận chuyển 20000 VNĐ
Tổng thanh toán 150000 VNĐ

Đặt hàng

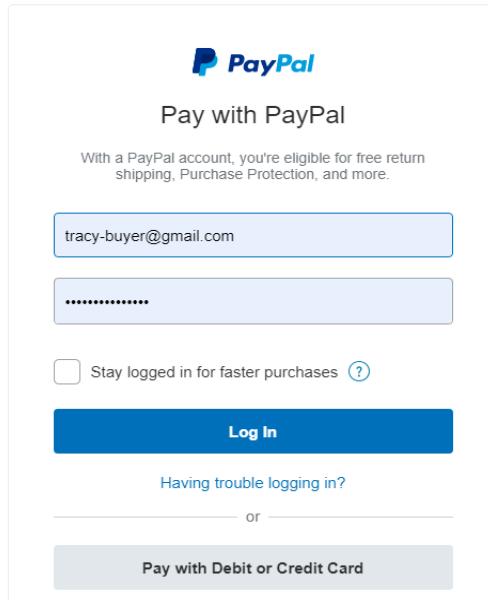
Hình 3.87: Giao diện đặt hàng của khách hàng

Nếu khách hàng chọn phương thức thanh toán khi nhận hàng và nhấp vào “Đặt hàng”, hệ thống sẽ hiển thị giao diện thông báo cho người dùng biết đã đặt hàng thành công.



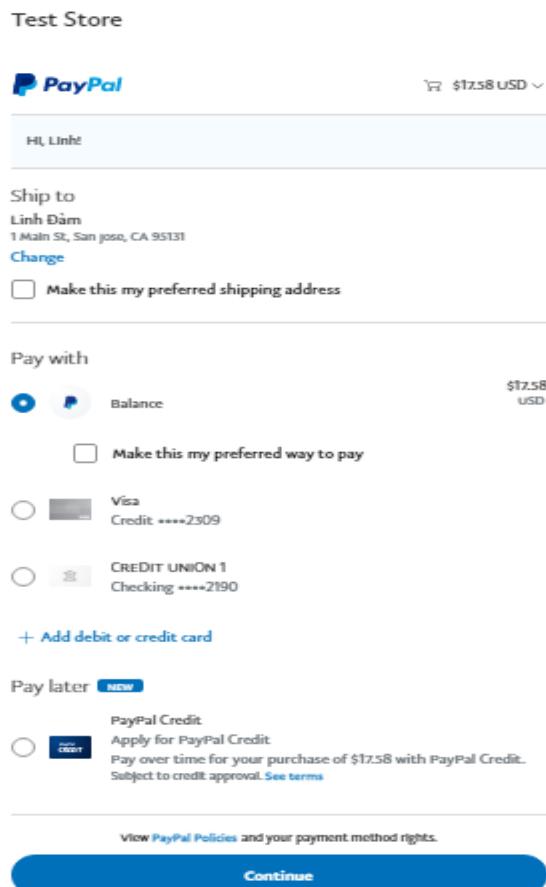
Hình 3.88: Giao diện thông báo đặt hàng thành công

Nếu khách hàng chọn phương thức thanh toán PayPal, hệ thống sẽ chuyển đến trang thanh toán của PayPal. Khách hàng sẽ đăng nhập tài khoản PayPal để tiến hành thanh toán:



Hình 3.89: Giao diện đăng nhập thanh toán PayPal

Sau khi đăng nhập sẽ hiển thị hóa đơn thanh toán PayPal trên giao diện. Khách hàng chọn nguồn tiền thanh toán và nhấp “Continue”, hệ thống thanh toán PayPal sẽ xử lý thanh toán và trả về kết quả thanh toán thành công hoặc thất bại.



Hình 3.90: Giao diện thanh toán trên PayPal

3.3.3.9. Chức năng Phản hồi đánh giá

Chức năng này cho phép khách hàng phản hồi và đánh giá sản phẩm:

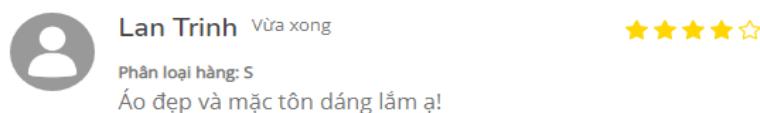
The screenshot shows a form for leaving a review. At the top, it says "Áo croptop xoắn eo body". Below that is a dropdown menu for "Phân loại hàng" set to "S". A placeholder text "Cho chúng tôi biết ý kiến của bạn" is followed by a 5-star rating. A text input field contains the review text "Áo đẹp và mặc tôn dáng lắm ạ!". Below the input field are buttons for "Thêm ảnh" (Add photo), "Choose File" (No file chosen), and "Gửi" (Send). To the right of the form is a photograph of a woman wearing a white crop top and blue denim shorts.

Hình 3.91: Giao diện phản hồi và đánh giá sản phẩm của khách hàng

Khách hàng chọn phân loại hàng, nhập phản hồi và đánh giá sản phẩm, sau đó nhấp vào nút “Gửi”. Phản hồi và đánh giá của khách hàng sẽ hiển thị ở phần đánh giá sản phẩm của trang chi tiết sản phẩm.

Đánh giá sản phẩm

[Thêm đánh giá](#)



Hình 3.92: Giao diện xem đánh giá sản phẩm của khách hàng

3.3.3.10. Chức năng Xem lịch sử mua hàng

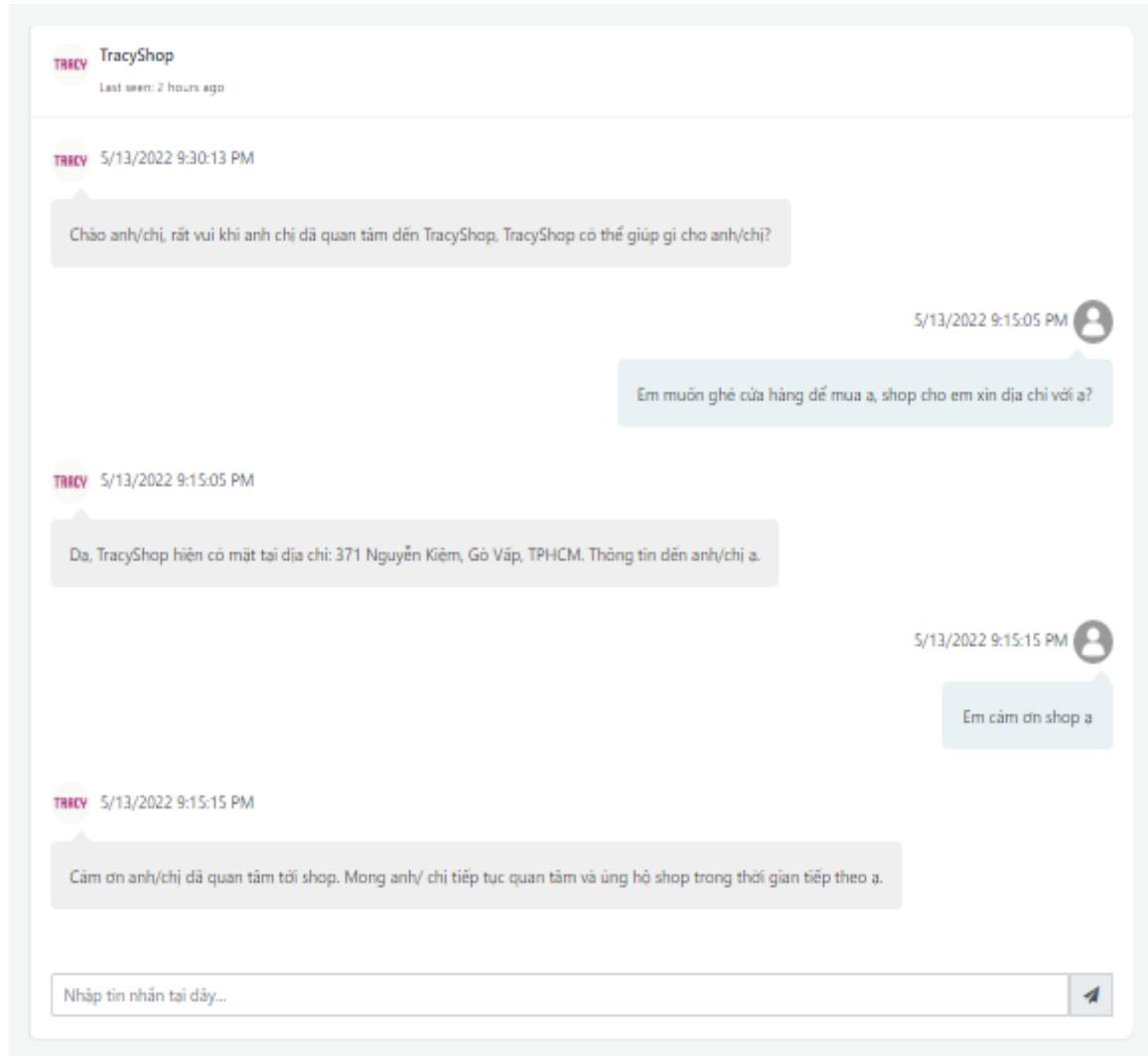
Chức năng này cho phép khách hàng xem lịch sử mua hàng bằng cách nhấp vào “Lịch sử mua hàng” ở trang cá nhân.



Hình 3.93: Giao diện lịch sử mua hàng của khách hàng

3.3.3.11. Chức năng Chatbot

Chức năng này cho phép khách hàng gửi tin nhắn đến cho admin hoặc nhân viên của shop để nhờ tư vấn, hỗ trợ và xem các câu trả lời từ phía admin hoặc nhân viên.



Hình 3.94: Giao diện chức năng chatbot của khách hàng

Chương 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI

4.1. Kết luận

Đề tài “Xây dựng website bán quần áo nữ online TracyShop” đã hoàn thành các chức năng cơ bản của một website bán quần áo nữ online ứng với từng loại người dùng là admin, nhân viên và khách hàng. Ngoài ra, đề tài cũng tích hợp các chức năng mới như:

- Chức năng gửi mail qua Mailtrap để tiến hành xác nhận email đăng ký của người dùng và lấy lại mật khẩu khi người dùng quên mật khẩu.
- Đăng nhập qua Facebook và Google được hỗ trợ bởi ASP.NET Core Identity.
- Thanh toán qua dịch vụ thanh toán trực tuyến PayPal.
- Chức năng Chat giữa khách hàng và admin hoặc nhân viên có sử dụng tin nhắn tự động.

Bên cạnh đó, đề tài “Xây dựng website bán quần áo nữ online TracyShop” vẫn còn tồn tại những vấn đề cần được khắc phục như:

- Hiệu suất của hệ thống chưa được tối ưu, một số chức năng xử lý còn chậm.
- Trong quá trình hệ thống đang hoạt động, một số lỗi có thể xuất hiện ở một số chức năng.

4.2. Hướng phát triển đề tài

Trong tương lai, website bán quần áo nữ online TracyShop sẽ cố gắng khắc phục được những vấn đề còn tồn tại của hệ thống như vấn đề về hiệu suất xử lý ở một số chức năng để tăng tính trải nghiệm cho người dùng và sửa một số lỗi có thể xuất hiện trong đề tài. Ngoài ra, chức năng Chat giữa khách hàng với admin hoặc nhân viên chưa sử dụng realtime (thời gian thực), trong tương lai em sẽ khắc phục cho tính năng Chat này.

TÀI LIỆU THAM KHẢO

Tài liệu điện tử

- [1] TutorialsTeacher, “.NET Core Overview,”. [Trực tuyến]. Địa chỉ: <https://www.tutorialsteacher.com/core/dotnet-core> [Truy cập 5/10/2021]
- [2] Trọng Đức, “Phân biệt .NET Core, .NET Framework, .NET Standard,” 18/11/2019. [Trực tuyến]. Địa chỉ: <https://tuhocict.com/phan-biet-net-cor-net-framework-net-standard/> [Truy cập 5/10/2021]
- [3] Nguyễn Minh Tuấn, “Tổng quan về ASP.NET Core.”. [Trực tuyến]. Địa chỉ: <https://netcore.vn/bai-viet/tong-quan-ve-aspnet-core> [Truy cập 6/10/2021]
- [4] Steve Smith, “Overview of ASP.NET Core MVC,” 12/2/2020. [Trực tuyến]. Địa chỉ: <https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-5.0> [Truy cập 6/10/2021]
- [5] Bạch Ngọc Toàn, “Cơ chế Routing trong ASP.NET Core,” 7/7/2019. [Trực tuyến]. Địa chỉ: <https://tedu.com.vn/lap-trinh-aspnet-core/co-che-routing-trong-aspnet-core-227.html> [Truy cập 6/10/2021]
- [6] Bạch Ngọc Toàn, “Cơ chế Model Binding: Truyền dữ liệu từ View lên Controller,” 13/8/2019. [Trực tuyến]. Địa chỉ: <https://tedu.com.vn/lap-trinh-aspnet-core/co-che-model-binding-truyen-du-lieu-tu-view-len-controller-252.html> [Truy cập 7/10/2021]
- [7] Bạch Ngọc Toàn, “Model Validation trong ASP.NET Core,” 19/8/2019. [Trực tuyến]. Địa chỉ: <https://tedu.com.vn/lap-trinh-aspnet-core/model-validation-trong-aspnet-core-253.html> [Truy cập 7/10/2021]
- [8] Admin, “Dependency Injection là gì?,” 7/2019. [Trực tuyến]. Địa chỉ: <https://www.dotnetcoban.com/2019/07/introduction-to-dependency-injection.html> [Truy cập 8/10/2021]
- [9] Bạch Ngọc Toàn, “Cơ chế Dependency Injection trong ASP.NET Core,” 20/8/2019. [Trực tuyến]. Địa chỉ: <https://tedu.com.vn/lap-trinh-aspnet-core/co-che-dependency-injection-trong-aspnet-core-256.html> [Truy cập 8/10/2021]
- [10] Rick Anderson, “Introduction to Identity on ASP.NET Core,” 15/9/2021. [Trực tuyến]. Địa chỉ: <https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-5.0&tabs=visual-studio> [Truy cập 9/10/2021]
- [11] Piotr Malek, “Send and Receive Emails in ASP.NET C#,” 27/7/ 2019. [Trực tuyến]. Địa chỉ: <https://mailtrap.io/blog/send-emails-in-net/> [Truy cập 11/10/2021]

- [12] Jeff Reifman, “Introduction to Mailtrap: A Fake SMTP Server for Pre-Production Testing,” 30/5/2015. [Trực tuyến]. Địa chỉ: <https://code.tutsplus.com/tutorials/introduction-to-mailtrap-a-fake-smtp-server-for-pre-production-testing--cms-23279> [Truy cập 11/10/2021]
- [13] https://vi.wikipedia.org/wiki/Microsoft_SQL_Server [Truy cập 30/10/2021]

Tài liệu sách

- [14] Andrew Lock, *ASP.NET Core in Action*. Nhà xuất bản Án phẩm Manning, 2018.
- [15] ThS. Nguyễn Minh Đạo, *Giáo trình Lập trình web với ASP.NET*. Nhà xuất bản Đại học Quốc gia Thành phố Hồ Chí Minh, 2014.

PHỤ LỤC

GIỚI THIỆU VỀ .NET CORE

.NET Core là một phiên bản mới của .NET Framework. Đây là một nền tảng phát triển miễn phí, mã nguồn mở, được duy trì và phát triển bởi Microsoft và cộng đồng .NET ở trên Github. Nó là một khung nền tảng chéo, chạy trên hệ điều hành Windows, macOS và cả Linux [1]. Về cấu trúc, .NET Core được tổ chức theo hướng module hóa nên các ứng dụng chạy bằng .NET Core thường nhẹ, nhanh, dung lượng bộ nhớ thấp, hiệu suất ứng dụng được tối ưu hóa.

CoreCLR được gọi là môi trường thực thi của .NET Core có thể chạy được trên nhiều hệ điều hành như Windows, macOS và Linux.

CoreCLR sử dụng trình biên dịch JIT compiler. Trình biên dịch này của CoreCLR có thể dịch được mã IL sang mã máy của ba nền tảng mà nó hỗ trợ là Windows, macOS và Linux [2].

GIỚI THIỆU VỀ ASP.NET CORE

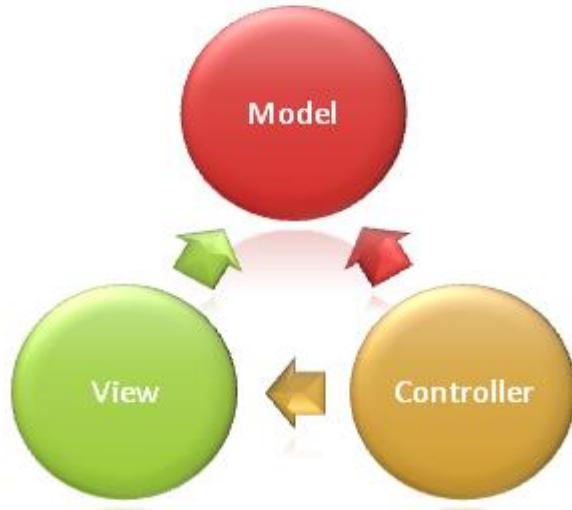
ASP.NET Core là một framework miễn phí, mã nguồn mở và chạy đa nền tảng. Giống như .NET Core, nó có thể chạy được trên nhiều hệ điều hành như Windows, Linux, macOS, có cấu trúc theo module hóa với chi phí thấp. Nếu người dùng muốn tích hợp thêm các tính năng nâng cao khác thì chỉ cần thêm vào dưới dạng gói NuGet [3]. ASP.NET Core cũng được hỗ trợ và phát triển bởi Microsoft và cộng đồng trên Github.

Vì ASP.NET Core có cấu trúc module hóa và chỉ cần thêm các gói NuGet khi cần thiết nên các ứng dụng sử dụng ASP.NET Core thường có hiệu suất cao, yêu cầu ít bộ nhớ hơn, kích thước triển khai cũng ít hơn và dễ bảo trì. Bên cạnh đó, với kích thước triển khai của ứng dụng ít hơn thì ứng dụng sẽ có tính bảo mật cao và chặt chẽ hơn.

MÔ HÌNH MVC

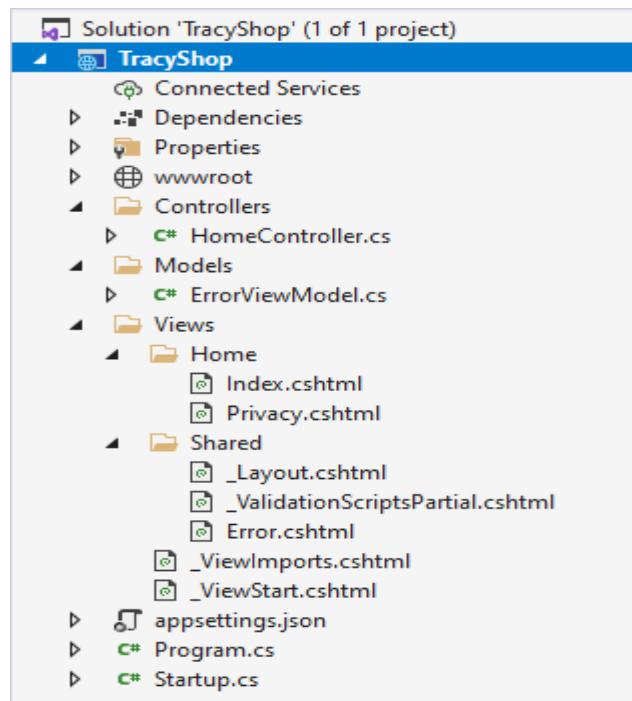
Mô hình MVC (Model – View – Controller) là mô hình kiến trúc gồm 3 phần: Model, View, Controller. Mỗi phần thực hiện một chức năng cụ thể. Với mô hình này, các yêu cầu của người dùng sẽ được chuyển đến Controller. Controller sẽ tiếp nhận yêu

cầu của người dùng và tiến hành làm việc với Model để thực hiện những yêu cầu của người dùng và có thể lấy kết quả dữ liệu của các câu truy vấn. Sau đó, Controller sẽ chọn một View tương ứng trong các View để hiển thị ra cho người dùng và cung cấp cho nó những dữ liệu của Model mà nó yêu cầu [4].



Hình 4.1: Các thành phần trong mô hình MVC và sự tương tác giữa chúng (nguồn: [4])

Việc phân chia chức năng từng thành phần trong mô hình MVC giúp cho việc viết code, kiểm tra và sửa lỗi trở nên trực quan và dễ dàng hơn . Đồng thời, nó cũng giúp cho chương trình có tính mở rộng và dễ bảo trì.



Hình 4.2: Mô hình MVC trong dự án ASP.NET Core MVC

USECASE XEM THÔNG TIN SẢN PHẨM

Mô tả	Use case cho phép khách hàng xem thông tin sản phẩm
Actor chính	Khách hàng
Actor phụ	Không có
Luồng hoạt động	<ol style="list-style-type: none"> 1. Khách hàng nhấp chọn xem trên sản phẩm 2. Hệ thống sẽ chuyển đến trang chi tiết sản phẩm và hiển thị thông tin của sản phẩm đó. 3. Kết thúc use case.

Bảng 4.1: Đặc tả usecase xem thông tin sản phẩm của khách hàng

USECASE XEM LỊCH SỬ MUA HÀNG

Mô tả	Use case cho phép khách hàng xem thông tin lịch sử mua hàng.
Actor chính	Khách hàng
Actor phụ	Không có
Tiền điều kiện	Khách hàng đã đăng nhập thành công vào hệ thống.
Luồng hoạt động	<ol style="list-style-type: none"> 1. Khách hàng nhấp chọn lịch sử mua hàng 2. Hệ thống sẽ chuyển đến trang lịch sử mua hàng và hiển thị thông tin các đơn hàng mà khách hàng đã mua. 3. Kết thúc use case.

Bảng 4.2: Đặc tả usecase lịch sử mua hàng

USECASE XEM THÔNG TIN PHẢN HỒI

Mô tả	Use case cho phép admin và nhân viên xem thông tin phản hồi của khách hàng.
Actor chính	Admin, nhân viên
Actor phụ	Không có
Tiền điều kiện	Admin hoặc nhân viên đã đăng nhập thành công vào hệ thống.

Luồng hoạt động	<p>1. Hệ thống hiển thị các phản hồi của khách hàng nếu có phản hồi, ngược lại sẽ hiển thị chưa có phản hồi.</p> <p>3. Kết thúc use case.</p>
-----------------	---

Bảng 4.3: Đặc tả usecase xem thông tin phản hồi

USE CASE TÌM KIẾM

Mô tả	Use case cho phép admin và nhân viên tìm kiếm sản phẩm theo tên sản phẩm, loại sản phẩm, nhãn hiệu, xuất xứ.
Actor chính	Admin, nhân viên
Actor phụ	Không có
Tiền điều kiện	Admin hoặc nhân viên đã đăng nhập thành công vào hệ thống.
Luồng hoạt động	<p>1. Admin hoặc nhân viên nhập từ khóa trên thanh tìm kiếm của trang sản phẩm dành cho quản lý và nhân tìm kiếm.</p> <p>2. Hệ thống sẽ tìm những sản phẩm có tên sản phẩm, loại sản phẩm, nhãn hiệu hoặc xuất xứ trùng với từ khóa mà admin hay nhân viên nhập vào và hiển thị ở trang sản phẩm. Nếu không tìm thấy thì sẽ không trả về sản phẩm nào.</p> <p>3. Kết thúc use case.</p>

Bảng 4.4: Đặc tả usecase tìm kiếm

USECASE QUẢN LÝ TÀI KHOẢN

Mô tả	Use case cho khách hàng dùng xem và cập nhật thông tin cá nhân.
Actor chính	Khách hàng
Actor phụ	Không có
Tiền điều kiện	Khách hàng phải đăng nhập thành công vào hệ thống
Luồng hoạt động	<p>1. Hệ thống hiển thị màn hình trang cá nhân. Ở trang này, có các tùy chọn chức năng để khách hàng có thể thực hiện quản lý tài khoản như: Hồ sơ (xem và chỉnh sửa thông tin cá nhân), Địa chỉ (xem và thay đổi địa chỉ), Đổi mật khẩu.</p>

	<p>2. Khách hàng chọn một trong các tùy chọn trên thì một trong các luồng tương ứng sau sẽ thực hiện:</p> <ul style="list-style-type: none"> - Nếu khách hàng chọn “Hồ sơ” thì luồng phụ Hồ sơ được thực hiện. - Nếu khách hàng chọn “Địa chỉ” thì luồng phụ Địa chỉ được thực hiện. - Nếu khách hàng chọn “Đổi mật khẩu” thì luồng phụ Đổi mật khẩu được thực hiện. <p>* Các luồng phụ:</p> <p>I – Hồ sơ</p> <ol style="list-style-type: none"> 1. Hệ thống hiển thị thông tin của khách hàng. 2. Nếu khách hàng cập nhật thông tin và chọn lưu thông tin, hệ thống sẽ cập nhật lại thông tin khách hàng và hiển thị thông báo cập nhật thành công. Ngược lại, thông tin khách hàng vẫn giữ nguyên. <p>II – Địa chỉ</p> <ol style="list-style-type: none"> 1. Hệ thống hiển thị thông tin địa chỉ của khách hàng. 2. Nếu khách hàng cập nhật địa chỉ thì hệ thống cập nhật lại thông tin địa chỉ của khách hàng, ngược lại địa chỉ của khách hàng vẫn giữ nguyên. <p>III – Đổi mật khẩu</p> <ol style="list-style-type: none"> 1. Hệ thống hiển thị giao diện đổi mật khẩu. 2. Khách hàng nhập mật khẩu hiện tại và mật khẩu mới và chọn lưu thay đổi thì hệ thống sẽ tiến hành cập nhật lại mật khẩu của khách hàng, ngược lại mật khẩu của khách hàng vẫn giữ nguyên.
--	---

Bảng 4.5: Đặc tả usecase quản lý tài khoản

USECASE QUẢN LÝ KHUYẾN MÃI

Mô tả	Use case cho phép admin thực hiện các chức năng quản lý khuyến mãi như thêm, xóa, sửa.
Actor chính	Admin
Actor phụ	Không có

Tiền điều kiện	Admin đã đăng nhập thành công vào hệ thống.
Luồng hoạt động	<p>1. Admin chọn chức năng quản lý khuyến mãi</p> <p>2. Hệ thống sẽ hiển thị các tùy chọn chức năng gồm chương trình khuyến mãi, thêm khuyến mãi.</p> <p>3. Nếu admin chọn “Chương trình khuyến mãi” thì hệ thống sẽ chuyển đến trang danh sách khuyến mãi và trang này sẽ hiển thị tất cả các khuyến mãi. Tại đây, với từng khuyến mãi admin có thể tiến hành chỉnh sửa hoặc xóa.</p> <p>Nếu admin chọn “Thêm khuyến mãi” thì luồng phụ Thêm khuyến mãi sẽ được thực hiện.</p> <p>* Các luồng phụ:</p> <p>I – Thêm khuyến mãi</p> <p>1. Hệ thống hiển thị giao diện thêm khuyến mãi.</p> <p>2. Admin nhập các thông tin của khuyến mãi và chọn thêm ưu đãi.</p> <p>3. Hệ thống tiến hành lưu thông tin của khuyến mãi.</p> <p>4. Kết thúc usecase.</p>

Bảng 4.6: Đặc tả usecase quản lý khuyến mãi

USECASE QUÊN MẬT KHẨU

Mô tả	Use case cho phép khách hàng đặt lại mật khẩu mới khi quên mật khẩu.
Actor chính	Khách hàng
Actor phụ	Không có
Tiền điều kiện	Khách hàng phải có tài khoản trên hệ thống.
Hậu điều kiện	Khách hàng đăng nhập để sử dụng một số chức năng.
Luồng hoạt động	<p>1. Hệ thống hiển thị giao diện để khách hàng nhập email đăng ký lấy lại mật khẩu.</p> <p>2. Khách hàng nhập email đăng ký và chọn tiếp tục hoặc có thể chọn đăng nhập để quay về trang đăng nhập.</p> <p>3. Nếu khách hàng chọn tiếp tục, hệ thống sẽ kiểm tra email vừa nhập có tồn tại với một trong những email đã đăng ký</p>

	<p>hay không. Nếu có, hệ thống sẽ gửi một mail đến email đó và hiển thị thông báo đã gửi mail. Ngược lại sẽ không gửi mail.</p> <p>4. Khách hàng nhập vào liên kết trong mail sẽ dẫn đến một trang cho phép khách hàng lấy lại mật khẩu. Khách hàng nhập mật khẩu mới và chọn đặt lại mật khẩu.</p> <p>5. Hệ thống sẽ xóa mật khẩu cũ và lưu mật khẩu mới của khách hàng vừa đặt.</p> <p>6. Kết thúc use case.</p>
--	--

Bảng 4.7: Đặc tả usecase quên mật khẩu