

TRƯỜNG ĐẠI HỌC BÀ RỊA – VŨNG TÀU
KHOA KỸ THUẬT CÔNG NGHỆ

BÁO CÁO ĐỒ ÁN TỐT NGHIỆP



BARIA VUNGTAU
UNIVERSITY
CAP SAINT JACQUES

ĐỀ TÀI: XÂY DỰNG WEBSITE BÁN SIM SỐ ĐẸP

Trình độ đào tạo : Đại học
Ngành : Công nghệ thông tin
Chuyên ngành : Lập trình ứng dụng di động và game
Khóa học : 2019 - 2023
Lớp : DH19LT
Sinh viên thực hiện : Hồ Trọng Vinh
Mã số sinh viên : 19033832
GVHD : TS. Phan Ngọc Hoàng

Vũng Tàu, năm 2023

LỜI CẢM ƠN

Thưa thầy Phan Ngọc Hoàng và toàn thể các thầy cô giáo của trường Đại học Bà Rịa - Vũng Tàu,

Em xin gửi lời cảm ơn chân thành nhất đến thầy và trường đã dạy và hỗ trợ em trong suốt quá trình học tập và làm đồ án tốt nghiệp.

Trước hết, em xin cảm ơn thầy Phan Ngọc Hoàng - người đã trực tiếp hướng dẫn và giúp đỡ em trong suốt quá trình làm đồ án. Nhờ có sự hướng dẫn tận tình của thầy, em đã học hỏi được nhiều kiến thức mới, từ đó nâng cao được khả năng làm việc và giải quyết vấn đề của mình. Thầy còn giúp đỡ em về mặt tinh thần, giúp em tự tin và kiên định với quyết tâm của mình.

Em cũng muốn gửi lời cảm ơn đến các thầy cô giáo của trường Đại học Bà Rịa - Vũng Tàu. Nhờ sự giảng dạy chuyên nghiệp và tận tình của các thầy cô, em đã có được kiến thức cần thiết để phục vụ cho việc học tập và nghiên cứu của mình.

Cuối cùng, em xin chúc thầy Phan Ngọc Hoàng và toàn thể các thầy cô giáo của trường Đại học Bà Rịa - Vũng Tàu luôn khỏe mạnh và thành công trong sự nghiệp giảng dạy. Em sẽ luôn cảm ơn và tự hào về những người thầy đã dạy dỗ và hướng dẫn mình trên con đường phát triển sự nghiệp của mình.

Trân trọng cảm ơn!

LỜI CAM ĐOAN

Tôi xin cam đoan rằng đồ án tốt nghiệp được trình bày dưới đây là công sức của bản thân tôi. Tôi cam đoan rằng các nội dung, kết quả và nhận xét được trình bày trong đồ án này là hoàn toàn chính xác và được lấy từ nhiều nguồn tài liệu khác nhau.

Trong quá trình làm đồ án, tôi đã tham khảo và sử dụng nhiều tài liệu khác nhau như sách, báo cáo, trang web, các tài liệu tham khảo từ giáo viên và các tác giả khác. Tất cả các nguồn tài liệu đều được liệt kê và tham khảo đầy đủ trong phần tài liệu tham khảo của đồ án.

Tôi xin cam đoan rằng đồ án tốt nghiệp này là sự nghiên cứu của bản thân tôi và không sao chép, đạo từ bất kỳ nguồn tài liệu nào khác. Nếu có bất kỳ điểm nào sai sót, tôi sẽ chịu trách nhiệm và sẵn sàng chấp nhận những hình thức kỷ luật và phạt tiền phù hợp.

Tôi cảm ơn giáo viên hướng dẫn của mình đã hỗ trợ và giúp đỡ tôi trong quá trình làm đồ án. Tôi cũng cảm ơn trường Đại học Bà Rịa – Vũng Tàu đã cung cấp cho tôi những kiến thức cần thiết để hoàn thành đồ án này.

Xin chân thành cảm ơn.

Vũng Tàu, ngày 05 tháng 05 năm 2023

Sinh viên thực hiện

Hồ Trọng Vinh

MỤC LỤC

LỜI CẢM ƠN.....	2
LỜI CAM ĐOAN.....	3
NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN.....	4
NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN.....	5
MỤC LỤC.....	6
DANH MỤC BẢNG.....	10
DANH MỤC SƠ ĐỒ.....	11
DANH MỤC HÌNH.....	12
LỜI NÓI ĐẦU.....	14
CHƯƠNG 1 TỔNG QUAN.....	15
1.1 Lý do chọn đề tài.....	15
1.2 Mục tiêu và chức năng của hệ thống.....	15
1.2.1 Mục tiêu.....	15
1.2.2 Chức năng.....	15
1.3 Mô tả bài toán.....	16
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT.....	19
2.1 Giới thiệu về Python.....	19
2.1.1 Tổng quát.....	19
2.1.2 Quá trình phát triển của ASP.NET.....	20
2.1.2.1 Tại sao có tên là Python.....	20
2.1.2.2 Ý tưởng của Python.....	20
2.1.2.3 Bản phát hành đầu tiên.....	20
2.1.2.4 Sự ra đời của Python 3.....	21
2.1.2.5 Sự tăng trưởng vượt bậc.....	22
2.1.3 Ứng dụng của Python.....	23
2.2 Giới thiệu Django.....	23
2.2.1 Django là gì?.....	23
2.2.2 Ưu điểm của Django.....	24
2.2.3 Nhược điểm.....	25
2.2.4 Vì sao nên sử dụng Django trong lập trình web?.....	25
2.2.5 Mô hình MVT của Django.....	26

2.2.6 Một số ứng dụng web nổi tiếng sử dụng Django.....	27
2.3 Giới thiệu MySQL	28
2.3.1 MySQL là gì?	28
2.3.2 MySQL hình thành như thế nào?.....	28
2.3.3 Cơ chế hoạt động của MySQL	29
2.3.4 Ưu và nhược điểm của MySQL.....	29
2.3.4.1 Ưu điểm	29
2.3.4.2 Nhược điểm	29
2.3.5 Các công nghệ khác	30
CHƯƠNG 3 PHÂN TÍCH HỆ THỐNG	31
3.1 Mô tả hệ thống	31
3.1.1 Website bán sim.....	31
3.1.2 Quản lý sim.....	31
3.1.3 Quản lý khách hàng	32
3.1.4 Quản lý đơn hàng.....	33
3.1.5 Quản lý bài viết.....	33
3.2 Xây dựng các chức năng của hệ thống	34
3.2.1 Mô tả các chức năng chi tiết	34
3.2.1.1 Website bán sim.....	34
3.2.1.2 Quản lý sim.....	34
3.2.1.3 Quản lý khách hàng	35
3.2.1.4 Quản lý đơn hàng.....	35
3.2.1.5 Quản lý bài viết.....	36
3.2.2 Xác định use case.....	36
3.2.2.1 Admin	36
3.2.2.2 Khách hàng	37
3.2.3 Đặc tả các use case	38
3.2.3.1 Mô tả quá trình đăng ký.....	38
3.2.3.2 Mô tả quá trình đăng nhập.....	38
3.2.3.3 Mô tả quá trình quản lý của admin	39
3.2.3.4 Mô tả quá trình thêm vào giỏ hàng.....	40
3.2.3.5 Mô tả quá trình bỏ ra khỏi giỏ hàng	40

3.2.3.6	Mô tả quá trình đặt hàng	41
3.2.3.7	Mô tả quá trình hủy hàng	41
3.3	Thiết kế hệ thống	43
3.3.1	Sơ đồ cơ sở dữ liệu	43
3.3.2	Chi tiết cơ sở dữ liệu.....	44
3.3.2.1	Quản lý sim và các thông tin liên quan	44
3.3.2.2	Quản lý khách hàng	44
3.3.2.3	Quản lý đơn hàng.....	45
3.3.2.4	Quản lý bài viết.....	46
CHƯƠNG 4	XÂY DỰNG WEBSITE	47
4.1	Giao diện và chức năng người dùng	47
4.1.1	Trang chủ	47
4.1.2	Trang cửa hàng	48
4.1.3	Trang chi tiết sản phẩm.....	53
4.1.4	Trang đăng ký	55
4.1.5	Trang đăng nhập	57
4.1.6	Trang quên mật khẩu	58
4.1.7	Trang thay đổi thông tin	59
4.1.8	Trang thay đổi mật khẩu	60
4.1.9	Trang giỏ hàng	60
4.1.10	Trang đặt hàng	62
4.1.11	Trang quản lý đơn hàng.....	63
4.1.12	Trang bài viết và danh sách bài viết	63
4.1.13	Trang quản lý của admin	65
CHƯƠNG 5	KẾT LUẬN	71
5.1	Kết quả đạt được	71
5.2	Hướng phát triển	71
CHƯƠNG 6	TÀI LIỆU THAM KHẢO	72
CHƯƠNG 7	PHỤ LỤC	73
7.1	Cấu trúc thư mục của project.....	73
7.2	Code xử lý.....	74
7.2.1	Cấu hình trong file settings.py.....	74

7.2.2 Lớp định nghĩa các đối tượng mà trang quản lý.....	78
7.2.3 Lớp định nghĩa các form nhập dữ liệu.....	88
7.2.4 Cấu hình url cho ứng dụng web.....	96
7.2.5 Xây dựng views tương tác giữa giao diện, người dùng, hệ thống.....	97

DANH MỤC BẢNG

Commented [PN1]: Sang trang mới

Bảng 3.1 Bảng mô tả chức năng quản lý giỏ hàng	34
Bảng 3.2 Bảng mô tả chức năng quản lý đơn hàng	34
Bảng 3.3 Bảng mô tả chức năng quản lý thông tin cá nhân	34
Bảng 3.4 Bảng mô tả chức năng quản lý sim	34
Bảng 3.5 Bảng mô tả chức năng quản lý nhà mạng	34
Bảng 3.6 Bảng mô tả chức năng quản lý phân loại	35
Bảng 3.7 Bảng mô tả chức năng quản lý khách hàng.....	35
Bảng 3.8 Bảng mô tả chức năng quản lý đơn hàng	35
Bảng 3.9 Bảng mô tả chức năng quản lý danh mục bài viết	36
Bảng 3.10 Bảng mô tả chức năng quản lý bài viết	36
Bảng 3.11 Bảng mô tả thuộc tính của đối tượng Network	44
Bảng 3.12 Bảng mô tả thuộc tính của đối tượng Tag (Phân loại kiểu sim số đẹp)	44
Bảng 3.13 Bảng mô tả thuộc tính của đối tượng Sim.....	44
Bảng 3.14 Bảng mô tả thuộc tính của đối tượng Khách hàng.....	45
Bảng 3.15 Bảng mô tả thuộc tính của đối tượng thành phần giỏ hàng.....	45
Bảng 3.16 Bảng mô tả thuộc tính của đối tượng đơn hàng	45
Bảng 3.17 Bảng mô tả thuộc tính của đối tượng chủ đề bài viết.....	46
Bảng 3.18 Bảng mô tả thuộc tính của đối tượng bài viết	46

DANH MỤC SƠ ĐỒ

Sơ đồ 3.1 Sơ đồ ER của chức năng quản lý sim.....	31
Sơ đồ 3.2 Mô hình ER quản lý khách hàng	32
Sơ đồ 3.3 Mô hình ER quản lý đơn hàng	33
Sơ đồ 3.4 Mô hình ER quản lý bài viết	33
Sơ đồ 3.5 Sơ đồ use case của đối tượng Admin	37
Sơ đồ 3.6. Sơ đồ use case của đối tượng khách hàng (Customer).....	37
Sơ đồ 3.7. Sơ đồ tuần tự quá trình đăng ký	38
Sơ đồ 3.8 Sơ đồ hợp tác quá trình đăng nhập.....	38
Sơ đồ 3.9 Sơ đồ tuần tự quá trình đăng nhập.....	39
Sơ đồ 3.10 Sơ đồ hợp tác quá trình đăng nhập.....	39
Sơ đồ 3.11 Sơ đồ tuần tự quá trình quản lý của admin với các dữ liệu mà trang web quản lý	39
Sơ đồ 3.12 Sơ đồ hợp tác quá trình quản lý của admin với các dữ liệu mà trang web quản lý.....	40
Sơ đồ 3.13 Sơ đồ tuần tự quá trình thêm vào giỏ hàng	40
Sơ đồ 3.14 Sơ đồ hợp tác quá trình thêm vào giỏ hàng.....	40
Sơ đồ 3.15 Sơ đồ tuần tự quá trình thêm vào giỏ hàng	41
Sơ đồ 3.16 Sơ đồ hợp tác quá trình thêm vào giỏ hàng.....	41
Sơ đồ 3.17 Sơ đồ tuần tự quá trình đặt hàng	41
Sơ đồ 3.18 Sơ đồ tuần tự quá trình đặt hàng	41
Sơ đồ 3.19 Sơ đồ tuần tự quá trình hủy đơn	42
Sơ đồ 3.20 Sơ đồ tuần tự quá trình hủy đơn	42
Sơ đồ 3.21 Sơ đồ cơ sở dữ liệu	43

DANH MỤC HÌNH

Hình 2.1 Biểu tượng của Python là hình 2 con rắn tạo thành 2 ký tự đầu là Py ..	19
Hình 2.2 Dòng thời gian phát triển của các phiên bản Python	21
Hình 2.3 Thống kê số câu hỏi trên Stack Overflow liên quan đến mỗi ngôn ngữ	22
Hình 2.4. Logo của Django.....	23
Hình 2.5 Giao diện trang quản lý của Django Admin	24
Hình 2.6. Cấu hình file settings.py	25
Hình 2.7 Mô hình MVT của Django	27
Hình 2.8 Logo của MySQL	28
Hình 4.1 Giao diện phần trên của trang chủ	47
Hình 4.2 Thanh header khi không có người đăng nhập.....	47
Hình 4.3 Thanh header khi có người đăng nhập.....	47
Hình 4.4 Danh sách sim đang khuyến mãi hiển thị ở trang chủ	48
Hình 4.5 Phần footer của trang web	48
Hình 4.6 Phần danh mục tìm kiếm sim theo 2 cách (nhà mạng hoặc kiểu số đẹp)	49
Hình 4.7 Phần filter phía trên để tìm kiếm sim.....	49
Hình 4.8 Phần filter phía dưới để tìm kiếm sim	50
Hình 4.9 Danh sách sim hiển thị theo dạng gridview.....	51
Hình 4.10 Trang web khi đang load thêm danh sách sim.....	51
Hình 4.11 Sau khi load thành công thêm sim	52
Hình 4.12 Giao diện tổng thể cả trang cửa hàng	53
Hình 4.13 Trang chi tiết sim	54
Hình 4.14 Phần hiển thị giỏ hàng	55
Hình 4.15 Giao diện phần đăng ký	56
Hình 4.16 Giao diện của trang đăng ký khi bị lỗi.....	56
Hình 4.17 Giao diện thông báo đăng ký thành công	57
Hình 4.18 Mail xác nhận đăng ký được gửi về.....	57
Hình 4.19 Giao diện trang web sau khi ấn đường link xác nhận.....	57
Hình 4.20 Giao diện trang đăng nhập	58
Hình 4.21 Giao diện trang lấy lại mật khẩu	58
Hình 4.22 Email yêu cầu lấy lại mật khẩu được gửi đến email đăng ký.....	59

Hình 4.23	Giao diện điền mật khẩu mới sau khi ấn đường link	59
Hình 4.24	Giao diện trang thay đổi thông tin	60
Hình 4.25	Giao diện trang thay đổi mật khẩu	60
Hình 4.26	Giao diện trang giỏ hàng khi trống	61
Hình 4.27	Giao diện trang giỏ hàng khi có sản phẩm	61
Hình 4.28	Giao diện trang giỏ hàng sau khi thêm 1 sim mới	61
Hình 4.29	Giao diện trang đặt hàng	62
Hình 4.30	Thông báo đặt hàng thành công	62
Hình 4.31	Giao diện trang quản lý đơn hàng	63
Hình 4.32	Giao diện trang chi tiết bài viết	64
Hình 4.33	Hiện thị danh sách bài viết	64
Hình 4.34	Giao diện trang quản lý chính của admin	65
Hình 4.35	Giao diện thanh điều hướng phần quản lý bài viết	65
Hình 4.36	Giao diện trang quản lý bài viết	66
Hình 4.37	Giao diện thêm hoặc chỉnh sửa nội dung bài viết	66
Hình 4.38	Giao diện danh sách chủ đề bài viết	66
Hình 4.39	Giao diện trang thêm hoặc sửa chủ đề	67
Hình 4.40	Phần sidebar truy cập quản lý khách hàng	67
Hình 4.41	Giao diện quản lý tài khoản	67
Hình 4.42	Giao diện thêm hoặc sửa thông tin tài khoản	68
Hình 4.43	Giao diện chức năng quản lý sim	68
Hình 4.44	Giao diện danh sách sim	68
Hình 4.45	Giao diện danh sách sim theo nhà mạng	69
Hình 4.46	Giao diện danh sách sim theo kiểu nhà mạng	69
Hình 4.47	Giao diện trang quản lý đơn hàng	69
Hình 4.48	Giao diện thêm hoặc chỉnh sửa thông tin đơn hàng	70

LỜI NÓI ĐẦU

Trong thời đại số hóa ngày nay, thương mại điện tử đang trở thành xu hướng phát triển mạnh mẽ và không ngừng mở rộng. Đặc biệt, với tình hình dịch COVID-19 lan rộng khiến nhiều người phải tạm dừng hoạt động ngoài đời thực và chuyển sang tiếp cận và mua sắm trực tuyến, việc xây dựng website bán hàng trực tuyến đã trở thành một sự lựa chọn hợp lý và tiềm năng để phát triển kinh doanh.

Với mong muốn cung cấp một giải pháp hiệu quả cho việc xây dựng website bán hàng trực tuyến, chúng ta có thể sử dụng Django - một framework phổ biến trong lập trình web. Điểm nổi bật của Django chính là khả năng tích hợp các tính năng tiện ích để giúp quản lý dữ liệu trên website trở nên đơn giản và tiết kiệm thời gian. Điều này cho phép chúng ta tập trung vào việc phát triển các chức năng khác, nhằm đem lại trải nghiệm tốt nhất cho người dùng.

Chẳng hạn, nếu chúng ta muốn xây dựng một website bán sim số đẹp, Django có thể giúp chúng ta quản lý thông tin về các sim số, giá cả, mô tả sản phẩm và các thông tin khác liên quan. Đồng thời, Django cũng hỗ trợ tính năng tìm kiếm, sắp xếp và lọc dữ liệu, giúp người dùng dễ dàng tìm kiếm và lựa chọn sản phẩm phù hợp.

CHƯƠNG 1 TỔNG QUAN

1.1 Lý do chọn đề tài

Việc xây dựng website bán sim số đẹp là một đề tài hấp dẫn và tiềm năng trong lĩnh vực thương mại điện tử. Dưới đây là một số lý do nổi bật cho việc lựa chọn đề tài này:

- **Sự phổ biến của sim số đẹp:** Xã hội ngày càng phát triển, nhiều doanh nghiệp được thành lập, với mong muốn của các doanh nghiệp được khách hàng nhớ đến, họ cũng muốn thông tin liên hệ họ
- **Sự tiện lợi cho khách hàng:** Với việc mua sắm trực tuyến, khách hàng có thể lựa chọn và mua hàng từ bất cứ đâu, bất cứ lúc nào. Không chi tiết kiệm thời gian và công sức, khách hàng còn được đảm bảo về chất lượng sản phẩm, đặc biệt là với các sim số đẹp có giá trị cao.
- **Sự linh hoạt trong quản lý sản phẩm:** Việc quản lý các sản phẩm sim số đẹp trên website giúp cho chủ cửa hàng dễ dàng cập nhật và kiểm soát thông tin sản phẩm. Điều này cũng giúp cho việc cung cấp thông tin chính xác và chi tiết hơn cho khách hàng, đồng thời giúp tăng tính chuyên nghiệp và uy tín cho cửa hàng.

Với những lý do trên, xây dựng website bán sim số đẹp trực tuyến sẽ mang lại nhiều cơ hội và tiềm năng kinh doanh cho chủ cửa hàng, đồng thời đem lại sự tiện lợi và chất lượng dịch vụ cho khách hàng.

1.2 Mục tiêu và chức năng của hệ thống

1.2.1 Mục tiêu

Xây dựng trang web giúp người mua có thể tìm kiếm được sản phẩm (ở đây là sim) đúng với nhu cầu mà bản thân đề ra.

Bên cạnh đó người bán (người quản trị trang web) có thể dễ dàng quản lý được sản phẩm đang bán, quản lý khách hàng, quản lý đơn hàng, quản lý bài viết.

1.2.2 Chức năng

+ Đối với người mua:

- Đăng ký, đăng nhập, lấy lại tài khoản khi quên mật khẩu qua email.
- Thay đổi thông tin cá nhân (bao gồm mật khẩu)

- Quản lý đơn hàng đã đặt
- Xem bài viết.
- Lọc và tìm kiếm sim theo nhu cầu.
- Xem chi tiết thông tin sim.
- Đặt sim qua web.

+ Đối với người mua, ngoài các chức năng mà khách hàng thực hiện, họ còn có thể:

- Quản lý người dùng.
- Quản lý danh mục và sản phẩm.
- Quản lý đơn đặt hàng qua web.
- Quản lý bài viết.

1.3 Mô tả bài toán

➤ Mục đích của website quản lý sản phẩm, bài đăng trên website, hiển thị và cung cấp công cụ hỗ trợ người dùng tìm kiếm sim theo nhu cầu của họ.

✓ Quản lý danh mục và sim:

- Quản lý thông tin nhà mạng: mã, tên nhà mạng, ảnh thương hiệu nhà mạng, ảnh sim card của nhà mạng
- Tìm kiếm sim theo nhà mạng

✓ Quản lý người dùng:

- Quản lý thông tin người dùng: Mã, tên người dùng, họ, tên, email, số điện thoại, địa chỉ, giới tính, ngày tạo, ngày đăng nhập gần nhất, quyền tài khoản (là admin), quyền đăng nhập
- Chặn người dùng đăng nhập

✓ Quản lý đơn hàng:

- Quản lý thông tin đơn hàng: thông tin liên hệ, trạng thái đơn hàng, thông tin vận chuyển, thông tin thanh toán, thông tin dùng để đăng ký sim.
- Hủy hoặc xóa đơn hàng tùy theo yêu cầu.

✓ Quản lý bài viết:

- Quản lý bài viết: Nội dung, tiêu đề, hình ảnh, tóm tắt
- Xóa bài viết.

- Mục đích của website bán sim là giúp người mua có thể dễ dàng tìm kiếm sim theo nhu cầu, xem chi tiết thông tin sim, để rồi từ đó có thể đưa ra quyết định đặt mua hay không, nếu muốn đặt có thể đặt qua web luôn, từ đó người bán có thể dễ dàng tiếp nhận yêu cầu và chủ động liên hệ, để xác nhận và lập giao dịch
 - ✓ Đăng ký, đăng nhập, thay đổi thông tin:
 - Đăng ký: Người dùng đăng ký tài khoản với thông tin cá nhân như số điện thoại với email.
 - Đăng nhập: Người dùng đăng nhập vào hệ thống sau khi đã xác nhận tài khoản với email.
 - Thay đổi thông tin đăng ký.
 - ✓ Lấy lại mật khẩu đăng nhập:
 - Sử dụng 1 SMTP server để gửi mail về tài khoản yêu cầu, từ đó người dùng có thể đặt lại mật khẩu.
 - ✓ Hiển thị danh sách sim dưới dạng gridview.
 - Mỗi sim được hiển thị dưới dạng thẻ với bố cục đơn giản hiển thị ngắn gọn thông tin bao gồm nhà mạng, số sim, giá tiền.
 - ✓ Tìm kiếm theo sim bằng công cụ hoặc thông qua danh mục sim:
 - Có thể tìm kiếm dựa vào số mong muốn, có thể tìm dựa vào số đầu, số cuối, né số xấu, tìm theo nhà mạng, kiểu sim đẹp hay là tìm theo ngày tháng năm sinh.
 - ✓ Hiển thị chi tiết :
 - Hiển thị chi tiết các thông tin liên quan đến sim, bên cạnh đó đề xuất các sim khác cùng nhà mạng mà trang web đang có
 - ✓ Quản lý giỏ hàng:
 - Khi mà người dùng có ý định mua sim, có thể ấn thêm vào giỏ hàng, để người dùng có thể cân nhắc trước khi quyết định đặt mua.
 - Khi cảm thấy không muốn mua nữa, họ có thể quyết định bỏ ra khỏi giỏ hàng, và thêm vào sim khác.
 - ✓ Đặt hàng:
 - Người dùng sau khi thêm sim mình cần mua, có thể chuyển qua đặt hàng.
 - Khi đặt hàng, khách hàng cần phải điền thông tin liên hệ (mặc định hệ thống sẽ dùng dữ liệu mà người dùng đăng ký), khách hàng cần chọn phương thức

thanh toán (bắt buộc), và có thể gửi cả dữ liệu ảnh đăng ký sim cho cửa hàng, để có thể nhanh chóng đăng ký.

- ✓ Xem bài viết:
 - Người dùng có thể xem bài viết đăng trên website.

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

2.1 Giới thiệu về Python

2.1.1 Tổng quát

Hiểu đơn giản, Python là một ngôn ngữ lập trình bậc cao, mã nguồn mở và đa nền tảng. Python được Guido van Rossum giới thiệu vào năm 1991 và đã trải qua 3 giai đoạn phát triển khác nhau tương ứng với các version, mới nhất hiện nay là Python version 3x.



Hình 2.1 Biểu tượng của Python là hình 2 con rắn tạo thành 2 ký tự đầu là Py

Python được thiết kế với tư tưởng giúp người học dễ đọc, dễ hiểu và dễ nhớ; vì thế ngôn ngữ Python có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học. Cấu trúc của Python cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu, nói cách khác thì so với các ngôn ngữ lập trình khác, chúng ta có thể sử dụng ít dòng code hơn để viết ra một chương trình trong Python.

Ban đầu, Python được phát triển để chạy trên nền Unix, vì thế nó là mã nguồn mở. Sau này qua thời gian phát triển, Python mở rộng và hiện nay đã hỗ trợ hầu hết các nền tảng khác như Window hay MacOS.

Python là một ngôn ngữ lập trình đa mẫu hình, nó hỗ trợ hoàn toàn mẫu lập trình hướng đối tượng và lập trình cấu trúc; ngoài ra về mặt tính năng, Python cũng hỗ trợ lập trình hàm và lập trình hướng khía cạnh. Nhờ vậy mà Python có thể làm được rất nhiều thứ, sử dụng trong nhiều lĩnh vực khác nhau.

2.1.2 Quá trình phát triển của ASP.NET

2.1.2.1 Tại sao có tên là Python

Python không phải được đặt theo tên của con rắn thần Python trong thần thoại Hy Lạp. Rossum là fan của một sê-ri chương trình hài cuối những năm 1970, và cái tên “Python” được lấy từ tên một phần trong sê-ri đó “Monty Python’s Flying Circus.”

2.1.2.2 Ý tưởng của Python

Python được hình thành vào cuối những năm 1980. Thời gian đó, Guido van Rossum làm việc trong một dự án tại CWI, có tên là Amoeba, một hệ điều hành phân tán. Trong một cuộc phỏng vấn với Bill Venners, Guido van Rossum cho biết: “Vào đầu những năm 1980, tôi làm việc với tư cách là người triển khai một nhóm xây dựng ngôn ngữ có tên là ABC tại Centrum voor Wiskunde en Informatica (CWI). Tôi không biết mọi người hiểu như thế nào về Ảnh hưởng của ABC đối với Python. Tôi cố gắng đề cập đến ảnh hưởng của ABC vì tôi mang ơn tất cả những gì tôi đã học được trong dự án đó và những người đã làm việc với nó.”

Sau đó trong cùng một Cuộc phỏng vấn, Guido van Rossum tiếp tục: “Tôi nhớ lại tất cả kinh nghiệm và những điều không hài lòng của mình với ABC. Tôi quyết định cố gắng thiết kế một ngôn ngữ kịch bản đơn giản có một số đặc tính tốt hơn của ABC. Vì vậy, tôi bắt đầu nhập. Tôi đã tạo một máy ảo đơn giản, một trình phân tích cú pháp đơn giản và một thời gian chạy đơn giản. Tôi đã tạo phiên bản của riêng mình cho các phần ABC khác nhau mà tôi thích. Tôi đã tạo một cú pháp cơ bản, sử dụng thụt lề để nhóm câu lệnh thay vì dấu ngoặc nhọn hoặc khối begin-end và phát triển một số lượng nhỏ các kiểu dữ liệu mạnh mẽ: bảng băm (hoặc từ điển, như chúng tôi gọi), danh sách, chuỗi và số.”

2.1.2.3 Bản phát hành đầu tiên.

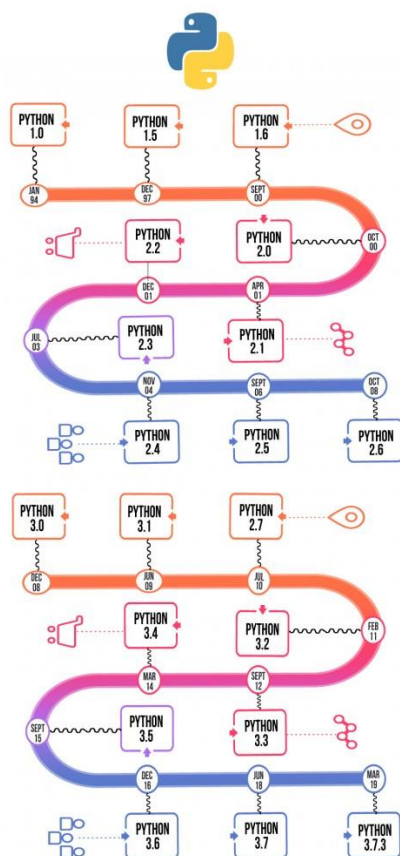
Guido Van Rossum đã xuất bản phiên bản đầu tiên của mã Python (phiên bản 0.9.0) tại alt.sources vào tháng 2 năm 1991. Bản phát hành này đã bao gồm xử lý ngoại lệ, các hàm và các kiểu dữ liệu cốt lõi của list, dict, str và các loại khác. Nó cũng hướng đối tượng và có một hệ thống mô-đun. Phiên bản Python 1.0 được phát hành vào tháng 1 năm 1994. Các tính năng mới chính trong bản phát hành này là các công cụ lập trình chức năng lambda, map, filter và reduce, những thứ mà Guido Van Rossum không bao giờ thích. Sáu năm rưỡi sau vào tháng 10 năm 2000, Python 2.0 được giới thiệu. Bản

phát hành này bao gồm toàn bộ danh sách, một bộ thu gom rác đầy đủ và nó hỗ trợ unicode.

2.1.2.4 Sự ra đời của Python 3.

Bắt đầu từ năm 2000, các nhà phát triển cốt lõi bắt đầu nghĩ về Python 3.0. Họ muốn hợp lý hóa ngôn ngữ, cắt các cấu trúc và chức năng ngôn ngữ không cần thiết mà Python đã tích lũy được trong gần 20 năm tồn tại của nó. Như Zen of Python nói: “Nên có một — và tốt nhất là chỉ một — cách rõ ràng để làm điều đó”.

Những nỗ lực của họ đã dẫn đến Python 3.0, một phiên bản không tương thích ngược của ngôn ngữ Python được phát hành vào tháng 12 năm 2008. Thật không may, bản phát hành đã mang lại một số phức tạp.



Hình 2.2 Dòng thời gian phát triển của các phiên bản Python

Các nhà phát triển đã không nhận ra bao nhiêu Python được sử dụng và bao nhiêu mã Python ngoài tự nhiên phụ thuộc vào các thư viện Python khác. Do đó, mặc dù dễ dàng chuyển các tập lệnh của một người sang Python 3, nhưng việc di chuyển các chương trình dựa vào thư viện của bên thứ ba lại khó hơn nhiều vì chúng không nâng cấp nhanh như vậy.

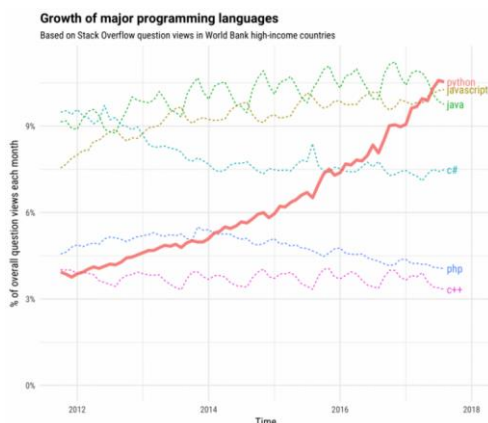
Điều này dẫn đến việc chuyển đổi khá rắc rối và gây đau đớn cho một số người nhưng đã cải thiện đáng kể ngôn ngữ. Python 2 cuối cùng đã ngừng hoạt động vào năm 2020.

2.1.2.5 Sự tăng trưởng vượt bậc

Năm 1999, Guido van Rossum xác định mục tiêu của mình cho Python:

- Một ngôn ngữ dễ dàng và trực quan cũng mạnh mẽ như ngôn ngữ của các đối thủ cạnh tranh lớn;
- Nguồn mở, vì vậy bất kỳ ai cũng có thể đóng góp vào sự phát triển của nó;
- Mã dễ hiểu như tiếng Anh đơn giản;
- Thích hợp cho các công việc hàng ngày, cho phép thời gian phát triển ngắn.

Khoảng 20 năm sau, rõ ràng là tất cả những dự định này đã được thực hiện. Mặc dù Python đang phát triển ổn định về cơ bản toàn bộ sự tồn tại của nó, từ khoảng năm 2010, nó bắt đầu trên một quỹ đạo phát triển khiến nó sớm cho phép nó sánh ngang với các ngôn ngữ lập trình hàng đầu khác, chẳng hạn như Java và JavaScript.



Hình 2.3 Thống kê số câu hỏi trên Stack Overflow liên quan đến mỗi ngôn ngữ

2.1.3 Ứng dụng của Python

Python là ngôn ngữ được ứng dụng đa dạng trong các lĩnh vực.

- **Làm Web với các Framework của Python:** Django và Flask là 2 framework phổ biến hiện nay dành cho các lập trình viên Python để tạo ra các website.
- **Tool tự động hóa:** các ứng dụng như từ điển, crawl dữ liệu từ website, tool giúp tự động hóa công việc được các lập trình viên ưu tiên lựa chọn Python để viết nhờ tốc độ code nhanh của nó.
- **Khoa học máy tính:** Trong Python có rất nhiều thư viện quan trọng phục vụ cho ngành khoa học máy tính như: OpenCV cho xử lý ảnh và machine learning, Scipy và Numpys cho lĩnh vực toán học, đại số tuyến tính, Pandas cho việc phân tích dữ liệu, ...
- **Lĩnh vực IoT:** Python có thể viết được các ứng dụng cho nền tảng nhúng, đồng thời cũng được lựa chọn cho việc xử lý dữ liệu lớn. Vì thế Python là một ngôn ngữ quen thuộc trong lĩnh vực Internet kết nối vạn vật.
- **Làm game:** Pygame là một bộ module Python cross-platform được thiết kế để viết game cho cả máy tính và các thiết bị di động.

2.2 Giới thiệu Django

2.2.1 Django là gì?

Django là một trong số những web framework bậc cao miễn phí, là mã nguồn mở được tạo ra bởi ngôn ngữ Python dựa trên mô hình mô hình MTV (gồm Model-Template-Views). Hiện framework này được phát triển, quản lý bởi Django Software Foundation.

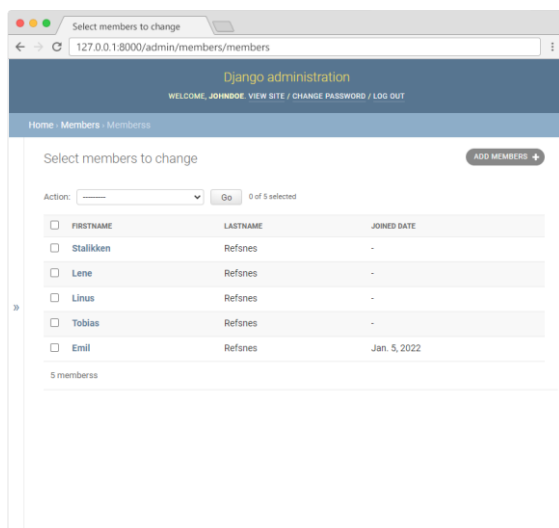


Hình 2.4. Logo của Django

Django ra đời với mục tiêu hỗ trợ thiết kế các website phức tạp dựa trên những CSDL có sẵn. Nó hoạt động dựa theo nguyên lý ‘cắm’ các thành phần và tái sử dụng để tạo nên các website với ít code, ít khớp nối, có khả năng phát triển và không bị trùng lặp.

2.2.2 Ưu điểm của Django

Lợi thế hàng đầu của Django là khả năng thiết kế, tạo lập website và các ứng dụng nhanh chóng đến bất ngờ. Ngoài ra, dưới đây Bizfly liệt kê những điểm cộng khiến framework này trở nên nổi bật và được lập trình viên sử dụng rộng rãi.



Hình 2.5 Giao diện trang quản lý của Django Admin

- **Bảo mật tốt:** Làm việc với Django, các lập trình viên gần như không có không gian để phạm bất cứ sai lầm về an ninh nào. Nó giúp các developer tránh được tất cả các lỗi thường gặp như nhấp chuột, kịch bản chéo trang, SQL tiêm, giả mạo yêu cầu,... Nhờ đó, sản phẩm được tạo bởi framework này có khả năng bảo mật cực tốt.
- **Mở rộng thoải thích:** Django có sẵn tính năng mở rộng nhằm hỗ trợ các lập trình viên quản lý lưu lượng người truy cập, thích hợp với các trang có traffic lớn.
- **Đễ sử dụng:** Django được tạo ra bởi ngôn ngữ lập trình Python và mô hình MVC nên rất dễ ứng dụng trong các dự án. Đa ngôn ngữ và được hỗ trợ Multi-Site

- **Dễ học:** Có nhiều tài liệu hỗ trợ việc học Django, bao gồm cả tài liệu miễn phí trên mạng và sách in. Cộng đồng sử dụng Django hiện nay đang phát triển mạnh mẽ, newbie có thể tìm kiếm sự giúp đỡ trên các trang facebook, diễn đàn, blog,...

2.2.3 Nhược điểm

Bên cạnh những ưu điểm kể trên, Django cũng tồn tại một số nhược điểm mà ta cần cân nhắc trước khi sử dụng chúng. Cụ thể sau đây:

- Có thể xảy ra một vài vấn đề khi phát triển các ứng dụng/website quy mô nhỏ
- Định tuyến tương đối khó
- Không đưa ra các cảnh báo khi xuất hiện lỗi trong mẫu.

2.2.4 Vì sao nên sử dụng Django trong lập trình web?

Đây là câu hỏi của khá nhiều newbie khi bắt đầu tìm hiểu Django là gì. Có nhiều nguyên nhân khiến framework này được ưu ái. Có thể kể đến vài lợi ích sau.

```

25 # SECURITY WARNING: don't run with debug turned on in production!
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30 # Application definition
31
32 INSTALLED_APPS = [
33     'django.contrib.admin',
34     'django.contrib.auth',
35     'django.contrib.contenttypes',
36     'django.contrib.sessions',
37     'django.contrib.messages',
38     'django.contrib.staticfiles',
39     'polls.apps.'
40 ]
41
42 MIDDLEWARE = [
43     'django.middleware.common.CommonMiddleware',
44     'django.middleware.csrf.CsrfViewMiddleware',
45 ]
46
47

```

Hình 2.6. Cấu hình file settings.py

- Nhanh chóng, đơn giản trong lập trình web/ứng dụng
- Đa dạng về tài liệu
- Sử dụng nguyên tắc DRY - không trùng lặp
- Hỗ trợ ORM, mang lại sự tương thích, khả năng hoạt động với các CSDL thường gặp như Oracle và SQL

- Sở hữu nhiều tính năng, giải pháp độc đáo, hữu ích như GPS, chatbot, ứng dụng CSDL, nguồn cung cấp dữ liệu RSS,... giúp phát triển, quản trị website, chăm sóc khách hàng,...
- Mang lại khả năng sửa đổi (thêm, xóa) một cách liền mạch nhiều thành phần
- Thân thiện với Seo
- Linh hoạt cao: Xây dựng nhiều dạng website, ứng dụng cần thiết cho mọi lĩnh vực, tổ chức, cơ quan.

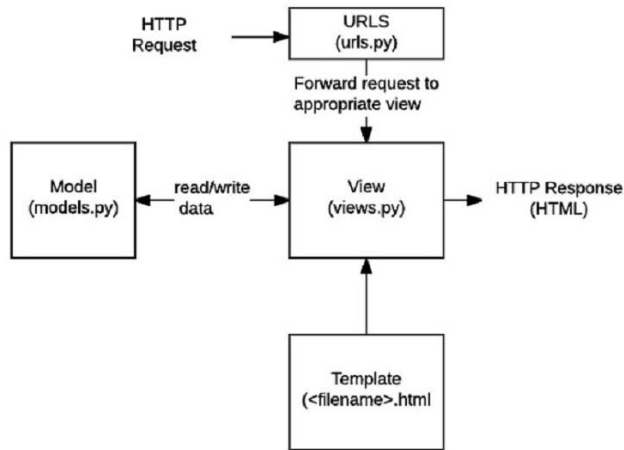
2.2.5 Mô hình MVT của Django

Django là một framework web phát triển theo mô hình MVT (Model-View-Template). Mô hình MVT tương tự với mô hình MVC (Model-View-Controller), nhưng có sự khác biệt nhỏ về cách thức hoạt động.

- **Model:** Chịu trách nhiệm cho việc xử lý các tương tác với cơ sở dữ liệu. Nó đại diện cho dữ liệu và quản lý truy cập vào cơ sở dữ liệu thông qua các truy vấn (query).
- **View:** Chịu trách nhiệm cho việc xử lý logic và điều khiển chương trình. Nó đại diện cho các chức năng của ứng dụng và xử lý các yêu cầu từ phía người dùng.
- **Template:** Chịu trách nhiệm hiển thị dữ liệu cho người dùng. Nó đại diện cho giao diện và quản lý việc hiển thị dữ liệu cho người dùng.

Django cung cấp một cơ chế hoạt động rất linh hoạt, cho phép người dùng xây dựng ứng dụng web một cách nhanh chóng và hiệu quả. Nó cung cấp nhiều tính năng hỗ trợ phát triển web như: quản lý đường dẫn (URL), tương tác với cơ sở dữ liệu, quản lý phiên (session), quản lý tài nguyên tĩnh, bảo mật và xác thực, hỗ trợ gửi email, và các tính năng khác.

Để xây dựng một ứng dụng web Django, người dùng cần định nghĩa các mô hình dữ liệu, cài đặt các chức năng xử lý (views), và thiết kế các template để hiển thị dữ liệu. Sau đó, Django sẽ kết hợp các phần này với các chức năng của nó để tạo ra một ứng dụng web hoàn chỉnh và chạy trên máy chủ web.



Hình 2.7 Mô hình MVT của Django

2.2.6 Một số ứng dụng web nổi tiếng sử dụng Django

Django là một trong những framework web phổ biến nhất hiện nay và được sử dụng bởi rất nhiều các website lớn. Dưới đây là một số ví dụ về những website nổi tiếng sử dụng Django:

- **Instagram:** Được mua lại bởi Facebook vào năm 2012, Instagram là một trong những ứng dụng chia sẻ hình ảnh và video phổ biến nhất trên thế giới, và nó được xây dựng bằng Django.
- **YouTube:** Trang web chia sẻ video lớn nhất thế giới cũng sử dụng Django để xây dựng hệ thống của mình.
- **Pinterest:** Là một trang web giúp người dùng khám phá và lưu trữ các ý tưởng, hình ảnh và video phổ biến, Pinterest cũng được xây dựng bằng Django.
- **Dropbox:** Dropbox là một dịch vụ lưu trữ đám mây phổ biến, và cũng sử dụng Django để xây dựng trang web và các ứng dụng của mình.
- **Mozilla:** Mozilla là một tổ chức phi lợi nhuận và được biết đến với trình duyệt web Firefox của mình. Website chính của Mozilla cũng được xây dựng bằng Django.

2.3 Giới thiệu MySQL

2.3.1 MySQL là gì?

MySQL là 1 hệ thống quản trị về cơ sở dữ liệu với mã nguồn mở (được gọi tắt là RDBMS) và đang hoạt động theo mô hình dạng client-server. Đối với RDBMS – Relational Database Management System thì MySQL đã được tích hợp apache và PHP.

Được phát hành chính thức từ thập niên 90s, MySQL hiện đang quản lý dữ liệu qua những cơ sở dữ liệu, với mỗi một cơ sở dữ liệu hoàn toàn có thể có rất nhiều những bản quan hệ có chứa dữ liệu. Ngoài ra, MySQL cũng có cùng 1 cách thức truy xuất cũng như mã lệnh tương tự cùng với ngôn ngữ SQL.



Hình 2.8 Logo của MySQL

2.3.2 MySQL hình thành như thế nào?

- Vào năm 1994, một công ty của Thụy Điển là MySQL AB đã phát triển nên MySQL. Khái niệm MySQL là gì này cũng bắt nguồn từ đây.
- Phiên bản MySQL đầu tiên đã phát hành chính thức vào năm 1995.
- Trong năm 2008, công ty Sun Microsystems đã mua lại MySQL AB.
- Tập đoàn Oracle vào năm 2010 đã thâu tóm Sun Microsystems. Đội ngũ phát triển ngay lúc đó đã tách MySQL ra trở thành một nhánh trở thành một nhánh riêng và được gọi là riêng và được gọi là MariaDB. Oracle đã phát triển tiếp tục MySQL lên với phiên bản 5.5.
- Năm 2013 MySQL đã phát hành ra phiên bản 5.6
- Năm 2015 MySQL đã phát hành ra phiên bản 5.7
- Ở thời điểm hiện tại MySQL đang phát triển lên thành phiên bản 8.0

- Hiện tại MySQL đang có hai phiên bản miễn phí đó chính là MySQL Community Server và có phí là Enterprise Server.

2.3.3 Cơ chế hoạt động của MySQL

Cách thức vận hành chính tại MySQL môi trường hiện tại là:

- MySQL đang tạo ra bảng để có thể lưu trữ dữ liệu và định nghĩa về sự liên quan giữa những bảng đó
- Client sẽ trực tiếp gửi yêu cầu SQL bằng 1 lệnh đặc biệt có trên MySQL.
- Ứng dụng tại server sẽ tiến hành phân hồi thông tin cũng như trả về những kết quả trên máy client.

2.3.4 Ưu và nhược điểm của MySQL

2.3.4.1 Ưu điểm

- **Nhanh chóng:** Nhờ vào việc đưa ra một số những tiêu chuẩn và cho phép MySQL làm việc hiệu quả cũng như tiết kiệm chi phí, giúp gia tăng tốc độ thực thi.
- **Mạnh mẽ và khả năng mở rộng:** MySQL hoàn toàn có thể xử lý số lượng lớn dữ liệu và đặc biệt hơn thế nữa thì nó còn có thể mở rộng nếu như cần thiết.
- **Đa tính năng:** Ưu điểm MySQL là gì? MySQL hiện đang hỗ trợ nhiều những chức năng SQL rất được mong chờ từ 1 hệ quản trị CSDL quan hệ cả gián tiếp cũng như trực tiếp.
- **Độ bảo mật cao:** MySQL là gì? Hiện tại nó đang rất thích hợp cho những ứng dụng truy cập CSDL thông qua internet khi sở hữu rất nhiều những tính năng về bảo mật và thậm chí là đang ở cấp cao.

2.3.4.2 Nhược điểm

- **Dung lượng hạn chế:** Trong trường hợp nếu như số lượng bản ghi đang lớn dần lên thì khi đó quá trình truy xuất dữ liệu sẽ diễn ra vô cùng khó khăn. Như vậy cần phải áp dụng rất nhiều những biện pháp khác nhau để có thể gia tăng được tốc độ truy xuất những dữ liệu ví dụ như tạo cache MySQL hoặc chia tải database ra nhiều server.
- **Độ tin cậy:** Nhược điểm MySQL là gì? Theo đó cách thức nhận chức năng cụ thể đang được xử lý cùng với MySQL (ví dụ như kiểm toán, những giao

dịch, tài liệu tham khảo,...) khiến cho nó trở nên kém tin cậy hơn một số những hệ quản trị về cơ sở dữ liệu có quan hệ khác.

- **Giới hạn:** Theo thiết kế thì MySQL không có ý định thực hiện toàn bộ và nó đang đi kèm cùng với những hạn chế liên quan tới chức năng mà một số ứng dụng có thể cần tới.

2.3.5 Các công nghệ khác

- **Bootstrap:** Bootstrap là một framework CSS phổ biến được sử dụng để thiết kế giao diện web. Nó cung cấp các thành phần chuẩn hóa và giao diện đẹp mắt, giúp việc phát triển website trở nên dễ dàng và nhanh chóng hơn.
- **jQuery:** jQuery là một thư viện JavaScript được sử dụng để tương tác với các thành phần trên website. Nó cung cấp các hàm xử lý sự kiện, hiệu ứng, AJAX và DOM, giúp cho việc tương tác với website trở nên linh hoạt hơn.
- **Fontawesome:** Fontawesome là một thư viện biểu tượng đẹp và phong phú được sử dụng để thêm các biểu tượng vào website. Nó cung cấp các biểu tượng đa dạng và phong phú, giúp cho việc thiết kế giao diện trở nên đẹp mắt và chuyên nghiệp hơn.
- **Django-jet:** Django-jet là một giao diện quản trị mở rộng cho Django, giúp cho việc quản trị và điều khiển website trở nên dễ dàng hơn. Nó cung cấp các tính năng tiện ích như tìm kiếm, sắp xếp, lọc và phân trang dữ liệu, giúp cho việc quản lý dữ liệu trên website trở nên tiện lợi hơn.
- **Venv:** Venv là một môi trường ảo được cung cấp bởi Python để quản lý các gói phụ thuộc trong một dự án Python. Nó giúp cho việc phát triển và triển khai ứng dụng Python trở nên dễ dàng và có thể tái sử dụng được các gói phụ thuộc.
- **Ajax (Asynchronous JavaScript and XML)** là một kỹ thuật lập trình web cho phép giao tiếp và trao đổi dữ liệu giữa trang web và máy chủ mà không cần phải tải lại trang web. Điều này cho phép người dùng tương tác với trang web một cách nhanh chóng và liền mạch hơn, cải thiện trải nghiệm người dùng. Ajax được sử dụng rộng rãi trong phát triển ứng dụng web đơn trang (Single Page Application) và các trang web động (dynamic web page).

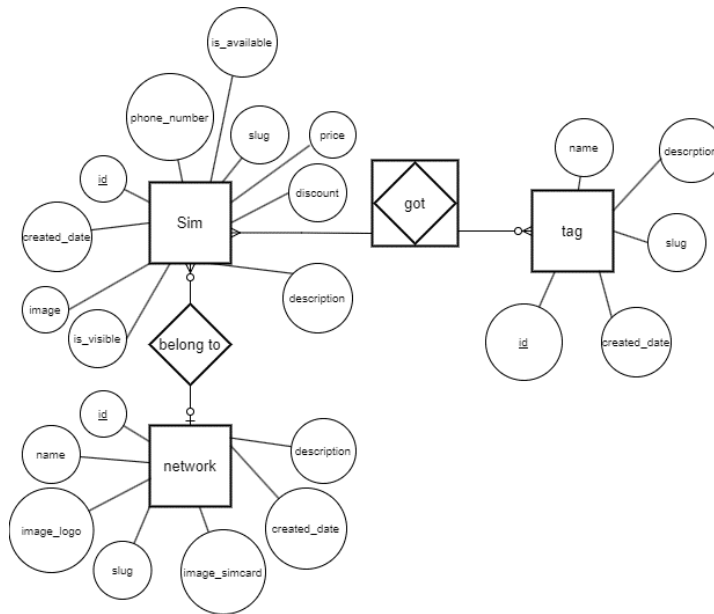
CHƯƠNG 3 PHÂN TÍCH HỆ THỐNG

3.1 Mô tả hệ thống

3.1.1 Website bán sim

- Hiển thị bài viết theo danh mục.
- Giới thiệu sim theo nhà mạng, kiểu sim số đẹp.
- Tìm kiếm sim theo yêu cầu: số đầu, số cuối, nhà mạng, kiểu số đẹp, ngày tháng năm sinh, loại bỏ số xấu.

3.1.2 Quản lý sim

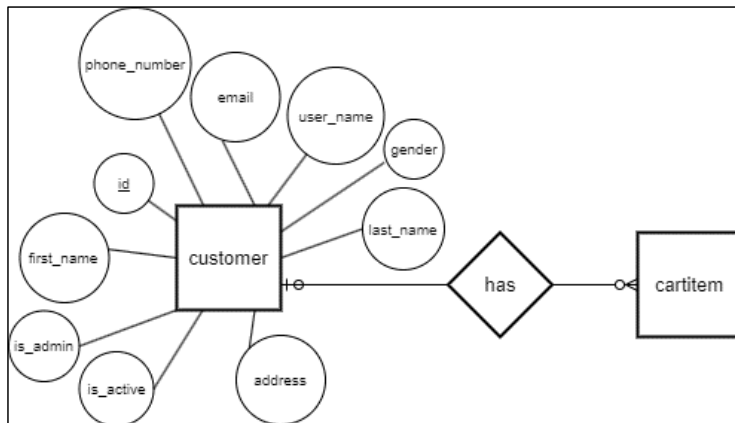


Sơ đồ 3.1 Sơ đồ ER của chức năng quản lý sim.

- Quản lý sim theo nhà mạng (network), kiểu số đẹp (tag).
- Quản lý sim thông tin bao gồm: id, số điện thoại (phone_number), slug, mô tả (description), ảnh sim (image), ngày tạo (created_date), hiển thị (is_visible), khả dụng (is_available), giá (price), giảm (discount).
- Quản lý nhà mạng thông tin bao gồm: id, số điện thoại (phone_number), slug, mô tả (description), ảnh logo nhà mạng (image_logo), ảnh phôi sim (image_simcard), ngày tạo (created_date).

- Quản lý sim thông tin bao gồm: id, số điện thoại (phone_number), slug, mô tả (description), ảnh sim (image), ngày tạo (created_date), hiển thị (is_visible), khả dụng (is_available), giá (price), giảm (discount).

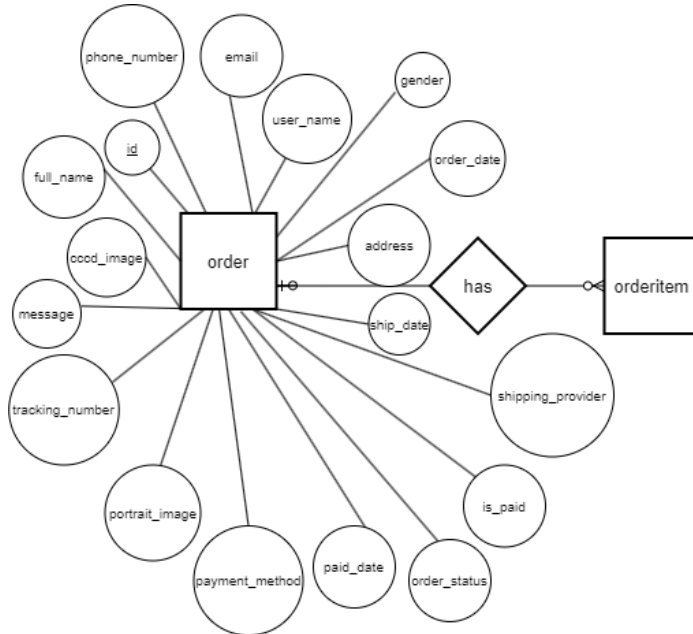
3.1.3 Quản lý khách hàng



Sơ đồ 3.2 Mô hình ER quản lý khách hàng

- Quản lý thông tin đăng ký của khách hàng: họ (last_name), tên (first_name), tên đăng nhập (username), email, số điện thoại (phone_number), giới tính (gender), địa chỉ (gender), hoạt động (is_active), là admin (is_admin).
- Mục giỏ hàng: là các sim mà khách hàng thêm.

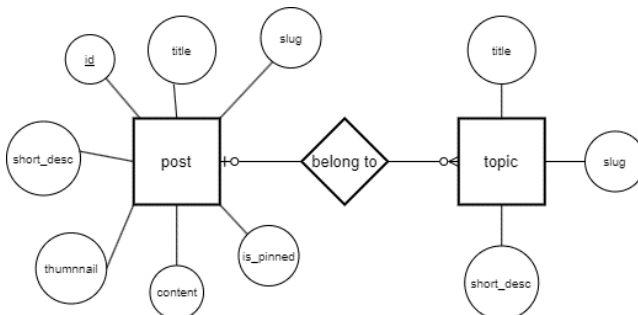
3.1.4 Quản lý đơn hàng



Sơ đồ 3.3 Mô hình ER quản lý đơn hàng

- Quản lý đơn hàng bao gồm: thông tin liên hệ, tình trạng đơn hàng, thông tin vận chuyển, thông tin thanh toán, thông tin đăng ký sim và danh sách sim mua trong đơn đó.

3.1.5 Quản lý bài viết



Sơ đồ 3.4 Mô hình ER quản lý bài viết

- Quản lý bài viết bao gồm: tiêu đề bài viết, mô tả, nội dung, hình ảnh, danh mục bài viết.

3.2 Xây dựng các chức năng của hệ thống

3.2.1 Mô tả các chức năng chi tiết

3.2.1.1 Website bán sim

- Quản lý giỏ hàng: Người dùng có thể thêm sim vào giỏ hàng.

STT	Chức năng	Diễn giải
1	Thêm mới	Cho phép người dùng thêm sim vào giỏ hàng
2	Xóa bỏ	Bỏ sim ra khỏi giỏ hàng

Bảng 3.1 Bảng mô tả chức năng quản lý giỏ hàng

- Quản lý đơn hàng: Người dùng từ giỏ hàng có thể đặt hàng.

STT	Chức năng	Diễn giải
1	Đặt hàng	Cho phép đặt hàng
2	Hủy đơn hàng	Cho phép hủy đơn hàng khi đang ở trạng thái <i>Chuẩn bị tiếp nhận</i>
3	Tra cứu	Xem chi tiết, theo dõi tình trạng đơn hàng đã đặt

Bảng 3.2 Bảng mô tả chức năng quản lý đơn hàng

- Quản lý thông tin cá nhân.

STT	Chức năng	Diễn giải
1	Thay đổi thông tin	Cho phép thay đổi thông tin cá nhân
2	Đổi mật khẩu	Cho phép người dùng cập nhật lại mật khẩu

Bảng 3.3 Bảng mô tả chức năng quản lý thông tin cá nhân

3.2.1.2 Quản lý sim

- Quản lý sim: Admin có thể thêm, sửa, xóa, ẩn, tìm kiếm sim

STT	Chức năng	Diễn giải
1	Thêm mới	Cho phép thêm mới sim vào phần quản lý
2	Cập nhật thông tin	Cho phép thay đổi thông tin của sim
3	Xóa	Cho phép xóa sim đó khỏi cơ sở dữ liệu
4	Tìm kiếm	Cho phép tìm kiếm đang có

Bảng 3.4 Bảng mô tả chức năng quản lý sim

- Quản lý nhà mạng: Cho phép admin thêm, sửa, xóa, tìm kiếm sim liên quan theo nhà mạng

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép thêm mới nhà mạng
2	Cập nhật thông tin	Cho phép cập nhật thông tin nhà mạng
3	Xóa	Cho phép xóa nhà mạng khỏi danh sách
4	Tìm kiếm	Cho phép tìm kiếm nhà mạng, và hiển thị sim thuộc nhà mạng.

Bảng 3.5 Bảng mô tả chức năng quản lý nhà mạng

- Quản lý phân loại sim số đẹp: Cho phép admin thêm, sửa, xóa, tìm kiếm sim liên quan theo phân loại

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép thêm phân loại mới
2	Cập nhật thông tin	Cho phép cập nhật thông tin phân loại mới
3	Xóa	Cho phép xóa phân loại mới
4	Tìm kiếm	Cho phép tìm kiếm sim thuộc phân loại đó.

Bảng 3.6 Bảng mô tả chức năng quản lý phân loại

3.2.1.3 Quản lý khách hàng

- Quản lý khách hàng: Admin có thể quản lý khách hàng đăng ký trên website

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới khách hàng
2	Cập nhật thông tin	Cho phép cập nhật thông tin khách hàng
3	Xóa	Cho phép xóa, hoặc khóa tài khoản không cho phép đăng nhập
4	Gửi lại mail xác minh	Cho phép gửi lại mail kích hoạt tài khoản khi link cũ hết phiên

Bảng 3.7 Bảng mô tả chức năng quản lý khách hàng

3.2.1.4 Quản lý đơn hàng

- Quản lý đơn hàng: Sau khi khách hàng thêm sim muốn mua vào giỏ hàng, người dùng có thể tiến hành đặt hàng, khi đặt hàng, hệ thống sẽ tự động lấy thông tin tài khoản làm thông tin liên hệ, khách hàng điền đầy đủ thông tin bắt buộc rồi tiến hành đặt hàng.

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép đặt hàng, hoặc tạo đơn hàng mới đối với admin
2	Cập nhật thông tin	Cho phép admin thay đổi thông tin liên quan đơn hàng
3	Xóa	Cho phép admin xóa đơn đặt
4	Tra cứu	Cho phép khách hàng tra cứu đơn hàng đã đặt và admin có thể tra cứu tất cả đơn đã đặt trên web để xử lý
5	Hủy đơn	Cho phép khách hàng hủy đơn hàng đã đặt nếu đang ở trạng thái Chuẩn bị tiếp nhận

Bảng 3.8 Bảng mô tả chức năng quản lý đơn hàng

3.2.1.5 Quản lý bài viết

- Quản lý danh mục bài viết

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới danh mục bài viết
2	Cập nhật thông tin	Cập nhật thông tin danh mục bài viết
3	Xóa	Cho phép xóa danh mục bài viết

Bảng 3.9 Bảng mô tả chức năng quản lý danh mục bài viết

- Quản lý bài viết: Cho phép quản lý bài viết.

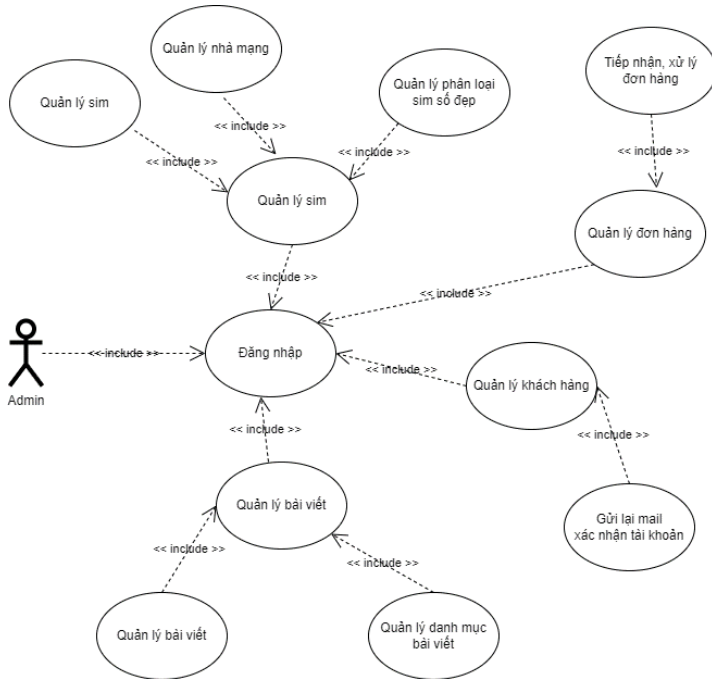
STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới bài viết
2	Cập nhật thông tin	Cho phép chỉnh sửa thông tin bài viết
3	Xóa	Cho phép xóa bài viết

Bảng 3.10 Bảng mô tả chức năng quản lý bài viết

3.2.2 Xác định use case

3.2.2.1 Admin

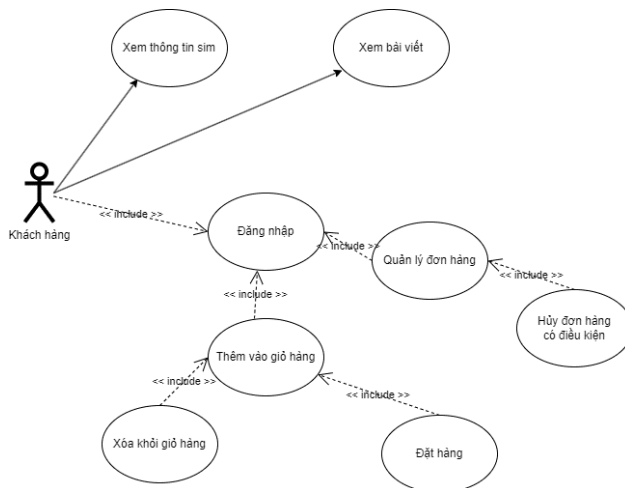
- Là người quản trị toàn bộ hệ thống
- Đăng nhập hệ thống
- Quản lý nhà mạng, phân loại sim số đẹp, sim
- Quản lý khách hàng
- Quản lý đơn hàng
- Quản lý bài viết và danh mục bài viết



Sơ đồ 3.5 Sơ đồ use case của đối tượng Admin

3.2.2.2 Khách hàng

- Khách hàng có thể truy cập website để xem thông tin công ty, địa chỉ liên hệ, danh sách sản phẩm, tìm kiếm sản phẩm, đặt hàng.



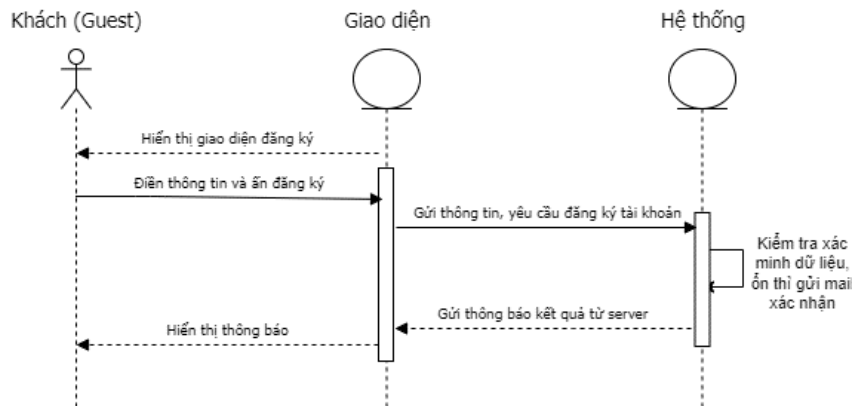
Sơ đồ 3.6. Sơ đồ use case của đối tượng khách hàng (Customer)

Commented [PN2]: Cho hình tăng kích thước lên để đọc dễ hơn
Tương tự cho các hình

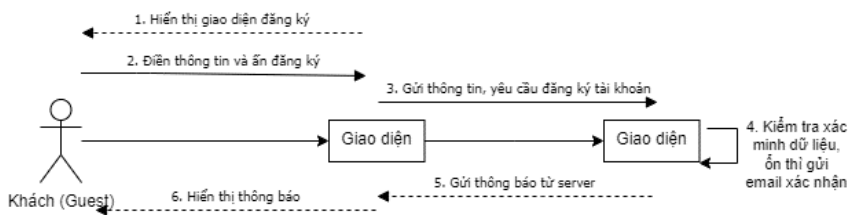
3.2.3 Đặc tả các use case

3.2.3.1 Mô tả quá trình đăng ký

Đặc tả: Người dùng khách (Guest) khi chưa có tài khoản có thể ấn đăng ký, để đăng ký tài khoản thì mới tiến hành thêm vào giỏ hàng để thanh toán được



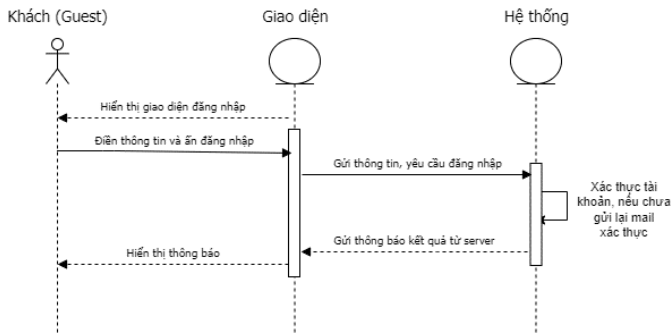
Sơ đồ 3.7. Sơ đồ tuần tự quá trình đăng ký



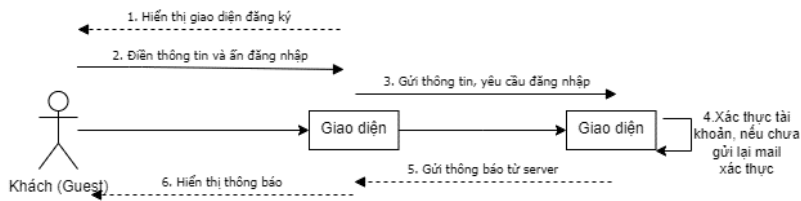
Sơ đồ 3.8 Sơ đồ hợp tác quá trình đăng nhập

3.2.3.2 Mô tả quá trình đăng nhập

Đặc tả: Người dùng khi đăng nhập đúng thông tin và tài khoản đã xác thực qua đường link gửi email sẽ đăng nhập thành công, còn nếu không tài khoản sẽ báo đăng nhập thất bại, nếu hệ thống phát hiện tài khoản chưa được kích hoạt, sẽ gửi lại thêm 1 mail nhắc nhở kèm đường link kích hoạt.



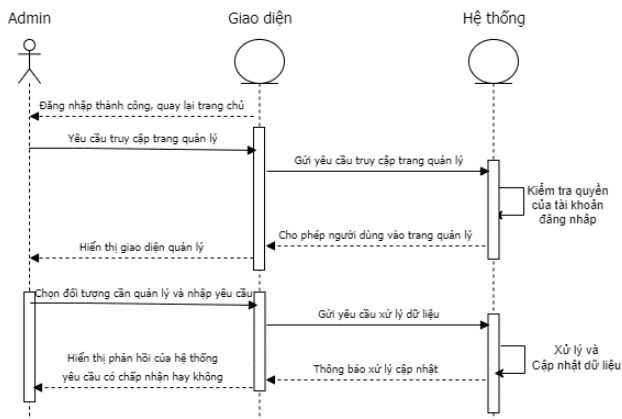
Sơ đồ 3.9 Sơ đồ tuần tự quá trình đăng nhập



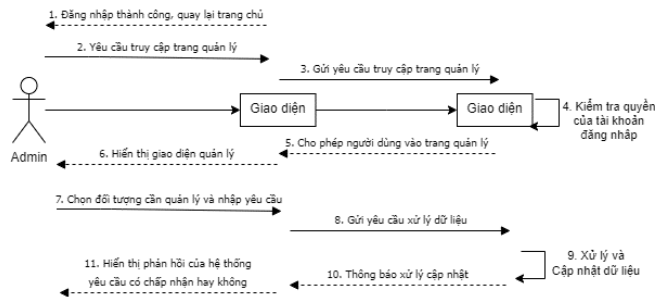
Sơ đồ 3.10 Sơ đồ hợp tác quá trình đăng nhập

3.2.3.3 Mô tả quá trình quản lý của admin

Đặc tả: Khi người dùng đăng nhập tài khoản có quyền admin có thể quản lý mọi đối tượng mà trang web có, ở đây là: sim, nhà mạng, phân loại nhà mạng, bài viết, danh mục bài viết, đơn hàng, tài khoản.



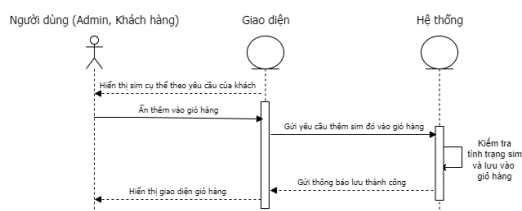
Sơ đồ 3.11 Sơ đồ tuần tự quá trình quản lý của admin với các dữ liệu mà trang web quản lý



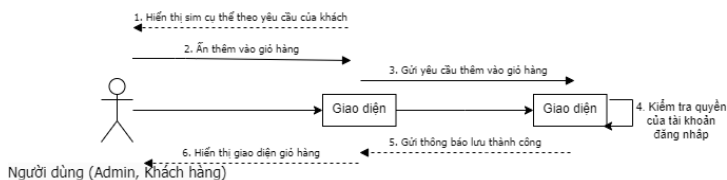
Sơ đồ 3.12 Sơ đồ hợp tác quá trình quản lý của admin với các dữ liệu mà trang web quản lý

3.2.3.4 Mô tả quá trình thêm vào giỏ hàng

Đặc tả: Khi người dùng có nhu cầu mua một sim, người dùng sẽ thêm sim vào giỏ hàng, để có thể tiến hành đặt hàng



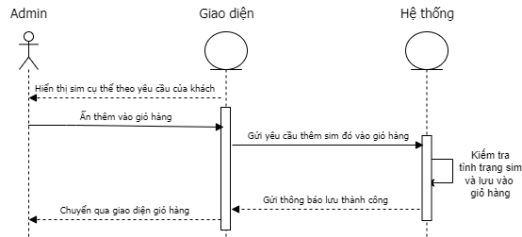
Sơ đồ 3.13 Sơ đồ tuần tự quá trình thêm vào giỏ hàng



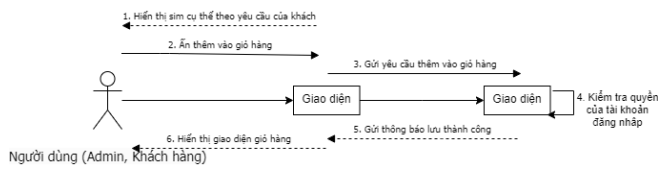
Sơ đồ 3.14 Sơ đồ hợp tác quá trình thêm vào giỏ hàng

3.2.3.5 Mô tả quá trình bỏ ra khỏi giỏ hàng

Đặc tả: Khi người dùng thay đổi quyết định, có thể bỏ sim ra khỏi giỏ hàng. Để thực hiện ta sẽ vào giỏ hàng và ấn nút bỏ ra, để sim đó ra khỏi giỏ hàng.



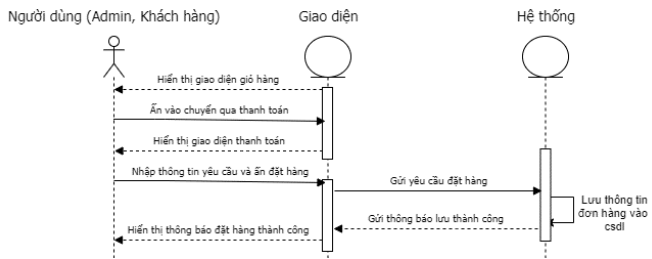
Sơ đồ 3.15 Sơ đồ tuần tự quá trình thêm vào giỏ hàng



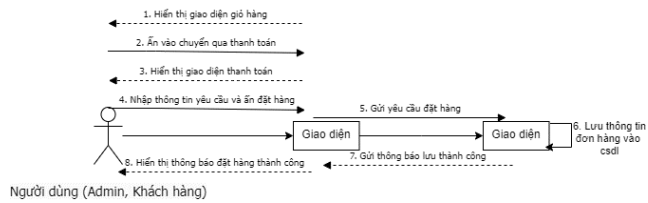
Sơ đồ 3.16 Sơ đồ hợp tác quá trình thêm vào giỏ hàng

3.2.3.6 Mô tả quá trình đặt hàng

Đặc tả: Sau khi thêm sim vào giỏ hàng, ta ấn đặt hàng để tiến hành đặt hàng, điền đầy đủ thông tin ta có thể đặt hàng (không yêu cầu ảnh thông tin đăng ký sim).



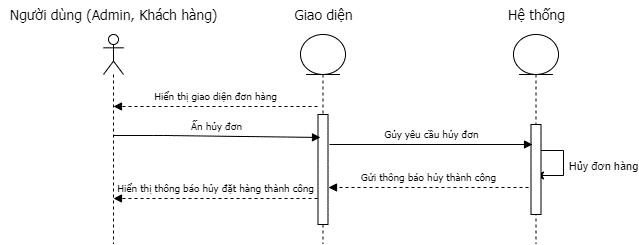
Sơ đồ 3.17 Sơ đồ tuần tự quá trình đặt hàng



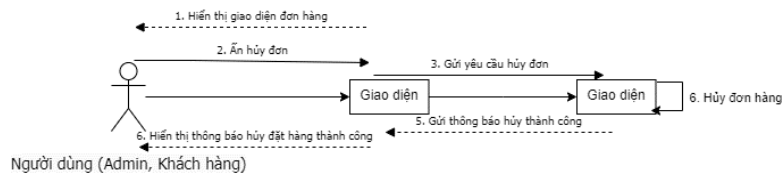
Sơ đồ 3.18 Sơ đồ tuần tự quá trình đặt hàng

3.2.3.7 Mô tả quá trình hủy hàng

Đặc tả: Sau khi đặt hàng, nếu như khách hàng có nhu cầu hủy và trạng thái đơn hàng đang là chuẩn bị tiếp nhận, khách hàng có thể vào đơn hàng đó và thao tác hủy.



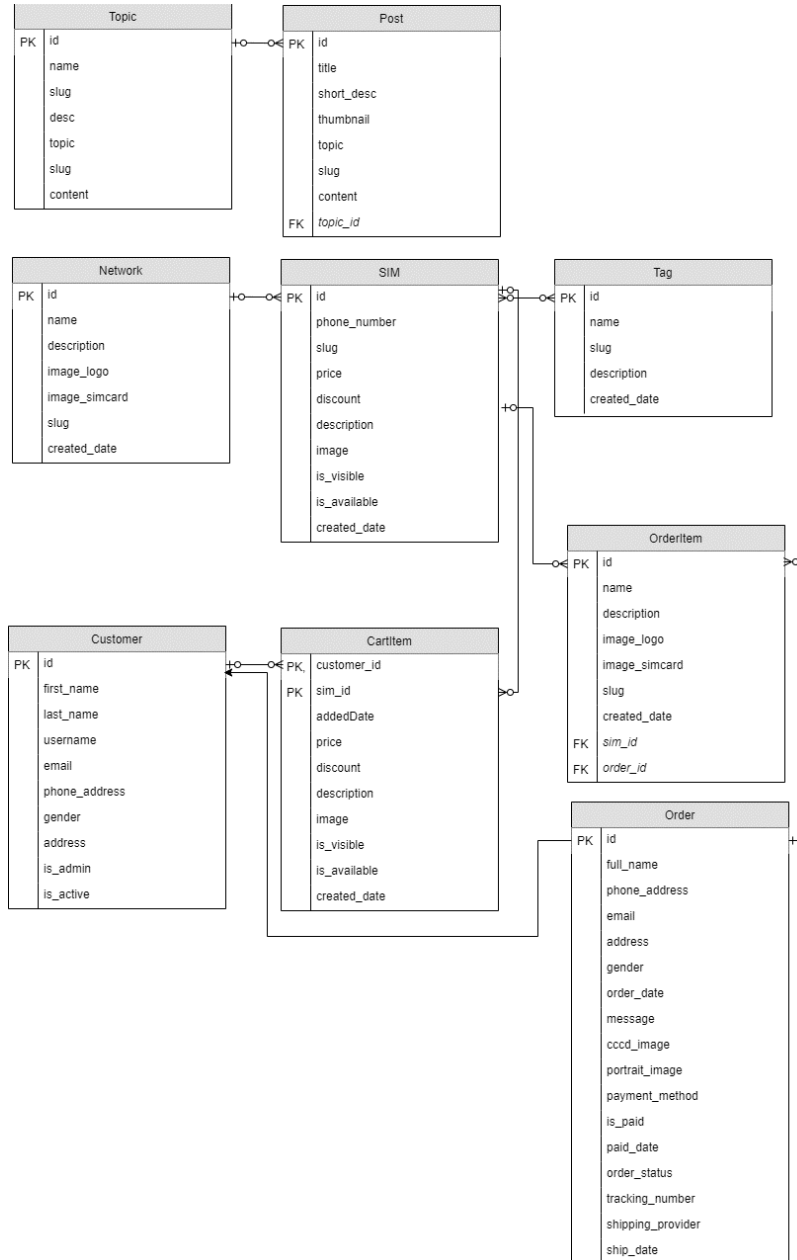
Sơ đồ 3.19 Sơ đồ tuần tự quá trình hủy đơn



Sơ đồ 3.20 Sơ đồ tuần tự quá trình hủy đơn

3.3 Thiết kế hệ thống

3.3.1 Sơ đồ cơ sở dữ liệu



Sơ đồ 3.21 Sơ đồ cơ sở dữ liệu

Commented [PN3]: Order không có liên kết với Customer???

3.3.2 Chi tiết cơ sở dữ liệu

Commented [PN4]: Đẩy lên trang trên, chỉ có chương mới ở trang mới

3.3.2.1 Quản lý sim và các thông tin liên quan

Đối tượng nhà mạng (Network)

Trường	Kiểu dữ liệu	Mô tả
name	CharField	Tên của nhà mạng.
description	TextField	Mô tả về nhà mạng (có thể để trống).
image_logo	ImageField	Hình ảnh logo của nhà mạng, được lưu trữ trong thư mục "photos/networks".
image_simcard	ImageField	Hình ảnh thẻ SIM của nhà mạng, được lưu trữ trong thư mục "photos/networks".
slug	SlugField	Đường dẫn tĩnh (slug) để định danh cho nhà mạng (unique).
created_date	DateTimeField	Ngày và giờ tạo bản ghi (tự động thêm khi lưu vào database).

Bảng 3.11 Bảng mô tả thuộc tính của đối tượng Network

Đối tượng phân loại sim số đẹp (Tag)

Trường	Kiểu dữ liệu	Mô tả
name	CharField	Tên của loại SIM.
slug	SlugField	Đường dẫn tĩnh (slug) để định danh cho loại SIM (unique).
description	TextField	Mô tả về loại SIM (có thể để trống).
created_date	DateTimeField	Ngày và giờ tạo bản ghi (tự động thêm khi lưu vào database).

Bảng 3.12 Bảng mô tả thuộc tính của đối tượng Tag (Phân loại kiểu sim số đẹp)

Đối tượng Sim

Trường	Kiểu dữ liệu	Mô tả
phone_number	CharField	Số điện thoại của SIM (unique).
slug	SlugField	Đường dẫn tĩnh (slug) để định danh cho SIM (unique).
price	DecimalField	Giá của SIM.
discount	PositiveIntegerField	Giá trị phần trăm giảm giá của SIM (giá trị từ 0 đến 100).
description	TextField	Mô tả về SIM (có thể để trống).
image	ImageField	Ảnh đại diện cho SIM (có thể để trống).
is_available	BooleanField	Trạng thái khả dụng của SIM (True hoặc False).
is_visible	BooleanField	Trạng thái hiển thị của SIM (True hoặc False).
created_date	DateTimeField	Ngày và giờ tạo bản ghi (tự động thêm khi lưu vào database).
network	ForeignKey	Khóa ngoại liên kết với một nhà mạng (ForeignKey tới model Network).
tags	ManyToManyField	Khóa ngoại liên kết với một hoặc nhiều tag (ManyToManyField tới model Tag).

Bảng 3.13 Bảng mô tả thuộc tính của đối tượng Sim

3.3.2.2 Quản lý khách hàng

Đối tượng Khách hàng (Customer)

Trường	Kiểu dữ liệu	Mô tả
first_name	CharField(max_length=30)	Tên của khách hàng
last_name	CharField(max_length=30)	Họ của khách hàng
gender	CharField(max_length=1)	Giới tính của khách hàng
address	CharField(max_length=200)	Địa chỉ của khách hàng
email	EmailField(unique=True)	Địa chỉ email của khách hàng

Trường	Kiểu dữ liệu	Mô tả
phone_number	CharField(max_length=30)	Số điện thoại của khách hàng
username	CharField(max_length=30, unique=True)	Tên đăng nhập của khách hàng
is_active	BooleanField(default=True)	Trạng thái hoạt động của tài khoản khách hàng
is_admin	BooleanField(default=False)	Trạng thái quản trị viên của tài khoản khách hàng
objects	CustomerAccountManager()	Manager để quản lý các tài khoản khách hàng
USERNAME_FIELD	CharField	Trường dữ liệu được sử dụng để xác định tài khoản người dùng
REQUIRED_FIELDS	List	Danh sách các trường bắt buộc được yêu cầu khi đăng ký tài khoản

Bảng 3.14 Bảng mô tả thuộc tính của đối tượng Khách hàng

Đối tượng sim trong giỏ hàng (CartItem)

Trường	Kiểu dữ liệu	Mô tả
Sim	ForeignKey(SIM)	Tham chiếu tới model SIM
customer	ForeignKey(Customer)	Tham chiếu tới model Customer
added_date	DateTimeField(auto_now_add=True, editable=False)	Ngày thêm vào giỏ hàng

Bảng 3.15 Bảng mô tả thuộc tính của đối tượng thành phần giỏ hàng

3.3.2.3 Quản lý đơn hàng

Đối tượng đơn hàng (Order)

Trường	Kiểu dữ liệu	Mô tả
full_name	CharField(100)	Họ và tên khách hàng
phone_number	CharField(20)	Số điện thoại khách hàng
email	EmailField()	Địa chỉ email của khách hàng
address	CharField(250)	Địa chỉ khách hàng
gender	CharField(1)	Giới tính khách hàng (Nam hoặc Nữ)
order_date	DateTimeField(auto_now_add=True)	Ngày giờ đặt hàng
message	CharField(500)	Yêu cầu đặt hàng của khách hàng
customer	ForeignKey(Customer)	Khóa ngoại đến bảng khách hàng
ccd_image	ImageField(upload_to='ccd_images/')	Ảnh CCCD đăng ký của khách hàng
portrait_image	ImageField(upload_to='portrait_images/')	Ảnh chân dung đăng ký của khách hàng
payment_method	CharField(10)	Phương thức thanh toán (COD hoặc TRANSFER)
is_paid	BooleanField()	Trạng thái thanh toán (Đã thanh toán hay chưa)
paid_date	DateTimeField(blank=True, null=True)	Ngày giờ thanh toán
order_status	CharField(20)	Trạng thái của đơn hàng
tracking_number	CharField(100)	Mã đơn vận khi vận chuyển đơn hàng
shipping_provider	CharField(100)	Nhà vận chuyển đơn hàng
ship_date	DateTimeField(blank=True, null=True)	Ngày giờ giao hàng

Bảng 3.16 Bảng mô tả thuộc tính của đối tượng đơn hàng

Commented [PN5]: Đây lên trang trên, chỉ có chương mới ở trang mới

3.3.2.4 Quản lý bài viết

Bảng cho model **Topic**

Trường	Kiểu dữ liệu	Mô tả
id	Integer	Khóa chính của bảng
name	CharField	Tên chủ đề
slug	SlugField	Đường dẫn đẹp
desc	TextField	Mô tả về chủ đề

Bảng 3.17 Bảng mô tả thuộc tính của đối tượng chủ đề bài viết

Bảng cho model **Post**

Trường	Kiểu dữ liệu	Mô tả
id	Integer	Khóa chính của bảng
title	CharField	Tiêu đề bài viết
short_desc	CharField	Mô tả ngắn gọn của bài viết
thumbnail	ImageField	Hình ảnh minh họa
topic	ForeignKey	Khóa ngoại đến chủ đề
slug	SlugField	Đường dẫn đẹp
content	RichTextField	Nội dung bài viết
is_pinned	BooleanField	Đánh dấu bài viết đã ghim hay chưa

Bảng 3.18 Bảng mô tả thuộc tính của đối tượng bài viết

CHƯƠNG 4 XÂY DỰNG WEBSITE

4.1 Giao diện và chức năng người dùng.

4.1.1 Trang chủ

Khi khách hàng truy cập trang web, hiển thị đầu tiên là trang chủ.

Giao diện trang chủ chia làm các khối riêng biệt.



Hình 4.1 Giao diện phần trên của trang chủ

Đây là thanh ngang trên cùng khi không có ai đăng nhập



Hình 4.2 Thanh header khi không có người đăng nhập

Đây là thanh tiêu đề khi có người đăng nhập



Hình 4.3 Thanh header khi có người đăng nhập

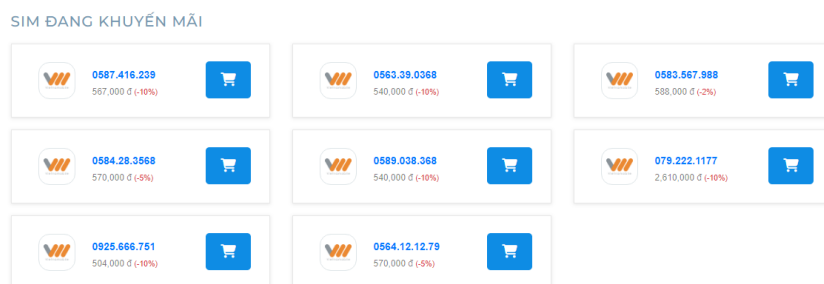
Khối đầu tiên là logo phần trên của trang web hiển thị logo, thanh tìm kiếm, và tương tác của người dùng.

Sau đó là đến thanh điều hướng, nơi đề xuất những nội dung mà khách hàng có thể muốn xem.

Từ đó người dùng, có thể tìm sim theo nhà mạng, hay kiểu sim đẹp

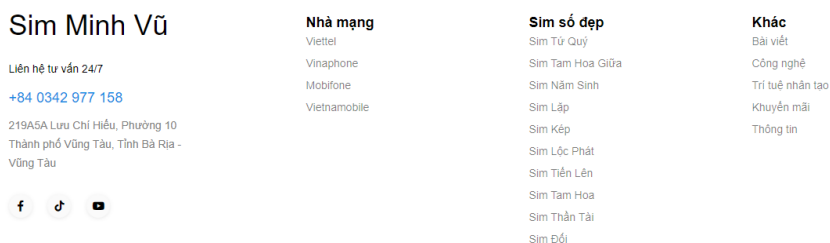
Hay đọc 1 bài viết có trên trang web

Ở ngay phía dưới trang web là 1 số sim đang khuyến mãi được hiển thị theo dạng gridview



Hình 4.4 Danh sách sim đang khuyến mãi hiển thị ở trang chủ

Ở ngay phía dưới cùng của trang web sẽ là thông tin liên hệ của website, và những lối tắt giúp người dùng thao tác nhanh chóng mà không cần phải kéo lên trên lại



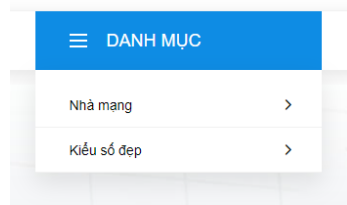
Hình 4.5 Phần footer của trang web

Đó là về phần trang chủ

4.1.2 Trang cửa hàng.

Sẽ chia ra làm 2 thành phần chính

Bố cục phía trên vẫn sẽ là thanh header và phía dưới là footer giống như trang chủ vậy, chỉ có 1 điều duy nhất khác là ở trang chủ phần danh mục luôn sỏ ra để người dùng có thể có những lựa chọn đầu tiên ngay từ khi truy cập vào trang web



Hình 4.6 Phần danh mục tìm kiếm sim theo 2 cách (nhà mạng hoặc kiểu số đẹp)

Còn ở trang cửa hàng, ta sẽ thay đổi lại bố cục ở phần chính chia ra làm 2 bên. Một phần bên trái bao gồm bộ lọc giúp người dùng tìm sim theo yêu cầu tìm.

Kiểu sim đẹp

- Sim Tứ Quý (8)
- Sim Tam Hoa Giữa (7)
- Sim Năm Sinh (7)
- Sim Lặp (15)
- Sim Kép (24)
- Sim Lộc Phát (7)
- Sim Tiến Lên (6)
- Sim Tam Hoa (2)
- Sim Thần Tài (14)
- Sim Đối (4)

Nhà mạng

- Viettel (21)
- Vinaphone (20)
- Mobifone (20)
- Vietnamobile (22)

Hình 4.7 Phần filter phía trên để tìm kiếm sim

Kiểu sim đẹp

Nhà mạng

Đầu số

Tránh số

Cuối số

Sinh Nhật

Khoảng giá

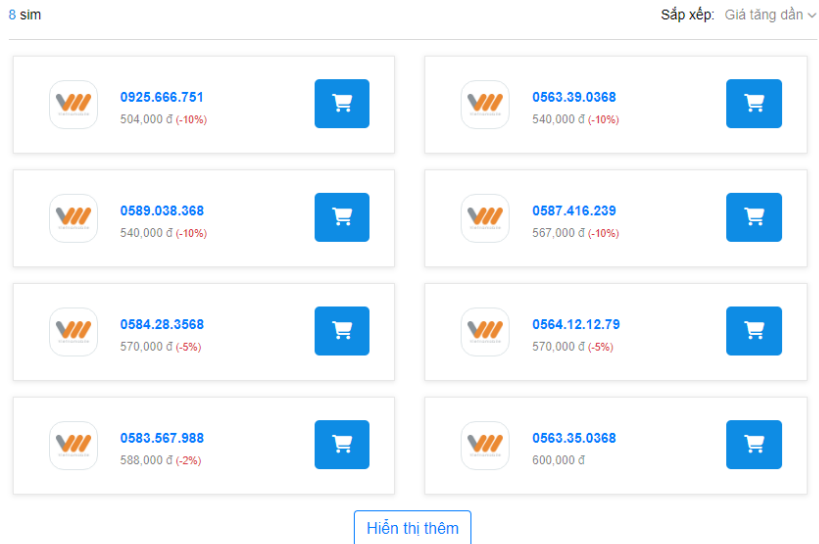
Từ: 100.000 ₫ - 299.100.000 ₫

Tim SIM

Hình 4.8 Phần filter phía dưới để tìm kiếm sim

Ở đây để cho trang web giảm tải việc phải load 1 lần hết tất cả sim, ta sẽ dùng ajax để khi nào cần hiển thị hoặc tìm kiếm theo bộ lọc, nó sẽ giúp ta đẩy yêu cầu và lấy dữ liệu về một cách nhanh chóng.

Ở bên phải là phần nội dung hiển thị danh sách sim có trong cửa hàng, danh sách này chỉ hiển thị dựa theo những sim được cho phép hiển thị (`is_visible = True`)

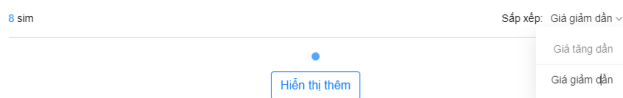


Hình 4.9 Danh sách sim hiển thị theo dạng gridview

Quan sát thật kỹ thật ra ở đây phần nội dung được chia làm 3, 1 phần là thanh ngang cho phép ta sắp xếp sim theo thứ tự tăng dần hoặc giảm dần, đồng thời cho ta biết được có bao nhiêu sim đang được hiển thị trên màn hình.

Đây là hình ảnh khi ta thao tác ấn sắp xếp tăng dần.

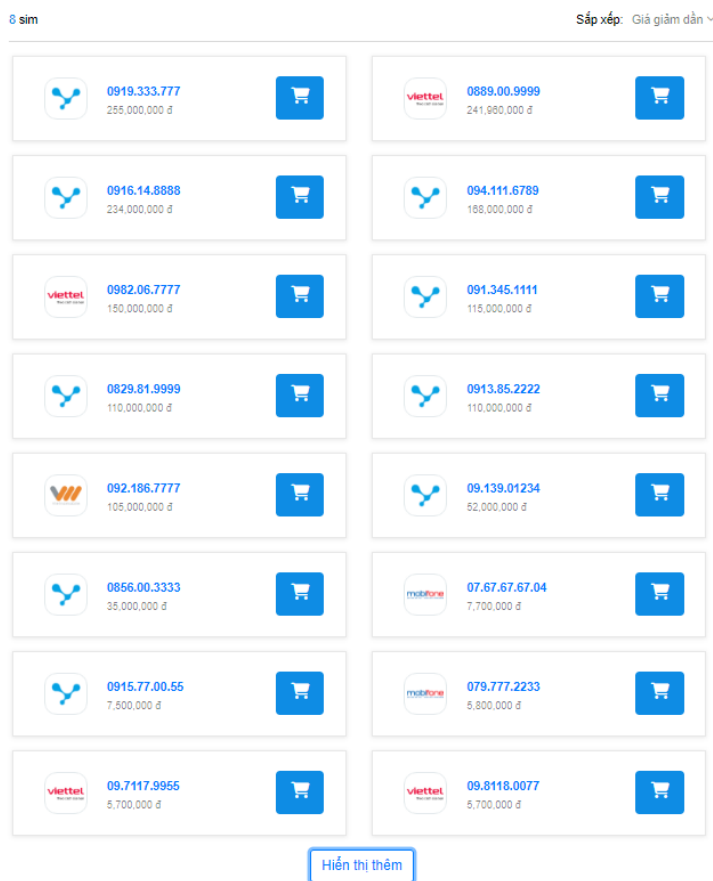
Đầu tiên nó sẽ có một hiệu ứng hình tròn nổi



Hình 4.10 Trang web khi đang load thêm danh sách sim

Nó chỉ dừng lại khi mà trang web đã lấy được dữ liệu cần thiết về bằng ajax

Sau khi lấy dữ liệu thành công, nó sẽ bắt đầu tiến hành hiển thị dữ liệu mới.



Hình 4.11 Sau khi load thành công thêm sim

Nếu như không còn gì để hiển thị thêm, trang web sẽ tự động ngừng load thêm cho dù có ấn nút hiển thị thêm bao nhiêu lần đi nữa, đảm bảo dữ liệu load sẽ không trùng lặp

Và giờ sau khi đã đến đây rồi, người dùng có thể tích điền vào cột bên trái để có thể tìm sim theo đúng nhu cầu.

Ở đây sẽ tra thử tìm sim mạng Viettel có số đuôi là 77

Kiểu sim đẹp

- Sim Tứ Quý (8)
- Sim Tam Hoa Giữa (7)
- Sim Năm Sinh (7)
- Sim Lập (15)
- Sim Kép (24)
- Sim Lộc Phát (7)
- Sim Tiến Lên (6)
- Sim Tam Hoa (2)
- Sim Thần Tài (14)
- Sim Đồi (4)

Nhà mạng

- Viettel (21)
- Vinaphone (20)
- Mobifone (20)
- Vietnamobile (22)

Đầu số

Tránh số

Cuối số

Sinh Nhật

Khoảng giá

Tìm SIM

0 sim
Sắp xếp: Giá giảm dần ▾

0982.06.7777
150,000,000 đ

09.8118.0077
5,700,000 đ

096.123.2277
5,700,000 đ

Hiện thị thêm

Hình 4.12 Giao diện tổng thể cá trang của hàng

4.1.3 Trang chi tiết sản phẩm

Trang chi tiết sản phẩm ở phần chính sẽ hiển thị như sau.

Kiểu sim đẹp

Sim Tứ Quý (8)
Sim Tam Hoa Giữa (7)
Sim Năm Sinh (7)
Sim Lập (15)
Sim Kép (24)
Sim Lộc Phát (7)
Sim Tiến Lên (6)
Sim Tam Hoa (2)
Sim Thần Tài (14)
Sim Đối (4)

Nhà mạng

Viettel (21)
Vinaphone (20)
Mobifone (20)
Vietnamobile (22)



Thuộc tính	Thông số
Số thuê bao:	0982.06.7777
Nhà mạng:	Viettel
Kiểu số đẹp:	Sim Tứ Quý
Giá bán:	150.000.000 đ

[Thêm vào giỏ hàng](#)

Một số SIM khác cùng nhà mạng

09771.15.0202 1.600.000 đ	0961.83.3030 1.700.000 đ
0973.07.07.01 2.200.000 đ	0961.77.3131 2.400.000 đ
0971.28.3030 2.900.000 đ	0981.61.3030 2.900.000 đ
0961.80.5050 3.400.000 đ	09.6116.8484 3.900.000 đ

[Hiện thị thêm](#)

Hình 4.13 Trang chi tiết sim

Phần bố cục ở trang này sẽ có phần tương đồng với trang cửa hàng, chỉ khác ở 1 vài điểm sau.

Ở bên trái sẽ là phần lối tắt dẫn đến những cách lựa sim phổ biến phân ra theo nhà mạng hoặc kiểu số đẹp.

Còn phần bên phải sẽ chia ra làm 2 phần trên dưới, phần phía dưới nhìn sơ sẽ biết đó là hiển thị một số sim khác đang khả dụng cùng nhà mạng, ta có thể xem thêm bằng cách ấn hiển thị thêm, hoạt động giống phần trang cửa hàng.

Còn phần phía trên sẽ chứa thông tin chiếc sim ấy.

Phần bố cục ấy lại được chia làm 2 bên: 1 bên là ảnh của chiếc sim card, nếu như trong cơ sở dữ liệu không có ảnh của chính chiếc sim đó, ảnh sẽ được hiển thị thay là ảnh mẫu chung của những chiếc sim card cùng nhà mạng (trường hợp ảnh trên là do sim không ảnh riêng trong cơ sở dữ liệu), bên phải đó sẽ là những thông tin quan trọng ảnh hưởng nhiều đến quyết định mua hay không của khách hàng.

Trên những thông tin đó ta cũng có thể truy cập nhanh tra cứu sim cùng nhà mạng hoặc sim cùng kiểu số đẹp.

Nếu như người dùng đang đăng nhập và ấn vào Thêm vào giỏ hàng thì sim sẽ được thêm vào giỏ hàng của người đó.

Còn nếu chưa đăng nhập sẽ bị chuyển qua trang đăng nhập. Đăng nhập là bắt buộc khi muốn thực hiện hành động này.

Sau khi thêm xong người dùng sẽ được chuyển qua trang giỏ hàng.

Giỏ hàng

Số điện thoại	Nhà mạng	Giá gốc	Giảm	Thành tiền
0982.06.7777	Viettel	150.000.000 đ	0%	150.000.000 đ

Tổng tiền: 150,000,000 đ

Hình 4.14 Phần hiển thị giỏ hàng

4.1.4 Trang đăng ký

Dưới đây là giao diện trang đăng ký

Commented [PN6]: Đây nội dung lên trên
Chỉ có chương mới sang trang mới
Rà soát toàn bộ để không có khoảng trắng nhiều trong một trang



Đăng ký

Họ	Tên
<input type="text" value="Họ"/>	<input type="text" value="Tên"/>
Tên đăng nhập	
<input type="text" value="Tên đăng nhập"/>	
Email	
<input type="text" value="name@example.com"/>	
Số điện thoại	
<input type="text" value="Số điện thoại"/>	
Địa chỉ	
<input type="text" value="Địa chỉ"/>	
Giới tính	
<input type="text" value="Nam"/>	
Mật khẩu	
<input type="text" value="Mật khẩu"/>	
Xác nhận mật khẩu	
<input type="text" value="Xác nhận mật khẩu"/>	
<input type="button" value="ĐĂNG KÝ"/>	

Nếu đã có tài khoản

[Đăng nhập](#)

Nếu quên mật khẩu hãy ấn vào đây

[Quên mật khẩu?](#)

Hình 4.15 Giao diện phần đăng ký

Nếu nhập thiếu hoặc dữ liệu không phù hợp nó sẽ báo lỗi

Họ	Tên
<input type="text" value="Họ"/>	<input type="text" value="Duc"/>
Tên đăng nhập	
<input type="text" value="laikhoan"/>	
Email	
<input type="text" value="hotrongvinh1234@gmail.com"/>	
Email đã được đăng ký.	
Số điện thoại	
<input type="text" value="0942977158"/>	
Số điện thoại đã được đăng ký.	
Địa chỉ	
<input type="text" value="41/23 Nguyễn Bảo"/>	
Giới tính	
<input type="text" value="Nam"/>	
Mật khẩu	
<input type="text" value="Mật khẩu"/>	
Xác nhận mật khẩu	
<input type="text" value="Xác nhận mật khẩu"/>	
<input type="button" value="ĐĂNG KÝ"/>	

Hình 4.16 Giao diện của trang đăng ký khi bị lỗi

Sau khi ấn đăng ký thành công, nó sẽ hiển thị thông báo kèm email kích hoạt mật khẩu.



Mail xác nhận đã được gửi, vui lòng truy cập vào hòm và ấn đường link xác nhận để có thể đăng nhập.

Nếu vẫn không thấy tài khoản

[Ấn vào đây để gửi lại link xác nhận email](#)

Nếu đã có tài khoản

[Đăng nhập](#)

Nếu quên mật khẩu

[Quên mật khẩu?](#)

Hình 4.17 Giao diện thông báo đăng ký thành công

Email xác nhận tài khoản sẽ có thời hạn trong vòng 4 tiếng, nếu như không nhận được hoặc đường link hết hạn có thể ấn nút phía dưới ở giao diện thông báo trước đó



Hình 4.18 Mail xác nhận đăng ký được gửi về

Sau khi ấn đường link nó sẽ báo kích hoạt thành công và chuyển về trang chủ



Tài khoản đã được xác nhận thành công, vui lòng đăng nhập để thực hiện đặt hàng.

Sẽ tự động chuyển đến trang chủ trong

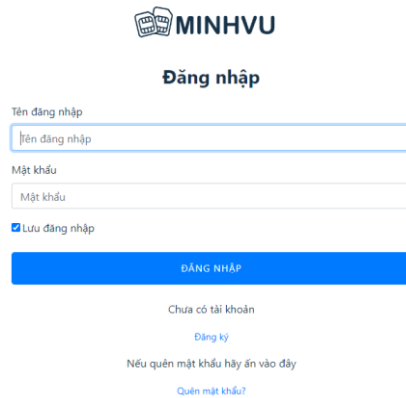
3

Hình 4.19 Giao diện trang web sau khi ấn đường link xác nhận

Ngoài tự động chuyển về nó sẽ tự đăng nhập luôn bằng tài khoản đó.

4.1.5 Trang đăng nhập

Dưới đây là giao diện trang đăng nhập



Tên đăng nhập

Mật khẩu

Lưu đăng nhập

ĐĂNG NHẬP

Chưa có tài khoản

[Đăng ký](#)

Nếu quên mật khẩu hãy ấn vào đây

[Quên mật khẩu?](#)

Hình 4.20 Giao diện trang đăng nhập

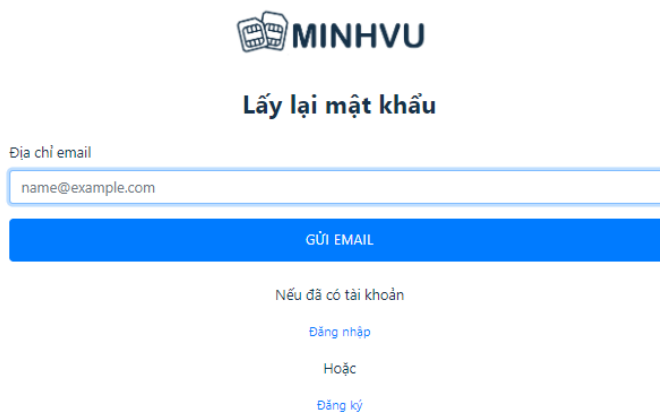
Để đăng nhập ta sử dụng tên tài khoản và mật khẩu đã đăng ký sử dụng trước đó.

Nếu như tài khoản chưa kích hoạt nó cũng sẽ tự động gửi về email một thư tương tự như trên để kích hoạt tài khoản.

Sau khi đăng nhập thành công, sẽ tự động chuyển về trang chủ.

4.1.6 Trang quên mật khẩu

Điền email vào trang này hệ thống sẽ gửi một đường link đặt lại mật khẩu mới qua email.



Địa chỉ email

name@example.com

GỬI EMAIL

Nếu đã có tài khoản

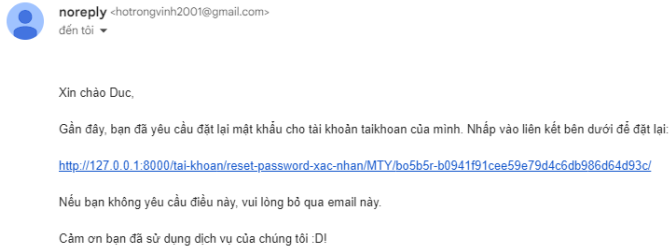
[Đăng nhập](#)

Hoặc

[Đăng ký](#)


Hình 4.21 Giao diện trang lấy lại mật khẩu

Email gửi về sẽ có nội dung như sau



Hình 4.22 Email yêu cầu lấy lại mật khẩu được gửi đến email đăng ký

Sau khi ấn vào đường link sẽ trả về một giao diện như sau:

 **MINHVU**

Cập nhật mật khẩu

Mật khẩu:

Xác nhận mật khẩu:

THAY ĐỔI

[Nếu đã có tài khoản](#)

[Đăng nhập](#)

[Hoặc](#)

[Đăng ký](#)

Hình 4.23 Giao diện điền mật khẩu mới sau khi ấn đường link

4.1.7 Trang thay đổi thông tin

Đây là giao diện trang thay đổi thông tin tài khoản cá nhân, thông tin được đăng ký với tài khoản sẽ được dùng để tự động làm thông tin liên hệ mặc định khi tiến hành đặt hàng, giao diện gần giống trang đăng ký, chi khác không có mật khẩu và mục đích sử dụng là để cập nhật thông tin.

MINHVU

Thay đổi thông tin

Họ: Tên:

Email:

Số điện thoại:

Địa chỉ:

Giới tính:

LƯU THAY ĐỔI

Hình 4.24 Giao diện trang thay đổi thông tin

4.1.8 Trang thay đổi mật khẩu

Người dùng có thể chủ động thay đổi mật khẩu của tài khoản

MINHVU

Thay đổi mật khẩu

Mật khẩu cũ:

Mật khẩu mới:

Xác nhận mật khẩu mới:

THAY ĐỔI MẬT KHẨU

Hình 4.25 Giao diện trang thay đổi mật khẩu

4.1.9 Trang giỏ hàng.

Sau khi đăng nhập và thêm vào giỏ hàng, người dùng có thể thêm hoặc bỏ ra khỏi giỏ hàng khi bấm vào nút tương ứng, ở đây khách hàng cũng có thể thấy được tổng số tiền phải trả là bao nhiêu.

Khi không có sản phẩm

Giỏ hàng trống

[Tiếp tục mua hàng](#)

Hình 4.26 Giao diện trang giỏ hàng khi trống

Khi giỏ hàng có sản phẩm ta có thể xem tổng giá hoặc bỏ sản phẩm đó ra khỏi giỏ

Giỏ hàng

Số điện thoại	Nhà mạng	Giá gốc	Giảm	Thành tiền
0925.666.751	Vietnamobile	560,000 đ	10%	504,000 đ Bỏ ra
				Tổng tiền: 504,000 đ

[Tiếp tục mua hàng](#)

[Chuyển qua đặt hàng](#)

Hình 4.27 Giao diện trang giỏ hàng khi có sản phẩm

Sau khi thêm mới một sim trong cửa hàng

Giỏ hàng

Số điện thoại	Nhà mạng	Giá gốc	Giảm	Thành tiền
0925.666.751	Vietnamobile	560,000 đ	10%	504,000 đ Bỏ ra
0563.39.0368	Vietnamobile	600,000 đ	10%	540,000 đ Bỏ ra
				Tổng tiền: 1,044,000 đ

[Tiếp tục mua hàng](#)

[Chuyển qua đặt hàng](#)

Hình 4.28 Giao diện trang giỏ hàng sau khi thêm 1 sim mới

4.1.10 Trang đặt hàng.

Sau khi ấn chuyển qua đặt hàng từ giỏ hàng, ta sẽ ra giao diện như dưới, ở giao diện này ta vẫn có thể bỏ sim ra khỏi giỏ hàng. Sau đó ta điền các thông tin cần và ấn đặt hàng.

Đặt hàng

Thông tin giỏ hàng

Số điện thoại	Nhà mạng	Giá gốc	Giảm	Thành tiền	
0925.666.751	Vietnamobile	560,000 đ	10%	504,000 đ	Bỏ ra
0563.39.0368	Vietnamobile	600,000 đ	10%	540,000 đ	Bỏ ra

Tổng tiền: 1,044,000 đ

Thông tin liên hệ

Họ và tên: Số điện thoại: Email:

Giới tính:

Địa chỉ:

Phương thức thanh toán:

Lời nhắn:

Thông tin đăng ký sim

Ảnh CCCD 2 mặt: Chọn ảnh CCCD đăng ký Ảnh chân dung: Chọn ảnh chân dung đăng ký

Hình 4.29 Giao diện trang đặt hàng

Giao diện sau khi đặt hàng thành công

Đặt hàng thành công
Bấm vào đây để theo dõi đơn hàng

Hình 4.30 Thông báo đặt hàng thành công

4.1.11 Trang quản lý đơn hàng

Đây là trang hiển thị thông tin chi tiết của đơn hàng

Thông tin đơn hàng ngày

13/05/2023 - 17:06:45

Tình trạng đơn hàng: Chuẩn bị tiếp nhận

Thông tin giỏ hàng

Số điện thoại	Nhà mạng	Giá gốc	Giảm	Thành tiền
0925.666.751	Vietnamobile	560,000 đ	10%	504,000 đ
0563.39.0368	Vietnamobile	600,000 đ	10%	540,000 đ

Thông tin liên hệ:Tổng tiền: **1,044,000 đ**

Ho Duc, Nam
0342977157, hotrongvinh123@gmail.com
41/23 Nguyễn Bảo

Phương thức thanh toán: Trả tiền khi nhận hàng

[Xem hóa đơn](#)

[Hủy đơn](#)

Hình 4.31 Giao diện trang quản lý đơn hàng

Ở trang quản lý hóa đơn người dùng sẽ xem được toàn bộ danh sách hóa đơn đã đặt.

4.1.12 Trang bài viết và danh sách bài viết

Đây là trang hiển thị nội dung bài viết

Thông tin cửa hàng

Công ty TNHH TM DV KT Minh Vũ là đơn vị chuyên cung cấp các dịch vụ bán sim chất lượng và uy tín tại Việt Nam. Với nhiều năm kinh nghiệm trong lĩnh vực kỹ thuật và đội ngũ nhân viên tay nghề cao, chúng tôi cam kết mang lại cho khách hàng những sản phẩm và dịch vụ tốt nhất.



Chúng tôi cung cấp dịch vụ bán sim chuyên nghiệp, đảm bảo cho khách hàng được sử dụng các loại sim ưng ý và đáp ứng nhu cầu sử dụng của mình. Bên cạnh đó, chúng tôi còn cung cấp các dịch vụ hỗ trợ khác như thay sim, đăng ký thông tin sim, kiểm tra sim và các dịch vụ liên quan đến sim khác.

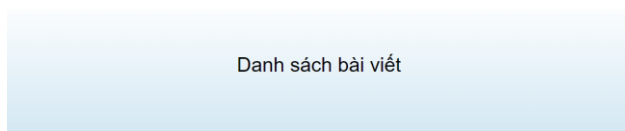
Nếu quý khách có nhu cầu sử dụng dịch vụ bán sim, vui lòng liên hệ với chúng tôi theo thông tin dưới đây:

Địa chỉ: 219A5A Lưu Chí Hiếu, Phường 10, Thành phố Vũng Tàu, Tỉnh Bà Rịa - Vũng Tàu, Việt Nam
Điện thoại: 025482281879

Chúng tôi rất mong được hợp tác và phục vụ quý khách!

Hình 4.32 Giao diện trang chi tiết bài viết

Đây là trang hiển thị danh sách bài viết



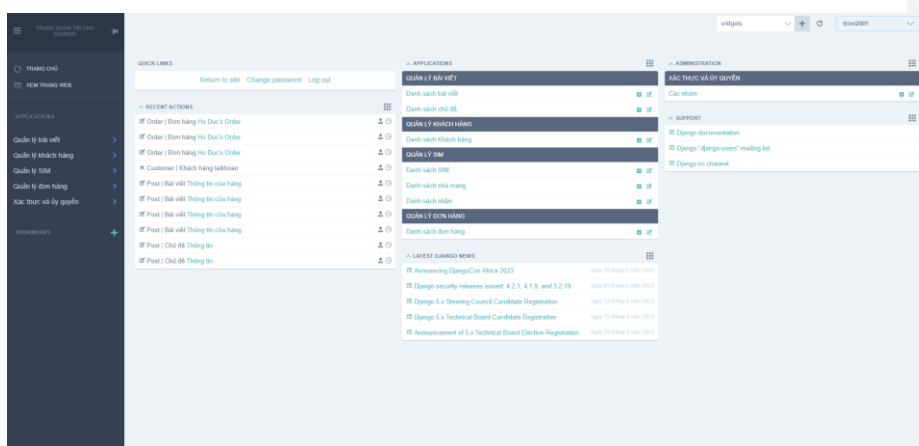
Hình 4.33 Hiển thị danh sách bài viết

4.1.13 Trang quản lý của admin

Trang quản lý của admin được xây dựng bằng Django Admin với thiết kế template của Django Jet.

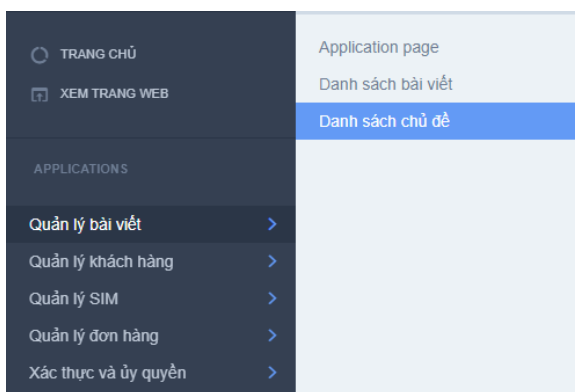
Tạo nên giao diện đẹp mắt và cuốn hút hơn.

Django Jet cung cấp bộ công cụ để xây dựng trang dashboard 1 cách dễ dàng bằng các phương thức và gói có sẵn



Hình 4.34 Giao diện trang quản lý chính của admin

Quản lý bài viết và chủ đề bài viết



Hình 4.35 Giao diện thanh điều hướng phần quản lý bài viết

Danh sách bài viết

The screenshot shows a web interface for managing articles. At the top, there are tabs for 'TIÊU ĐỀ', 'CHỦ ĐỀ', and 'CMM'. Below the tabs, there are input fields for 'Thống tin cửa hàng' and 'Thống tin'. The main form contains the following fields:

- Tiêu đề:** Input field containing 'Thống tin cửa hàng'.
- Mô tả ngắn:** Input field containing 'Thống tin cửa hàng'.
- Hình ảnh minh họa:** Shows a preview of 'images/minhvu.jpg' and a 'Choose File' button.
- Chủ đề:** Dropdown menu with 'Thống tin' selected.
- Đường dẫn đẹp:** Input field containing 'thong-tin-cua-hang'.
- Nội dung:** A rich text editor with a toolbar (Bold, Italic, Underline, etc.) and a preview of the article content. The content includes a paragraph about 'Công ty TNHH TM DV KT Minh Vũ' and an image of a building with a sign that says 'MINH VŨ'.

At the bottom of the form, there is a 'Ghim' checkbox and three buttons: 'LƯU LẠI', 'Lưu và thêm mới', and 'Lưu và tiếp tục chỉnh sửa'.

Hình 4.37 Giao diện thêm hoặc chỉnh sửa nội dung bài viết

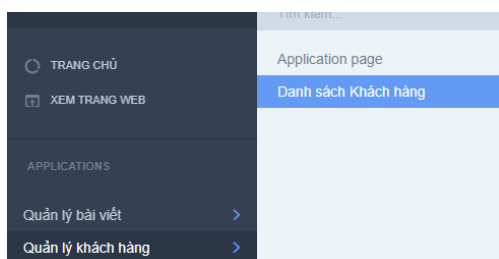
Danh sách chủ đề:

TÊN CHỦ ĐỀ	ĐƯỜNG DẪN ĐẸP	MÔ TẢ	SỐ BÀI VIẾT
<input type="checkbox"/> Thống tin	thong-tin	Thống tin trang	1
<input type="checkbox"/> Khuyến mãi	khuyen-mai	Khuyến mãi	0
<input type="checkbox"/> Trí tuệ nhân tạo	tri-tue-nhan-ao	Trí tuệ nhân tạo	0
<input type="checkbox"/> Công nghệ	cong-nghe	Công nghệ	0

Hình 4.38 Giao diện danh sách chủ đề bài viết

Hình 4.39 Giao diện trang thêm hoặc sửa chủ đề

Quản lý tài khoản: Admin có thể thêm, sửa, xóa, gửi lại email kích hoạt hay cấp quyền cho tài khoản được truy cập vào trang admin



Hình 4.40 Phần sidebar truy cập quản lý khách hàng

TÊN CẢNH NHẬP	EMAIL	SỐ ĐIỆN THOẠI	TÊN	HỌ	LẦN CUỐI ĐĂNG NHẬP	ĐANG HOẠT ĐỘNG	LA ADMIN
<input type="checkbox"/> talkhoan	hotronginh123@gmail.com	0342977157	Duc	Ho	16:13 Ngày 13 tháng 5 năm 2023	✓	✗
<input type="checkbox"/> talkhoan2	hotronginh12345@gmail.com	0342977159	Ngân	Tô	13:18 Ngày 12 tháng 5 năm 2023	✓	✗
<input type="checkbox"/> troi0001	hotronginh1234@gmail.com	0342977158	Vinh	Hồ	15:09 Ngày 13 tháng 5 năm 2023	✓	✓

Hình 4.41 Giao diện quản lý tài khoản

GENERAL GIỜ HÀNG

Mật khẩu: **thuật toán: pbkdf2_sha256 lặp lại: 390000 Mã salt: CDGpUj***** băm: SgUGf1*******
 Mật khẩu không được lưu trữ, vì vậy không có cách nào để xem mật khẩu của người dùng, nhưng bạn có thể dùng **biểu mẫu** này để thay đổi.

Tên:

Họ:

Giới tính:*

Địa chỉ:

Địa chỉ email:

Số điện thoại:

Tên đăng nhập:*

Đang hoạt động

Là Admin

Lần cuối đăng nhập: 16:13 Ngày 13 tháng 5 năm 2023

Hình 4.42 Giao diện thêm hoặc sửa thông tin tài khoản

Trang quản lý sim và danh mục sim.

APPLICATION MODELS	
Danh sách SIM	<input type="checkbox"/> <input type="checkbox"/>
Danh sách nhà mạng	<input type="checkbox"/> <input type="checkbox"/>
Danh sách nhãn	<input type="checkbox"/> <input type="checkbox"/>

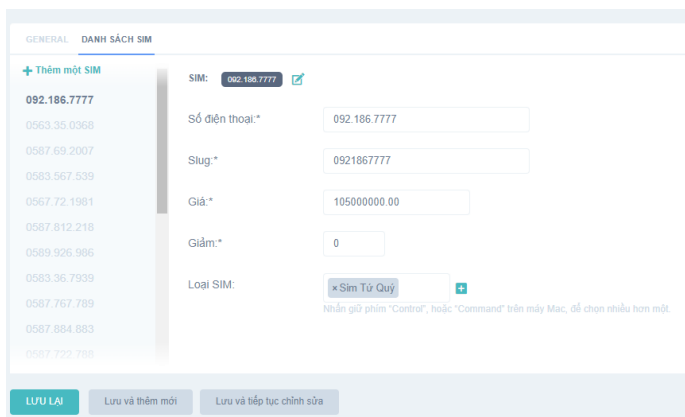
Hình 4.43 Giao diện chức năng quản lý sim

Dách sách sim

<input type="checkbox"/>	SỐ ĐIỆN THOẠI	GIÁ ĐÓC	GIÁ QUẢN	NHÀ MẠNG	LOẠI SIM	GIẢM	NGÀY TẠO	KHÓA THÈ	HẸN THÈ	<input type="button" value="Đi đến trang"/>
<input type="checkbox"/>	09771 15 0202	1,600,000 đ	1,600,000 đ	Viettel	Sim Năm Sinh	0%	18:45 Ngày 21 tháng 3 năm 2023	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Đi đến trang"/>
<input type="checkbox"/>	096 123 2277	5,700,000 đ	5,700,000 đ	Viettel	Sim Kép, Sim Tiền Lớn	0%	18:45 Ngày 21 tháng 3 năm 2023	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Đi đến trang"/>
<input type="checkbox"/>	097 123 1414	4,500,000 đ	4,500,000 đ	Viettel	Sim Lập, Sim Lộc Phát	0%	18:44 Ngày 21 tháng 3 năm 2023	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Đi đến trang"/>
<input type="checkbox"/>	0973 07 07 01	2,200,000 đ	2,200,000 đ	Viettel	Sim Năm Sinh	0%	18:44 Ngày 21 tháng 3 năm 2023	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Đi đến trang"/>
<input type="checkbox"/>	09 6116 4141	5,300,000 đ	5,300,000 đ	Viettel	Sim Lập, Sim Đổi	0%	18:41 Ngày 21 tháng 3 năm 2023	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Đi đến trang"/>
<input type="checkbox"/>	0961 22 4040	3,690,000 đ	3,690,000 đ	Viettel	Sim Lập	0%	18:40 Ngày 21 tháng 3 năm 2023	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Đi đến trang"/>
<input type="checkbox"/>	0961 17 7722	4,000,000 đ	4,000,000 đ	Viettel	Sim Kép	0%	18:40 Ngày 21 tháng 3 năm 2023	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Đi đến trang"/>

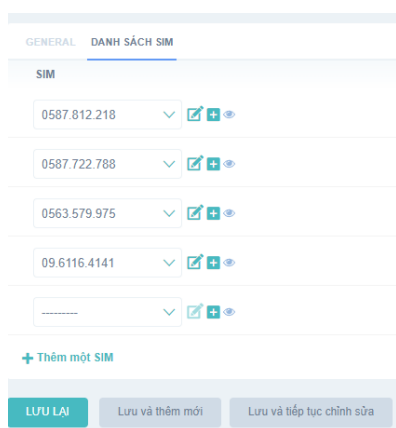
Hình 4.44 Giao diện danh sách sim

Quản lý sim theo nhà mạng



Hình 4.45 Giao diện danh sách sim theo nhà mạng

Quản lý sim theo kiểu số đẹp



Hình 4.46 Giao diện danh sách sim theo kiểu nhà mạng

Quản lý đơn hàng

<input type="checkbox"/>	HỌ VÀ TÊN	SỐ ĐIỆN THOẠI	EMAIL	ĐỊA CHỈ	KHÁCH HÀNG	THỜI GIAN ĐẶT	TRẠNG THÁI ĐƠN	NGƯỜI ĐẶT
<input type="checkbox"/>	Ho Duc	0342977157	hotronginh123@gmail.com	4123 Nguyễn Bào	talkhoan	17:29 Ngày 13 tháng 5 năm 2023	Chưa bị tiếp nhận	talkhoan
<input type="checkbox"/>	Ho Duc	0342977157	hotronginh123@gmail.com	4123 Nguyễn Bào	talkhoan	17:23 Ngày 13 tháng 5 năm 2023	Chưa bị tiếp nhận	talkhoan
<input type="checkbox"/>	Ho Duc	0342977157	hotronginh123@gmail.com	4123 Nguyễn Bào	talkhoan	17:16 Ngày 13 tháng 5 năm 2023	Chưa bị tiếp nhận	talkhoan
<input type="checkbox"/>	Ho Duc	0342977157	hotronginh123@gmail.com	4123 Nguyễn Bào	talkhoan	17:06 Ngày 13 tháng 5 năm 2023	Chưa bị tiếp nhận	talkhoan
<input type="checkbox"/>	Hà Vinh	0342977158	hotronginh123@gmail.com	4123 Nguyễn Bào	triv2001	09:41 Ngày 13 tháng 5 năm 2023	Chưa bị tiếp nhận	triv2001
<input type="checkbox"/>	Hà Vinh	0342977158	hotronginh123@gmail.com	4123 Nguyễn Bào	triv2001	09:28 Ngày 13 tháng 5 năm 2023	Chưa bị tiếp nhận	triv2001
<input type="checkbox"/>	Tô Ngân	0312345678	hotronginh12345@gmail.com	4123 Nguyễn Bào	talkhoan2	21:43 Ngày 11 tháng 5 năm 2023	Đã hủy	talkhoan2

Hình 4.47 Giao diện trang quản lý đơn hàng

Tra cứu, cập nhật thông tin đơn hàng

THÔNG TIN ĐƠN HÀNG THỐNG TIN ĐĂNG KÝ SIM THANH TOÁN TRẠNG THÁI ĐƠN HÀNG THÔNG TIN VẬN CHUYỂN DANH SÁCH SIM

Họ và tên:*




Số điện thoại:*

Email:*

Địa chỉ:*

Giới tính:

Yêu cầu khách hàng:

Khách hàng:*   

LƯU LẠI Lưu và thêm mới Lưu và tiếp tục chỉnh sửa

Hình 4.48 Giao diện thêm hoặc chỉnh sửa thông tin đơn hàng

CHƯƠNG 5 KẾT LUẬN

5.1 Kết quả đạt được

- Vận dụng kiến thức về ngôn ngữ Python vào dự án thực tế
- Thực hành và nghiên cứu sâu vào framework Django
- Kỹ năng phân tích yêu cầu thực tế để xây dựng website.
- Xây dựng website giới thiệu đẹp mắt, đầy đủ thông tin, dễ thao tác truy cập
- Xây dựng và chuẩn hóa dữ liệu đảm bảo tính đúng đắn, giúp dễ dàng quản lý dữ liệu.
- Sử dụng công cụ mà framework cung cấp để tạo nên giao diện quản lý dễ nhìn, rõ ràng để sử dụng.
- Cung cấp công cụ tìm kiếm hiệu quả, giúp khách hàng dễ dàng tìm kiếm sản phẩm mà họ mong muốn.
- Ứng dụng ajax để giảm tải khả năng phải gửi dữ liệu lượng lớn 1 lúc.

5.2 Hướng phát triển

- Kết nối với phần mềm xuất hóa đơn bên thứ 3.
- Tích hợp khả năng thanh toán, giúp tiết kiệm thời gian.
- Mở rộng cơ sở dữ liệu, quản lý chi tiết hơn nhằm đáp ứng nhu cầu của doanh nghiệp.
- Hỗ trợ nhập xuất với excel để hạn chế công việc lặp

Do giới hạn về thời gian thực hiện cùng với kiến thức và kinh nghiệm chưa đầy đủ, bài báo cáo của em có thể vẫn còn chứa đựng những sai sót. Em rất mong nhận được những đóng góp ý kiến quý báu từ các thầy cô trong Khoa Kỹ thuật - Công nghệ, nơi em đã được học hỏi và rèn luyện trong suốt chặng đường đại học.

CHƯƠNG 6 TÀI LIỆU THAM KHẢO

[1] "Django - The Web framework for perfectionists with deadlines" - Django, www.djangoproject.com.

[2] "Django Jet - Modern responsive template for Django" - GitHub, github.com/geex-arts/django-jet.

[3] "Django Migrations - A Primer" - Real Python, realpython.com/django-migrations-a-primer.

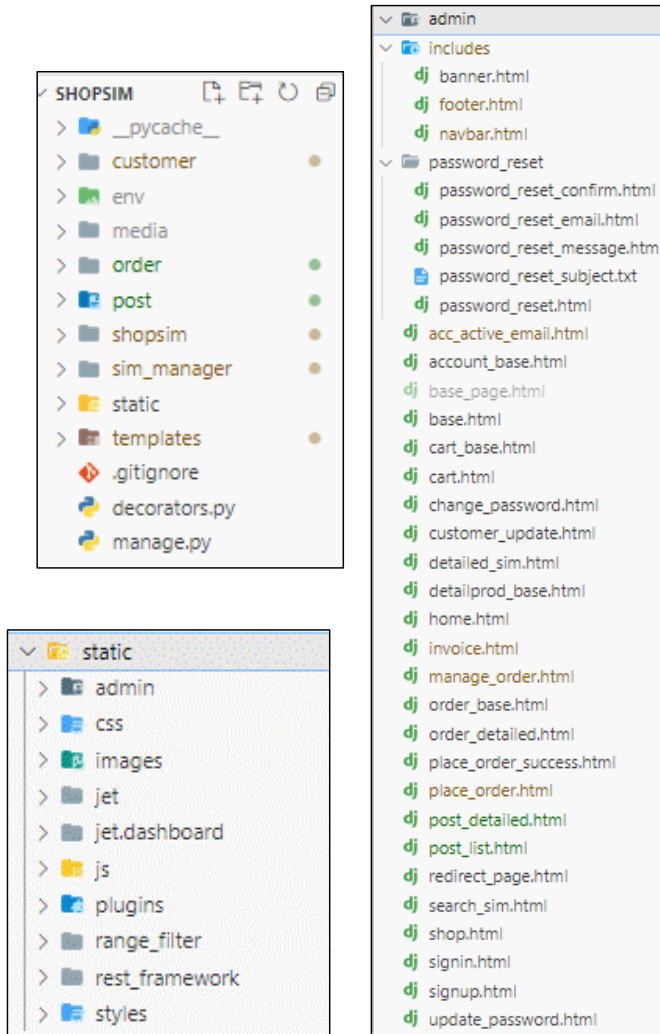
[4] "HTML to PDF Converter - Python & Django" - GitHub, github.com/nitesh7171/HTML-To-PDF-Converter-Python-Django.

[5] "MySQL :: MySQL Documentation" - MySQL, dev.mysql.com/doc/.

[6] "SMTP Server" - Wikipedia, en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol.

CHƯƠNG 7 PHỤ LỤC

7.1 Cấu trúc thư mục của project



Mô tả các thư mục và vai trò của chúng

- Thư mục "templates": Chứa các file template (mẫu giao diện) của ứng dụng Django, được sử dụng để hiển thị dữ liệu cho người dùng.

- Thư mục "static": Chứa các tài nguyên tĩnh của ứng dụng Django, bao gồm các file CSS, JavaScript, hình ảnh, font chữ, ... Tài nguyên này được sử dụng để tạo ra giao diện cho người dùng.
- Thư mục "media": Chứa các file đa phương tiện (hình ảnh, video, tài liệu, ...) được người dùng tải lên từ website. Thư mục này được sử dụng để lưu trữ các file đa phương tiện này.
- Thư mục "migrations": Chứa các file migration của ứng dụng Django. Migration được sử dụng để đồng bộ cấu trúc cơ sở dữ liệu với các model được định nghĩa trong ứng dụng Django.
- Thư mục "env": Chứa môi trường ảo (virtual environment) của ứng dụng Django, bao gồm các thư viện cần thiết để chạy ứng dụng Django.

Project được chia ra thành 4 app nhỏ khác nhau lần lượt là

- Sim_manager: Quản lý sim.
- Order: Đặt hàng và quản lý đơn đặt hàng.
- Customer: quản lý khách hàng và giỏ hàng.
- Post: Quản lý bài viết và chủ đề bài viết.

7.2 Code xử lý

7.2.1 Cấu hình trong file settings.py

Cấu hình khai báo các ứng dụng sẽ dùng trong project, đối tượng authentication, cấu hình SMTP Server, cấu hình đường dẫn static và media.

```

"""
Django settings for shopsim project.

Generated by 'django-admin startproject' using Django 4.1.7.

For more information on this file, see
https://docs.djangoproject.com/en/4.1/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/4.1/ref/settings/
"""

from pathlib import Path
import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.

```

```

BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-zdee08f#)$8_)j_hxg6s!y%k2yii*6^rf8oy-
p_w@=2118br^'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ['*']

# Application definition

INSTALLED_APPS = [
    'chartjs',
    'jet.dashboard',
    'jet',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'rest_framework',
    'sim_manager',
    'customer',
    'post',
    'order',
    'ckeditor',
    'django.contrib.humanize',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
    'django.middleware.locale.LocaleMiddleware',
]

ROOT_URLCONF = 'shopsim.urls'

TEMPLATES = [

```

```

    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': ['templates'],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'shopsim.wsgi.application'

AUTH_USER_MODEL = "customer.Customer"

# Database
# https://docs.djangoproject.com/en/4.1/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'shopsim',
        'USER': 'root',
        'PASSWORD': '',
        'OPTIONS': {
            'sql_mode': 'STRICT_TRANS_TABLES',
        },
    }
}

# Password validation
# https://docs.djangoproject.com/en/4.1/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',

```

```

    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/4.1/topics/i18n/

LANGUAGE_CODE = 'vi-VN'

LOCALE_PATHS = [
    os.path.join(BASE_DIR, 'locale'),
]

TIME_ZONE = 'Asia/Ho_Chi_Minh'

USE_I18N = True

USE_TZ = True

USE_L10N = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.1/howto/static-files/

STATIC_URL = '/static/'
STATIC_ROOT = BASE_DIR / 'static'
STATICFILES_DIRS = [
    'shopsim/static',
]
# Default primary key field type
# https://docs.djangoproject.com/en/4.1/ref/settings/#default-auto-field

MEDIA_URL = '/media/'
MEDIA_ROOT = BASE_DIR / 'media'

# Default primary key field type
# https://docs.djangoproject.com/en/4.1/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

#EMAIL CONFIG
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = 587
EMAIL_USE_TLS = True
EMAIL_HOST_USER = 'hotrongvinh2001@gmail.com'

```

```
EMAIL_HOST_PASSWORD = 'gtchqbbphcboifws'

DEFAULT_FROM_EMAIL = 'noreply<no_reply@domain.com>'

FILE_UPLOAD_MAX_MEMORY_SIZE = 52428800
```

7.2.2 Lớp định nghĩa các đối tượng mà trang quản lý

```
from django.db import models
from ckeditor.fields import RichTextField

class Topic(models.Model):
    name = models.CharField(max_length=100, verbose_name='Tên chủ đề')
    slug = models.SlugField(unique=True, verbose_name='Đường dẫn đẹp')
    desc = models.TextField(verbose_name='Mô tả', null=True, blank=True)

    def __str__(self):
        return self.name

    class Meta:
        verbose_name = 'Chủ đề'
        verbose_name_plural = 'Danh sách chủ đề'

class Post(models.Model):
    title = models.CharField(max_length=200, verbose_name='Tiêu đề')
    short_desc = models.CharField(max_length=500, verbose_name='Mô tả ngắn')
    thumbnail = models.ImageField(upload_to='images/', verbose_name='Hình ảnh minh họa')
    topic = models.ForeignKey(Topic, on_delete=models.CASCADE, related_name='posts', verbose_name='Chủ đề')
    slug = models.SlugField(unique=True, verbose_name='Đường dẫn đẹp')
    content = RichTextField(verbose_name='Nội dung')
    is_pinned = models.BooleanField(default=False, verbose_name='Ghim')

    def __str__(self):
        return self.title

    class Meta:
        verbose_name = 'Bài viết'
        verbose_name_plural = 'Danh sách bài viết'

from django.db import models
from django.utils.html import mark_safe
from django.core.validators import MaxValueValidator, MinValueValidator

class Tag(models.Model):
    name = models.CharField(max_length=50, unique=True, verbose_name='Loại SIM')
```

```

slug = models.SlugField(unique=True, verbose_name='Slug')
description = models.TextField(null=True, verbose_name='Mô tả',
blank=True)
created_date = models.DateTimeField(auto_now_add=True, verbose_name='Ngày
khởi tạo')

class Meta:
    verbose_name = 'Nhân'
    verbose_name_plural = "Danh sách nhân"

def sim_count(self):
    return self.sims.count()
sim_count.short_description = "Số lượng SIM"

def goto_tag(self): # new

    return mark_safe(f"""<a href="/sim-so-dep/{self.slug}" style="display:
inline-block; padding: 7px 15px; border: none; border-radius: 5px; color:
#fff; background-color: #47bac1; text-decoration: none; transition: all 0.3s
ease;"
                                onmouseover="this.style.backgroundColor='#639af5'"
                                onmouseout="this.style.backgroundColor='#47bac1'"
>
        Đi đến trang
    </a>""")
goto_tag.short_description = ""

def __str__(self):
    return self.name

class Network(models.Model):
    name = models.CharField(max_length=50, unique=True, verbose_name='Tên nhà
mạng')
description = models.TextField(null=True, verbose_name='Mô tả',
blank=True)
image_logo = models.ImageField(upload_to='photos/networks', null=True,
verbose_name='Logo nhà mạng:')
image_simcard = models.ImageField(upload_to='photos/networks', null=True,
verbose_name='Ảnh thẻ SIM:')
slug = models.SlugField(unique=True, verbose_name='Slug')
created_date = models.DateTimeField(auto_now_add=True, verbose_name='Ngày
khởi tạo')

def logo_preview(self): # new
    return mark_safe(f'<img src = "{self.image_logo.url}" width = "50"/>')
logo_preview.short_description = "Logo"

def simcard_img_preview(self): # new

```

```

        return mark_safe(f'<img src = "{self.image_simcard.url}" width =
"200"/>')
        simcard_img_preview.short_description = "Ảnh thẻ SIM"

    def sim_count(self):
        return self.sims.count()
    sim_count.short_description = "Số lượng SIM khả dụng"

    def goto_net(self): # new
        return mark_safe(f'""<a href="/nha-mang/{self.slug}" style="display:
inline-block; padding: 7px 15px; border: none; border-radius: 5px; color:
#fff; background-color: #47bac1; text-decoration: none; transition: all 0.3s
ease;"
                                onmouseover="this.style.backgroundColor='#639af5"'
                                onmouseout="this.style.backgroundColor='#47bac1'"
>
        Đi đến trang
        </a>""")
    goto_net.short_description = ""

    class Meta:
        verbose_name = 'Nhà mạng'
        verbose_name_plural = "Danh sách nhà mạng"

    def __str__(self):
        return self.name

class SIM(models.Model):
    phone_number = models.CharField(max_length=20, unique=True,
verbose_name='Số điện thoại')
    slug = models.SlugField(unique=True, verbose_name='Slug')
    price = models.DecimalField(max_digits=12, decimal_places=2,
verbose_name='Giá')
    discount = models.PositiveIntegerField(
        default=0,
        validators=[
            MaxValueValidator(100),
            MinValueValidator(0)
        ], verbose_name='Giảm')
    description = models.TextField(verbose_name='Mô tả', null=True,
blank=True)
    image = models.ImageField(upload_to='sim_images/', verbose_name='Ảnh SIM',
null=True, blank=True)

    is_available = models.BooleanField(default=True, verbose_name='Khả thi')
    is_visible = models.BooleanField(default=True, verbose_name='Hiển thị')
    created_date = models.DateTimeField(auto_now_add=True, verbose_name='Ngày
tạo')
```



```

network = models.ForeignKey(
    Network, on_delete=models.CASCADE, related_name='sims',
verbose_name='Nhà mạng')

tags = models.ManyToManyField(
    Tag, related_name='sims', verbose_name='Loại SIM', blank=True)

class Meta:
    verbose_name = 'SIM'
    verbose_name_plural = "Danh sách SIM"

def sim_img_preview(self): # new
    return mark_safe(f"<img src = \"{self.image.url}\" width = \"200\"/>")
sim_img_preview.short_description = "Ảnh thẻ SIM"

def goto_sim(self): # new
    return mark_safe(f"<\"\"<a href=\"/sim/{self.slug}\" style=\"display:
inline-block; padding: 7px 15px; border: none; border-radius: 5px; color:
#fff; background-color: #47bac1; text-decoration: none; transition: all 0.3s
ease;\"
                                onmouseover=\"this.style.backgroundColor='#639af5'\"
                                onmouseout=\"this.style.backgroundColor='#47bac1'\"
                                >
        Đi đến trang
    </a>\"\"\"")
goto_sim.short_description = ""

def __str__(self):
    return self.phone_number

def get_curr_price(self):
    new_price = self.price * (100 - self.discount) / 100
    return new_price
get_curr_price.short_description = "Giá hiện tại"

def get_salepricestr(self):
    new_price = self.price * (100 - self.discount) / 100
    return "{:, .0f}".format(new_price) + " đ"
get_salepricestr.short_description = 'Giá giảm'

def get_originalpricestr(self):
    return "{:, .0f}".format(self.price) + " đ"
get_originalpricestr.short_description = 'Giá gốc'

def get_tags(self):
    return ", ".join([t.name for t in self.tags.all()])
get_tags.short_description = 'Loại SIM'

def get_discount(self):
    return str(self.discount) + '%'

```

```

get_discount.short_description = 'Giảm'

from datetime import datetime
from django.db import models
from django.contrib.auth.models import AbstractBaseUser, BaseUserManager
from django.utils.html import mark_safe
from sim_manager.models import SIM

# Tạo manager để quản lý tài khoản khách hàng
class CustomerAccountManager(BaseUserManager):
    def create_user(self, email, username, first_name, last_name, gender,
address, phone_number, password=None):
        if not email:
            raise ValueError("Khách hàng phải có email.")
        if not username:
            raise ValueError("Khách hàng phải có tên đăng nhập.")
        if not first_name:
            raise ValueError("Khách hàng phải điền tên.")
        if not last_name:
            raise ValueError("Khách hàng phải điền họ.")
        if not gender:
            raise ValueError("Khách hàng phải điền giới tính.")
        if not address:
            raise ValueError("Khách hàng phải điền địa chỉ.")

        user = self.model(
            email=self.normalize_email(email),
            username=username,
            first_name = first_name,
            last_name = last_name,
            phone_number = phone_number,
            gender = gender,
            address = address,
        )

        user.set_password(password)
        user.save(using=self._db)

        return user

    def create_superuser(self, email, username, password, first_name,
last_name, gender, address, phone_number):
        if not email:
            raise ValueError("Khách hàng phải có email.")
        if not username:
            raise ValueError("Khách hàng phải có tên đăng nhập.")
        if not first_name:
            raise ValueError("Khách hàng phải điền tên.")
        if not last_name:

```

```

        raise ValueError("Khách hàng phải điền họ.")
    if not gender:
        raise ValueError("Khách hàng phải điền giới tính.")
    if not address:
        raise ValueError("Khách hàng phải điền địa chỉ.")

    user = self.create_user(
        email=self.normalize_email(email),
        username=username,
        first_name = first_name,
        last_name = last_name,
        phone_number = phone_number,
        gender = gender,
        address = address,
    )

    user.set_password(password)
    user.is_admin = True
    user.save(using=self._db)
    return user

# Model Khách hàng
class Customer(AbstractBaseUser):
    MALE = 'M'
    FEMALE = 'F'
    GENDER_CHOICES = (
        (MALE, 'Nam'),
        (FEMALE, 'Nữ'),
    )

    first_name = models.CharField(max_length=30, blank=True,
verbose_name='Tên')
    last_name = models.CharField(max_length=30, blank=True, verbose_name='Họ')
    gender = models.CharField(max_length=1, choices=GENDER_CHOICES,
blank=False, verbose_name='Giới tính')
    address = models.CharField(max_length=200, blank=True, verbose_name='Địa
chỉ')
    email = models.EmailField(unique=True, verbose_name='Địa chỉ email',
blank=True)
    phone_number = models.CharField(max_length=30, verbose_name='Số điện
thoại', unique=True, blank=True)
    username = models.CharField(max_length=30, unique=True, verbose_name='Tên
đăng nhập')
    is_active = models.BooleanField(default=True, verbose_name='Đang hoạt
động')
    is_admin = models.BooleanField(default=False, verbose_name='Là Admin')

    objects = CustomerAccountManager()

    USERNAME_FIELD = 'username'

```

```

REQUIRED_FIELDS = ['email', 'first_name', 'last_name', 'gender',
'address']

class Meta:
    verbose_name = 'Khách hàng'
    verbose_name_plural = "Danh sách Khách hàng"

def __str__(self):
    return self.username

def has_perm(self, perm, obj=None):
    return True

def has_module_perms(self, app_label):
    return True

@property
def is_staff(self):
    return self.is_admin

def resend_activate_email(self): # new
    return mark_safe(f"""<a href="/tai-khoan/gui-lai-email-xac-
nhan/{self.username}" style="display: inline-block; padding: 7px 15px; border:
none; border-radius: 5px; color: #fff; background-color: #47bac1; text-
decoration: none; transition: all 0.3s ease;"
                onmouseover="this.style.backgroundColor='#639af5'"
                onmouseout="this.style.backgroundColor='#47bac1'"
>
        Gửi lại mail kích hoạt
    </a>""")
    resend_activate_email.short_description = ""

class CartItem(models.Model):
    sim = models.ForeignKey(SIM, on_delete=models.CASCADE, verbose_name="Sim",
related_name="cart_items")
    customer = models.ForeignKey(Customer, on_delete=models.CASCADE,
verbose_name="Khách hàng", related_name="cart_items")
    added_date = models.DateTimeField(auto_now_add=True, editable=False)

def __str__(self):
    return self.sim.phone_number

class Meta:
    verbose_name = 'Sim'
    verbose_name_plural = "Giỏ hàng"
    unique_together = ['sim', 'customer']

def get_phone_number(self):
    return self.sim.phone_number
get_phone_number.short_description = "Số điện thoại"

```

```

def get_network_name(self):
    return self.sim.network.name
get_network_name.short_description = "Nhà mạng"

def get_original_price_str(self):
    return self.sim.get_originalpricestr()
get_original_price_str.short_description = "Giá gốc"

def get_sale_price_str(self):
    return self.sim.get_salepricestr()
get_sale_price_str.short_description = "Giá sau giảm"

def get_discount(self):
    return self.sim.get_discount()
get_discount.short_description = "Giảm"

def get_total_price(self):
    total_price = 0
    for item in CartItem.objects.filter(customer=self.customer):
        total_price += item.sim.get_curr_price()
    return "{:,.0f}".format(total_price) + " đ"

get_total_price.short_description = 'Tổng giá'

from django.db import models
from customer.models import Customer
from sim_manager.models import SIM

class Order(models.Model):
    # Thông tin đơn hàng
    full_name = models.CharField(max_length=100, blank=False, null=False,
verbose_name='Họ và tên')
    phone_number = models.CharField(max_length=20, blank=False, null=False,
verbose_name='Số điện thoại')
    email = models.EmailField(blank=False, null=False, verbose_name='Email')
    address = models.CharField(max_length=250, blank=False, null=False,
verbose_name='Địa chỉ')
    gender = models.CharField(max_length=1, choices=(( 'M', 'Nam'), ('F',
'Nữ')), blank=True, null=True, verbose_name='Giới tính')
    order_date = models.DateTimeField(auto_now_add=True, editable=False,
blank=False, null=False, verbose_name='Thời gian đặt')
    message = models.CharField(max_length=500, blank=True, null=True,
verbose_name="Yêu cầu khách hàng")

    #Tài khoản đặt
    customer = models.ForeignKey(Customer, on_delete=models.CASCADE,
verbose_name="Khách hàng", related_name="orders")

    #Thông tin đăng ký sim

```

```

    cccd_image = models.ImageField(upload_to='cccd_images/', blank=True,
null=True, verbose_name='Ảnh CCCD đăng ký')
    portrait_image = models.ImageField(upload_to='portrait_images/',
blank=True, null=True, verbose_name= 'Ảnh chân dung đăng ký')

    # Thông tin thanh toán
    PAYMENT_METHODS = (
        ('COD', 'Trả tiền khi nhận hàng'),
        ('TRANSFER', 'Chuyển khoản'),
    )

    payment_method = models.CharField(choices=PAYMENT_METHODS, max_length=10,
verbose_name="Phương thức thanh toán")
    is_paid = models.BooleanField(default=False, verbose_name='Trạng thái
thanh toán')
    paid_date = models.DateTimeField(blank=True, null=True, verbose_name='Ngày
thanh toán')

    # Trạng thái đơn hàng
    ORDER_STATUSES = (
        ('PENDING', 'Chuẩn bị tiếp nhận'),
        ('PROCESSING', 'Đang xử lý'),
        ('SHIPPING', 'Đang giao hàng'),
        ('DELIVERED', 'Đã giao hàng'),
        ('REFUNDED', 'Đã hoàn'),
        ('CANCELLED', 'Đã hủy'),
    )
    order_status = models.CharField(choices=ORDER_STATUSES, max_length=20,
verbose_name='Trạng thái đơn')

    # Thông tin vận chuyển
    tracking_number = models.CharField(max_length=100, blank=True, null=True,
verbose_name='Mã đơn vận')
    shipping_provider = models.CharField(max_length=100, blank=True,
null=True, verbose_name='Nhà vận chuyển')
    ship_date = models.DateTimeField(blank=True, null=True, verbose_name='Ngày
giao hàng')

    def __str__(self):
        return f"{self.full_name}'s Order"

    class Meta:
        verbose_name = 'Đơn hàng'
        verbose_name_plural = "Danh sách đơn hàng"

    def get_customer_username(self):
        return self.customer.username
    get_customer_username.short_description = "Người đặt"

```

```

def get_gender_display(self):
    if self.gender == 'M':
        return 'Nam'
    return 'Nữ'
get_gender_display.short_description = "Giới tính"

def get_order_status_display(self):
    return dict(Order.ORDER_STATUSES).get(self.order_status, '')
get_order_status_display.short_description = "Trạng thái đơn hàng"

def get_payment_method_display(self):
    return dict(Order.PAYMENT_METHODS).get(self.payment_method, '')
get_payment_method_display.short_description = "Phương thức thanh toán"

def get_paid_state_display(self):
    if self.is_paid:
        return "Đã thanh toán"
    return "Chưa thanh toán"
get_payment_method_display.short_description = "Trạng thái thanh toán"

def get_total_price(self):
    total_price = 0
    for item in OrderItem.objects.filter(order = self):
        total_price += item.sim.get_curr_price()
    return "{:,0f}".format(total_price) + " đ"
get_total_price.short_description = 'Tổng giá'

class OrderItem(models.Model):
    sim = models.ForeignKey(SIM, on_delete=models.CASCADE, verbose_name="Sim",
related_name="order_item")
    order = models.ForeignKey(Order, on_delete=models.CASCADE,
verbose_name="Sim", related_name="order_items")

    def __str__(self):
        return self.sim.phone_number

    class Meta:
        verbose_name = 'Sim'
        verbose_name_plural = "Danh sách SIM"
        unique_together = ['sim', 'order']

def get_phone_number(self):
    return self.sim.phone_number
get_phone_number.short_description = "Số điện thoại"

def get_network_name(self):
    return self.sim.network.name
get_network_name.short_description = "Nhà mạng"

def get_original_price_str(self):

```

```

        return self.sim.get_originalpricestr()
    get_original_price_str.short_description = "Giá gốc"

    def get_sale_price_str(self):
        return self.sim.get_salepricestr()
    get_sale_price_str.short_description = "Giá sau giảm"

    def get_discount(self):
        return self.sim.get_discount()
    get_discount.short_description = "Giảm"

    def get_total_price(self):
        total_price = 0
        for item in OrderItem.objects.filter(id = self.order.id):
            total_price += item.sim.get_curr_price()
        return "{:, .0f}".format(total_price) + " đ"
    get_total_price.short_description = 'Tổng giá'

```

7.2.3 Lớp định nghĩa các form nhập dữ liệu

```

from django import forms
from django.core.exceptions import ValidationError
from .models import Order

class PlaceOrderForm(forms.ModelForm):
    class Meta:
        model = Order
        fields = ['full_name', 'phone_number', 'email', 'address', 'gender',
'message',
                'cccd_image', 'portrait_image', 'payment_method']
        labels = {
            'full_name': 'Họ và tên',
            'phone_number': 'Số điện thoại',
            'email': 'Email',
            'address': 'Địa chỉ',
            'gender': 'Giới tính',
            'message': 'Yêu cầu khách hàng',
            'cccd_image': 'Ảnh CCCD đăng ký',
            'portrait_image': 'Ảnh chân dung đăng ký',
            'payment_method': 'Phương thức thanh toán',
        }
        widgets = {
            'full_name': forms.TextInput(attrs={'class': 'form-control text-
dark'}),
            'phone_number': forms.TextInput(attrs={'class': 'form-control
text-dark'}),
            'email': forms.EmailInput(attrs={'class': 'form-control text-
dark'}),
            'address': forms.TextInput(attrs={'class': 'form-control text-
dark'}),

```



```

        'gender': forms.Select(attrs={'class': 'form-control text-dark ml-0'}),
        'message': forms.Textarea(attrs={'class': 'form-control text-dark', 'rows': 4,}),
        'ccd_image': forms.ClearableFileInput(attrs={'class': 'custom-file-input'}),
        'portrait_image': forms.ClearableFileInput(attrs={'class': 'custom-file-input', 'id': 'customFile'}),
        'payment_method': forms.Select(attrs={'class': 'form-control text-dark ml-0'}),
    }

    def clean_phone_number(self):
        # Hàm kiểm tra số điện thoại hợp lệ
        phone_number = self.cleaned_data['phone_number']
        if not phone_number.isnumeric():
            raise ValidationError('Số điện thoại không hợp lệ')
        return phone_number

from django import forms
from django.contrib.auth.forms import UserCreationForm
from .models import Customer
from django.core.exceptions import ValidationError
from django.contrib.auth.forms import AuthenticationForm, PasswordResetForm, SetPasswordForm, PasswordChangeForm

class SignupForm(UserCreationForm):
    email = forms.EmailField(max_length=200, required=True, error_messages={
        'unique': 'Email đã được sử dụng',
        'invalid': 'Email không hợp lệ',
        'required': 'Vui lòng nhập email',
    })
    phone_number = forms.CharField(max_length=30, required=True, error_messages={
        'unique': 'Số điện thoại đã được sử dụng',
        'required': 'Vui lòng nhập số điện thoại',
    })
    address = forms.CharField(max_length=300, required=True, error_messages={
        'required': 'Vui lòng nhập địa chỉ',
    })

    MALE = 'M'
    FEMALE = 'F'
    GENDER_CHOICES = (
        (MALE, 'Nam'),
        (FEMALE, 'Nữ'),
    )
    gender = forms.ChoiceField(choices=GENDER_CHOICES)

class Meta:

```

```

    model = Customer
    fields = ('username', 'email', 'password1', 'password2', 'first_name',
'last_name', 'gender', 'address', 'phone_number')
    labels = {
        'username': 'Tên đăng nhập',
        'email': 'Email',
        'password1': 'Mật khẩu',
        'password2': 'Xác nhận mật khẩu',
        'first_name': 'Tên',
        'last_name': 'Họ',
        'phone_number': 'Số điện thoại',
        'address': 'Địa chỉ',
        'gender': 'Giới tính',
    }
}

def __init__(self, *args, **kwargs):
    super().__init__(*args, **kwargs)

    # Update HTML attributes for each field
    self.fields['username'].widget.attrs.update({
        'class': 'form-control',
        'id': 'floatingInput',
        'placeholder': 'Tên đăng nhập',
    })
    self.fields['email'].widget.attrs.update({
        'class': 'form-control',
        'id': 'floatingInput',
        'placeholder': 'name@example.com',
    })
    self.fields['password1'].widget.attrs.update({
        'class': 'form-control',
        'id': 'floatingPassword',
        'placeholder': 'Mật khẩu',
    })
    self.fields['password2'].widget.attrs.update({
        'class': 'form-control',
        'id': 'floatingPassword',
        'placeholder': 'Xác nhận mật khẩu',
    })
    self.fields['first_name'].widget.attrs.update({
        'class': 'form-control',
        'id': 'floatingInput',
        'placeholder': 'Tên',
    })
    self.fields['last_name'].widget.attrs.update({
        'class': 'form-control',
        'id': 'floatingInput',
        'placeholder': 'Họ',
    })
    self.fields['phone_number'].widget.attrs.update({

```

```

        'class': 'form-control',
        'id': 'floatingInput',
        'placeholder': 'Số điện thoại',
    })
    self.fields['address'].widget.attrs.update({
        'class': 'form-control',
        'id': 'floatingInput',
        'placeholder': 'Địa chỉ',
    })
    self.fields['gender'].widget.attrs.update({
        'class': 'form-control',
        'id': 'floatingInput',
        'placeholder': 'Giới tính',
    })

def clean_email(self):
    email = self.cleaned_data.get('email')
    if Customer.objects.filter(email=email).exists():
        raise ValidationError('Email đã được đăng ký.')
    return email

def clean_username(self):
    username = self.cleaned_data.get('username')
    if Customer.objects.filter(username=username).exists():
        raise ValidationError('Tên đăng nhập đã được sử dụng.')
    return username

def clean_phone_number(self):
    phone_number = self.cleaned_data.get('phone_number')
    if Customer.objects.filter(phone_number=phone_number).exists():
        raise ValidationError('Số điện thoại đã được đăng ký.')
    return phone_number

def clean_password2(self):
    password1 = self.cleaned_data.get('password1')
    password2 = self.cleaned_data.get('password2')

    if password1 and password2 and password1 != password2:
        raise forms.ValidationError('Mật khẩu xác nhận không khớp với mật
khẩu đã nhập.')

    # check if the password meets the minimum length requirement
    if len(password2) < 8:
        raise forms.ValidationError('Mật khẩu phải dài ít nhất 8 ký tự.')

    # check if the password is too common
    common_passwords = ['123456', 'password', 'qwerty', '123456789',
'12345678', '12345', '1234567', 'password1', '123123', 'admin', 'welcome',
'123qwe', '1234567890', 'password123']
    if password2.lower() in common_passwords:

```

```

        raise forms.ValidationError('Mật khẩu quá dễ đoán, vui lòng chọn
mật khẩu khác.')

    return password2

class LoginForm(AuthenticationForm):
    username = forms.CharField(label="Tên đăng nhập", max_length=30,
                               widget=forms.TextInput(attrs={'class': 'form-
control',
                                                             'placeholder':
'Tên đăng nhập'}))
    password = forms.CharField(label="Mật khẩu", max_length=30,
                               widget=forms.PasswordInput(attrs={'class':
'form-control',
                                                             'placeholder'
:'Mật khẩu'}))

    remember_me = forms.BooleanField(required=False, initial=True,
widget=forms.CheckboxInput(attrs={'class': 'form-check-input', 'style':
'margin-left: 0px'}))

    error_messages = {
        'invalid_login': 'Tài khoản hoặc mật khẩu không chính xác. Vui lòng
thử lại!',
        'inactive': 'Tài khoản chưa được kích hoạt, vui lòng vào mail để xác
nhận lại.',
    }

    def __init__(self, request=None, *args, **kwargs):
        super().__init__(request, *args, **kwargs)
        self.fields['username'].widget.attrs.update({'autofocus': True,
'class': 'form-control'})
        self.fields['password'].widget.attrs.update({'autocomplete': 'current-
password', 'class': 'form-control'})
        self.fields['remember_me'].widget.attrs.update({'class': 'form-check-
input'})

class CustomPasswordResetForm(PasswordResetForm):
    email = forms.EmailField(
        label='Địa chỉ email',
        max_length=254,
        widget=forms.EmailInput(attrs={
            'class': 'form-control',
            'id': 'floatingInput',
            'placeholder': 'name@example.com',}),
        error_messages={
            'required': 'Vui lòng nhập địa chỉ email.',

```

```

        'invalid': 'Địa chỉ email không hợp lệ.',
        'max_length': 'Địa chỉ email phải nhỏ hơn hoặc bằng 254 kí tự.',
    }
)

class CustomPasswordResetConfirmForm(SetPasswordForm):
    new_password1 = forms.CharField(
        label='Mật khẩu',
        widget=forms.PasswordInput(attrs={'class': 'form-control', 'id':
'floatingInput'})),
    )
    new_password2 = forms.CharField(
        label='Xác nhận mật khẩu',
        widget=forms.PasswordInput(attrs={'class': 'form-control', 'id':
'floatingInput'})),
    )

    error_messages = {
        'password_mismatch': ("Mật khẩu không khớp"),
        'password_match_username': ("Mật khẩu mới không được trùng với tên
đăng nhập."),
    }

class UpdateCustomerForm(forms.ModelForm):
    email = forms.EmailField(max_length=200, required=True, error_messages={
        'unique': 'Email đã được sử dụng',
        'invalid': 'Email không hợp lệ',
        'required': 'Vui lòng nhập email',
    })
    phone_number = forms.CharField(max_length=30, required=True,
error_messages={
        'unique': 'Số điện thoại đã được sử dụng',
        'required': 'Vui lòng nhập số điện thoại',
    })
    address = forms.CharField(max_length=300, required=True, error_messages={
        'required': 'Vui lòng nhập địa chỉ',
    })

    MALE = 'M'
    FEMALE = 'F'
    GENDER_CHOICES = (
        (MALE, 'Nam'),
        (FEMALE, 'Nữ'),
    )
    gender = forms.ChoiceField(choices=GENDER_CHOICES)

class Meta:
    model = Customer

```

```

        fields = ('email', 'first_name', 'last_name', 'gender', 'address',
'phone_number')
        labels = {
            'email': 'Email',
            'first_name': 'Tên',
            'last_name': 'Họ',
            'phone_number': 'Số điện thoại',
            'address': 'Địa chỉ',
            'gender': 'Giới tính',
        }

def __init__(self, *args, **kwargs):
    super().__init__(*args, **kwargs)

    # Update HTML attributes for each field
    self.fields['email'].widget.attrs.update({
        'class': 'form-control',
        'id': 'floatingInput',
        'placeholder': 'name@example.com',
    })
    self.fields['first_name'].widget.attrs.update({
        'class': 'form-control',
        'id': 'floatingInput',
        'placeholder': 'Tên',
    })
    self.fields['last_name'].widget.attrs.update({
        'class': 'form-control',
        'id': 'floatingInput',
        'placeholder': 'Họ',
    })
    self.fields['phone_number'].widget.attrs.update({
        'class': 'form-control',
        'id': 'floatingInput',
        'placeholder': 'Số điện thoại',
    })
    self.fields['address'].widget.attrs.update({
        'class': 'form-control',
        'id': 'floatingInput',
        'placeholder': 'Địa chỉ',
    })
    self.fields['gender'].widget.attrs.update({
        'class': 'form-control',
        'id': 'floatingInput',
        'placeholder': 'Giới tính',
    })

def clean_email(self):
    email = self.cleaned_data.get('email')
    if
Customer.objects.filter(email=email).exclude(id=self.instance.id).exists():

```

```

        raise ValidationError('Email đã được đăng ký.')
    return email

def clean_phone_number(self):
    phone_number = self.cleaned_data.get('phone_number')
    if
Customer.objects.filter(phone_number=phone_number).exclude(id=self.instance.id
).exists():
        raise ValidationError('Số điện thoại đã được đăng ký.')
    return phone_number

class ChangePasswordForm(PasswordChangeForm):
    old_password = forms.CharField(widget=forms.PasswordInput(attrs={
        'class': 'form-control',
        'placeholder': 'Mật khẩu cũ',
    })), label='Mật khẩu cũ')

    new_password1 = forms.CharField(widget=forms.PasswordInput(attrs={
        'class': 'form-control',
        'placeholder': 'Mật khẩu mới',
    })), label='Mật khẩu mới')

    new_password2 = forms.CharField(widget=forms.PasswordInput(attrs={
        'class': 'form-control',
        'placeholder': 'Xác nhận mật khẩu mới',
    })), label='Xác nhận mật khẩu mới')

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)

        self.fields['new_password1'].help_text = 'Mật khẩu mới phải dài ít
nhất 8 ký tự.'

    class Meta:
        model = None

    def clean_new_password1(self):
        new_password1 = self.cleaned_data.get('new_password1')

        # check if the password meets the minimum length requirement
        if len(new_password1) < 8:
            raise forms.ValidationError('Mật khẩu mới phải dài ít nhất 8 ký
tự.')

        return new_password1
from django import forms
from .models import Network, Tag

class SIMFilterForm(forms.Form):

```

```

min_price = forms.DecimalField(label='Giá tối thiểu', required=False)
max_price = forms.DecimalField(label='Giá tối đa', required=False)
networks = forms.ModelMultipleChoiceField(queryset=Network.objects.all(),
widget=forms.CheckboxSelectMultiple, required=False)
tags = forms.ModelMultipleChoiceField(queryset=Tag.objects.all(),
widget=forms.CheckboxSelectMultiple, required=False)

```

7.2.4 Cấu hình url cho ứng dụng web

```

from django.contrib import admin
from django.conf import settings
from . import views
from django.conf.urls.static import static
from django.urls import include, path

urlpatterns = [
    path('jet/', include('jet.urls', 'jet')), # Django JET URLs
    path('jet/dashboard/', include('jet.dashboard.urls',
        'jet-dashboard')), # Django JET dashboard URLs
    path('admin/', admin.site.urls),
    path('cua-hang/', views.shop, name='cua-hang'),
    path('trang-chu/', views.home, name='trang-chu'),
    path('', views.home, name='home'),
    path('sim/<slug:slug_sim>', views.sim, name='sim'),
    path('tim-sim', views.search_sim, name='search-sim'),
    path('nha-mang/<slug:slug_network>', views.network, name='network'),
    path('sim-so-dep/<slug:slug_tag>', views.tag, name='tag'),
    path('load-more-sim/', views.SIMListView.as_view(), name='load-more-sim'),
    path('filter-sim/', views.SIMFilterListView.as_view(), name='filter-sim'),
    path('tai-khoan/', include('customer.urls')),
    path('bai-viet/', include('post.urls')),
    path('don-hang/', include('order.urls')),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

from django.urls import path
from . import views

urlpatterns = [
    path('dang-ky/', views.signup, name = 'signup'),
    path('dang-nhap/', views.login_view, name = 'signin'),
    path('thay-doi-thong-tin/', views.customer_update,
name='customer_update'),
    path('thay-doi-mat-khau/', views.change_password, name='pass-change'),
    path('gio-hang/', views.cart, name='cart'),
    path('them-vao-gio-hang/', views.add_to_cart, name='add-to-cart'),
    path('bo-khoi-gio-hang/', views.remove_from_cart, name='remove-from-
cart'),
    path('kich-hoat/(?P<uidb64>[0-9A-Za-z_\-]+)/(?P<token>[0-9A-Za-z]{1,13}-
[0-9A-Za-z]{1,20})/$',

```



```

        views.activate, name='activate'),
    path('gui-lai-email-xac-nhan/<slug:username>', views.resend_valid_mail,
name = 'resend-email-valid'),
    path('dang-xuat/', views.logout_view, name='logout'),
    path('reset-password/',
        views.CustomPasswordResetView.as_view(),
        name='password-reset'),
    path('reset-password/hoan-thanh/',
        views.CustomPasswordResetDoneView.as_view(),
        name='password_reset_done'),
    path('reset-password-xac-nhan/<uidb64>/<token>/',
        views.CustomPasswordResetConfirmView.as_view(),
        name='password_reset_confirm'),
    path('reset-password-hoan-thanh/',
        views.CustomPasswordResetCompleteView.as_view(),
        name='password_reset_complete'),
]

from django.urls import path
from . import views

urlpatterns = [
    path('dat-hang/', views.place_order, name = 'place-order'),
    path('tra-cuu/<int:id>/', views.order_tracker, name = 'order-tracker'),
    path('xem-hoa-don/<int:id>/', views.order_pdf, name = 'order-invoice'),
    path('quan-ly-don/', views.manage_order, name = 'manage-order'),
    path('huy-don/', views.cancel_order, name = 'cancel-order'),
]

from . import views
from django.urls import include, path

urlpatterns = [
    path('', views.all_post, name='all-posts'),
    path('<slug:topic_slug>/', views.topic_post, name='post-topic'),
    path('<slug:topic_slug>/<slug:post_slug>/', views.post, name='post'),
]

```

7.2.5 Xây dựng views tương tác giữa giao diện, người dùng, hệ thống

```

from django.shortcuts import render, get_object_or_404, redirect
from sim_manager.models import SIM, Tag, Network
from sim_manager.serializers import SIMSerializer, TagSerializer,
NetworkSerializer
from django.views.generic import View, TemplateView

```

```

from django.core.paginator import Paginator
from django.http import JsonResponse
from django.db.models import F, FloatField, ExpressionWrapper
from django.db.models import Q
from customer.models import CartItem

from post.models import Post, Topic
from post.serializers import PostSerializer, TopicSerializer

def home(request):
    tagSerializer = TagSerializer(
        Tag.objects.all(), many=True)

    networkSerializer = NetworkSerializer(
        Network.objects.all(), many=True)

    sale_sims = SIM.objects.all().filter(is_available=True, is_visible=True,
discount__gt=0)[:9]
    sale_simSerializer = SIMSerializer(sale_sims, many=True)

    pinned_posts = Post.objects.filter(is_pinned=True)
    topics = Topic.objects.all()
    cart_count = 0
    if request.user.is_authenticated:
        cart_count = CartItem.objects.all().filter(customer =
request.user).count()

    context = {
        "tags": tagSerializer.data,
        "networks": networkSerializer.data,
        "sale_sims": sale_simSerializer.data,
        "cart_count": cart_count,
        "pinned_posts": PostSerializer(pinned_posts, many = True).data,
        "topics": TopicSerializer(topics, many = True).data,
    }

    return render(request, 'home.html', context)

def shop(request):

    pinned_posts = Post.objects.filter(is_pinned=True)
    topics = Topic.objects.all()
    tagSerializer = TagSerializer(
        Tag.objects.all(), many=True)

    networkSerializer = NetworkSerializer(
        Network.objects.all(), many=True)

    simSerializer = SIMSerializer(

```

```

SIM.objects.all().filter(is_available=True, is_visible=True).
annotate(
    current_price=ExpressionWrapper(
        F('price') - (F('discount') / 100) * F('price'),
        output_field=FloatField()
    )
).order_by('current_price')[:8], many=True)

cart_count = 0
if request.user.is_authenticated:
    cart_count = CartItem.objects.all().filter(customer =
request.user).count()

context = {
    "tags": tagSerializer.data,
    "networks": networkSerializer.data,
    "sims": simSerializer.data,
    "cart_count": cart_count,
    "pinned_posts": PostSerializer(pinned_posts, many = True).data,
    "topics": TopicSerializer(topics, many = True).data,
}

return render(request, 'shop.html', context)

def network(request, slug_network):
    tagSerializer = TagSerializer(
        Tag.objects.all(), many=True)

    networkSerializer = NetworkSerializer(
        Network.objects.all(), many=True)

    network = get_object_or_404(Network, slug=slug_network)
    simSerializer = SIMSerializer(
        SIM.objects.all().filter(is_available=True, is_visible=True,
network=network).
        annotate(
            current_price=ExpressionWrapper(
                F('price') - (F('discount') / 100) * F('price'),
                output_field=FloatField()
            )
        ).order_by('current_price'), many=True)

    pinned_posts = Post.objects.filter(is_pinned=True)
    topics = Topic.objects.all()
    cart_count = 0
    if request.user.is_authenticated:

```

```

        cart_count = CartItem.objects.all().filter(customer =
request.user).count()

    context = {
        "tags": tagSerializer.data,
        "networks": networkSerializer.data,
        "sims": simSerializer.data,
        "curr_net": network.slug,
        "cart_count": cart_count,
        "pinned_posts": PostSerializer(pinned_posts, many = True).data,
        "topics": TopicSerializer(topics, many = True).data,
    }

    return render(request, 'shop.html', context)

def tag(request, slug_tag):
    pinned_posts = Post.objects.filter(is_pinned=True)
    topics = Topic.objects.all()
    tagSerializer = TagSerializer(
        Tag.objects.all(), many=True)

    networkSerializer = NetworkSerializer(
        Network.objects.all(), many=True)

    tag = get_object_or_404(Tag, slug=slug_tag)
    simSerializer = SIMSerializer(
        SIM.objects.all().filter(is_available=True, is_visible=True,
tags=tag).
        annotate(
            current_price=ExpressionWrapper(
                F('price') - (F('discount') / 100) * F('price'),
                output_field=FloatField()
            )
        ).order_by('current_price'), many=True)

    cart_count = 0
    if request.user.is_authenticated:
        cart_count = CartItem.objects.all().filter(customer =
request.user).count()

    context = {
        "tags": tagSerializer.data,
        "networks": networkSerializer.data,
        "sims": simSerializer.data,
        "curr_tag": tag.slug,
        "cart_count": cart_count,
        "pinned_posts": PostSerializer(pinned_posts, many = True).data,
        "topics": TopicSerializer(topics, many = True).data,

```

```

    }

    return render(request, 'shop.html', context)

def sim(request, slug_sim):
    tagSerializer = TagSerializer(
        Tag.objects.all(), many=True)

    networkSerializer = NetworkSerializer(
        Network.objects.all(), many=True)

    sim = SIM.objects.get(slug=slug_sim)

    if not sim.is_visible and not request.user.is_admin:
        return home(request)

    other_sims = SIM.objects.all().filter(is_available=True,
network=sim.network).exclude(id=sim.id).annotate(
        current_price=ExpressionWrapper(
            F('price') - (F('discount') / 100) * F('price'),
            output_field=FloatField()
        )
    ).order_by('current_price')

    simSerializer = SIMSerializer(
        other_sims[:8], many=True)
    addedState = False

    pinned_posts = Post.objects.filter(is_pinned=True)
    topics = Topic.objects.all()
    cart_count = 0
    if request.user.is_authenticated:
        cart_item = CartItem.objects.filter(sim=sim,
customer=request.user).first()
        if cart_item:
            addedState = True
        else:
            addedState = False
        cart_count = CartItem.objects.all().filter(customer =
request.user).count()

    context = {
        "tags": tagSerializer.data,
        "networks": networkSerializer.data,
        "sim": sim,
        "addedState": addedState,
        "othersims_samenetwork": simSerializer.data,
        "cart_count": cart_count,
    }

```

```

        "pinned_posts": PostSerializer(pinned_posts, many = True).data,
        "topics": TopicSerializer(topics, many = True).data,
    }

    return render(request, 'detailed_sim.html', context)

def search_sim(request):
    search_input = request.POST.get('search_input').strip().replace('.', '')

    tagSerializer = TagSerializer(
        Tag.objects.all(), many=True)

    networkSerializer = NetworkSerializer(
        Network.objects.all(), many=True)

    sims = SIM.objects.filter(slug__contains=search_input,
is_visible=True).annotate(
        current_price=ExpressionWrapper(
            F('price') - (F('discount') / 100) * F('price'),
            output_field=FloatField()
        )
    ).order_by('current_price')

    pinned_posts = Post.objects.filter(is_pinned=True)
    topics = Topic.objects.all()
    cart_count = 0
    if request.user.is_authenticated:
        cart_count = CartItem.objects.all().filter(customer =
request.user).count()

    context = {
        "tags": tagSerializer.data,
        "networks": networkSerializer.data,
        "sims": SIMSerializer(sims, many=True).data,
        "cart_count": cart_count,
        "pinned_posts": PostSerializer(pinned_posts, many = True).data,
        "topics": TopicSerializer(topics, many = True).data,
    }

    return render(request, 'search_sim.html', context)

class SIMListView(View):
    def post(self, *args, **kwargs):
        page_number = self.request.POST.get('page_number')
        sim_slug = self.request.POST.get('sim_slug')
        sim = SIM.objects.get(slug=sim_slug)

```

```

        other_sims = SIM.objects.all().filter(is_available=True,
is_visible=True, network=sim.network).exclude(id=sim.id)
paginator = Paginator(other_sims, 8)
page_obj = paginator.get_page(page_number)
# print(page_number)
data = SIMSerializer(page_obj, many=True).data
if (int(page_number) > paginator.num_pages):
    data = None
return JsonResponse({
    'data': data,
})

class SIMFilterListView(View):
    def post(self, *args, **kwargs):
        request = self.request
        networks = request.POST.getlist('networks[]')
        tags = request.POST.getlist('tags[]')
        page_number = request.POST.get('page_number')
        dateofBirth = request.POST.get('dateofBirth')
        curr_net = request.POST.get('curr_net')
        curr_tag = request.POST.get('curr_tag')
        order = request.POST.get('order')
        price_range = request.POST.get('priceRange')
        avoidNum = request.POST.get('avoidNum')
        startNum = request.POST.get('startNum')
        endNum = request.POST.get('endNum')

        if order == "price_increasing":
            queryset = SIM.objects.annotate(
                current_price=ExpressionWrapper(
                    F('price') - (F('discount') / 100) * F('price'),
                    output_field=FloatField()
                )
            ).order_by(F('current_price').asc())
        else:
            queryset = SIM.objects.annotate(
                current_price=ExpressionWrapper(
                    F('price') - (F('discount') / 100) * F('price'),
                    output_field=FloatField()
                )
            ).order_by(F('current_price').desc())

        queryset = queryset.filter(is_available=True, is_visible=True)

        if price_range:
            minPrice, maxPrice = price_range.split('-')
            queryset = queryset.filter(current_price__gte=float(minPrice),
current_price__lte=float(maxPrice))

```

```

if dateofBirth:
    year, month, day = dateofBirth.split("-")
    dateofBirth = day + month + year
    print(dateofBirth)
    queryset = queryset.filter(slug__contains=dateofBirth)

if curr_net:
    networks = [curr_net]
if curr_tag:
    tags = [curr_tag]

if networks:
    queryset = queryset.filter(Q(network__slug__in=networks))

if tags:
    queryset = queryset.filter(Q(tags__slug__in=tags))

if avoidNum:
    avoidNum = avoidNum.replace(' ', '')
    if ',' in avoidNum:
        avoidNums = avoidNum.split(',')
    else:
        avoidNums = [avoidNum]
    q_objects = Q()

    for substring in avoidNums:
        q_objects |= Q(my_field__icontains=substring)

    queryset = queryset.exclude(q_objects)

if startNum:
    startNum = startNum.replace(' ', '')
    if ',' in startNum:
        startNums = startNum.split(',')
    else:
        startNums = [startNum]
    q_start = Q()
    for c in startNums:
        q_start |= Q(slug__startswith=c)
    queryset = queryset.filter(q_start)

if endNum:
    startNum = endNum.replace(' ', '')
    if ',' in endNum:
        endNums = endNum.split(',')
    else:
        endNums = [endNum]
    q_end = Q()
    for c in endNums:

```



```

        q_end |= Q(phone_number__endswith=c)
        queryset = queryset.filter(q_end)

    if order == "price_increasing":
        queryset.order_by(F('current_price').asc())
    else:
        queryset.order_by(F('current_price').desc())

    # sim = SIM.objects.get(slug=sim_slug)
    # other_sims = SIM.objects.all().filter(is_available=True,
network=sim.network).exclude(id=sim.id)
    paginator = Paginator(queryset, 8)
    page_obj = paginator.get_page(page_number)
    data = SIMSerializer(page_obj, many=True).data

    print(data)
    if (int(page_number) > paginator.num_pages):
        data = None

    return JsonResponse({
        'data': data,
    })
from django.shortcuts import render
from sim_manager.models import SIM, Network, Tag
from sim_manager.serializers import SIMSerializer, NetworkSerializer,
TagSerializer
from customer.models import CartItem

from post.models import Post, Topic
from post.serializers import PostSerializer, TopicSerializer
# Create your views here.

def post(request, topic_slug, post_slug):
    tagSerializer = TagSerializer(
        Tag.objects.all(), many=True)

    networkSerializer = NetworkSerializer(
        Network.objects.all(), many=True)

    post = Post.objects.get(slug = post_slug)

    pinned_posts = Post.objects.filter(is_pinned=True)
    topics = Topic.objects.all()

    context = {
        "tags": tagSerializer.data,
        "networks": networkSerializer.data,
        "cart_count": CartItem.objects.filter(customer=request.user).count(),
        "pinned_posts": PostSerializer(pinned_posts, many = True).data,

```

```

        "topics": TopicSerializer(topics, many = True).data,
        "post": post,
    }

    return render(request, 'post_detailed.html', context=context)

def topic_post(request, topic_slug):
    tagSerializer = TagSerializer(
        Tag.objects.all(), many=True)

    networkSerializer = NetworkSerializer(
        Network.objects.all(), many=True)

    posts = Post.objects.filter(topic__slug = topic_slug)

    pinned_posts = Post.objects.filter(is_pinned=True)
    topics = Topic.objects.all()

    context = {
        "tags": tagSerializer.data,
        "networks": networkSerializer.data,
        "cart_count": CartItem.objects.filter(customer=request.user).count(),
        "pinned_posts": PostSerializer(pinned_posts, many = True).data,
        "topics": TopicSerializer(topics, many = True).data,
        "posts": PostSerializer(posts, many = True).data
    }

    return render(request, 'post_list.html', context=context)

def all_post(request):
    tagSerializer = TagSerializer(
        Tag.objects.all(), many=True)

    networkSerializer = NetworkSerializer(
        Network.objects.all(), many=True)

    posts = Post.objects.all()

    pinned_posts = Post.objects.filter(is_pinned=True)
    topics = Topic.objects.all()

    context = {
        "tags": tagSerializer.data,
        "networks": networkSerializer.data,
        "cart_count": CartItem.objects.filter(customer=request.user).count(),
        "pinned_posts": PostSerializer(pinned_posts, many = True).data,
        "topics": TopicSerializer(topics, many = True).data,
        "posts": PostSerializer(posts, many = True).data
    }
}

```

```

        return render(request, 'post_list.html', context=context)

from datetime import datetime
from django.http import HttpResponse
from django.shortcuts import get_object_or_404, render, redirect
from django.contrib.auth import login, authenticate

from customer.serializers import CartItemSerializer
from .forms import SignupForm, LoginForm, CustomPasswordResetForm,
CustomPasswordResetConfirmForm, UpdateCustomerForm, ChangePasswordForm
from django.contrib.auth.decorators import login_required
from .models import Customer
from django.contrib.sites.shortcuts import get_current_site
from django.utils.encoding import force_bytes, force_str
from django.utils.http import urlsafe_base64_encode, urlsafe_base64_decode
from django.contrib.auth import update_session_auth_hash
from django.template.loader import render_to_string
from .tokens import account_activation_token
from django.contrib.auth import logout
from django.contrib.auth.models import User
from django.core.mail import EmailMessage
from sim_manager.models import SIM, Tag, Network
from sim_manager.serializers import SIMSerializer, TagSerializer,
NetworkSerializer
from .models import Customer, CartItem

from django.contrib.auth.views import (
    PasswordResetView,
    PasswordResetDoneView,
    PasswordResetConfirmView,
    PasswordResetCompleteView
)

from post.models import Post, Topic
from post.serializers import PostSerializer, TopicSerializer

def signup(request):
    if request.user.is_authenticated:
        return redirect('/')
    if request.method == 'POST':
        form = SignupForm(request.POST)
        if form.is_valid():
            user = form.save(commit=False)
            user.is_active = False
            user.save()
            current_site = get_current_site(request)

```

```

mail_subject = 'Kích hoạt tài khoản trên simminhvu'
message = render_to_string('acc_active_email.html', {
    'user': user,
    'domain': current_site.domain,
    'uid':urlsafe_base64_encode(force_bytes(user.pk)),
    'token':account_activation_token.make_token(user),
})
to_email = form.cleaned_data.get('email')
email = EmailMessage(
    mail_subject, message, to=[to_email]
)
email.send()
return render(request, 'redirect_page.html', {'message': 'Mail xác
nhận đã được gửi, vui lòng truy cập vào hòm và ấn đường link xác nhận để có
thể đăng nhập.', 'username': user.username})

else:
return render(request, 'signup.html', {'form': form, 'title':
'Dăng ký'})
else:
form = SignupForm()
return render(request, 'signup.html', {'form': form, 'title': 'Đăng ký'})

def activate(request, uidb64, token):
try:
uid = force_str(urlsafe_base64_decode(uidb64))
user = Customer.objects.get(pk=uid)
except(TypeError, ValueError, OverflowError, Customer.DoesNotExist):
user = None
if user is not None and account_activation_token.check_token(user, token):
user.is_active = True
user.save()
login(request, user)
return render(request, 'redirect_page.html', {'message': 'Tài khoản đã
được xác nhận thành công, vui lòng đăng nhập để thực hiện đặt hàng.'})
else:
return render(request, 'redirect_page.html', {'message': 'Link xác
nhận không khả dụng, vui lòng thực hiện lại', 'username': user.username})

def resend_valid_mail(request, username):
if not username:
return render(request, 'redirect_page.html', {'message': 'Tài khoản
không hợp lệ.'})
current_site = get_current_site(request)
user = Customer.objects.get(username = username)
mail_subject = 'Kích hoạt tài khoản trên simminhvu'
message = render_to_string('acc_active_email.html', {
    'user': user,

```

```

        'domain': current_site.domain,
        'uid':urlsafe_base64_encode(force_bytes(user.pk)),
        'token':account_activation_token.make_token(user),
    })
    to_email = user.email
    email = EmailMessage(
        mail_subject, message, to=[to_email]
    )
    email.send()
    return render(request, 'redirect_page.html', {'message': 'Mail xác nhận đã
được gửi, vui lòng truy cập vào hòm và ấn đường link xác nhận để có thể đăng
nhập.', 'username': user.username, 'title': 'Xác nhận tài khoản'})

def logout_view(request):
    logout(request)
    return redirect('/') # Redirect to your desired page after logout

def login_view(request):
    if request.user.is_authenticated:
        return redirect('/')
    if request.method == 'POST':
        form = LoginForm(request, data=request.POST)
        if form.is_valid():
            username = form.cleaned_data.get('username')
            password = form.cleaned_data.get('password')
            remember_me = form.cleaned_data.get('remember_me')

            if not Customer.objects.get(username=username).is_active:
                current_site = get_current_site(request)
                user = Customer.objects.get(username = username)
                mail_subject = 'Kích hoạt tài khoản trên siminhvu'
                message = render_to_string('acc_active_email.html', {
                    'user': user,
                    'domain': current_site.domain,
                    'uid':urlsafe_base64_encode(force_bytes(user.pk)),
                    'token':account_activation_token.make_token(user),
                })
                to_email = user.email
                email = EmailMessage(
                    mail_subject, message, to=[to_email]
                )
                email.send()
                form.add_error(None, 'Tài khoản chưa được kích hoạt, đã gửi
mail về lại, vui lòng ấn để kích hoạt')
                return render(request, 'signin.html', {'form': form, 'title':
'Dăng nhập'})

            user = authenticate(request, username=username, password=password)
            if user is not None:

```

```

        login(request, user)
        if not remember_me:
            request.session.set_expiry(0)
            return redirect('/')
        else:
            form.add_error(None, 'Tài khoản hoặc mật khẩu không chính
xác.')
    else:
        form = LoginForm(request)
        return render(request, 'signin.html', {'form': form, 'title': 'Đăng
nhập'})

class CustomPasswordResetView>PasswordResetView):
    template_name = 'password_reset/password_reset.html'
    form_class = CustomPasswordResetForm
    email_template_name = 'password_reset/password_reset_email.html'
    subject_template_name = 'password_reset/password_reset_subject.txt'
    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['title'] = 'Lấy lại mật khẩu'
        return context

class CustomPasswordResetDoneView>PasswordResetDoneView):
    template_name='password_reset/password_reset_message.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['title'] = 'Lấy lại mật khẩu'
        context['message'] = 'Đã gửi link reset mật khẩu vào email.'
        return context

class CustomPasswordResetConfirmView>PasswordResetConfirmView):
    template_name='password_reset/password_reset_confirm.html'
    form_class = CustomPasswordResetConfirmForm
    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['title'] = 'Cập nhật mật khẩu'
        return context

class CustomPasswordResetCompleteView>PasswordResetCompleteView):
    template_name='password_reset/password_reset_message.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['title'] = 'Cập nhật mật khẩu'
        context['message'] = 'Thay đổi mật khẩu thành công.'
        return context

def customer_update(request):

```

```

customer = request.user

if not customer.is_authenticated:
    return redirect('/')

if request.method == 'POST':
    form = UpdateCustomerForm(request.POST, instance=customer)
    if form.is_valid():
        form.save()
        return redirect('/')
    else:
        form = UpdateCustomerForm(instance=customer)
    return render(request, 'customer_update.html', {'form': form, 'title':
'Thay đổi mật khẩu'})

def customer_update(request):
    customer = request.user

    if not customer.is_authenticated:
        return redirect('/')

    if request.method == 'POST':
        form = UpdateCustomerForm(request.POST, instance=customer)
        if form.is_valid():
            form.save()
            return redirect('/')
        else:
            form = UpdateCustomerForm(instance=customer)
        return render(request, 'customer_update.html', {'form': form, 'title':
'Thay đổi thông tin'})

def change_password(request):
    customer = request.user

    if not customer.is_authenticated:
        return redirect('/')

    if request.method == 'POST':
        form = ChangePasswordForm(request.user, request.POST)

        if form.is_valid():
            customer = form.save()
            update_session_auth_hash(request, customer) # Important!
            return redirect('/')

    else:
        form = ChangePasswordForm(customer)

```

```
    return render(request, 'change_password.html', {'form': form, 'title':
'Thay đổi mật khẩu'})
```

```
def cart(request):
    if not request.user.is_authenticated:
        return redirect('signin')

    tagSerializer = TagSerializer(
        Tag.objects.all(), many=True)

    networkSerializer = NetworkSerializer(
        Network.objects.all(), many=True)

    cart_items = CartItem.objects.all().filter(customer=request.user)
    cart_itemSerializer = CartItemSerializer(cart_items, many=True)

    cart_count = cart_items.count()

    pinned_posts = Post.objects.filter(is_pinned=True)
    topics = Topic.objects.all()

    available_count = 0
    for item in cart_items:
        if item.sim.is_available == True:
            available_count += 1
        else:
            item.delete()

    context = {
        "tags": tagSerializer.data,
        "networks": networkSerializer.data,
        "cart_items": cart_itemSerializer.data,
        "cart_count": cart_count,
        "pinned_posts": PostSerializer(pinned_posts, many = True).data,
        "topics": TopicSerializer(topics, many = True).data,
    }

    return render(request, 'cart.html', context)
```

```
def add_to_cart(request):
    customer = request.user

    if not customer.is_authenticated:
        return redirect('signin')

    sim_id = request.POST.get('hiddenID')
    sim = get_object_or_404(SIM, pk=sim_id, is_visible=True)

    # Check if the customer already has a cart item for this sim
```



```

    cart_item = CartItem.objects.filter(sim=sim, customer=customer).first()

    # If the customer already has a cart item for this sim, increment the
    quantity
    if not cart_item:
        # Otherwise, create a new cart item
        cart_item = CartItem(sim=sim, customer=customer)
        cart_item.save()

    return redirect('cart')

def remove_from_cart(request):
    customer = request.user

    if not customer.is_authenticated:
        return redirect('signin')

    cart_id = request.POST.get('hiddenID')
    cart_item = CartItem.objects.get(id=cart_id)

    # delete
    cart_item.delete()

    return redirect('cart')

from django.shortcuts import render, redirect
from .forms import PlaceOrderForm
from sim_manager.models import SIM, Tag, Network
from sim_manager.serializers import SIMSerializer, TagSerializer,
NetworkSerializer
from customer.models import Customer, CartItem
from customer.serializers import CartItemSerializer
from .models import Order, OrderItem
from .serializers import OrderSerializer, OrderItemSerializer
from io import StringIO
from django.http import HttpResponse
from django.template.loader import get_template
from xhtml2pdf import pisa
from io import BytesIO

from post.models import Post, Topic
from post.serializers import PostSerializer, TopicSerializer

def place_order(request):
    if not request.user.is_authenticated:
        return redirect('signin')

    tagSerializer = TagSerializer(

```

```

        Tag.objects.all(), many=True)

networkSerializer = NetworkSerializer(
    Network.objects.all(), many=True)

pinned_posts = Post.objects.filter(is_pinned=True)
topics = Topic.objects.all()

cart_items = CartItem.objects.all().filter(customer=request.user)
cart_itemSerializer = CartItemSerializer(cart_items, many=True)

available_count = 0
for item in cart_items:
    if item.sim.is_available == True:
        available_count += 1
    else:
        item.delete()

if available_count < cart_items.count():
    if request.method == 'POST':
        form = PlaceOrderForm(request.POST, request.FILES)
    else:
        initial_values = {
            'full_name': request.user.last_name + ' ' +
request.user.first_name,
            'phone_number': request.user.phone_number,
            'email': request.user.email,
            'address': request.user.address,
            'gender': request.user.gender,
        }
        form = PlaceOrderForm(initial=initial_values)

    return render(request, 'place_order.html',
        context={
            "tags": tagSerializer.data,
            "networks": networkSerializer.data,
            "cart_count": cart_items.count(),
            "cart_items": cart_itemSerializer.data,
            "pinned_posts": PostSerializer(pinned_posts, many
= True).data,
            "topics": TopicSerializer(topics, many =
True).data,
            "form": request.form,
        })

if request.method == 'POST':
    form = PlaceOrderForm(request.POST, request.FILES)
    if form.is_valid():
        # Lưu đơn đặt hàng
        if cart_items.count() == 0:

```

```

        return redirect('/')

order = form.save(commit=False)
order.order_status = 'PENDING'
order.cccd_image = form.cleaned_data['ccd_image']
order.portrait_image = form.cleaned_data['portrait_image']
order.customer = request.user
order.save()
for item in cart_items:
    sim = SIM.objects.get(id = item.sim.id)
    orderItem = OrderItem(
        order = order,
        sim = sim
    )
    sim.is_available = False
    sim.is_visible = False
    sim.save()
    orderItem.save()
    item.delete()
# Thông báo đặt hàng thành công
return render(request, 'place_order_success.html', context={
    "tags": tagSerializer.data,
    "networks": networkSerializer.data,
    "cart_count": 0,
    "pinned_posts": PostSerializer(pinned_posts, many =
True).data,
    "topics": TopicSerializer(topics, many = True).data,
    "order_id": order.id,
})
else:
    render(request, 'place_order.html',
        context={
            "tags": tagSerializer.data,
            "networks": networkSerializer.data,
            "cart_count": cart_items.count(),
            "cart_items": cart_itemSerializer.data,
            "pinned_posts": PostSerializer(pinned_posts, many =
True).data,
            "topics": TopicSerializer(topics, many = True).data,
            "form": form,
        })
else:
    initial_values = {
        'full_name': request.user.last_name + ' ' +
request.user.first_name,
        'phone_number': request.user.phone_number,
        'email': request.user.email,
        'address': request.user.address,
        'gender': request.user.gender,
    }

```

```

        form = PlaceOrderForm(initial=initial_values)

    return render(request, 'place_order.html',
                  context={
                      "tags": tagSerializer.data,
                      "networks": networkSerializer.data,
                      "cart_count": cart_items.count(),
                      "cart_items": cart_itemSerializer.data,
                      "pinned_posts": PostSerializer(pinned_posts, many =
True).data,
                      "topics": TopicSerializer(topics, many = True).data,
                      "form": form,
                  })

def cancel_order(request):
    if not request.user.is_authenticated:
        return redirect('signin')

    tagSerializer = TagSerializer(
        Tag.objects.all(), many=True)

    networkSerializer = NetworkSerializer(
        Network.objects.all(), many=True)

    pinned_posts = Post.objects.filter(is_pinned=True)
    topics = Topic.objects.all()

    cart_items = CartItem.objects.all().filter(customer=request.user)
    cart_itemSerializer = CartItemSerializer(cart_items, many=True)

    if request.method == 'POST':
        id = request.POST.get('hiddenID')

        order = Order.objects.get(id = id)

        if order.order_status == 'PENDING':
            order.order_status = 'CANCELLED'

        orderItems = OrderItem.objects.all().filter(order = order)

        for item in orderItems:
            sim = item.sim
            sim.is_available = True
            sim.is_visible = True
            sim.save()

        order.save()
        return redirect('manage-order')

```

```

    return redirect('/')

def order_tracker(request, id):
    if not request.user.is_authenticated:
        return redirect('signin')

    if id is None:
        return redirect('/')

    order = Order.objects.get(id = id)

    if order.customer != request.user:
        return redirect('/')

    tagSerializer = TagSerializer(
        Tag.objects.all(), many=True)

    networkSerializer = NetworkSerializer(
        Network.objects.all(), many=True)

    pinned_posts = Post.objects.filter(is_pinned=True)
    topics = Topic.objects.all()

    cart_items = CartItem.objects.all().filter(customer=request.user)

    orderItem = OrderItem.objects.all().filter(order=order)
    orderItemSerializer = OrderItemSerializer(orderItem, many=True)

    return render(request, 'order_detailed.html',
        context={
            "tags": tagSerializer.data,
            "networks": networkSerializer.data,
            "cart_count": cart_items.count(),
            "pinned_posts": PostSerializer(pinned_posts, many =
True).data,
            "topics": TopicSerializer(topics, many = True).data,
            "order": order,
            "orderItem": orderItemSerializer.data,
        })

# defining the function to convert an HTML file to a PDF file
def html_to_pdf(template_src, context_dict={}):
    template = get_template(template_src)
    response = HttpResponse(content_type='application/pdf')
    response['Content-Disposition'] = f'attachment; filename="invoice.pdf"'
    html = template.render(context_dict)

    pisa_status = pisa.CreatePDF(html, dest=response)

```

```

    if pisa_status.err:
        return HttpResponse(f'We had some errors <pre>{html}</pre>')

    return response

def order_pdf(request, id):
    if not request.user.is_authenticated:
        return redirect('signin')

    if id is None:
        return redirect('/')

    order = Order.objects.get(id = id)

    if order.customer != request.user:
        return redirect('/')

    orderItem = OrderItem.objects.all().filter(order=order)
    orderItemSerializer = OrderItemSerializer(orderItem, many=True)

    context={
        "order": order,
        "orderItem": orderItemSerializer.data,
    }
    # html = template.render(context)
    # response = HttpResponse(content_type='application/pdf')
    # response['Content-Disposition'] = 'attachment; filename="invoice.pdf"'

    pdf = html_to_pdf('invoice.html', context)
    return HttpResponse(pdf, content_type='application/pdf')

def manage_order(request):
    pinned_posts = Post.objects.filter(is_pinned=True)
    topics = Topic.objects.all()
    if not request.user.is_authenticated:
        return redirect('signin')

    tagSerializer = TagSerializer(
        Tag.objects.all(), many=True)

    networkSerializer = NetworkSerializer(
        Network.objects.all(), many=True)

    cart_items = CartItem.objects.all().filter(customer=request.user)

    order = Order.objects.filter(customer=request.user)
    orderSerializer = OrderSerializer(order, many=True)

```

```
return render(request, 'manage_order.html',
               context={
                   "tags": tagSerializer.data,
                   "networks": networkSerializer.data,
                   "cart_count": cart_items.count(),
                   "orders": orderSerializer.data,
                   "pinned_posts": PostSerializer(pinned_posts, many =
True).data,
                   "topics": TopicSerializer(topics, many = True).data,
               })
```