

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÀ RỊA - VŨNG TÀU
KHOA KỸ THUẬT - CÔNG NGHỆ



BARIA VUNGTAU
UNIVERSITY
CAP SAINT JACQUES

ĐỒ ÁN TỐT NGHIỆP

Tên đề tài:

**PHÁT TRIỂN WEB ỨNG DỤNG
WEB MỞ RỘNG - GROWPRO**

GVHD: ThS. Nguyễn Tân Phương

Sinh viên: Ngô Minh Thông

MSSV : 16031519

Lớp : DH19LT

Vũng Tàu, tháng 5 năm 2023

LỜI CẢM ƠN

Đầu tiên, em xin chân thành cảm ơn trường Đại Học Bà Rịa Vũng Tàu đã cho em cơ hội học tập tại môi trường giáo dục thân thiện và trình độ cao.

Tiếp theo em xin gửi lời cảm ơn tới tất cả các quý thầy cô trong trường đặc biệt là thầy Nguyễn Tấn Phương đã hỗ trợ em về mọi khía cạnh từ học tập, đời sống, đạo đức, kỷ luật,...

Em rất vui được học tập và phát triển dưới sự tận tâm, nhiệt huyết của các quý thầy cô trong trường nói chung, các thầy cô đã hướng dẫn em nói riêng và các bạn bè thân thiện trong học tập và cuộc sống. Từ đó giúp em dễ dàng tiếp thu các kiến thức một cách thoải mái và học được các kỹ năng cần thiết để tiếp tục phát triển.

Em mong rằng những gì em đã học được tại trường sẽ là hành trang quý giá để em vững bước trên con đường tương lai. Với nó, em có thể tạo ra nhiều lợi ích cho xã hội, đóng góp thêm một phần cho công cuộc phát triển đất nước.

Em xin chân thành cảm ơn!

LỜI CAM ĐOAN

Em xin cam đoan rằng những gì em đã và sẽ thực hiện trong đồ án này hoàn toàn là do bản thân dưới sự hướng dẫn của giáo viên hướng dẫn, không sao chép từ bất kỳ nguồn nào khác. Em đã thu thập các thông tin, dữ liệu từ những nguồn mở, cộng đồng hợp pháp và luôn đảm bảo các thông tin này được trích dẫn đầy đủ và chính xác. Em không có bất kỳ hành vi nào vi phạm quy định về đạo đức, bản quyền hoặc pháp luật. Nếu có bất kỳ sự không chính xác hay vi phạm quy định nào, Em sẽ chịu hoàn toàn trách nhiệm về những hậu quả pháp lý và hình thức kỷ luật theo quy định của trường và pháp luật.

Vũng Tàu, ngày 11 tháng 05 năm 2023

Sinh viên thực hiện

Ngô Minh Thống

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Vũng Tàu, ngày ... tháng ... năm 2023

Người hướng dẫn

ThS.Nguyễn Tấn Phương

Mục lục

DANH MỤC ẢNH.....	9
I. GIỚI THIỆU	13
1. Ý TƯỞNG:	14
2. CÁC YÊU CẦU ĐỂ SẢN PHẨM ĐƯỢC CÔNG NHẬN.....	15
3. CÁC FRAMEWORK WEB CHÍNH	16
4. CÁC LOẠI API TRUY VẤN DỮ LIỆU	16
5. CÁC LOẠI NGÔN NGỮ ĐƯỢC SỬ DỤNG	16
- Ngôn ngữ chính : Python.	16
- Ngôn ngữ hỗ trợ:	16
6. LOAD THÔNG TIN HIỂN THỊ THEO 3 TẦNG	16
II. PHÂN TÍCH HỆ THỐNG	17
1. SƠ ĐỒ TỔNG QUÁT	17
2. SƠ ĐỒ TƯƠNG TÁC TỔNG QUÁT	18
3. CHỨC NĂNG CỦA CÁC LOẠI NGƯỜI DÙNG	19
Khách tham quan (không có tài khoản)	19
Khách hàng tiềm năng (đã đăng nhập)	20
Người bán hàng	23
Nhà cung cấp	25
Đối tác (hỗ trợ quản lý trang)	26
Người quản lý (admin)	28
4. SƠ ĐỒ CƠ SỞ DỮ LIỆU	29
III. PHÂN TÍCH CÁC TÀI NGUYÊN VÀ CÁC LUỒNG DỮ LIỆU.....	40
1. Các loại ngôn ngữ.....	40
- Python:	40
- Jinja2 :	41
- HTML:	42
- CSS:	42
- JavaScript:	42
- GraphQL:	43
2. Các loại framework.....	43

- Django	43
- Flask	44
3. Phương thức load thông tin	45
- Tầng một:	45
- Tầng hai:.....	45
- Tầng ba:.....	45
4. Các trình tự giải quyết tương tác của người dùng.....	46
Phương thức 1:	46
Phương thức 2:	46
Phương thức 3:	47
Phương thức 4:	47
Phương thức 5:	47
5. Các loại API được sử dụng	48
- RESTful API.....	48
- API GraphQL.....	49
6. Các kiểu response trả về.....	50
- Return Template/view.....	50
- Return Redirect/url.....	50
- Return Response(API GraphQL)/dir data.....	50
- Return JsonResponse(result).....	50
- Return Response(json.dumps(context), 200).....	50
- Return jsonify(rep).....	50
Lưu ý: khi dùng với fetch từ javacript.....	50
7. Các nơi lưu trữ dữ liệu	51
- Database	51
- Server	51
- Máy trạm	51
IV. XÂY DỰNG HỆ THỐNG	52
1) Server API (Django):.....	52
a) Các bộ thư viện sử dụng:	52
b) Hướng dẫn cài đặt:.....	54
c) Cấu trúc thư mục và chức năng:	55
d) Các giao diện người dùng:.....	64
2) Server Client(Flask):	114

a)	Các bộ thư viện sử dụng:	114
b)	Hướng dẫn cài đặt:.....	116
c)	Cấu trúc thư mục và chức năng:	117
d)	Các giao diện người dùng trên laptop/pc:	120
e)	Các giao diện người dùng trên điện thoại:	129
3)	Server Admin(Flask)	140
a)	Các thư viện sử dụng	140
b)	Hướng dẫn cài đặt.....	140
c)	Cấu trúc thư mục và chức năng:	141
d)	Các giao diện người dùng:.....	143
V.	KẾT LUẬN	150
1)	Kết quả đạt được	150
2)	Hướng phát triển	150
VI.	TÀI LIỆU THAM KHẢO	150

DANH MỤC ẢNH

Hình 1 sơ đồ tổng quát	17
Hình 2 Sơ đồ tương tác tổng quát.....	18
Hình 3 sơ đồ chức năng khách tham quan.....	19
Hình 4 sơ đồ chức năng khách hàng tìm năng	20
Hình 5 sơ đồ chức năng người bán hàng	23
Hình 6 sơ đồ chức năng nhà cung cấp.....	25
Hình 7 sơ đồ chức năng đối tác	26
Hình 8 sơ đồ chức năng người quản trị	28
Hình 9 sơ đồ cơ sở dữ liệu.....	29
Hình 10 sơ đồ luồng dữ liệu theo tầng 1	45
Hình 11 sơ đồ luồng dữ liệu theo tầng 2	45
Hình 12 sơ đồ luồng dữ liệu theo tầng 3	45
Hình 13 sơ đồ luồng dữ liệu theo phương thức 1	46
Hình 14 sơ đồ luồng dữ liệu theo phương thức 2.....	46
Hình 15 sơ đồ luồng dữ liệu theo phương thức 3.....	47
Hình 16 sơ đồ luồng dữ liệu theo phương thức 4.....	47
Hình 17 sơ đồ luồng dữ liệu theo phương thức 5.....	47
Hình 18 kết quả từ một câu lệnh query	59
Hình 19 kết quả từ câu lệnh mutation	61
Hình 20 giao diện danh sách các đối tượng trong hệ thống	64
Hình 21 giao diện thêm đối tượng người dùng	68
Hình 22 giao diện cập nhật đối tượng người dùng	69
Hình 23 giao diện danh sách các đối tượng quyền trong hệ thống	70
Hình 24 giao diện thêm đối tượng quyền trong hệ thống.....	70
Hình 25 giao diện cập nhật đối tượng quyền trong hệ thống	70
Hình 26 giao diện danh sách các đối tượng nội dung trong hệ thống	71
Hình 27 giao diện thêm đối tượng nội dung trong hệ thống	71
Hình 28 giao diện danh sách các đối tượng nội dung trong hệ thống	71
Hình 29 giao diện danh sách các đối tượng người bán	72
Hình 30 giao diện thêm đối tượng người bán.....	72
Hình 31 giao diện cập nhật đối tượng người bán	72
Hình 32 giao diện danh sách các đối tượng người bán	73
Hình 33 giao diện thêm đối tượng người bán.....	73
Hình 34 giao diện cập nhật đối tượng người bán	73
Hình 35 giao diện danh sách các đối tượng đối tác.....	74
Hình 36 giao diện thêm đối tượng đối tác	74
Hình 37 giao diện cập nhật đối tượng đối tác.....	74
Hình 38 giao diện danh sách các đối tượng nhà cung cấp	75
Hình 39 giao diện thêm đối tượng nhà cung cấp.....	75
Hình 40 giao diện cập nhật đối tượng nhà cung cấp	75
Hình 41 giao diện danh sách các đối tượng trạng thái người dùng.....	76
Hình 42 giao diện thêm đối tượng trạng thái người dùng	76

Hình 43	giao diện cập nhật đối tượng trạng thái người dùng.....	76
Hình 44	giao diện danh sách các đối tượng trang hiển thị	77
Hình 45	giao diện thêm đối tượng trang hiển thị	77
Hình 46	giao diện cập nhật đối tượng trang hiển thị	77
Hình 47	giao diện danh sách các đối tượng menu.....	78
Hình 48	giao diện thêm đối tượng menu	78
Hình 49	giao diện cập nhật đối tượng menu	79
Hình 50	giao diện danh sách các đối tượng thành phần trang.....	79
Hình 51	giao diện thêm đối tượng thành phần trang.....	81
Hình 52	giao diện thêm đối tượng thành phần trang.....	83
Hình 53	giao diện danh sách các đối tượng thẻ loại hiển thị thành phần trang.....	84
Hình 54	giao diện thêm đối tượng thẻ loại hiển thị thành phần trang.....	84
Hình 55	giao diện cập nhật đối tượng thẻ loại hiển thị thành phần trang	84
Hình 56	giao diện danh sách các đối tượng thành phần con trong thành phần trang....	85
Hình 57	giao diện thêm đối tượng thành phần con trong thành phần trang.....	85
Hình 58	giao diện cập nhật đối tượng thành phần con trong thành phần trang	86
Hình 59	giao diện danh sách các đối tượng quy định các thành phần trong trang.....	86
Hình 60	giao diện thêm đối tượng quy định các thành phần trong trang.....	87
Hình 61	giao diện cập nhật đối tượng quy định các thành phần trong trang	87
Hình 62	giao diện danh sách các đối tượng thẻ loại sản phẩm	87
Hình 63	giao diện thêm đối tượng thẻ loại sản phẩm.....	88
Hình 64	giao diện cập nhật đối tượng thẻ loại sản phẩm	88
Hình 65	giao diện danh sách các đối tượng thẻ loại.....	89
Hình 66	giao diện thêm đối tượng thẻ loại.....	89
Hình 67	giao diện cập nhật đối tượng thẻ loại.....	89
Hình 68	giao diện danh sách các đối tượng sản phẩm	90
Hình 69	giao diện thêm đối tượng sản phẩm.....	91
Hình 70	giao diện cập nhật đối tượng sản phẩm	92
Hình 71	giao diện danh sách các đối tượng sản phẩm	93
Hình 72	giao diện thêm đối tượng sản phẩm.....	93
Hình 73	giao diện cập nhật đối tượng sản phẩm	93
Hình 74	giao diện danh sách các đối tượng thích các sản phẩm.....	94
Hình 75	giao diện thêm đối tượng thích các sản phẩm	94
Hình 76	giao diện cập nhật đối tượng thích các sản phẩm.....	94
Hình 77	giao diện danh sách các đối tượng sản phẩm bán ra	95
Hình 78	giao diện thêm đối tượng sản phẩm bán ra.....	95
Hình 79	giao diện cập nhật đối tượng sản phẩm bán ra	96
Hình 80	giao diện danh sách các đối tượng thành phần trang theo sản phẩm	96
Hình 81	giao diện thêm đối tượng thành phần trang theo sản phẩm.....	97
Hình 82	giao diện cập nhật đối tượng thành phần trang theo sản phẩm	97
Hình 83	giao diện danh sách các đối tượng thông tin giao hàng	97
Hình 84	giao diện thêm đối tượng thông tin giao hàng.....	98
Hình 85	giao diện cập nhật đối tượng thông tin giao hàng	98
Hình 86	giao diện danh sách các đối tượng sản phẩm trong hóa đơn.....	98
Hình 87	giao diện thêm đối tượng sản phẩm trong hóa đơn	99

Hình 88	giao diện cập nhật đối tượng sản phẩm trong hóa đơn.....	99
Hình 89	giao diện danh sách các đối tượng nhóm người dùng có thể tham gia	99
Hình 90	giao diện thêm đối tượng nhóm người dùng có thể tham gia.....	100
Hình 91	giao diện cập nhật đối tượng nhóm người dùng có thể tham gia	100
Hình 92	giao diện danh sách các đối tượng hội thoại	100
Hình 93	giao diện thêm đối tượng hội thoại.....	101
Hình 94	giao diện cập nhật đối tượng hội thoại	101
Hình 95	giao diện danh sách các đối tượng thành viên tham gia nhóm người dùng ..	101
Hình 96	giao diện thêm đối tượng thành viên tham gia nhóm người dùng	102
Hình 97	giao diện cập nhật đối tượng thành viên tham gia nhóm người dùng	102
Hình 98	giao diện danh sách các đối tượng lịch sử thao tác trên giao diện admin server API.....	102
Hình 99	giao diện thêm đối tượng lịch sử thao tác trên giao diện admin server API .	103
Hình 100	giao diện cập nhật đối tượng lịch sử thao tác trên giao diện admin server API	103
Hình 101	giao diện danh sách các đối tượng file đã tải lưu trên server lưu trữ	104
Hình 102	giao diện thêm đối tượng file đã tải lưu trên server lưu trữ	104
Hình 103	giao diện cập nhật đối tượng file đã tải lưu trên server lưu trữ.....	104
Hình 104	giao diện danh sách các đối tượng lịch sử hóa đơn.....	105
Hình 105	giao diện thêm đối tượng lịch sử hóa đơn	105
Hình 106	giao diện cập nhật đối tượng lịch sử hóa đơn.....	105
Hình 107	giao diện danh sách các đối tượng lịch sử xem của khách hàng	106
Hình 108	giao diện thêm đối tượng lịch sử xem của khách hàng	106
Hình 109	giao diện cập nhật đối tượng lịch sử xem của khách hàng.....	106
Hình 110	giao diện danh sách các lịch sử hành động của server supadmin.....	107
Hình 111	giao diện thêm đối tượng lịch sử hành động của server supadmin	107
Hình 112	giao diện cập nhật đối tượng lịch sử hành động của server supadmin.....	108
Hình 113	giao diện danh sách các đối tượng nhóm phân quyền người dùng	108
Hình 114	giao diện thêm đối tượng nhóm phân quyền người dùng.....	109
Hình 115	giao diện cập nhật đối tượng nhóm phân quyền người dùng	109
Hình 116	giao diện hướng dẫn sử dụng api grapql	110
Hình 117	giao diện xem lại lịch sử hướng dẫn sử dụng api grapql.....	110
Hình 118	giao diện hướng dẫn sử dụng query	111
Hình 119	giao diện hướng dẫn sử dụng mutation	111
Hình 120	giao diện trang chủ	120
Hình 121	giao diện trang thể loại	121
Hình 122	giao diện sản phẩm khi trở chuột.....	122
Hình 123	giao diện sản phẩm khi xem chi tiết.....	123
Hình 124	giao diện trang trò chuyện	124
Hình 125	giao diện trang giỏ hàng	124
Hình 126	giao diện trang đăng ký	124
Hình 127	giao diện trang đăng nhập với username	125
Hình 128	giao diện trang đăng nhập với email	125
Hình 129	giao diện trang lấy lại mật khẩu	125
Hình 130	giao diện trang xác thực email.....	126

Hình 131	giao diện trang thông tin người dùng	126
Hình 132	giao diện thay đổi họ	126
Hình 133	giao diện thay đổi tên	127
Hình 134	giao diện trang lịch sử mua hàng – sản phẩm	127
Hình 135	giao diện yêu cầu hủy đơn hàng	128
Hình 136	giao diện trang chủ trên điện thoại	129
Hình 137	giao diện trang thẻ loại trên điện thoại	130
Hình 138	giao diện trang trò chuyện trên điện thoại	130
Hình 139	giao diện trang chủ trên điện thoại	131
Hình 140	giao diện trang giỏ hàng trên điện thoại	132
Hình 141	giao diện trang đăng ký trên điện thoại	133
Hình 142	giao diện trang đăng nhập trên điện thoại	134
Hình 143	giao diện trang quên mật khẩu trên điện thoại	135
Hình 144	giao diện trang xác thực email trên điện thoại	136
Hình 145	giao diện trang thông tin người dùng trên điện thoại	137
Hình 146	giao diện thay đổi thông tin người dùng trên điện thoại	138
Hình 147	giao diện trang lịch sử mua hàng trên điện thoại	139
Hình 148	giao diện trang đăng nhập với username	143
Hình 149	giao diện trang thông tin người dùng	144
Hình 150	giao diện trang danh sách layout	144
Hình 151	giao diện thêm layout	144
Hình 152	giao diện cập nhật layout	147
Hình 153	giao diện thêm layout con	147
Hình 154	giao diện thêm item layout	148
Hình 155	giao diện popup thay đổi giá trị kiểu combobox	148
Hình 156	giao diện popup thay đổi giá trị kiểu text	149
Hình 157	giao diện popup thay đổi giá trị kiểu file	149

I. GIỚI THIỆU

Mỗi người đều có một phong cách riêng của bản thân! Em mong muốn: Tạo nên một hệ thống có thể để mỗi người tham gia đều có thể phát huy được tối đa ưu điểm, phong cách của bản thân mình với vai trò/cương vị của họ.

Với sự phát triển mạnh mẽ về kinh doanh điện tử như hiện nay, việc tạo dựng hình ảnh thương hiệu luôn là vấn đề hàng đầu của các doanh nghiệp, ko chỉ mặt hàng, logo, in ấn mà ngay cả trang web cũng cần mang đậm dấu ấn của doanh nghiệp, tạo dấu ấn trong lòng khách hàng. Thì em tin tưởng hệ thống phần mềm này sẽ là một giải pháp cho vấn đề này. Với việc thiết kế trang web đại trà như hiện nay thì rất khó để tạo nên sự khác biệt dưới cái nhìn của khách hàng, nên một hệ thống tạo nên một website mà các thành phần của nó hoàn toàn có thể thay đổi dựa trên yêu cầu doanh nghiệp, bạn cần như thế nào tôi sẽ đáp ứng hết '**your website, your style**'.

Nó không lệ thuộc bởi bất kỳ khuôn mẫu nào nên bạn có thể tùy thích sáng tạo, phát huy tối đa sức tưởng tượng của mình và thể hiện nó mà không bị bất kỳ sự gò bó, ràng buộc. Bạn có thể thiết kế một trang, một phần thậm chí mỗi sản phẩm, mỗi thông tin được đưa lên là một phong cách riêng của mỗi cá nhân, mỗi lúc, mỗi nơi,...

1. Ý TƯỞNG:

- Xây dựng một hệ thống quản lý nội dung lần giao diện hiển thị theo đặc trưng của người dùng. Mọi thông tin hiển thị lên có thể được tùy chỉnh bởi sở thích, ý muốn của người sử dụng.
- Hệ thống dễ dàng nâng cấp, cập nhật mọi lúc mọi nơi, có thể mở rộng, tích hợp nếu cần.
- Hệ thống có thể cập nhật trực tiếp nội dung lần giao diện mà không làm gián đoạn hoạt động của các máy chủ và người sử dụng.
- Dễ dàng khắc phục nếu gặp lỗi.
- Thông tin bảo mật, phòng/chống các loại tấn công cơ bản.
- Thu thập thông tin người dùng nhằm hướng tới phát triển hơn.
- Dễ dàng chuyển đổi giữa các loại dịch vụ: quảng cáo, thông tin, bán hàng, blog cá nhân,...
- Tối ưu hóa tài nguyên :
 - Dung lượng: Khi tải file lên, nếu file đã lưu trên máy chủ thì trả về đường dẫn. Ngược lại lưu file và trả về đường dẫn
 - Tốc độ: Phân chia công việc trên các máy chủ, máy trạm (ưu tiên xử lý công việc trên máy trạm), sử dụng đa luồng để tăng hiệu suất trên máy chủ. Chia 3 tầng hiển thị với máy trạm.

2. CÁC YÊU CẦU ĐỂ SẢN PHẨM ĐƯỢC CÔNG NHẬN

❖ 3 câu hỏi nhanh để bắt đầu xây dựng 1 sản phẩm:

- Khách hàng/xã hội có cần hay không?

Với sự phát triển mạnh mẽ về kinh doanh điện tử như hiện nay, việc tạo dựng hình ảnh thương hiệu luôn là vấn đề hàng đầu của các doanh nghiệp, ko chỉ mặt hàng, logo, in ấn mà ngay cả trang web cũng cần mang đậm dấu ấn của doanh nghiệp, tạo dấu ấn trong lòng khách hàng. Thì em tin tưởng hệ thống phần mềm này sẽ là một giải pháp hữu hiệu.

- Tại sao khách hàng phải chọn sản phẩm của mình mà không phải của người khác/ sản phẩm này có ưu thế hơn người khác chỗ nào, lợi ích khách hàng khi sử dụng sản phẩm?

Hệ thống có thể để mỗi người tham gia đều có thể phát huy được tối đa ưu điểm, phong cách của bản thân mình với vai trò/cương vị của họ.

Với việc thiết kế trang web đại trà như hiện nay thì rất khó để tạo nên sự khác biệt dưới cái nhìn của khách hàng, nên một hệ thống tạo nên một website mà các thành phần của nó hoàn toàn có thể thay đổi dựa trên yêu cầu doanh nghiệp, bạn cần như thế nào tôi sẽ đáp ứng hết 'your website, your style'.

Nó không lệ thuộc bởi bất kỳ khuôn mẫu nào nên bạn có thể tùy thích sáng tạo, phát huy tối đa sức tưởng tượng của mình và thể hiện nó mà không bị bất kỳ sự gò bó, ràng buộc. Bạn có thể thiết kế một trang, một phần thậm chí mỗi sản phẩm, mỗi thông tin được đưa lên là một phong cách riêng của mỗi cá nhân, mỗi lúc, mỗi nơi,...

Với hệ thống được phân chia ra nhiều máy chủ khác nhau được tối ưu theo chức năng riêng nhằm tối ưu hiệu năng làm việc cũng như nâng cao tính bảo mật giảm thiểu thiệt hại khi bị tấn công/ gặp lỗi. Dễ dàng khắc phục khi gặp vấn đề với thời gian và chi phí thấp nhất.

Dễ dàng mở rộng, nâng cấp khi khách hàng có nhu cầu.

- Sản phẩm của mình có dễ bị sao chép/thay thế hay không?

Vì hệ thống hoạt động trên nhiều server với các tính năng tối ưu riêng biệt, các tầng làm việc phân lớp, đang xen với nhau. Nên với vai trò là khách hàng thì dễ sử dụng cũng như phục hồi lại nhưng với vai trò là người lập trình sẽ khó khăn trong việc mở rộng khi chưa nắm rõ cấu trúc toàn bộ hệ thống.

Để thay thế hoặc nâng cấp thì chi phí có thể cao hơn nếu chọn dùng sản phẩm của em/trả phí cho em nâng cấp. Trong khi đó nếu tùy tiện nâng cấp có thể gây ra lỗi không đồng bộ trong hệ thống.

❖ 3 câu hỏi để duy trì quảng bá sản phẩm:

- Doang thu mang lại/chi phí xây dựng của sản phẩm khi mang ra thị trường

- + Chi phí xây dựng hệ thống và chi phí mở rộng: sức lao động, server chạy hệ thống
- + Doang thu:
 - Phí cài đặt, hỗ trợ, tư vấn về hệ thống.
 - Phí duy trì, bảo dưỡng, khắc phục hệ thống khi hoạt động.
 - Phí nâng cấp, thêm chức năng, chuyển đổi tính năng, cung cấp dịch vụ,...

- Rủi ro gặp phải khi sản phẩm mang ra thị trường

- + Lượt truy cập: nâng cấp server của chức năng đó như nâng cấp phần cứng/ thêm server xử lý chức năng đó
- + Dung lượng bộ nhớ:
 - Kiểm soát file tải lên
 - Kết nối thêm database (sử dụng multiple database)/ nâng cấp database

- Cách để mọi người biết tới sản phẩm (marketing)

- + Miễn phí hệ thống với các tính năng cơ bản
- + Liên kết, hợp tác với các doang nghiệp, studio, cá nhân trong lĩnh vực it/marketing quảng bá sản phẩm và chia lợi nhuận từ nó.

3. CÁC FRAMEWORK WEB CHÍNH

- Xây dựng trang quản lý tài nguyên với Django (web framework)
- Xây dựng các trang hiển thị và tương tác với Flask (web framework)

4. CÁC LOẠI API TRUY VẤN DỮ LIỆU

- **API REST** : Truy vấn file, truy vấn giữa máy trạm-máy chủ.
- **API GraphQL** : Truy vấn dữ liệu kiểu json giữa các máy chủ.

5. CÁC LOẠI NGÔN NGỮ ĐƯỢC SỬ DỤNG

- Ngôn ngữ chính : Python.

- Ngôn ngữ hỗ trợ:

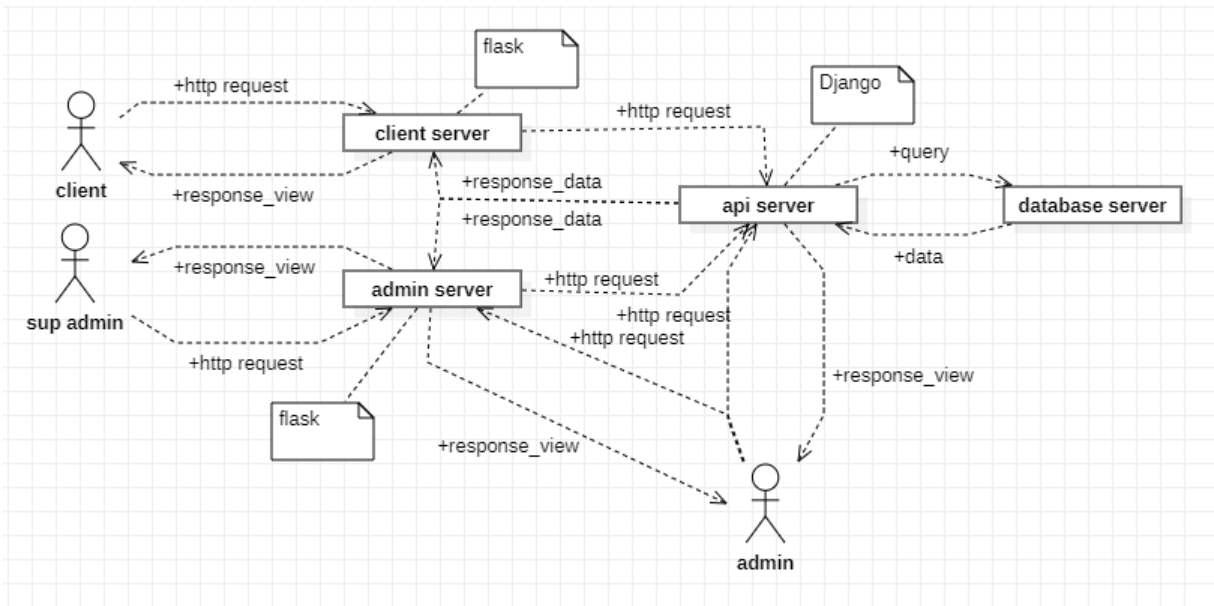
- ❖ Jinja2
- ❖ HTML
- ❖ CSS
- ❖ JavaScript
- ❖ GraphQL

6. LOAD THÔNG TIN HIỂN THỊ THEO 3 TẦNG

- **Tầng một:** Thông tin hiển thị ngay khi người dùng yêu cầu hiển thị trang
- **Tầng hai:** Tải các thông tin chưa được hiển thị (thông tin ẩn)
- **Tầng ba:** Các thông tin sẽ hiển thị theo tương tác của người dùng với trang mà không cần tải lại toàn trang.

II. PHÂN TÍCH HỆ THỐNG

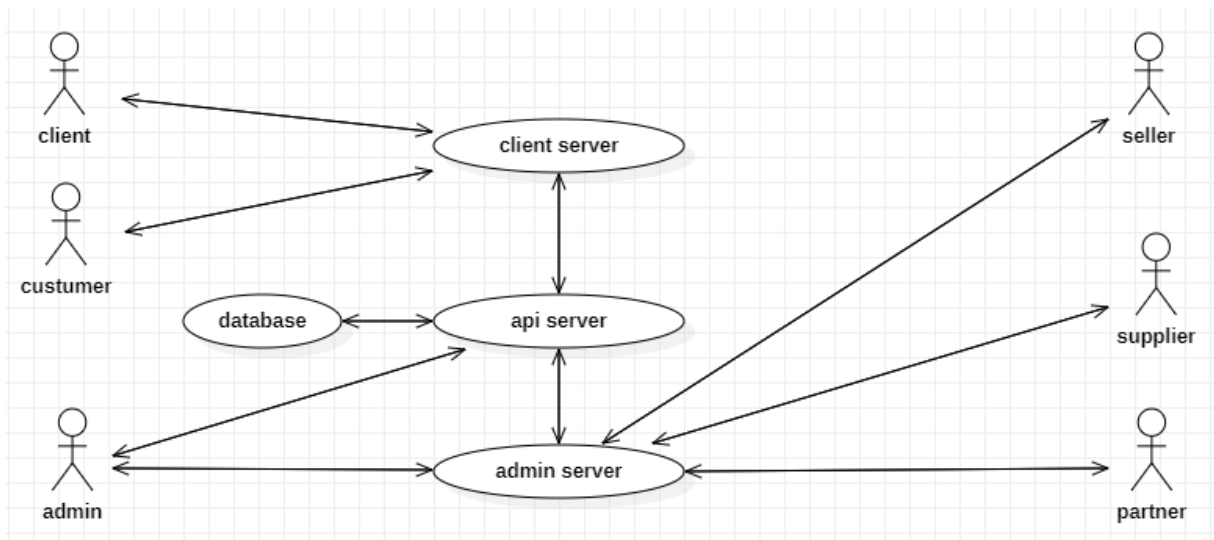
1. SƠ ĐỒ TỔNG QUÁT



Hình 1 sơ đồ tổng quát

- Server API: là máy chủ/nhóm máy chủ kết nối trực tiếp với database để tương tác với dữ liệu, xử lý các yêu cầu, cung cấp tài nguyên cho toàn hệ thống thông qua các api. Người quản trị chính có thể chỉnh sửa cơ bản thông qua view hỗ trợ, nhưng ít bị ràng buộc, khó thao tác, dễ gây lỗi hệ thống. Nên view trên máy chủ này chủ yếu là xem/theo dõi là chính. Khi chỉnh sửa, yêu cầu người quản trị phải có kiến thức và hiểu rõ về hệ thống. Đây là máy chủ chỉ sử dụng tài khoản quản trị mới truy cập được. Khi gặp vấn đề khó khắc phục, các vấn đề đều ảnh hưởng với toàn bộ hệ thống. Vì vậy cần theo dõi, giám sát, backup dữ liệu thường xuyên trong quá trình hoạt động.
- Server admin: là máy chủ/nhóm máy chủ cung cấp các chức năng quản lý thông qua view cho người dùng với các ràng buộc chặt chẽ và hoàn thiện hơn máy chủ API. Đây là máy chủ được sử dụng chính khi muốn quản lý cả hệ thống. Khi truy cập, hệ thống sẽ dựa trên các quyền người dùng mà cung cấp các chức năng cụ thể. Khi máy chủ bị tấn công/vấn đề thì dễ dàng khắc phục (reload lại server => kiểm tra lại lỗi sau) và ít ảnh hưởng tới toàn hệ thống.
- Server client: là máy chủ/nhóm máy chủ quản lý tương tác chính với người dùng /khách hàng. Đây là máy chủ sẽ được sử dụng/ truy cập nhiều nhất nhưng ít được bảo mật nhất trong toàn bộ hệ thống (dễ bị tấn công). Máy chủ phải xử lý nhiều tác vụ cùng lúc nhưng yêu cầu độ trễ thấp, tính liên tục cao, cũng như phải dễ khắc phục nếu xảy ra vấn đề/tấn công, dễ dàng mở rộng, nâng cấp, dễ dàng tương tác/truy cập. Và khi bị tấn công/vấn đề thì hạn chế thấp nhất ảnh hưởng tới trải nghiệm của người dùng (khắc phục nhanh) và hoạt động của toàn bộ hệ thống (cách ly/tạm dừng khi bị tấn công/vấn đề).

2. SƠ ĐỒ TƯƠNG TÁC TỔNG QUÁT



Hình 2 Sơ đồ tương tác tổng quát

Người dùng hệ thống sẽ được chia làm ba nhóm chính:

- Nhóm client: tương tác với hệ thống thông qua server client:

Client: khách hàng tham quan/vãng lai (không có tài khoản)

Customer: khách hàng thân thiết/tiềm năng (đã/sẽ có tài khoản)

- Nhóm supadmin: tương tác với hệ thống thông qua server admin:

Seller: người bán hàng

Supplier: nhà cung cấp hàng hóa

Partner: đối tác/ người hỗ trợ quản lý trang

- Nhóm admin: tương tác qua server api và server admin

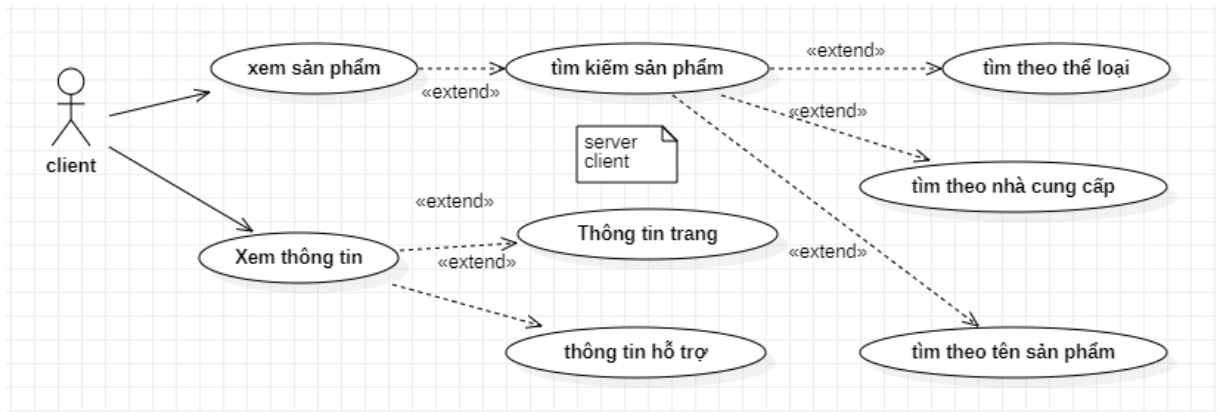
Admin: người quản trị chính của hệ thống, quản lý tất cả thông tin/chức năng của hệ thống

- ❖ Phân cấp quyền: Admin > Partner > Supplier > Seller > Customer > Client

Phân cấp lớn hơn thì sẽ bao gồm các quyền của phân cấp nhỏ hơn nó.

3. CHỨC NĂNG CỦA CÁC LOẠI NGƯỜI DÙNG

Khách tham quan (không có tài khoản)



Hình 3 sơ đồ chức năng khách tham quan

- Xem sản phẩm:

+ Xem tổng quan với một danh sách các sản phẩm với các thông tin tóm tắt:

- Tên sản phẩm
- Thẻ loại
- Số lượng bán
- Giá bán
- Giá khuyến mãi
- Tỷ lệ khuyến mãi
- Người bán
- Hình ảnh sản phẩm

+ Xem chi tiết sản phẩm:

❖ Bao gồm các thông tin tổng quan:

- Nhà cung cấp
- Thông tin giới thiệu (do nhà cung cấp giới thiệu)
- Các thông tin liên quan (do người bán cung cấp)

- Tìm kiếm sản phẩm:

+ Tìm kiếm theo thẻ loại: người dùng có thể chọn thẻ loại thông qua tag thẻ loại để lọc lấy các sản phẩm cùng thẻ loại.

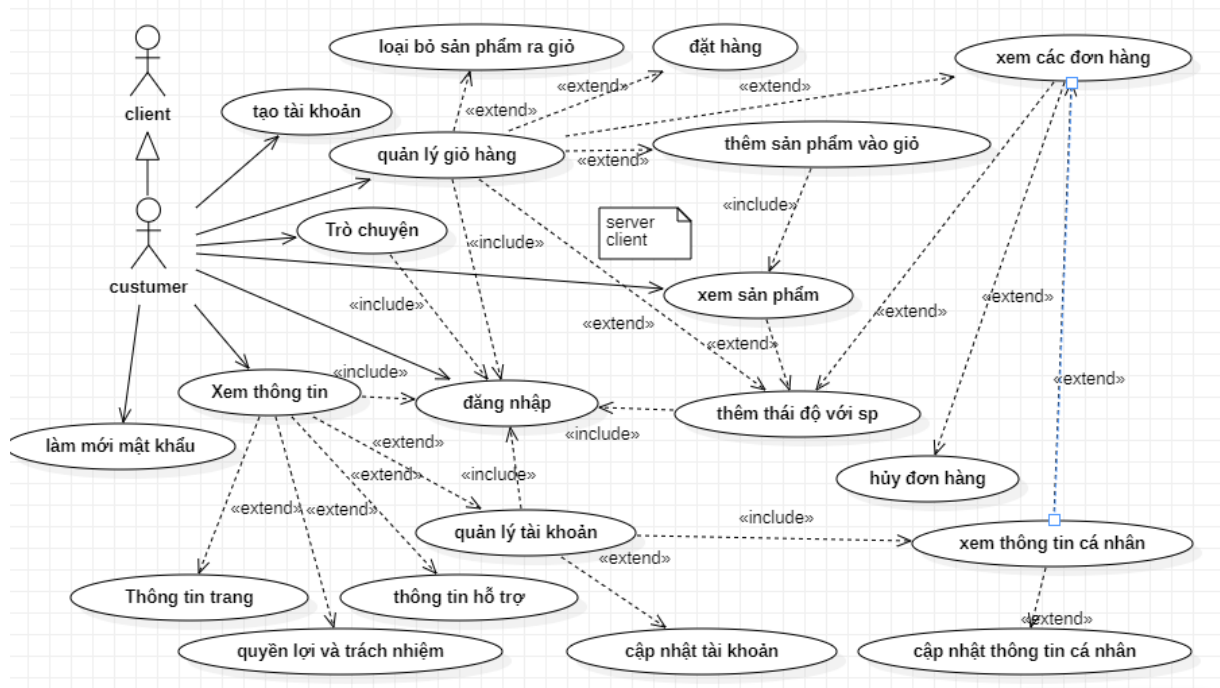
+ Tìm kiếm theo nhà cung cấp: người dùng có thể chọn nhà cung cấp thông qua tag nhà cung cấp để lọc lấy các sản phẩm cùng nhà cung cấp.

+ Tìm kiếm theo tên sản phẩm: người dùng nhập một phần/ toàn bộ tên sản phẩm qua thanh tìm kiếm để tìm kiếm một/ nhiều sản phẩm giống hoặc có chứa các từ/ký tự do người dùng cung cấp.

- Xem thông tin: xem các thông tin về trang

Xem thông tin hỗ trợ: xem các thông tin liên quan nếu cần liên hệ hỗ trợ

Khách hàng tiềm năng (đã đăng nhập)



Hình 4 sơ đồ chức năng khách hàng tiềm năng

❖ Bao gồm các chức năng của khách tham quan.

- Tạo tài khoản: Tạo mới một tài khoản người dùng cho khách mới
 - Username
 - Email
 - Password
 - Password confirm
- Làm mới mật khẩu: Lấy lại mật khẩu cho người dùng quên mật khẩu
 - Email
- Đăng nhập: Đăng nhập vào trang để phân biệt các người dùng khác nhau

+ Đăng nhập với username

- Username
- Password

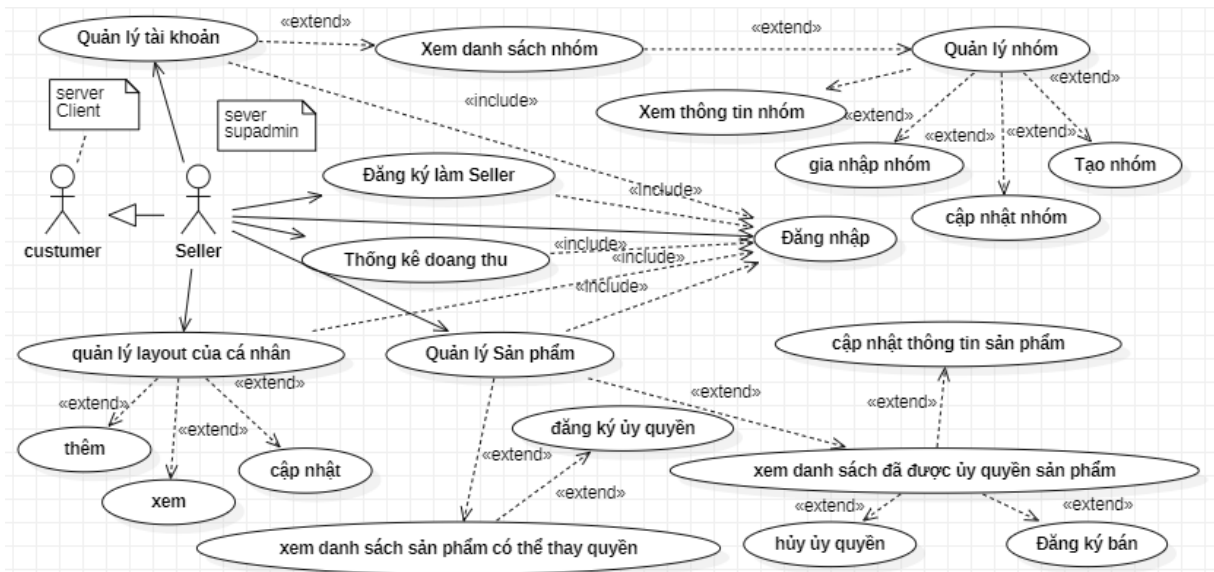
+ Đăng nhập với Email

- Email
- Password

- Trò chuyện: yêu cầu đăng nhập, liên hệ/ trao đổi tin nhắn
 - Tên người dùng
 - Nội dung tin nhắn
 - Thời gian tin nhắn
 - Trạng thái tin nhắn
 - Các thái độ với tin nhắn
 - Tên nhóm
 - Quyền hạn
- + Gửi tin nhắn
- + Thu hồi tin nhắn (nếu có quyền)
- + Xóa tin nhắn (nếu có quyền)
- + Thêm thái độ với tin nhắn
- + Gửi lại (nếu gửi thất bại)
- + Chia sẻ tin nhắn
- + Tạo phòng/nhóm
- + Thay đổi phòng/nhóm (nếu có quyền)
- + Xóa phòng/nhóm (nếu có quyền)
- + Thêm thành viên
- + Đẩy thành viên ra khỏi phòng (nếu có quyền)
- + Chặn
 - Thêm thái độ với sản phẩm: yêu cầu đăng nhập
 - Thái độ với sản phẩm
- + Thích sản phẩm : thông qua icon trên giao diện thông tin sản phẩm
- + Ghét sản phẩm: yêu cầu đã đặt mua sản phẩm, thông qua icon trên giao diện sản phẩm trong lịch sử đơn hàng

- Quản lý giỏ hàng: yêu cầu đăng nhập
- + Sản phẩm
 - Tên sản phẩm
 - Số lượng muốn đặt hàng
 - Giá bán
 - Người bán
 - Hình ảnh sản phẩm
 - Thái độ với sản phẩm
- + Thông tin đơn hàng
 - Tên người nhận
 - Địa chỉ giao hàng
 - Số điện thoại liên hệ khi giao
 - Email liên hệ trong quá trình giao
 - Yêu cầu với đơn hàng
 - Tổng số tiền phải thanh toán nếu nhận hàng
- + Thêm sản phẩm: thông qua số lượng sản phẩm
- + Loại bỏ sản phẩm ra khỏi giỏ hàng
- + Xem lịch sử đơn hàng: xem lại tất cả đơn hàng
 - Xem lịch sử đơn hàng
 - ❖ Bao gồm các thuộc tính của đơn hàng.
 - Trạng thái đơn hàng
 - Thời gian tạo đơn hàng
 - Thời gian cập nhật trạng thái
 - Xem thông tin:
- + Xem thông tin cơ bản như khách tham quan thì không cần đăng nhập
- + Xem thông tin về quyền lợi và trách nhiệm của người dùng, yêu cầu đăng nhập
- + Quản lý tài khoản: yêu cầu đăng nhập
 - Quản lý tài khoản
- + Cập nhật tài khoản: thay đổi mật khẩu
- + Xem thông tin cá nhân: xem các thông tin về người dùng
 - Họ
 - Tên
 - Địa chỉ
 - Tài khoản
 - Email
 - Số điện thoại
- + Xem lịch sử các đơn hàng của người dùng

Người bán hàng



Hình 5 sơ đồ chức năng người bán hàng

❖ Bao gồm các chức năng của khách hàng trên server client.

Trên server admin

- Thông tin cá nhân:

❖ Bao gồm các thuộc tính thông tin của khách hàng

- Thuộc tính title khi bán hàng
- Thống kê doanh thu: theo ngày/tháng/năm/tất cả
- Loại seller

- Đăng ký làm Seller

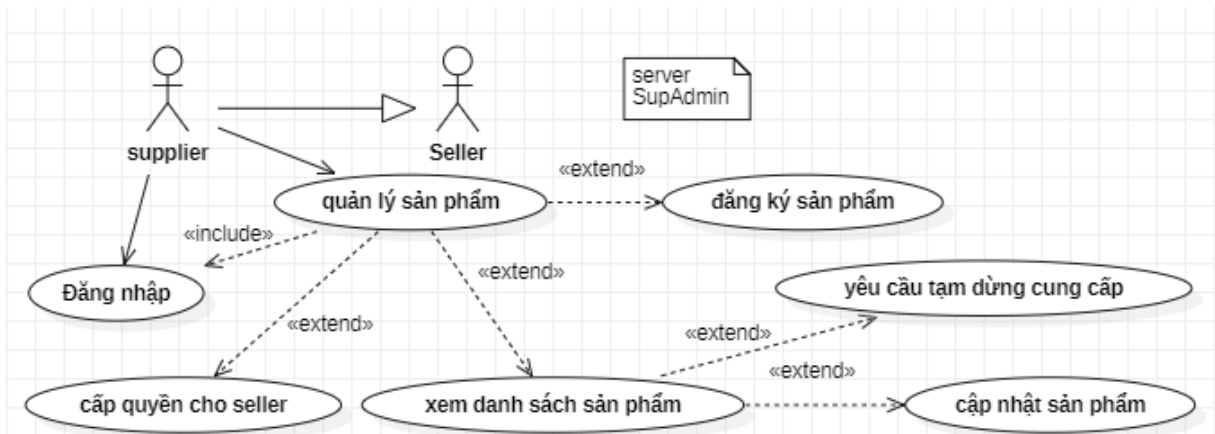
- Quản lý nhóm

❖ Tin nhắn trong nhóm: bao gồm các thuộc tính về tin nhắn

- Tên nhóm
- Thông tin nhóm
- Loại nhóm
- Thành viên nhóm
- Quyền hạn trong nhóm

- Quản lý sản phẩm
 - ❖ Bao gồm các thuộc tính về sản phẩm
 - Số lượng tồn kho
 - Số lượng đã bán
 - Trạng thái với sản phẩm kèm số lượng
- + Xem danh sách có thể đăng ký thay quyền
- + Đăng ký ủy quyền
- + Xem danh sách đã được ủy quyền
- + Cập nhật thông tin các sản phẩm đã được ủy quyền
- + Hủy ủy quyền
- + Đăng ký bán
 - Quản lý layout của cá nhân(theo sản phẩm)
 - Tên layout(name)
 - Phong cách thể hiện tên(name_stype)
 - Tiêu đề layout(title)
 - Phong cách thể hiện tiêu đề(title_stype)
 - Trạng thái layout(show)
 - Chế độ (active)
 - Mức độ ưu tiên
 - Mô tả
 - Phong cách thể hiện mô tả
 - Người tạo
 - Phong cách thể hiện layout
 - Thuộc quyền quản lý của layout nào
 - Loại layout
 - Thời gian chuyển cảnh(đối với layout là carousel)
 - Ảnh nền
 - Thuộc sản phẩm nào
- + Xem danh sách/chi tiết layout
- + Thêm layout
- + Thay đổi
- + Xóa

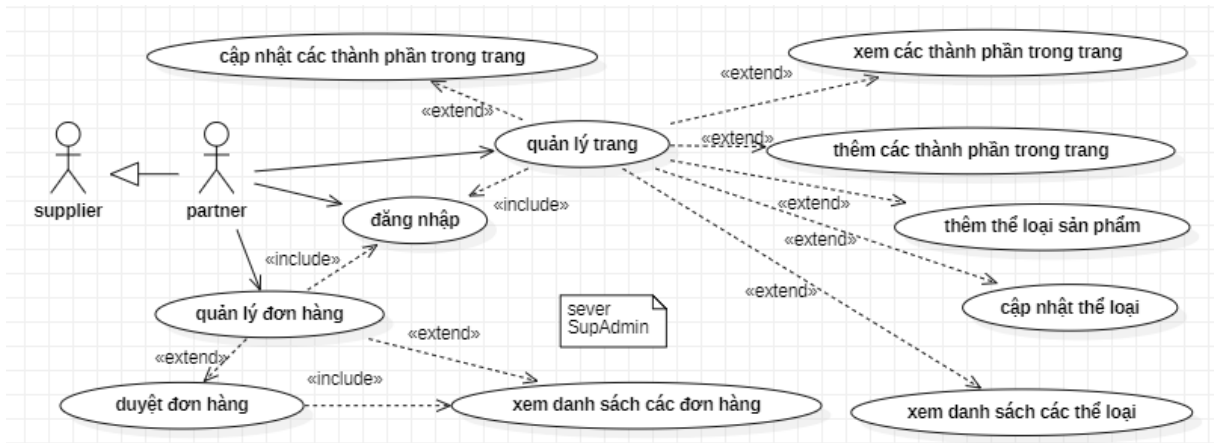
Nhà cung cấp



Hình 6 sơ đồ chức năng nhà cung cấp

- ❖ Bao gồm các quyền của seller
 - Thông tin cá nhân
 - Tên gọi khi là nhà cung cấp
 - Loại nhà cung cấp
 - Tổng doanh thu theo ngày/tháng/năm/tổng
 - Quản lý sản phẩm
 - ❖ Bao gồm các thuộc tính sản phẩm
- + Xem danh sách các sản phẩm cá nhân/tất cả
 - + Đăng sản phẩm mới
 - + Cập nhật các sản phẩm cá nhân đã đăng
 - + Yêu cầu tạm dừng cung cấp với các sản phẩm cá nhân
 - + Xem danh sách các seller đăng ký ủy quyền với sản phẩm
 - + Cấp quyền cho seller
 - Xem danh sách sản phẩm
 - + Lọc theo loại sản phẩm
 - + Lọc theo trạng thái sản phẩm: tồn kho/trao quyền/đã bán,...
 - + Tìm kiếm theo tên
 - Đăng ký sản phẩm: thêm sản phẩm mới
 - Cập nhật sản phẩm: thay đổi các thuộc tính sản phẩm
 - Yêu cầu tạm dừng cung cấp
 - Xem danh sách các seller đăng ký ủy quyền
 - + Lọc theo trạng thái (đang chờ, đã cấp, thu hồi)
 - + Tìm kiếm theo tên seller
 - + Tìm kiếm theo tên sản phẩm
 - Cấp quyền cho seller

Đối tác (hỗ trợ quản lý trang)

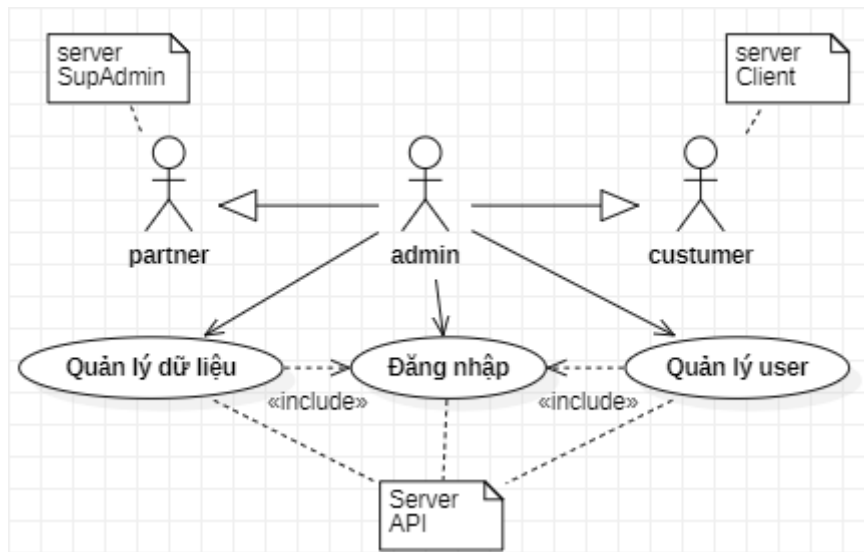


Hình 7 sơ đồ chức năng đối tác

- ❖ Bao gồm các quyền của nhà cung cấp(supplier)
 - Quản lý sản phẩm
- ❖ Bao gồm các thuộc tính sản phẩm
 - Tên thể loại(name)
 - Phong cách thể hiện tên(name_stype)
 - Tiêu đề thể loại(title)
 - Phong cách thể hiện tiêu đề(title_stype)
 - Trạng thái (show)
 - Chế độ (active)
 - Đối tượng quản lý nó
- + Xem danh sách thể loại sản phẩm
- + Thêm thể loại sản phẩm
- + Cập nhật thể loại sản phẩm
 - Xem danh sách thể loại sản phẩm
 - Thêm thể loại sản phẩm
 - Cập nhật thể loại sản phẩm
 - Quản lý đơn hàng
- ❖ Bao gồm tất cả thông tin sản phẩm
 - Mã đơn hàng
 - Số lượng theo sản phẩm
 - Giá theo sản phẩm
 - Trạng thái đơn hàng
 - Ngày tạo
 - Ngày cập nhật
 - Người cập nhật sau cùng
 - Người đặt hàng
 - Người nhận hàng
 - Địa chỉ nhận hàng
 - Số điện thoại liên hệ khi nhận
 - Email nhận thông báo về đơn hàng

- Yêu cầu nếu có
- + Xem danh sách các đơn hàng
- + Duyệt đơn hàng
 - Xem danh sách các đơn hàng
 - Duyệt đơn hàng
 - Quản lý trang
 - Tên trang(name)
 - Phong cách thể hiện tên(name_stype)
 - Tiêu đề trang (title)
 - Phong cách thể hiện tiêu đề(title_stype)
 - Trạng thái (show)
 - Chế độ (active)
 - Phong cách thể hiện trang
- + Thêm trang
- + Cập nhật trang
 - Thêm trang
 - Cập nhật trang
 - Quản lý menu
 - Tên menu (name)
 - Phong cách thể hiện tên(name_stype)
 - Tiêu đề menu (title)
 - Phong cách thể hiện tiêu đề(title_stype)
 - Trạng thái (show)
 - Chế độ (active)
 - Phong cách thể hiện menu
 - Mức độ ưu tiên
 - Thuộc quyền quản lý của đối tượng
 - Liên kết với trang
 - Ảnh
- + Thêm menu
- + Cập nhật menu
 - Thêm menu
 - Cập nhật menu
 - Xem các thành phần trong trang
 - ❖ Bao gồm các thuộc tính layout
 - Thuộc quyền quản lý trang
 - Thuộc quyền quản lý sản phẩm
- + Thêm các thành phần trong trang
- + Cập nhật các thành phần trong trang
 - Thêm các thành phần trong trang
 - Cập nhật các thành phần trong trang

Người quản lý (admin)



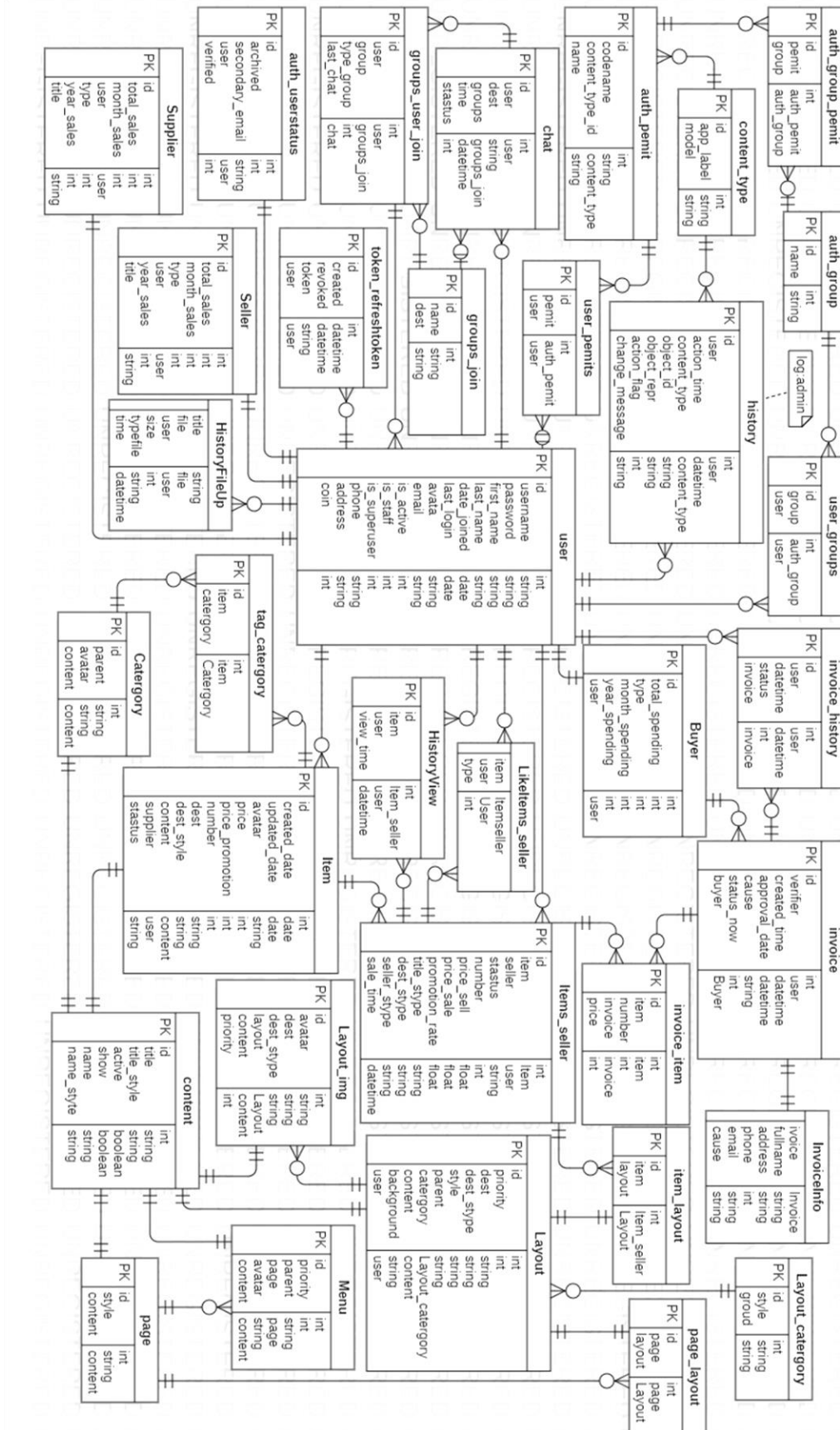
Hình 8 sơ đồ chức năng người quản trị

❖ Bao gồm các quyền của tất cả với server admin và server client

Trên server API

- Quản lý user
- + Cấp quyền cho các user
- + Xóa các user
- + Thêm các user
- + Cập nhật các thông tin cho các user
 - Quản lý tất cả dữ liệu
- + Xem các dữ liệu
- + Xem các lịch sử hoạt động
- + Cập nhật các dữ liệu

4. SƠ ĐỒ CƠ SỞ DỮ LIỆU



Hình 9 sơ đồ cơ sở dữ liệu

Bảng user:

Thông tin người dùng

Tên cột	Kiểu	Mô tả
id	int	Khóa chính
username	string	Tên đăng nhập
password	string	Mật khẩu
first_name	string	Họ người dùng
last_name	string	Tên người dùng
date_joined	datetime	Ngày tạo
last_login	datetime	Thời gian đăng nhập gần nhất
avata	file	ảnh đại diện, lưu ảnh vào thư mục, đường dẫn vào csdl
email	string	Email chính của người dùng, dùng để đăng nhập/nhận mail
is_active	int	Trạng thái hoạt động
is_staff	int	Có phải nhân viên
is_superuser	int	Có phải quản trị viên
phone	string	Số điện thoại
address	string	Địa chỉ liên hệ
coin	int	Số xu trong hệ thống

Bảng auth_group

Nhóm người dùng để phân quyền trong hệ thống

Cột	Kiểu	Mô tả
id	int	Khóa chính
name	string	Tên nhóm

Bảng user_groups

Người dùng tham gia nhóm phân quyền

Cột	Kiểu	Mô tả
Id	Int	Khóa chính
user	user	Thành viên tham gia, liên kết bảng user
group	auth_gruop	Nhóm phân quyền người dùng, liên kết với bảng auth_group

Bảng content_type

Bảng tham chiếu các đối tượng/bảng trong cơ sở dữ liệu trên toàn bộ hệ thống, dữ liệu sẽ được hệ thống tự động khởi tạo khi chạy

Cột	Kiểu	Mô tả
id	int	Khóa chính
app_label	string	Tên của app
model	string	Tên bảng trong model

Bảng auth_pemit

Bảng quyền tham chiếu trên đối tượng trong hệ thống, dữ liệu sẽ được hệ thống tự động khởi tạo khi chạy

Cột	Kiểu	Mô tả
Id	Int	Khóa chính
codename	string	Tên chức năng: “(add/change/delete)_” +”Tên của bảng trong model”
content_type_id	content_type	Đối tượng tham chiếu trong model, liên kết với bảng content_type

Bảng auth_group_pemit

Bảng phân quyền của nhóm người dùng

Cột	Kiểu	Mô tả
id	int	Khóa chính
pemit	auth_pemit	Quyền tham chiếu, liên kết với bảng auth_pemit
group	auth_group	Nhóm được áp dụng, liên kết với bảng auth_group

Bảng history(log admin):

Lịch sử hành động trên server API

Cột	Kiểu	Mô tả
Id	int	Khóa chính
action_time	datetime	Thời gian thao tác
user	user	Người dùng được áp dụng, liên kết với bảng user
content_type	content_type	Loại thao tác, liên kết với bảng content_type
object_id	string	Id đối tượng tương tác
object_repr	string	Đối tượng tương tác
action_flag	int	Hành động (1: add, 2: change, 3: delete)
change_message	string	Nội dung thay đổi

Bảng history:

Lịch sử hành động thông qua API do server API cung cấp

Cột	Kiểu	Mô tả
Id	int	Khóa chính
action_time	datetime	Thời gian thao tác
user	user	Người dùng được áp dụng, liên kết với bảng user
content_type	content_type	Loại thao tác, liên kết với bảng content_type
object_id	string	Id đối tượng tương tác
object_repr	string	Đối tượng tương tác
action_flag	int	Hành động (1: add, 2: change, 3: delete)
change_message	string	Nội dung thay đổi

Bảng HistoryFileUp

Bảng quản lý file được upload bằng API do server API cung cấp

Cột	Kiểu	Mô tả
Id	int	Khóa chính
user	user	Người upload file, liên kết với bảng user
title	string	Tên file
file	file	File lưu trữ và folder, cơ sở dữ liệu lưu đường dẫn
size	int	Kích thước của file
typefile	string	Loại file
time	datetime	Thời gian upload file

Bảng token_refreshtoken

Quản lý mã định danh để hệ thống phân biệt và cấp quyền cho người dùng (không cần đăng nhập)

Cột	Kiểu	Mô tả
id	int	Khóa chính
created	datetime	Thời gian tạo
user	user	Người dùng được áp dụng, liên kết với bảng user
revoked	datetime	Thời gian hết hiệu lực

Bảng auth_userstatus

Bảng quản lý trạng thái kích hoạt của người dùng

Cột	Kiểu	Mô tả
Id	int	Khóa chính
archived	int	Trạng thái lưu trữ
user	user	Người dùng được áp dụng, liên kết với bảng user
secondary_email	string	Email phụ(email thứ 2 có thể dùng thay đổi email chính nếu quên email chính)
verified	int	Trạng thái xác nhận, 0: chưa xác nhận, 1: xác nhận, mặc định là 0

Bảng groups_join

Bảng quản lý nhóm liên hệ

Cột	Kiểu	Mô tả
id	int	Khóa chính
name	string	Tên nhóm
dest	string	Mô tả nhóm

Bảng chat

Bảng quản lý nội dung tin nhắn

Cột	Kiểu	Mô tả
id	int	Khóa chính
user	user	Người tạo
dest	string	Nội dung tin nhắn
groups	groups_join	Thuộc về nhóm, liên kết với bảng groups_join
time	datetime	Thời gian tin nhắn gửi
stastus	int	Trạng thái tin nhắn

Bảng groups_user_join

Bảng quản lý thành viên tham gia nhóm liên hệ

Cột	Kiểu	Mô tả
id	int	Khóa chính
user	user	Thành viên tham gia, liên kết với bảng user
type_group	int	Loại nhóm
group	groups_join	Thuộc về nhóm, liên kết với bảng groups_join
last_chat	chat	Tin nhắn cuối cùng đã đọc trong nhóm của người dùng, liên kết với bảng chat

Bảng Supplier

Bảng quản lý nhà cung cấp

Cột	Kiểu	Mô tả
id	int	Khóa chính
user	user	Người dùng được tham chiếu tới, liên kết với bảng user
title	string	Tên hiển thị khi là nhà cung cấp
total_sales	int	Tổng doanh thu
month_sales	int	Tổng doanh thu trong tháng
year_sales	int	Tổng doanh thu trong năm
type	int	Loại nhà cung cấp

Bảng Buyer

Bảng quản lý người mua hàng

Cột	Kiểu	Mô tả
id	int	Khóa chính
user	user	Người dùng được tham chiếu tới, liên kết với bảng user
total_spending	int	Tổng chi
month_spending	int	Tổng chi trong tháng
year_spending	int	Tổng chi trong năm
type	int	Loại khách hàng

Bảng page

Bảng quản lý trang

Cột	Kiểu	Mô tả
id	int	Khóa chính
style	string	Phong cách hiển thị trang
content	content	Kê thừa lại các thuộc tính trong bảng content, liên kết với bảng content

Bảng Seller

Bảng quản lý người bán hàng

Cột	Kiểu	Mô tả
id	int	Khóa chính
user	user	Người dùng được tham chiếu tới, liên kết với bảng user
title	string	Tên hiển thị khi là người bán
total_sales	int	Tổng doanh thu
month_sales	int	Tổng doanh thu trong tháng
year_sales	int	Tổng doanh thu trong năm
type	int	Loại người bán

Bảng content

Bảng thuộc tính chung cho các bảng sau kế thừa

Cột	Kiểu	Mô tả
id	int	Khóa chính
title	string	Tiêu đề đối tượng
title_style	string	Phong cách hiển thị tiêu đề
active	boolean	Trạng thái hoạt động
show	boolean	Trạng thái hiển thị
name	string	Tên nhóm
name_style	string	Phong cách hiển thị tên

Bảng Layout_category

Bảng quản lý kiểu hiển thị của các thành phần trong trang

Cột	Kiểu	Mô tả
id	int	Khóa chính
style	string	Phong cách hiển thị
group	string	Nhóm hiển thị

Bảng Menu

Bảng quản lý menu

Cột	Kiểu	Mô tả
id	int	Khóa chính
parent	string	Thuộc quyền quản lý của đối tượng
content	content	Kế thừa lại các thuộc tính trong bảng content, liên kết với bảng content
priority	int	Độ ưu tiên của đối tượng
avatar	file	ảnh đại diện, lưu ảnh vào thư mục, đường dẫn vào csdl
page	page	Trang tham chiếu tới, liên kết với bảng page

Bảng user_pemits

Bảng phân quyền của cá nhân người dùng

Cột	Kiểu	Mô tả
id	int	Khóa chính
permit	auth_permit	Quyền tham chiếu, liên kết với bảng auth_permit
user	user	Người dùng được áp dụng, liên kết với bảng user

Bảng Layout

Bảng quản lý các thành phần trong trang

Cột	Kiểu	Mô tả
Id	int	Khóa chính
parent	string	Thuộc quyền quản lý của đối tượng
content	content	Kế thừa lại các thuộc tính trong bảng content, liên kết với bảng content
priority	int	Độ ưu tiên của đối tượng
background	file	ảnh nền, lưu ảnh vào thư mục, đường dẫn vào csdl
dest	string	Mô tả đối tượng
dest_stype	string	Phong cách hiển thị của mô tả
stype	string	Phong cách hiển thị của đối tượng
category	layout_category	Loại hiển thị, liên kết với bảng Layout_category
user	user	Người khởi tạo, liên kết với bảng user

Bảng Layout_img

Bảng quản lý các hình ảnh trong layout

Cột	Kiểu	Mô tả
Id	int	Khóa chính
parent	string	Thuộc quyền quản lý của đối tượng
content	content	Kế thừa lại các thuộc tính trong bảng content, liên kết với bảng content
priority	int	Độ ưu tiên của đối tượng
avatar	file	ảnh đại diện, lưu ảnh vào thư mục, đường dẫn vào csdl
dest	string	Mô tả đối tượng
dest_stype	string	Phong cách hiển thị của mô tả
stype	string	Phong cách hiển thị của đối tượng
catergory	layout_catery	Loại hiển thị, liên kết với bảng Layout_catery
user	user	Người khởi tạo, liên kết với bảng user
layout	layout	Thuộc quyền quản lý của đối tượng, liên kết với bảng layout

Bảng Item

Bảng quản lý thông tin sản phẩm

Tên cột	Kiểu	Mô tả
id	int	Khóa chính
created_date	datetime	Ngày tạo
avata	file	Ảnh đại diện, lưu ảnh vào thư mục, đường dẫn vào csdl
updated_date	datetime	Ngày cập nhật
price	int	Giá
price_promotion	int	Giá khuyến mãi
number	Int	Số lượng
dest	string	Mô tả sản phẩm
dest_style	string	Phong cách hiển thị mô tả
content	content	Kế thừa tất cả thuộc tính từ bảng content, liên kết với bảng content
supplier	user	Nhà cung cấp, liên kết với bảng user
stastus	string	Trạng thái sản phẩm

Bảng invoice_history

Bảng theo dõi trạng thái hóa đơn

Cột	Kiểu	Mô tả
Id	Int	Khóa chính
user	user	Người duyệt, liên kết với bảng user
datetime	datetime	Thời gian duyệt
cause	string	Lý do
status	int	Trạng thái
invoice	invoice	Hóa đơn tham chiếu, liên kết với bảng hóa đơn

Bảng invoice

Bảng quản lý hóa đơn

Cột	Kiểu	Mô tả
Id	Int	Khóa chính
verifier	user	Người duyệt, liên kết với bảng user
created_time	datetime	Ngày tạo
approval_date	datetime	Thời gian duyệt
cause	string	Lý do
status_now	int	Trạng thái hiện tại
Buyer	Buyer	Người mua hàng, liên kết với bảng user

Bảng tag_category

Bảng quản lý thể loại sản phẩm trong một sản phẩm

Cột	Kiểu	Mô tả
Id	Int	Khóa chính
item	item	Hiện thị trong sản phẩm, liên kết với bảng sản phẩm
Category	Category	đối tượng hiển thị, liên kết với bảng Category

Bảng LikeItems_seller

Bảng quản lý thái độ đối với sản phẩm đã bán

Cột	Kiểu	Mô tả
id	int	Khóa chính
item	item_seller	Sản phẩm bán ra, liên kết với bảng item_seller
user	user	Người tương tác, liên kết với bảng user
type	int	Loại tương tác

Bảng Items_seller

Bảng quản lý sản phẩm bán ra

Tên cột	Kiểu	Mô tả
id	int	Khóa chính
item	item	Sản phẩm, liên kết với bảng item
seller	user	Người bán, liên kết với bản user
number	int	Số lượng
stastus	string	Trạng thái sản phẩm
price_sell	float	Giá bán
price_sale	float	Giá giảm giá
promotion_rate	float	Tỉ lệ giảm giá
title_stype	string	Phong cách hiển thị tiêu đề
dest_stype	string	Phong cách hiển thị mô tả
seller_stype	string	Phong cách hiển thị tên người bán
sale_time	datetime	Thời gian bán

Bảng item_layout

Bảng quản lý layout hiển thị theo sản phẩm

Cột	Kiểu	Mô tả
id	int	Khóa chính
item	item_seller	Sản phẩm bán ra, liên kết với bảng item_seller
layout	layout	đối tượng hiển thị, liên kết với bảng layout

Bảng HistoryView

Bản theo dõi tương tác

Cột	Kiểu	Mô tả
id	int	Khóa chính
item	item_seller	Sản phẩm bán ra, liên kết với bảng item_seller
user	user	Người tương tác, liên kết với bảng user
view_time	datetime	Thời gian tương tác

Bảng page_layout

Bảng quản lý các layout sẽ được hiển thị trong trang nào

Cột	Kiểu	Mô tả
id	int	Khóa chính
page	page	Trang sẽ hiển thị, liên kết với bảng page
layout	layout	đối tượng hiển thị, liên kết với bảng layout

Bảng invoice_item

Bảng quản lý các sản phẩm trong hóa đơn

Tên cột	Kiểu	Mô tả
id	int	Khóa chính
item	itemseller	Sản phẩm, liên kết với bảng itemseller
number	int	Số lượng
invoice	invoice	Thuộc hóa đơn, liên kết với bảng hóa đơn
price	float	Giá

Bảng InvoiceInfo

Bảng quản lý thông tin nhận hàng theo hóa đơn

Tên cột	Kiểu	Mô tả
id	int	Khóa chính
full_name	string	Họ tên người nhận
email	string	Email nhận mail
phone	string	Số điện thoại
address	string	Địa chỉ liên hệ
cause	string	Yêu cầu/lý do

Bảng Category

Bảng quản lý thể loại sản phẩm

Cột	Kiểu	Mô tả
id	int	Khóa chính
parent	String	Thuộc quyền quản lý của đối tượng
content	content	Kế thừa lại các thuộc tính trong bảng content, liên kết với bảng content
avatar	file	Ảnh đại diện, lưu ảnh vào thư mục, đường dẫn vào csdl

III. PHÂN TÍCH CÁC TÀI NGUYÊN VÀ CÁC LUỒNG DỮ LIỆU

1. Các loại ngôn ngữ

- Python:

Python là ngôn ngữ lập trình hướng đối tượng, cấp cao, mạnh mẽ, được tạo ra bởi Guido van Rossum. Python có cú pháp đơn giản và dễ sử dụng, do đó nó trở thành một trong những ngôn ngữ lập trình nhập môn tốt nhất cho người mới bắt đầu.

Python có cấu trúc dữ liệu cao cấp mạnh mẽ và cách tiếp cận đơn giản nhưng hiệu quả đối với lập trình hướng đối tượng. Cú pháp lệnh của Python là điểm cộng vô cùng lớn vì sự rõ ràng, dễ hiểu và linh động, làm cho nó nhanh chóng trở thành một ngôn ngữ lý tưởng để viết script và phát triển ứng dụng trong nhiều lĩnh vực, ở hầu hết các nền tảng.

✚ Tính năng chính của Python:

Ngôn ngữ lập trình đơn giản, dễ sử dụng

Python có cú pháp rất đơn giản, rõ ràng. Nó dễ đọc và viết hơn rất nhiều khi so sánh với những ngôn ngữ lập trình khác như C++, Java, C#. Python làm cho việc lập trình trở nên thú vị, cho phép bạn tập trung vào những giải pháp chứ không phải cú pháp.

Miễn phí, mã nguồn mở

Bạn có thể tự do sử dụng và phân phối Python, thậm chí là dùng nó cho mục đích thương mại. Vì Python là ứng dụng mã nguồn mở, bạn không những có thể sử dụng các phần mềm, chương trình được viết trong Python mà còn có thể thay đổi mã nguồn của nó. Python có một cộng đồng rộng lớn, không ngừng cải thiện nó mỗi lần cập nhật.

Khả năng di chuyển

Các chương trình Python có thể di chuyển từ nền tảng này sang nền tảng khác và chạy nó mà không có bất kỳ thay đổi nào. Nó chạy liền mạch trên hầu hết tất cả các nền tảng như Windows, macOS, Linux.

Khả năng mở rộng và có thể nhúng

Giả sử một ứng dụng đòi hỏi sự phức tạp rất lớn, bạn có thể dễ dàng kết hợp các phần code bằng C, C++ và những ngôn ngữ khác (có thể gọi được từ C) vào code Python. Điều này sẽ cung cấp cho ứng dụng của bạn những tính năng tốt hơn cũng như khả năng scripting mà những ngôn ngữ lập trình khác khó có thể làm được.

Ngôn ngữ thông dịch cấp cao

Không giống như C/C++, với Python, bạn không phải lo lắng những nhiệm vụ khó khăn như quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa,... Khi chạy code Python, nó sẽ tự động chuyển đổi code sang ngôn ngữ máy tính có thể hiểu. Bạn không cần lo lắng về bất kỳ hoạt động ở cấp thấp nào.

Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến

Python có một số lượng lớn thư viện tiêu chuẩn giúp cho công việc lập trình của bạn trở nên dễ thở hơn rất nhiều, đơn giản vì không phải tự viết tất cả code.

- **Jinja2** :

Jinja2 là một ngôn ngữ tạo template cung cấp cho các lập trình viên Python, được tạo ra dựa trên ý tưởng của Django template.

Jinja2 được sử dụng để tạo HTML, XML hoặc các định dạng file khác dựa trên nguyên tắc kết hợp các dữ liệu vào các vị trí đã được đánh dấu trong văn bản.

Các tính năng nổi bật:

Chế độ thực thi hộp cát. Mọi khía cạnh của quá trình thực thi mẫu đều được theo dõi và đưa vào danh sách trắng hoặc danh sách đen một cách rõ ràng, bất cứ điều gì được ưu tiên. Điều này cho phép thực thi các mẫu không đáng tin cậy.

Hệ thống thoát HTML tự động mạnh mẽ để ngăn chặn tập lệnh chéo trang.

Kế thừa mẫu giúp bạn có thể sử dụng cùng một bố cục hoặc bố cục tương tự cho tất cả các mẫu.

Xác định và nhập macro trong các mẫu để dễ dàng sử dụng khi cần

Hiệu suất cao với việc biên dịch kịp thời sang mã byte Python. Jinja2 sẽ dịch các nguồn mẫu của bạn trong lần tải đầu tiên sang mã byte Python để có hiệu suất thời gian chạy tốt nhất.

Biên dịch trước thời hạn tùy chọn

Dễ dàng gỡ lỗi với một hệ thống gỡ lỗi tích hợp các lỗi thời gian chạy và biên dịch mẫu vào hệ thống theo dõi Python tiêu chuẩn.

Các ngoại lệ trở đến đúng dòng trong các mẫu để giúp gỡ lỗi dễ dàng hơn.

Hỗ trợ AsyncIO để tạo mẫu và gọi các hàm không đồng bộ.

Hỗ trợ I18N với Babel.

Cú pháp có thể cấu hình. Chẳng hạn, bạn có thể cấu hình lại Jinja2 để phù hợp hơn với các định dạng đầu ra như LaTeX hoặc JavaScript.

Người trợ giúp thiết kế mẫu. Jinja2 cung cấp một loạt các trình trợ giúp nhỏ hữu ích giúp giải quyết các tác vụ phổ biến trong các mẫu, chẳng hạn như chia chuỗi các mục thành nhiều cột, v.v.

- **HTML:**


Ngôn ngữ đánh dấu siêu văn bản (HTML), HTML là khối dựng cơ bản của hầu hết các trang web. Tất cả các đoạn văn, phần mục, hình ảnh, tiêu đề và văn bản đều được viết bằng HTML.

- **CSS:**

Ngôn ngữ định kiểu theo tầng (CSS), CSS là một ngôn ngữ của các quy tắc định kiểu mà chúng được dùng để định kiểu cho nội dung HTML. Nó có thể sử dụng để thiết kế các phần tử của trang web như màu nền, phông chữ, cột và đường viền.

- **JavaScript:**

Là ngôn ngữ kịch bản phía máy khách .Nó được sử dụng để tạo trang web tương tác.

 Tính năng của JavaScript

Dễ dàng học và sử dụng

Cú pháp của JavaScript được lấy cảm hứng từ ngôn ngữ lập trình Java, rất dễ để học và viết mã. Các nhà phát triển sử dụng JavaScript trong hầu hết các trang web và ứng dụng di động để viết tập lệnh phía máy khách. Node.js cũng đã trở nên hết sức phổ biến đối với việc viết mã backend trong thập kỷ qua. Nhiều nền tảng phát trực tuyến và video phổ biến đã được viết mã bằng Node.js.

Có thể không phụ thuộc vào nền tảng

Không giống như các ngôn ngữ lập trình khác, bạn có thể chèn JavaScript vào bất kỳ trang web nào và sử dụng với nhiều ngôn ngữ và khung phát triển web khác. Sau khi viết, bạn có thể chạy mã JavaScript trên bất kỳ máy nào. Do đó, với JavaScript, việc phát triển ứng dụng có thể không phụ thuộc vào nền tảng.

Giảm tải máy chủ

Bạn có thể sử dụng JavaScript để giảm tải máy chủ và tắc nghẽn mạng vì JavaScript có thể chạy các phép toán logic và thực hiện nhiều công việc của máy chủ trên chính máy khách. Ví dụ: hãy xem xét quá trình điền một biểu mẫu đăng ký. JavaScript nhanh chóng kiểm tra xem bạn đã nhập số có 10 chữ số cho trường điện thoại di động hay chưa. Nếu những yêu cầu này được gửi đến máy chủ, trang của bạn sẽ tải lại mỗi khi có lỗi, làm cho quá trình đăng ký rất chậm chạp và tẻ nhạt.

Cải thiện giao diện người dùng

JavaScript tạo ra các trang web tinh tế giúp thuận tiện trong việc tìm kiếm và xử lý thông tin phức tạp. Các nhà phát triển áp dụng JavaScript để mở rộng chức năng và độ dễ đọc cũng như để tương tác giữa người dùng và trang web hiệu quả hơn.

Hỗ trợ đồng thời

JavaScript có thể chạy song song nhiều tập hợp hướng dẫn khác nhau. Về phần backend, Node.js có thể giải quyết và xử lý các phản hồi của máy chủ được mở rộng quy mô mà không tiêu tốn cùng một lượng băng thông để làm việc đó.

- GraphQL:

Là ngôn ngữ thao tác và truy vấn dữ liệu nguồn mở cho API, cung cấp cho client một cách thức dễ dàng để truy vấn chính xác những gì họ cần, giúp việc phát triển dễ dàng và nhanh chóng hơn.

✚ GraphQL bao gồm 3 điểm đặc trưng bao gồm:

Cho phép client xác định chính xác những dữ liệu gì họ cần, không thừa không thiếu. (Các truy vấn GraphQL luôn trả về kết quả có thể dự đoán được, truy vấn với GraphQL rất nhanh và ổn định vì GraphQL kiểm soát dữ liệu mà nó nhận được chứ không phải máy chủ.)

GraphQL làm cho việc tổng hợp dữ liệu từ nhiều nguồn dễ dàng hơn (Các câu query GraphQL không chỉ có thể truy xuất các thuộc tính của một dữ liệu mà còn làm việc trộn tru với các đối tượng liên kết khác)

Sử dụng một type system để khai báo dữ liệu (API GraphQL được tổ chức theo types và fields, truy cập toàn bộ dữ liệu từ một endpoint duy nhất.).

❖ Trong khi các API REST chúng ta hay dùng phải yêu cầu request từ nhiều URL thì API GraphQL lại có thể lấy tất cả dữ liệu mà ứng dụng của bạn cần trong một request duy nhất.

2. Các loại framework

- Django

Django là một khung web Python cấp cao cho phép phát triển nhanh chóng các trang web an toàn và có thể bảo trì. Được xây dựng bởi các nhà phát triển có kinh nghiệm, Django giải quyết phần lớn rắc rối của quá trình phát triển web, vì vậy bạn có thể tập trung vào việc viết ứng dụng của mình mà không cần phải xây dựng lại bộ khung. Nó là nguồn mở và miễn phí, có một cộng đồng thịnh vượng và tích cực, tài liệu chi tiết và nhiều. Có thể tùy chọn hỗ trợ miễn phí và trả phí.

✚ Các tính năng:

Hoàn thiện

Django tuân theo triết lý "Bao gồm pin" và cung cấp hầu hết mọi thứ mà các nhà phát triển có thể muốn thực hiện "ngoài luồng". Bởi vì mọi thứ bạn cần đều là một phần của một "sản phẩm", tất cả đều hoạt động liền mạch với nhau, tuân theo các nguyên tắc thiết kế nhất quán và có tài liệu mở rộng và cập nhật.

Cầm tay

Django được viết bằng Python, chạy trên nhiều nền tảng. Điều đó có nghĩa là bạn không bị ràng buộc với bất kỳ nền tảng máy chủ cụ thể nào và có thể chạy các ứng dụng của mình trên nhiều phiên bản Linux, Windows và macOS. Hơn nữa, Django được hỗ trợ tốt bởi nhiều nhà cung cấp dịch vụ lưu trữ web, họ thường cung cấp cơ sở hạ tầng và tài liệu cụ thể để lưu trữ các trang web Django.

Linh hoạt

Django có thể (và đã) được sử dụng để xây dựng hầu hết mọi loại trang web — từ hệ thống quản lý nội dung và wiki, cho đến mạng xã hội và trang tin tức. Nó có thể hoạt động với mọi khung phía máy khách và có thể phân phối nội dung ở hầu hết mọi định dạng (bao gồm HTML, nguồn cấp RSS, JSON và XML).

Bên trong, mặc dù nó cung cấp các lựa chọn cho hầu hết mọi chức năng mà bạn có thể muốn (ví dụ: một số cơ sở dữ liệu phổ biến, công cụ tạo khuôn mẫu, v.v.), nhưng nó cũng có thể được mở rộng để sử dụng các thành phần khác nếu cần.

Chắc chắn

Django giúp các nhà phát triển tránh được nhiều lỗi bảo mật phổ biến bằng cách cung cấp một khung đã được thiết kế để "làm những điều đúng đắn" nhằm bảo vệ trang web một cách tự động. Ví dụ: Django cung cấp một cách an toàn để quản lý tài khoản người dùng và mật khẩu, tránh các lỗi phổ biến như đưa thông tin phiên vào cookie ở những nơi dễ bị tấn công (thay vào đó, cookie chỉ chứa khóa và dữ liệu thực tế được lưu trữ trong cơ sở dữ liệu) hoặc lưu trữ trực tiếp mật khẩu chứ không phải là một mật khẩu băm.

Có thể mở rộng

Django sử dụng kiến trúc "shared-nothing" dựa trên thành phần (mỗi phần của kiến trúc độc lập với các phần khác và do đó có thể được thay thế hoặc thay đổi nếu cần). Có sự tách biệt rõ ràng giữa các phần khác nhau có nghĩa là nó có thể mở rộng quy mô để tăng lưu lượng truy cập bằng cách thêm phần cứng ở mọi cấp độ: máy chủ bộ nhớ đệm, máy chủ cơ sở dữ liệu hoặc máy chủ ứng dụng.

Có thể bảo trì và tái sử dụng

Mã Django được viết bằng các nguyên tắc và mẫu thiết kế khuyến khích tạo mã có thể bảo trì và tái sử dụng. Đặc biệt, nó sử dụng nguyên tắc Don't Repeat Yourself (DRY) để không có sự trùng lặp không cần thiết, giảm số lượng mã. Django cũng thúc đẩy việc nhóm các chức năng liên quan thành các "ứng dụng" có thể tái sử dụng và, ở cấp độ thấp hơn, nhóm mã liên quan thành các mô-đun

- Flask

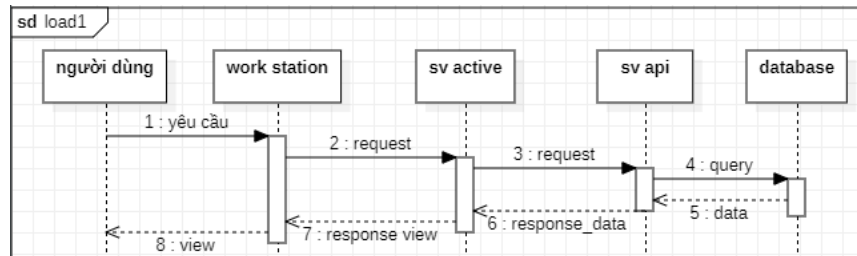
Flask là một micro web framework được viết bằng Python, không yêu cầu tool hay thư viện cụ thể nào. "Micro" không có nghĩa là thiếu chức năng, mà "micro" theo triết lý thiết kế là cung cấp một lõi chức năng "súc tích" nhất cho ứng dụng web nhưng người dùng có thể mở rộng bất cứ lúc nào. Flask luôn hỗ trợ các thành phần tiện ích mở rộng cho ứng dụng như tích hợp cơ sở dữ liệu, xác thực biểu mẫu, xử lý upload, các công nghệ xác thực, template, email, RESTful..., chỉ khác là khi nào bạn muốn thì bạn mới đưa vào thôi. Người dùng có thể tập trung xây dựng web application ngay từ đầu trong một khoảng thời gian rất ngắn và có thể phát triển quy mô của ứng dụng tùy theo yêu cầu.

3. Phương thức load thông tin

- Tầng một:

Thông tin hiển thị ngay khi người dùng yêu cầu load trang

Các thông tin sẽ được máy chủ xử lý thành một response trả về khi người dùng yêu cầu, thường là thông qua lệnh get url

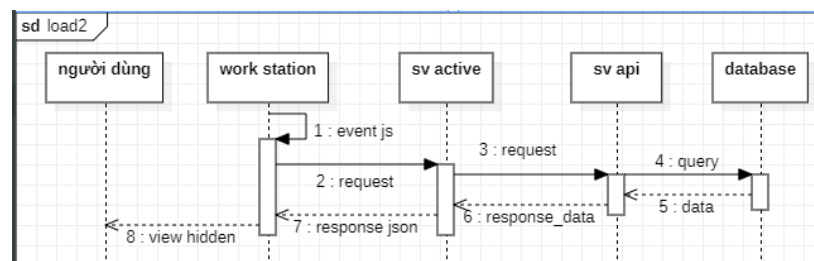


Hình 10 sơ đồ luồng dữ liệu theo tầng 1

- Tầng hai:

Khi các thông tin tầng 1 đã hiển thị xong thì sẽ load các thông tin chưa được hiển thị (thông tin ẩn, thông tin trong popup,...)

Khi các thông tin tầng 1 đã load xong mã lệnh của javascript sẽ bắt sự kiện (window.onload) và thực thi các lệnh load/cập nhật tiếp theo một cách tự động mà không có sự can thiệp của người dùng

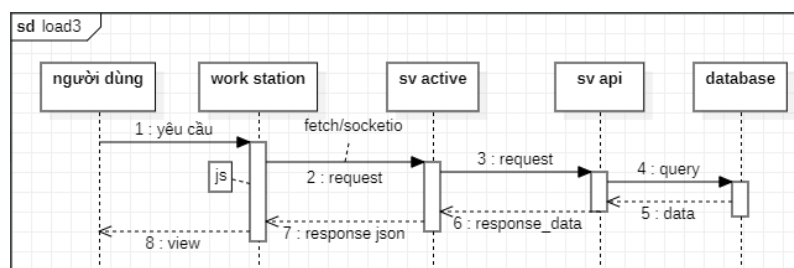


Hình 11 sơ đồ luồng dữ liệu theo tầng 2

- Tầng ba:

Các thông tin sẽ load theo tương tác của người dùng với trang thông qua các event được gọi.

Khi trang đã load xong, người dùng tương tác với các sự kiện thì các lệnh javascript được thực thi và thay đổi dữ liệu mà không cần load lại toàn bộ trang



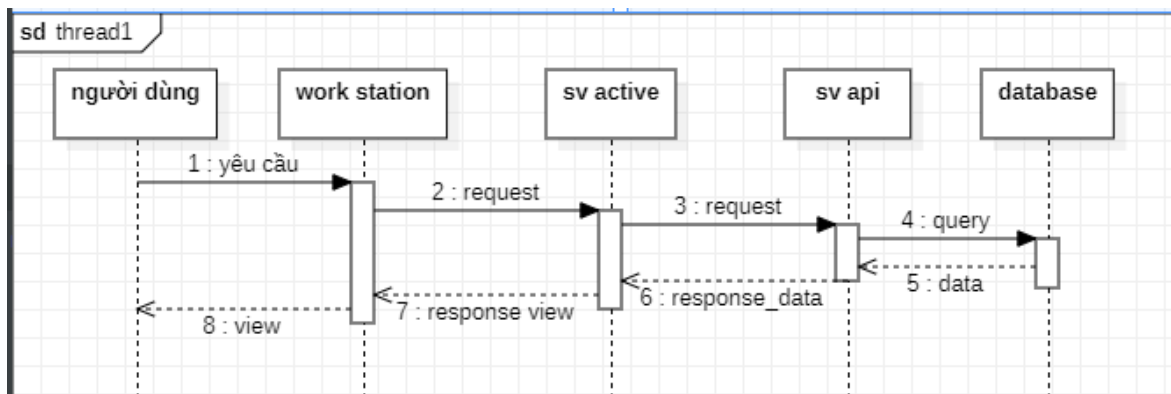
Hình 12 sơ đồ luồng dữ liệu theo tầng 3

4. Các trình tự giải quyết tương tác của người dùng

Phương thức 1:

Người dùng gửi yêu cầu đến máy chủ thông qua giao diện web trên máy trạm=> máy chủ nhận request => máy chủ sẽ gọi tới hàm xử lý tương ứng => gửi request lấy dữ liệu tới server API => server API xử lý yêu cầu => gửi query lấy dữ liệu tới cơ sở dữ liệu=> cơ sở dữ liệu gửi dữ liệu về server API=> server API xử lý rồi gửi response phản hồi lại server gửi=> server gửi sẽ xử lý dữ liệu vừa phản rồi và trả về response cho máy trạm=> máy trạm phản hồi dạng view cho người dùng

- Vd: Truy cập một trang thông qua menu
- Sơ đồ minh họa:

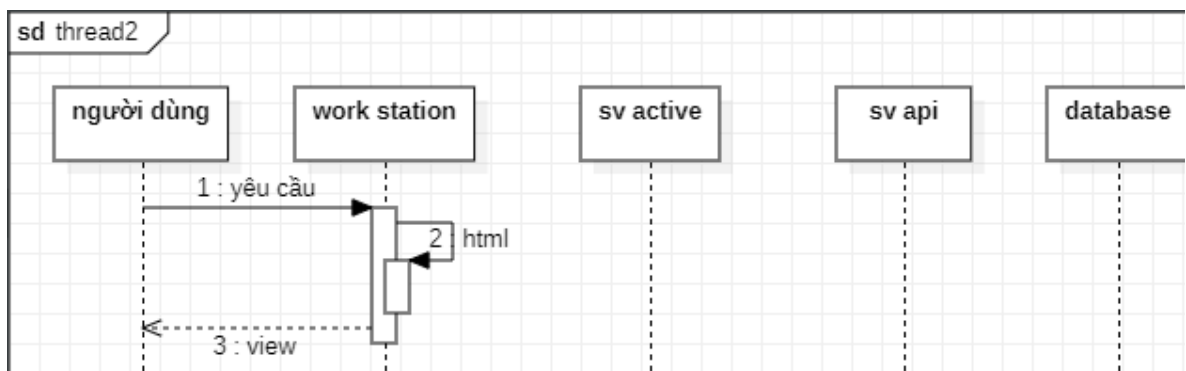


Hình 13 sơ đồ luồng dữ liệu theo phương thức 1

Phương thức 2:

Người dùng gửi yêu cầu thông qua giao diện web trên máy trạm=> máy trạm xử lý và phản hồi lại nếu cấu trúc không chính xác thông qua ràng buộc trên html

- VD: Nhập các giá trị trong một số ô input
- Sơ đồ minh họa

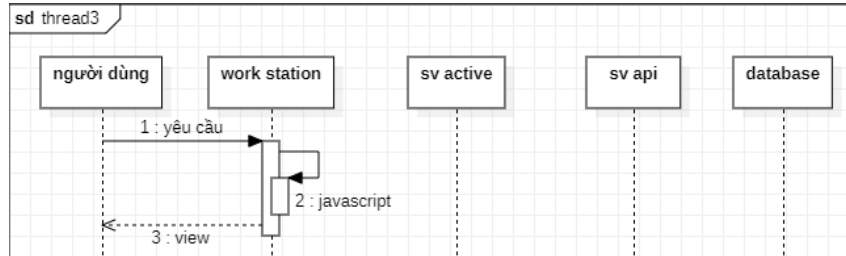


Hình 14 sơ đồ luồng dữ liệu theo phương thức 2

Phương thức 3:

Người dùng gửi yêu cầu thông qua giao diện trên máy trạm=> máy trạm xử lý, kiểm tra và phản hồi thông qua các hàm javascript

- VD: Mở/đóng một số popup
- Sơ đồ minh họa

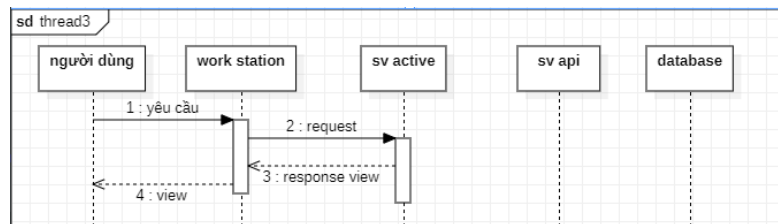


Hình 15 sơ đồ luồng dữ liệu theo phương thức 3

Phương thức 4:

Người dùng gửi yêu cầu thông qua giao diện trên máy trạm=> máy trạm kiểm tra => gửi request lên máy chủ=> máy chủ xử lý và phản hồi lại máy trạm=> máy trạm phản hồi lại người dùng qua view

- VD: Tương tác với một số trang mà không qua menu / không hợp lệ
- Sơ đồ minh họa

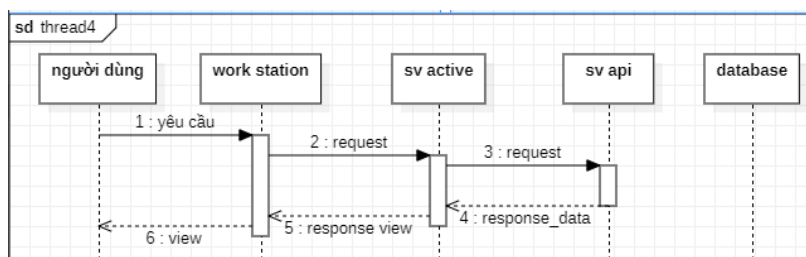


Hình 16 sơ đồ luồng dữ liệu theo phương thức 4

Phương thức 5:

Người dùng gửi yêu cầu thông qua giao diện trên máy trạm => máy trạm kiểm tra => gửi request lên máy chủ => máy chủ kiểm tra và xử lý gửi request lên server API => server API kiểm tra, xử lý và gửi phản hồi response lại server => Máy chủ server xử lý gửi phản hồi về máy trạm => máy trạm trả về view cho người dùng

- VD: Tương tác không hợp lệ/ thiếu thông tin
- Sơ đồ minh họa



Hình 17 sơ đồ luồng dữ liệu theo phương thức 5

5. Các loại API được sử dụng

- RESTful API

API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.

REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.

- API GraphQL

GraphQL là ngôn ngữ truy vấn dành cho API và là thời gian chạy để thực hiện các truy vấn đó với dữ liệu hiện có. GraphQL cung cấp mô tả đầy đủ và dễ hiểu về dữ liệu trong API, cho phép yêu cầu chính xác những gì cần và không cần gì, giúp phát triển API dễ dàng hơn theo thời gian và hỗ trợ các công cụ mạnh mẽ dành cho nhà phát triển.

Yêu cầu những gì cần, nhận được chính xác nó

Gửi truy vấn GraphQL tới API và nhận chính xác những gì cần, không hơn không kém. Truy vấn GraphQL luôn trả về kết quả có thể dự đoán được. Các ứng dụng sử dụng GraphQL rất nhanh và ổn định vì chúng kiểm soát dữ liệu nhận được chứ không phải máy chủ.

Nhận nhiều tài nguyên trong một yêu cầu

Các truy vấn GraphQL không chỉ truy cập các thuộc tính của một tài nguyên mà còn truy cập trọn vẹn các tham chiếu giữa chúng. Mặc dù API REST điển hình yêu cầu tải từ nhiều URL, API GraphQL nhận tất cả dữ liệu mà ứng dụng của bạn cần trong một yêu cầu. Các ứng dụng sử dụng GraphQL có thể nhanh chóng ngay cả trên các kết nối mạng di động chậm.

Mô tả những gì có thể với một hệ thống loại

API GraphQL được tổ chức theo loại và trường, không phải theo điểm cuối. Truy cập toàn bộ khả năng dữ liệu của bạn từ một điểm cuối duy nhất. GraphQL sử dụng các loại để đảm bảo Ứng dụng chỉ yêu cầu những gì có thể và cung cấp các lỗi rõ ràng và hữu ích. Ứng dụng có thể sử dụng các loại để tránh viết mã phân tích cú pháp thủ công

Di chuyển nhanh hơn với các công cụ mạnh mẽ dành cho nhà phát triển

Biết chính xác dữ liệu nào có thể yêu cầu từ API mà không cần rời khỏi trình chỉnh sửa, đánh dấu các sự cố tiềm ẩn trước khi gửi truy vấn và tận dụng lợi thế của mã thông minh được cải thiện. GraphQL giúp dễ dàng xây dựng các công cụ mạnh mẽ bằng cách tận dụng hệ thống các câu truy vấn.

Phát triển API của mà không cần phiên bản

Thêm các trường và loại mới vào API GraphQL mà không ảnh hưởng đến các truy vấn hiện có. Các trường lỗi hóa có thể không được dùng nữa và bị ẩn khỏi các công cụ. Bằng cách sử dụng một phiên bản phát triển duy nhất, API GraphQL cung cấp cho ứng dụng quyền truy cập liên tục vào các tính năng mới và khuyến khích mã máy chủ sạch hơn, dễ bảo trì hơn.

6. Các kiểu response trả về

- Return Template/view

Trả về một response mà browser có thể hiển thị thành giao diện

- Return Redirect/url

Trả về một chuyên hướng tới một url khác

- Return Response(API GraphQL)/dir data

Trả về một response kiểu response, muốn sử dụng cần ép về một kiểu khác

Vd ép về json: `data=json.loads(json.dumps(r.json()))`

- Return JsonResponse(result)

Trả về một response kiểu json

- Return Response(json.dumps(context), 200)

Trả về một response kiểu json

- Return jsonify(rep)

Trả về một response kiểu json

Lưu ý: khi dùng với fetch từ javacript

+ Với Return Response(json.dumps(context), 200)

Thì chỉ gọi hàm `then((data) => { })` nếu thành công

+ Với Return jsonify(rep)

Thì gọi cả hàm `then((data) => { })` và hàm `.catch((error) => { })` nếu thành công

Để khắc phục ta cần thêm điều kiện `if(!error){ }`

7. Các nơi lưu trữ dữ liệu

- Database

Lưu trữ các thông tin chung và cho phép tất cả người dùng có thể truy cập nếu có quyền thông qua các câu query

- Server

Lưu trữ các thông tin của các người dùng truy cập/sử dụng máy chủ đó

Thông tin được lưu trữ trong session/local đặt tại máy chủ

Các phiên làm việc(session) được xác định bằng mã cooki

- Máy trạm

Lưu trữ thông tin cá nhân người dùng tại thời điểm đó.

Thông tin được lưu trữ trong cooki/ local đặt tại máy trạm

Các cooki luôn được đóng gói gửi lên server mỗi khi gửi request. Từ đó phân biệt với các phiên làm việc khác.

IV. XÂY DỰNG HỆ THỐNG

1) Server API (Django):

a) Các bộ thư viện sử dụng:

Thư viện kèm theo khi cài đặt django

- ❖ Django==4.1.7
- ❖ aniso8601==7.0.0
- ❖ asgiref==3.6.0
- ❖ certifi==2022.12.7
- ❖ charset-normalizer==3.1.0
- ❖ deprecation==2.1.0
- ❖ django-js-asset==2.0.0
- ❖ gunicorn==20.1.0
- ❖ idna==3.4
- ❖ packaging==23.0
- ❖ Rx==1.6.3
- ❖ singledispatch==4.0.0
- ❖ six==1.16.0
- ❖ sqlparse==0.4.3
- ❖ text-unidecode==1.3
- ❖ tzdata==2022.7
- ❖ urllib3==1.26.14

Thư viện cài đặt thêm

- ❖ `django-ckeditor==6.5.1`

Cung cấp một số giao diện field hỗ trợ việc định dạng văn bản

- ❖ `django-filter==22.1`

Cung cấp bộ lọc queryset

- ❖ `django-graphql-auth==0.3.16`

Là một thư viện Python thường được sử dụng trong Dịch vụ web, ứng dụng GraphQL. `django-graphql-auth` không có lỗi hỏng, nó có sẵn tệp xây dựng, nó có Giấy phép cho phép và nó có mức hỗ trợ thấp.

- ❖ `django-graphql-jwt==0.3.0`

- ❖ `PyJWT==1.7.1`

Cung cấp khả năng xác thực với token

- ❖ `mysqlclient==2.1.1`

Cung cấp/ hỗ trợ tương tác dữ liệu với database kiểu mysql

- ❖ `Pillow==9.4.0`

Cung cấp/hỗ trợ các tương tác khi làm việc với ảnh

- ❖ `graphene==2.1.9`

- ❖ `graphene-django==2.15.0`

- ❖ `graphene-file-upload==1.3.0`

- ❖ `graphql-core==2.3.2`

- ❖ `graphql-relay==2.0.1`

Cung cấp các gói hỗ trợ việc xây dựng API graphene trên django

- ❖ `promise==2.3`

Cung cấp các tiện ích mở rộng để cho các thư viện khác gọi lại

- ❖ `psycopg2==2.9.5`

Cung cấp/ hỗ trợ tương tác dữ liệu với database kiểu PostgreSQL

- ❖ `requests==2.28.2`

Requests là một mô-đun Python có thể sử dụng để gửi tất cả các loại yêu cầu HTTP. Đây là một thư viện dễ sử dụng với nhiều tính năng khác nhau, từ việc truyền các tham số trong URL cho đến gửi các header tùy biến và Xác minh SSL.

- ❖ `whitenoise==6.3.0`

Tệp tĩnh được đơn giản hóa triệt để phục vụ cho các ứng dụng web Python chạy trên máy chủ

b) Hướng dẫn cài đặt:

- ❖ Cài biến môi trường : `python -m venv venv`
- ❖ Khởi động môi trường : `venv\Scripts\activate`
- ❖ Di chuyển tới thư mục root : `growpro`
- ❖ Cài đặt các bộ thư viện: `pip install -r requirements.txt`
- ❖ Đồng bộ các bộ thư viện : `pip freeze`
- ❖ Vào file `Growpro_django/settings.py` cấu hình thông tin pj:
 - Cấu hình thông tin database:

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'database_gro',  
        'USER': 'root',  
        'PASSWORD': '',  
        'HOST': '',  
        'OPTIONS': {'init_command': "SET  
sql_mode='STRICT_TRANS_TABLES'"},  
    }  
}
```

- Cấu hình địa chỉ xác thực

```
CSRF_TRUSTED_ORIGINS = ['http://127.0.0.1:8000/']
```

- Cấu hình thời gian duy trì đăng nhập

```
GRAPHQL_JWT => "JWT_EXPIRATION_DELTA": timedelta(days=7)
```

- Cấu hình thông tin mail hoạt động trong pj

```
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'  
EMAIL_HOST='smtp.gmail.com'  
EMAIL_HOST_USER='growprosp@gmail.com'  
EMAIL_HOST_PASSWORD='wzaobsnzkwwnbqr'  
EMAIL_PORT=587  
EMAIL_USE_TLS=True  
EMAIL_USE_SSL = False  
DEFAULT_FROM_EMAIL='growprosp@gmail.com'
```

- ❖ Vào file `Growpro_django/ app_grapql/ views.py` cấu hình thông tin trang frontend: `urlfe="http://127.0.0.1:5000/"`
- ❖ Tạo các file migration/kiểm tra lỗi: `python manage.py makemigrations`
- ❖ Di chuyển tới các file lỗi/lỗi thời trong thư viện và thay đổi :

```
Signal(providing_args=['request', 'refresh_token'])  
=> Signal(['request', 'refresh_token'])
```

Ví dụ:

```
from django.dispatch import Signal  
#token_issued = Signal(providing_args=['request', 'user'])  
token_issued = Signal(['request', 'user'])  
#token_refreshed = Signal(providing_args=['request', 'user'])
```

```
token_refreshed = Signal(['request', 'user'])
```

- ❖ Chạy lại lệnh: `python manage.py makemigrations` để tạo các file migration
- ❖ Thực hiện đồng bộ với database : `python manage.py migrate`
- ❖ Chạy chương trình nếu cần: `python manage.py runserver`

c) Cấu trúc thư mục và chức năng:

Thư mục project

	<ul style="list-style-type: none">- <code>app_grapql</code>: chứa các file liên quan app- <code>venv</code>: chứa các thư viện được cài đặt- <code>manage.py</code>: file quản lý project, chứa các tập lệnh thực thi hỗ trợ tương tác với project- <code>readme.md</code>: ghi chú- <code>requirements.txt</code>: chứa các tên thư viện + phiên bản đã cài đặt trong project- <code>Growpro_django</code>: thư mục quản lý file project+ <code>__pycache__</code>: chứa các file nhớ đệm được sinh ra khi thư mục đó là thư viện/gọi lại+ <code>__init__.py</code>: file khai báo/ thông báo thư mục này là một thư viện có thể gọi lại+ <code>asgi.py</code> : file cấu hình asgi cho project+ <code>wsgi.py</code>: file cấu hình wsgi cho project+ <code>schema.py</code>: chứa các khai báo/ hàm sử dụng trong api grapql+ <code>Settings.py</code>: chứa các thông số cấu hình project+ <code>urls.py</code>: chứa các đường dẫn routes
--	--

Lưu ý trong file `settings.py`:

Vì một số phiên bản thư viện không đồng bộ với nhau nên có thể một vài thư viện/hàm sẽ không được hỗ trợ, nên chúng ta có thể đặt lại/gán chúng với các giá trị/hàm đã định nghĩa/hỗ trợ. Không nên thay đổi trực tiếp trong thư viện vì nó có thể ảnh hưởng tới các hoạt động của các thư viện khác

Đối với các thư viện có thể đặt lại trên file settings.py:

Vd:

Django phiên bản 4.1 không hỗ trợ “django.utils.encoding.force_text” mà thay vào đó hỗ trợ “django.utils.encoding.force_str” trong khi các thư viện khác vẫn sử dụng/gọi “django.utils.encoding.force_text”. Nên chúng ta có thể gán “django.utils.encoding.force_text” = “django.utils.encoding.force_str”

```
from django.utils.encoding import force_str
```

```
django.utils.encoding.force_text = force_str
```

Tương tự với các thư viện khác

```
from django.utils.translation import gettext
```

```
django.utils.translation.ugettext=gettext
```

```
from django.utils.translation import gettext_lazy
```

```
django.utils.translation.ugettext_lazy=gettext_lazy
```

Với các thư viện không thể đặt lại thì chúng ta phải thay đổi trực tiếp trong thư viện

Vd:

Thư viện graphql_jwt

File graphql_jwt / signals.py vì cấu trúc hàm Signal đã thay đổi:

```
from django.dispatch import Signal
```

```
#token_issued = Signal(providing_args=['request', 'user'])
```

```
token_issued = Signal(['request', 'user'])
```

```
#token_refreshed = Signal(providing_args=['request', 'user'])
```

```
token_refreshed = Signal(['request', 'user'])
```

File graphql_jwt / refresh_token / signals.py cấu trúc hàm Signal đã thay đổi:

```
from django.dispatch import Signal
```

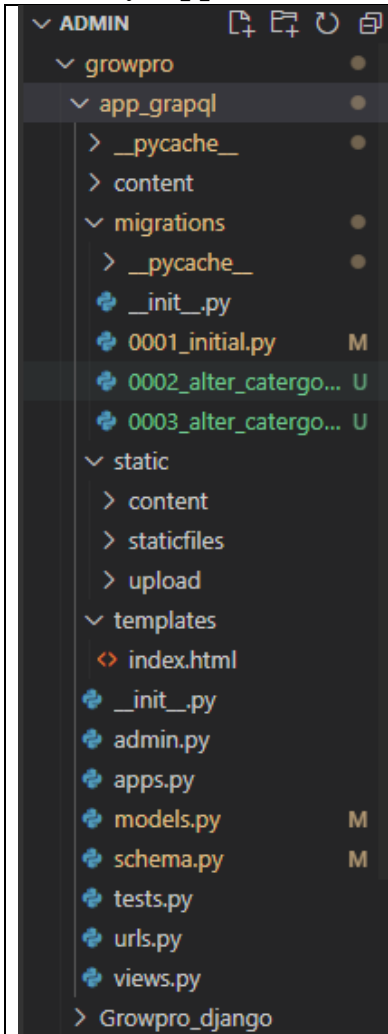
```
#token_issued = Signal(providing_args=['request', 'user'])
```

```
token_issued = Signal(['request', 'user'])
```

```
#token_refreshed = Signal(providing_args=['request', 'user'])
```

```
token_refreshed = Signal(['request', 'user'])
```


Thư mục app

	<ul style="list-style-type: none">- Thư mục <code>__pycache__</code>: chứa các file nhớ đệm được sinh ra khi thư mục đó là thư viện/gọi lại- File <code>__init__.py</code>: file khai báo/ thông báo thư mục này là một thư viện có thể gọi lại- Thư mục chứa các file do ckeditor upload- Thư mục migrations chứa các phiên bản làm việc với database- Thư mục static chứa các file không phải python, vd: upload, css, js,...- Thư mục templates: chứa các template- File <code>admin.py</code> : chứa các cấu hình trang admin hiển thị với người dùng- File <code>apps.py</code>: chứa cấu hình app- File <code>models.py</code>: chứa các cấu hình bảng cơ sở dữ liệu trong hệ thống. Lưu ý: chỉ chứa một phần không phải toàn bộ các bảng dữ liệu. Các bảng ở đây phần lớn do người dùng định nghĩa và phần lớn được quản lý bởi app- File <code>schema.py</code>: file chứa các kiểu/hàm/cấu hình định nghĩa api graphql- File <code>test.py</code>: nơi chứa các hàm kiểm tra- File <code>urls.py</code>: chứa các đường dẫn trong app- File <code>view.py</code>: chứa các hàm xử lý khi đường dẫn(url) được gọi do người dùng cấu hình.
--	---

Lưu ý:

- RESTful API : được cấu hình trong `views.py`

Vd: cấu hình upload file và quản lý các file đã upload bởi RESTful API

- `@csrf_exempt` : Trình quản lý với chế độ bỏ qua bảo vệ csrf mặc định để sử dụng phương thức “POST/PUT” trong django.
- `if request.method == 'POST'`: chỉ hoạt động khi phương thức là post
- `file = request.FILES['file']` : lấy các giá trị là file gán vào biến file
- `_file=file.read()` : đọc các giá trị trong file và gán kết quả vào biến `_file`
- `content_file = ContentFile(_file)` : lọc lấy các giá trị nội dung trong `_file`
- `..._user=request.POST['user']`: nhận các giá trị tham số từ request
- `u=str(_user).strip()`: loại bỏ tất cả khoảng trắng và gán kết quả cho biến u
- `user=User.objects.get(username=u)`: kiểm tra người dùng

```

@csrf_exempt
def uploadFile(request):
    if request.method == 'POST':
        file = request.FILES['file']
        _file=file.read()
        content_file = ContentFile(_file)
        _filename=request.POST['filename']
        _typefile=request.POST['filetype']
        _user=request.POST['user']
        u=str(_user).strip()
        if content_file:
            user=User.objects.get(username=u)
            if user:
                _size=len(content_file)
                _files_old=HistoryFileUp.objects.filter(size=_size)
                if _files_old:
                    for p in _files_old:
                        current_dir=""
                        url = os.path.abspath(os.path.join(current_dir,
os.pardir))+"/growpro/app_grapql%s"%p.file.url
                        if os.path.isfile(url):
                            with open(url,'rb') as f:
                                if _file == f.read():
                                    result = {'status': True,'url':p.file.url,'message':'file exists'}
                                    return JsonResponse(result)
                uploaded_file = UploadedFile(
                    file=content_file,
                    name=_filename,
                    content_type=_typefile
                )
                fileUpload=HistoryFileUp.objects.create(
                    title=_filename,
                    user=user,
                    file=uploaded_file,
                    typefile=_typefile,
                    size=_size
                )
                result = {'status': True,'url':fileUpload.file.url,'message':'saved files'}
            else:result = {'status': False,'url':None,'message':'user does not exist'}
        else:result = {'status': False,'url':None,'message':'file does not exist'}
        return JsonResponse(result)
return

```

- API GraphQL: được cấu hình trong schema.py

```

1 import graphene
2 import app_grapql.schema
3 from graphql_auth.schema import MeQuery
4 from graphql_auth import mutations
5
6 class AuthMutation(graphene.ObjectType): ...
7
8 class Query( MeQuery, app_grapql.schema.Query, graphene.ObjectType):
9     pass
10
11 class Mutation( app_grapql.schema.Mutation, AuthMutation, graphene.ObjectType,
12                 ):
13     pass
14
15 schema = graphene.Schema(query=Query, mutation=Mutation)

```

Các bước để cấu hình các queryset trong graphql:

+ Query: lấy dữ liệu

B1: import các model tham chiếu để lấy các kiểu dữ liệu

from app_grapql.models import *

B2: khai báo các Types : loại đối tượng truy vấn

```

class PanerType(DjangoObjectType):
    class Meta:
        model = Paner
        fields = '__all__'
        filter_fields = '__all__'
        interfaces = (graphene.relay.Node, )

```

B3: Khai báo các đối tượng và kiểu truy vấn

```

class Query(graphene.ObjectType):
    #model
    all_page = filter.DjangoFilterConnectionField(PageType)
    page = graphene.relay.Node.Field(PageType)

```

B4: Cấu hình lại dữ liệu đầu ra nếu muốn

```

def resolve_count_Items_seller(self, info):
    return Items_seller.objects.filter(status='Public').count()

```

Ví dụ một kết quả query

```

1 query{
2   allBuyer{
3     edges{
4       node{
5         id
6       }
7     }
8   }
9 }
10
11
12

```

```

{
  "data": {
    "allBuyer": {
      "edges": [
        {
          "node": {
            "id": "QnV5ZXJUeXB1OjE="
          }
        },
        {
          "node": {
            "id": "QnV5ZXJUeXB1OjI="
          }
        }
      ]
    }
  }
}

```

Hình 18 kết quả từ một câu lệnh query

+ Mutation: tương tác với dữ liệu

B1: Tạo lớp chứa hành động sẽ thực thi khi gọi:

```
#action
#user
class ChangeFirtNameAction(Output):
    """
    Change user FirtName without login.

    It needs a token to authenticate.
    """
    class Arguments:
        firtName=graphene.String(required=True)

    @classmethod
    def resolve_mutation(cls, root, info, **kwargs):
        user=info.context.user
        if user.is_authenticated:
            user.first_name=kwargs.get("firtName")
            user.save()
            return cls(success=True,errors=None)
        return cls(success=False,errors=[{"message":"User does not exist"}])
```

- Output: là lớp đệm với đầu ra là đối tượng cls với 2 thuộc tính cơ bản : success và errors. Nó được công khai mở rộng và được gọi nhiều với các hàm trong thư viện graphene. Nên kế thừa để đồng bộ các kiểu trả về/ nhận các kiểu trả về từ các thư viện có sẵn(có thể không kế thừa)
- ChangeFirtNameAction: tên lớp chứa hành động
- Arguments: Tham số truyền vào/trả về
- resolve_mutation: hàm sẽ được thực thi khi hành động
- cls: đối tượng được kế thừa, thường dùng làm kiểu trả về vì nó có thể nhận kèm theo các hành động khác
- info: thông tin request, thường dùng để xác thực người dùng
- kwargs: các tham số truyền vào

B2: chuyển hàm vừa tạo thành hàm Mutation với các ràng buộc, báo lỗi, hiển thị,...

```
#MutationFunction
#user
class ChangeFirtName(MutationMixin, DynamicArgsMixin,
ChangeFirtNameAction, graphene.Mutation):
    __doc__ = ChangeFirtNameAction.__doc__
    _required_args = ["firtName"]
```

- ChangeFirtName: tên hàm sử dụng khi khai báo mutation
- MutationMixin: nhận được thông báo ngoại lệ khi kế thừa lớp này
- DynamicArgsMixin: Lớp mở rộng để khởi tạo/ lấy các đối số

```
get args from
    cls._args
    cls._required_args
args is dict { arg_name: arg_type }
or list [arg_name,] -> defaults to String
```

- ChangeFirtNameAction lớp chứa hành động khi gọi tới
- graphene.Mutation: Chuyển sang thành hàm mutation khi kế thừa lớp này
- __doc__: ghi chú
- _required_args: các tham số bắt buộc phải có.

B3: Khai báo với hệ thống:

```
#Mutation
class Mutation(graphene.ObjectType):
    #user
    changeFirtName=ChangeFirtName.Field()
```

🚩 Lưu ý: kiểu trả về của mutation có thể là một đối tượng, và cho phép truy vấn tới các đối tượng khác nếu có liên kết trong query

Vd:Lấy/tạo ra hóa đơn với hóa đơn với trạng thái mới tạo của người dùng

Kết quả trả về:

- Kết quả thực hiện
- Lỗi xảy ra
- Id của hóa đơn nếu thành công
- Người tạo hóa đơn
- Tên username người tạo

```
mutation{
  getOrCreateInvoice{
    success
    errors
    invoice{
      id
      buyer{
        id
        user{
          id
          username
        }
      }
    }
  }
}
```

```
{
  "data": {
    "getOrCreateInvoice": {
      "success": false,
      "errors": {
        "nonFieldErrors": [
          {
            "message": "User does not exist"
          }
        ]
      },
      "invoice": null
    }
  }
}
```

chat(dest: String!, rom: String!): ChatRom
Send a chat to Rom without login.
It needs a token to authenticate.
getOrCreateInvoice: Get_or_createInvoice
Get invoice with invoice status = created.
If the invoice does not exist, the system will create a new invoice.
It needs a token to authenticate.

Hình 19 kết quả từ câu lệnh mutation

Trong file views.py

Chúng ta có thể sử dụng tính chất “Có thể bảo trì và tái sử dụng” để sử dụng lại các hàm mà các thư viện đã cài đặt. Từ đó giúp việc xây dựng nhanh chóng hơn và ít trùng lặp, giảm số lượng mã.

Vd:

Để xử lý yêu cầu xác nhận người dùng qua mail :

Cách 1: Dùng thư viện graphql_auth để xác thực

```
from graphql_auth.models import
def activate(request,token):
    #UserStatus.verify(token)
```

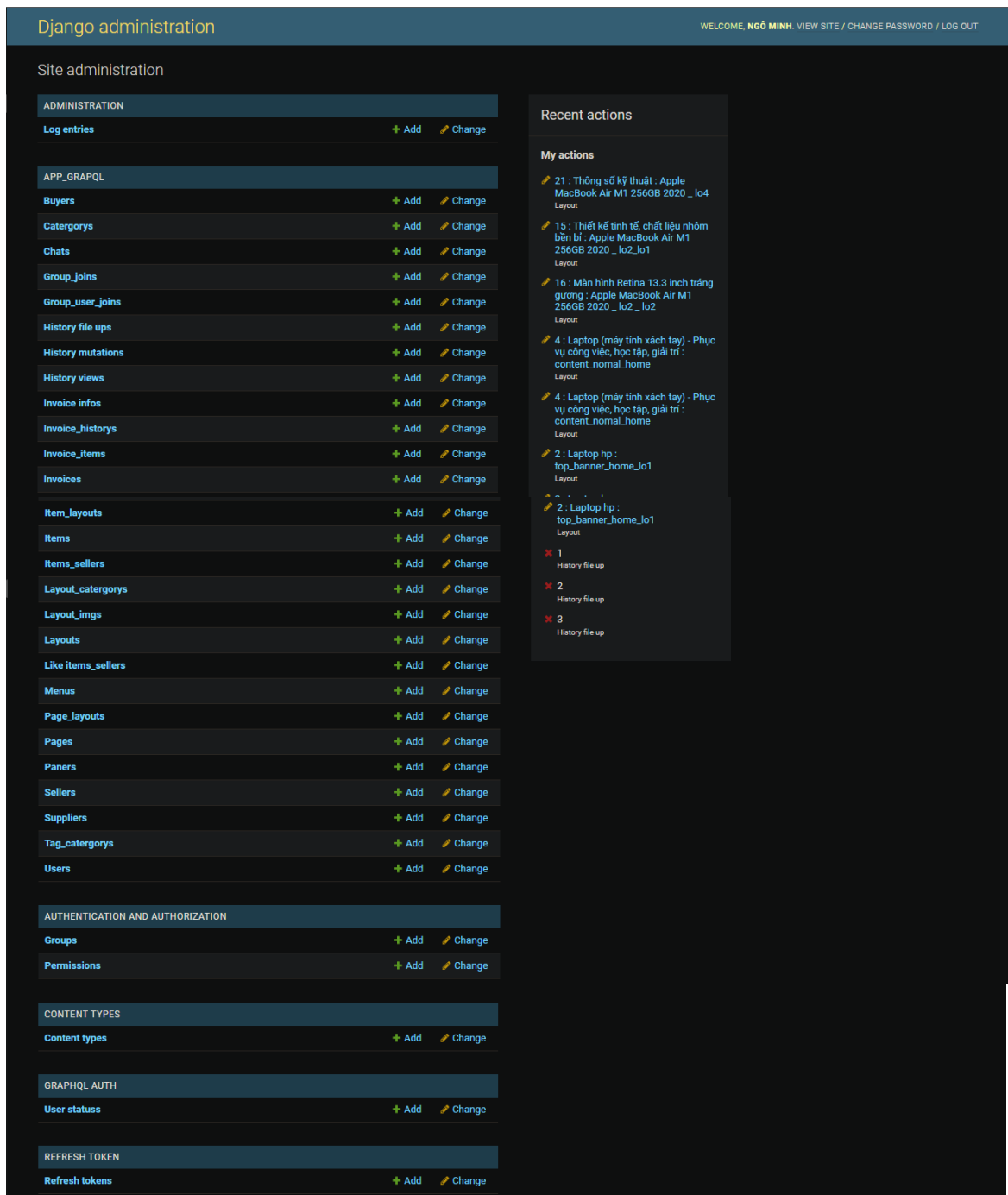
Cách 2 : Sử dụng kết hợp các bộ thư viện khác nhằm đưa ra kết quả như mong muốn dựa trên cấu trúc hàm verify trong thư viện graphql_auth

```
from django.shortcuts import render,redirect
from django.contrib.auth import get_user_model
from graphql_auth.models import UserStatus
from graphql_auth.utils import get_token_payload
from graphql_auth.settings import graphql_auth_settings as app_settings
from graphql_auth.constants import TokenAction
UserModel = get_user_model()
urlfe=http://127.0.0.1:5000/

def activate(request,token):
    #UserStatus.verify(token)
    payload = get_token_payload(token, TokenAction.ACTIVATION,
app_settings.EXPIRATION_ACTIVATION_TOKEN)
    user = UserModel._default_manager.get(**payload)
    user_status = UserStatus.objects.get(user=user)
    if user_status.verified is True:
        return redirect("%sverify"% urlfe)
    elif user_status.verified is False:
        user_status.verified = True
        user_status.save(update_fields=["verified"])
        return redirect("%sverify"% urlfe)
    else:
        return render(request, template_name='index.html',context={'name':'lỗi'})
```

- from django.shortcuts import render,redirect : thêm 2 kiểu trả về từ thư viện django
- from django.contrib.auth import get_user_model: thêm hàm lấy đối tượng người dùng trong thư viện xác thực django
- from graphql_auth.models import UserStatus : thêm đối tượng trạng thái người dùng trong thư viện graphql_auth
- from graphql_auth.utils import get_token_payload: thêm hàm giải mã dữ liệu từ đoạn mã cho trước trong thư viện graphql_auth
- from graphql_auth.settings import graphql_auth_settings as app_settings: thêm các giá trị cài đặt của graphql_auth
- from graphql_auth.constants import TokenAction : thêm đối tượng quản lý các mã quy định các hoạt động của hệ thống
- UserModel = get_user_model() : gán đối tượng người dùng vào biến để sử dụng
- payload = get_token_payload(token, TokenAction.ACTIVATION, app_settings.EXPIRATION_ACTIVATION_TOKEN): giải mã đoạn mã từ đường dẫn với thư viện graphql_auth và gán kết quả vào biến payload
- user = UserModel._default_manager.get(**payload): tách lấy người dùng từ các thông tin trong payload và gán kết quả thu được vào biến user bằng thư viện
- (tham khảo các thuộc tính có trong payload trong thư viện graphql_auth file utils.py :” payload = {user.USERNAME_FIELD: username, "action": action}”)

d) Các giao diện người dùng:
❖ Giao diện tương tác



Hình 20 giao diện danh sách các đối tượng trong hệ thống

Được quản lý trong file app_grapql/admin.py

```
from django import forms
from django.contrib import admin
from django.contrib.contenttypes.models import ContentType
from django.contrib.admin.models import LogEntry
from django.contrib.auth.models import Permission
from graphql_auth.models import UserStatus
from .models import *
from django.utils.safestring import mark_safe
# Register your models here.
#page
class layoutForm(forms.ModelForm):
    #dest=forms.CharField(widget=CKEditorUploadingWidget)
    class Meta:
        models=Layout
        fields='__all__'

class layout_imgInline(admin.StackedInline):
    model=Layout_img
    pk_name='layout'
    readonly_fields=['img_read']
    def img_read(self,obj):
        return mark_safe(u'' % (obj.avatar))

class page_layoutInline(admin.StackedInline):
    model=Page_layout
    pk_name='layout'

class item_layoutInline(admin.StackedInline):
    model=Item_layout
    pk_name='layout'

class layoutAdmin(admin.ModelAdmin):
    inlines=(layout_imgInline, page_layoutInline, item_layoutInline)
    form =layoutForm

#list_display=["id", "title", "show", "active", "priority", "parent", "page", "category"]
list_filter=["show", "active", "parent"]
search_fields=["id", "title", "parent"]

#product
class tag_categoryInline(admin.StackedInline):
    model=Tag_category
    pk_name='item'

class itemAdmin(admin.ModelAdmin):
    inlines =(tag_categoryInline,)
```

```

#list_display=["id","title","show","active","prite","prite_promotion"]
list_filter=["show","active"]
search_fields=["id","title"]
readonly_fields=['img_read']
def img_read(self,obj):
    return mark_safe(u'' % (obj.avatar))
#register
#user
admin.site.register(User)
admin.site.register(Permission)
admin.site.register(ContentType)
admin.site.register(Seller)
admin.site.register(Buyer)
admin.site.register(Paner)
admin.site.register(Supplier)
admin.site.register(UserStatus)
#page
admin.site.register(Page)
admin.site.register(Menu)
admin.site.register(Layout,layoutAdmin)
admin.site.register(Layout_catergory)
admin.site.register(Layout_img)
admin.site.register(Page_layout)
#product
admin.site.register(Catergory)
admin.site.register(Tag_catergory)
admin.site.register(Item,itemAdmin)
admin.site.register(Invoice)
admin.site.register(LikeItems_seller)
admin.site.register(Items_seller)
admin.site.register(Item_layout)
admin.site.register(InvoiceInfo)
admin.site.register(Invoice_item)
#group
admin.site.register(Group_join)
admin.site.register(Chat)
admin.site.register(Group_user_join)
#history
admin.site.register(LogEntry)
admin.site.register(HistoryFileUp)
admin.site.register(Invoice_history)
admin.site.register(HistoryView)
admin.site.register(HistoryMutation)

```

Trong đó:

- from django import forms

Thêm trình hiển thị, tương tác, ngoại lệ, lỗi Với Đối tượng là form từ thư viện django

- from django.contrib import admin

Thêm trình hiển thị, tương tác, ngoại lệ, lỗi với đối tượng được quản lý bởi trình giao diện admin của django

- from django.contrib.contenttypes.models import ContentType

Thêm kiểu đối tượng ContentType từ thư viện django

- from django.contrib.admin.models import LogEntry

Thêm kiểu đối tượng LogEntry từ thư viện django

- from django.contrib.auth.models import Permission

Thêm kiểu đối tượng Permission từ thư viện django

- from graphql_auth.models import UserStatus

Thêm kiểu đối tượng UserStatus từ thư viện graphql_auth

- from .models import *

Thêm tất cả các kiểu đối tượng đã được xây dựng trong file app_grapql/models.py

- from django.utils.safestring import mark_safe

Thêm hàm chuyển đổi dạng chuỗi sang dạng html từ thư viện django

- class layoutForm(forms.ModelForm):

Khai báo lớp đối tượng mới layoutForm từ đối tượng Layout

- class layout_imgInline(admin.StackedInline):

Khai báo lớp đối tượng con/thành phần mới cho phép nhúng vào các đối tượng khác khi gọi từ đối tượng Layout_img

- class page_layoutInline(admin.StackedInline):

Khai báo lớp đối tượng con/thành phần mới cho phép nhúng vào các đối tượng khác khi gọi từ đối tượng Page_layout

- class item_layoutInline(admin.StackedInline):

Khai báo lớp đối tượng con/thành phần mới cho phép nhúng vào các đối tượng khác khi gọi từ đối tượng Item_layout

- class layoutAdmin(admin.ModelAdmin):

Khai báo lớp đối tượng mới layoutAdmin

inlines=(layout_imgInline, page_layoutInline, item_layoutInline):nhúng các đối tượng con: layout_imgInline, page_layoutInline, item_layoutInline trong nó

form =layoutForm:Kiểu đối tượng chính là layoutForm

- class tag_caterygoryInline(admin.StackedInline):

Khai báo lớp đối tượng con/thành phần mới cho phép nhúng vào các đối tượng khác khi gọi từ đối tượng Tag_caterygory

- class itemAdmin(admin.ModelAdmin):

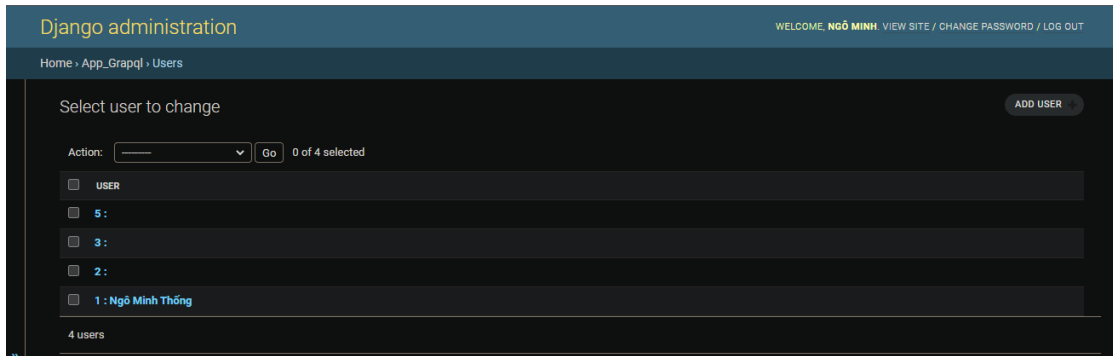
Khai báo lớp đối tượng mới itemAdmin

inlines =(tag_caterygoryInline,): nhúng đối tượng tag_caterygoryInline con vào trong nó

- admin.site.register(User)

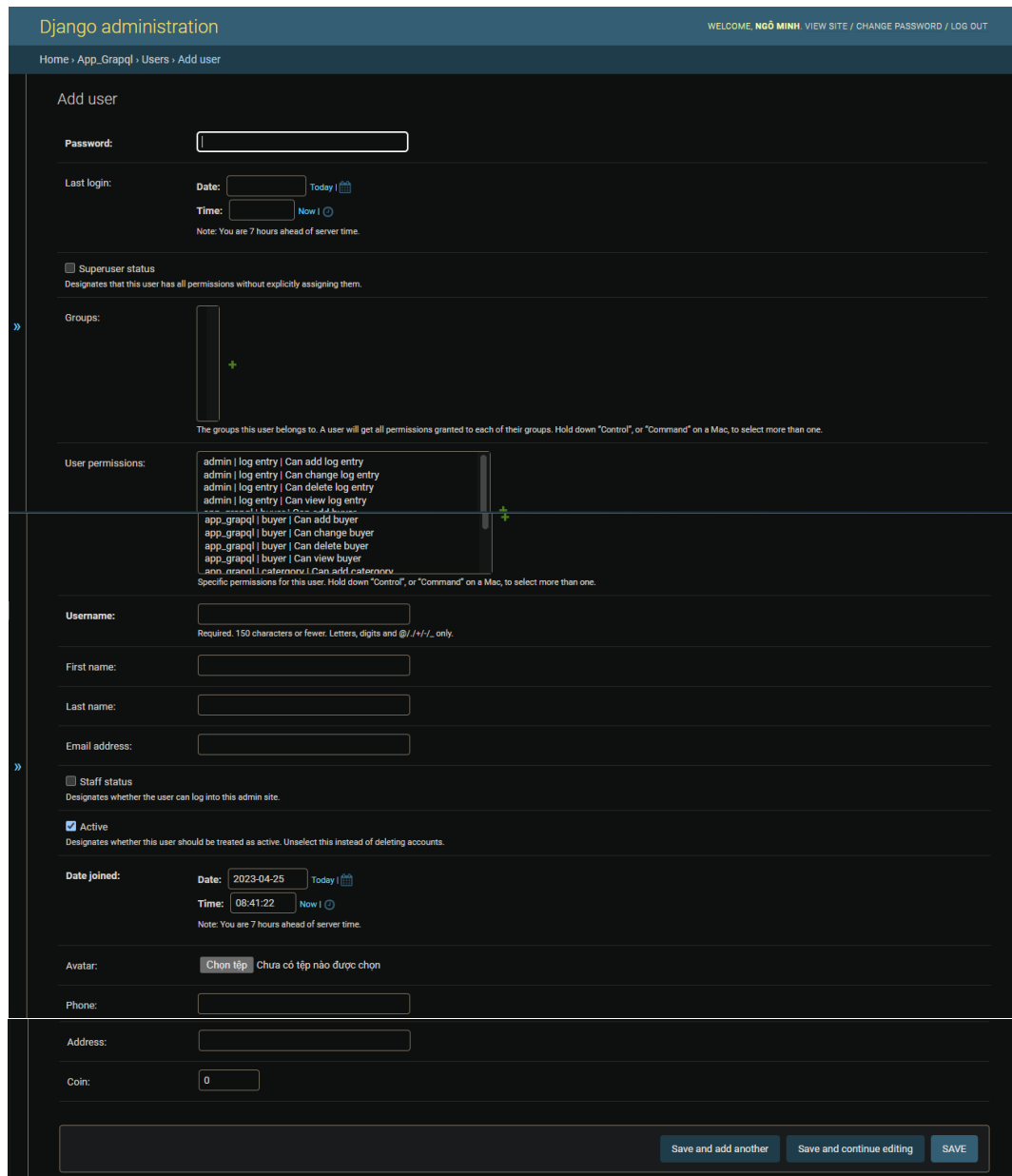
Đăng ký đối tượng User với trình quản lý admin trong django

+ Xem danh sách:



Hình IV.1.4 giao diện danh sách các đối tượng người dùng

+ Thêm:



Hình 21 giao diện thêm đối tượng người dùng

+ Cập nhật

Django administration WELCOME, NGÔ MINH. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > App_Grapql > Users > 1 : Ngô Minh Thống

Change user

1 : Ngô Minh Thống HISTORY

Password:

Last login: **Date:** Today | 📅
Time: Now | 🕒
Note: You are 7 hours ahead of server time.

Superuser status
Designates that this user has all permissions without explicitly assigning them.

Groups:

The groups this user belongs to. A user will get all permissions granted to each of their groups. Hold down "Control", or "Command" on a Mac, to select more than one.

User permissions:

Specific permissions for this user. Hold down "Control", or "Command" on a Mac, to select more than one.

Username:
Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

First name:

Last name:

Email address:

Staff status
Designates whether the user can log into this admin site.

Active
Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

Date joined: **Date:** Today | 📅
Time: Now | 🕒
Note: You are 7 hours ahead of server time.

Avatar: Chưa có tệp nào được chọn

Phone:

Address:

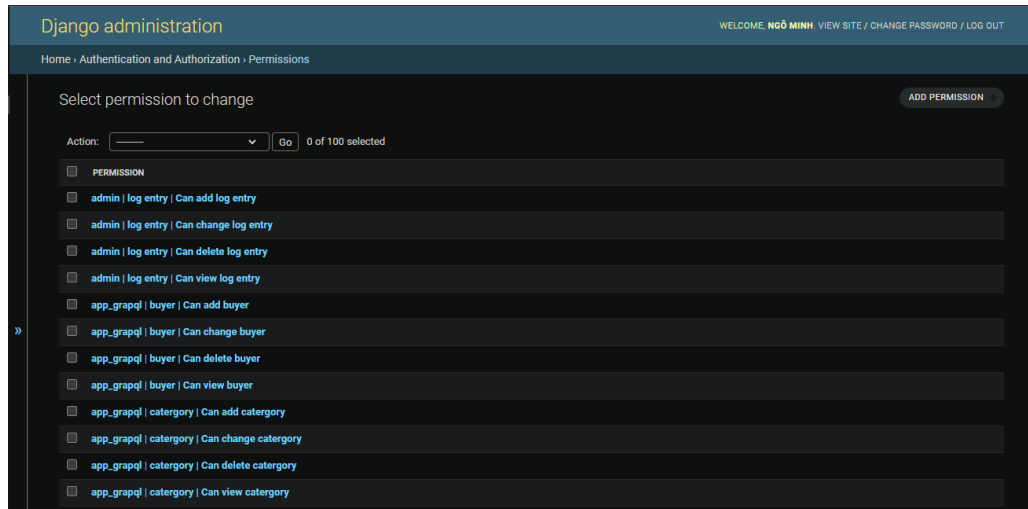
CoIn:

Hình 22 giao diện cập nhật đối tượng người dùng

- admin.site.register(Permission)

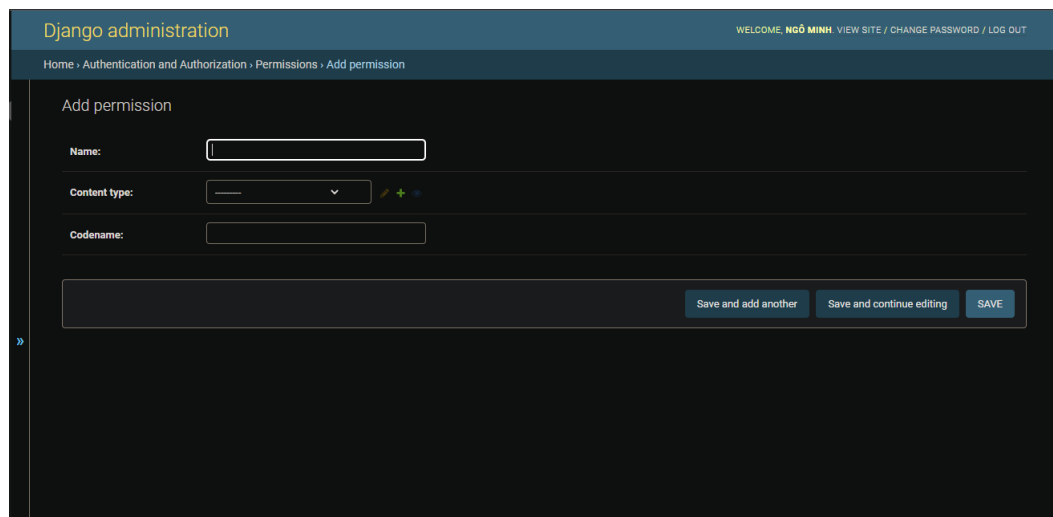
Đăng ký đối tượng Permission với trình quản lý admin trong django(chỉ nên xem)

+ Xem danh sách:



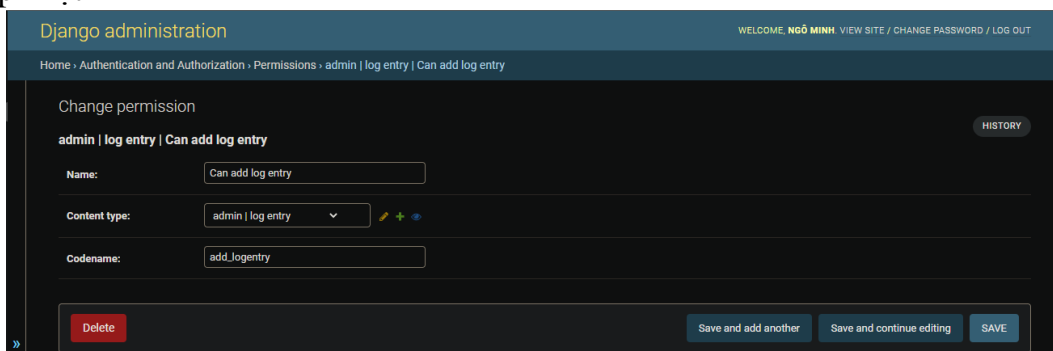
Hình 23 giao diện danh sách các đối tượng quyền trong hệ thống

+ Thêm:



Hình 24 giao diện thêm đối tượng quyền trong hệ thống

+ Cập nhật

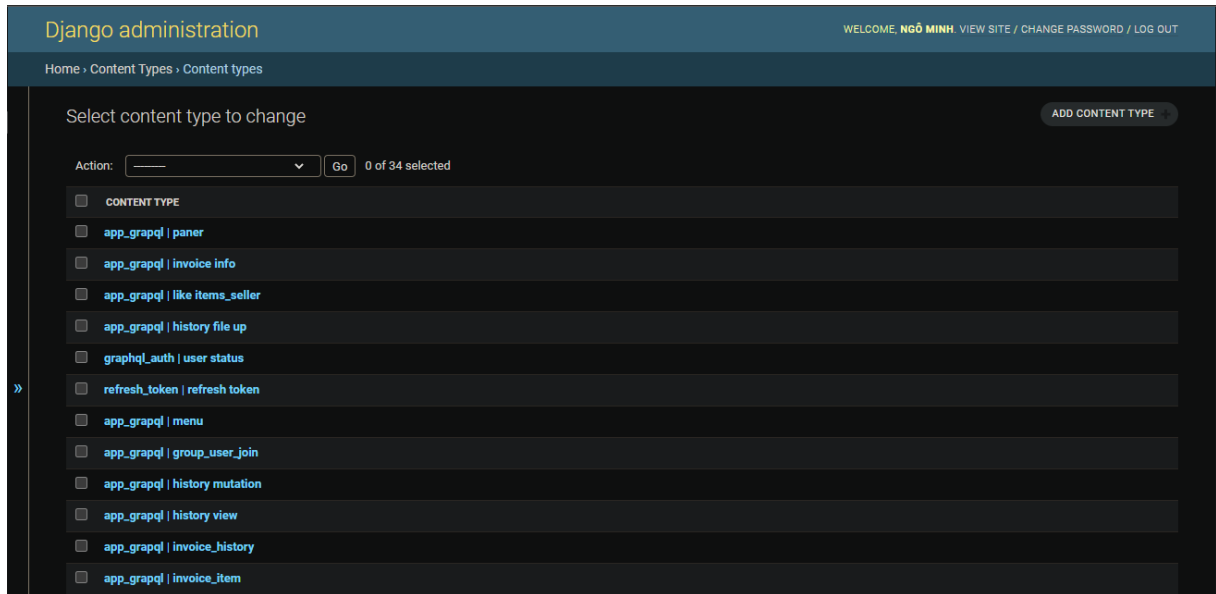


Hình 25 giao diện cập nhật đối tượng quyền trong hệ thống

- admin.site.register(ContentType)

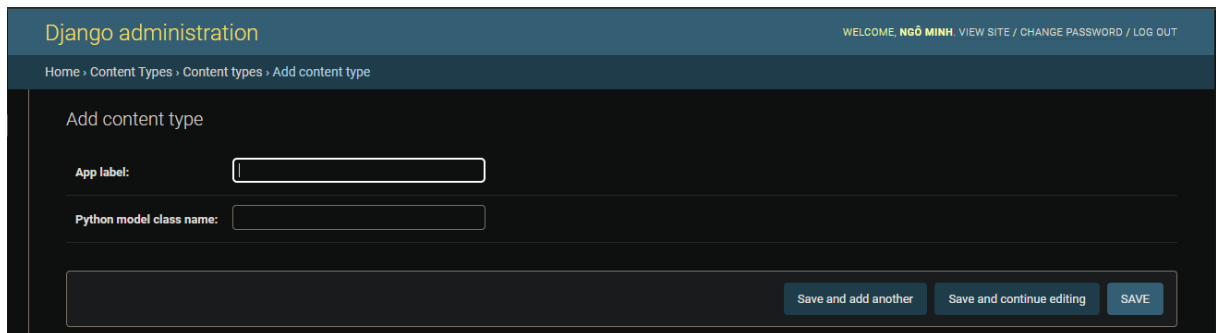
Đăng ký đối tượng ContentType với trình quản lý admin trong django(chỉ nên xem)

+ Xem danh sách:



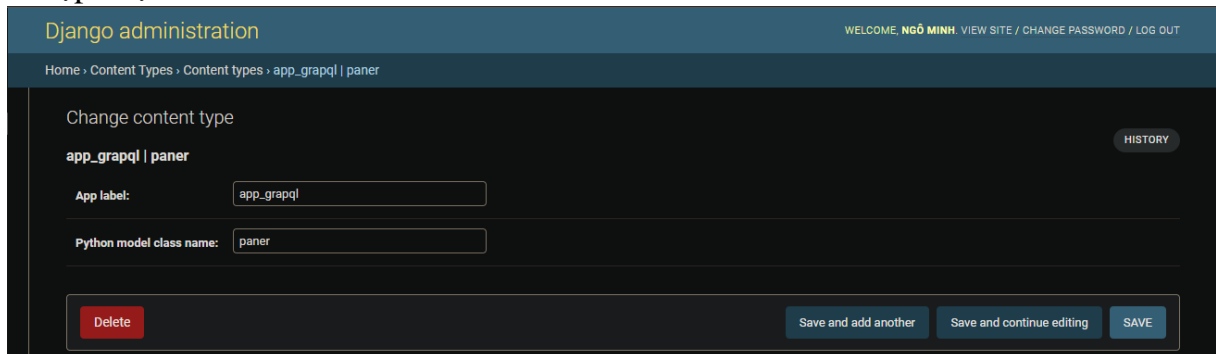
Hình 26 giao diện danh sách các đối tượng nội dung trong hệ thống

+ Thêm:



Hình 27 giao diện thêm đối tượng nội dung trong hệ thống

+ Cập nhật

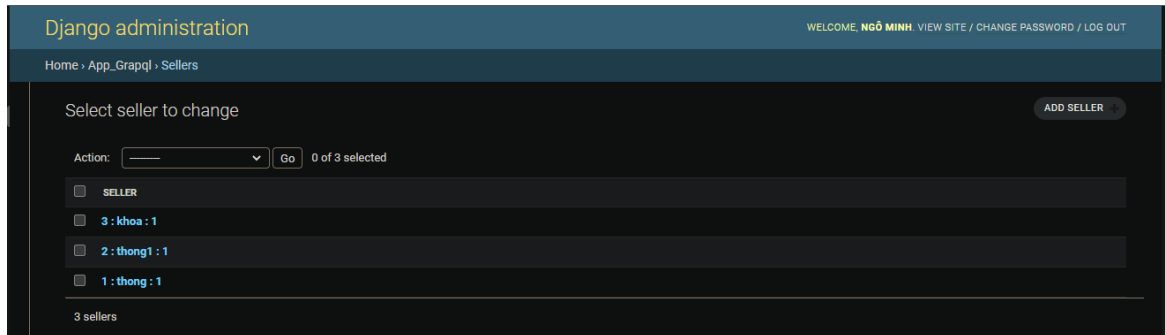


Hình 28 giao diện danh sách các đối tượng nội dung trong hệ thống

- admin.site.register(Seller)

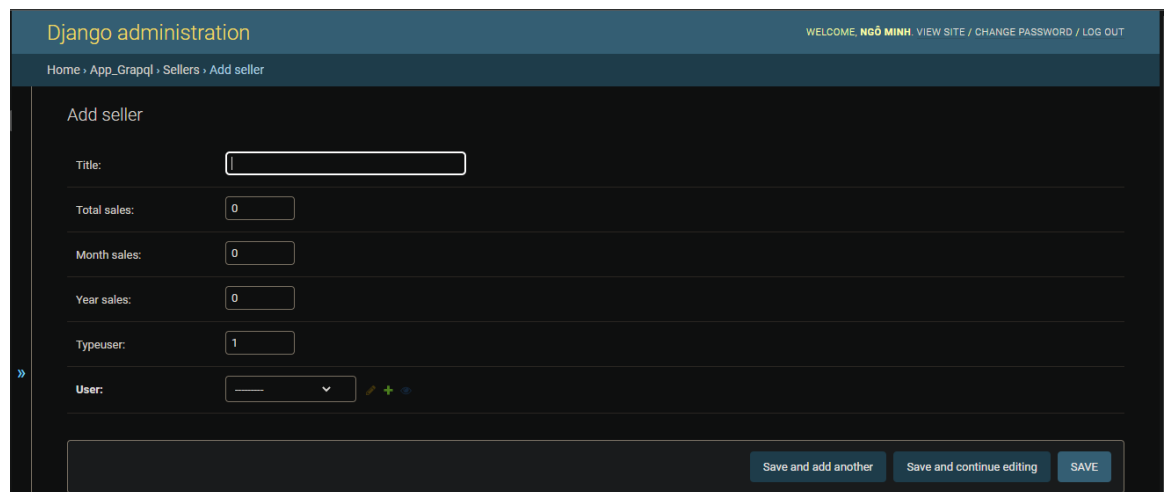
Đăng ký đối tượng Seller với trình quản lý admin trong django

+ Xem danh sách:



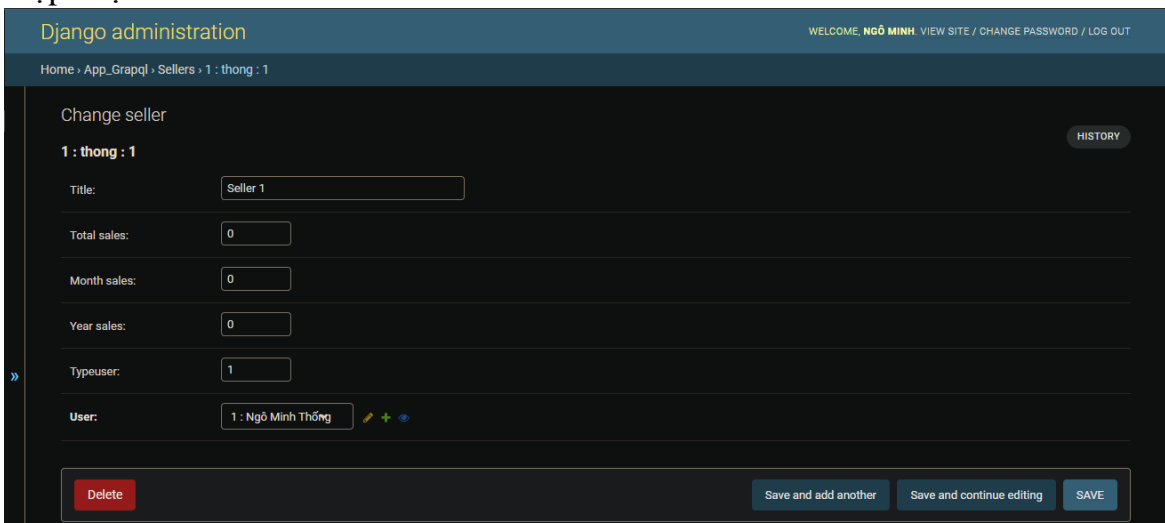
Hình 29 giao diện danh sách các đối tượng người bán

+ Thêm:



Hình 30 giao diện thêm đối tượng người bán

+ Cập nhật

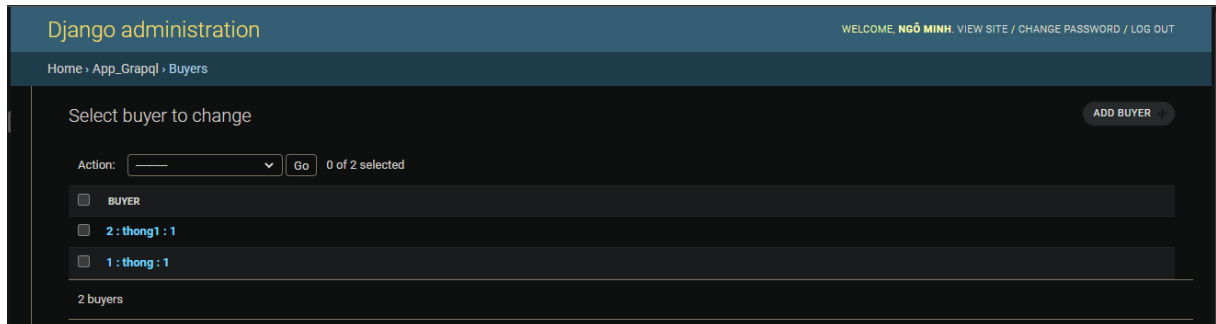


Hình 31 giao diện cập nhật đối tượng người bán

- admin.site.register(Buyer)

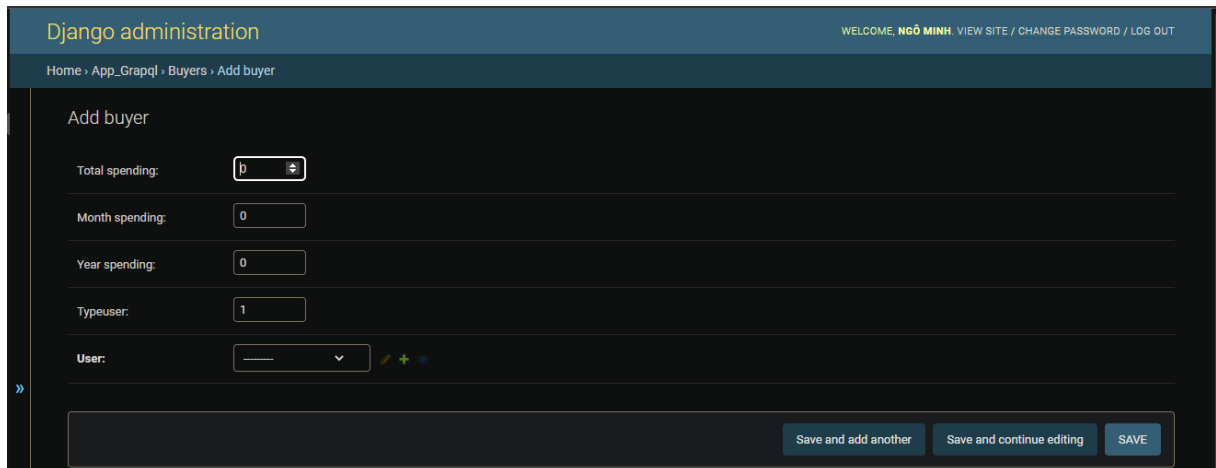
Đăng ký đối tượng Buyer với trình quản lý admin trong django

+ Xem danh sách:



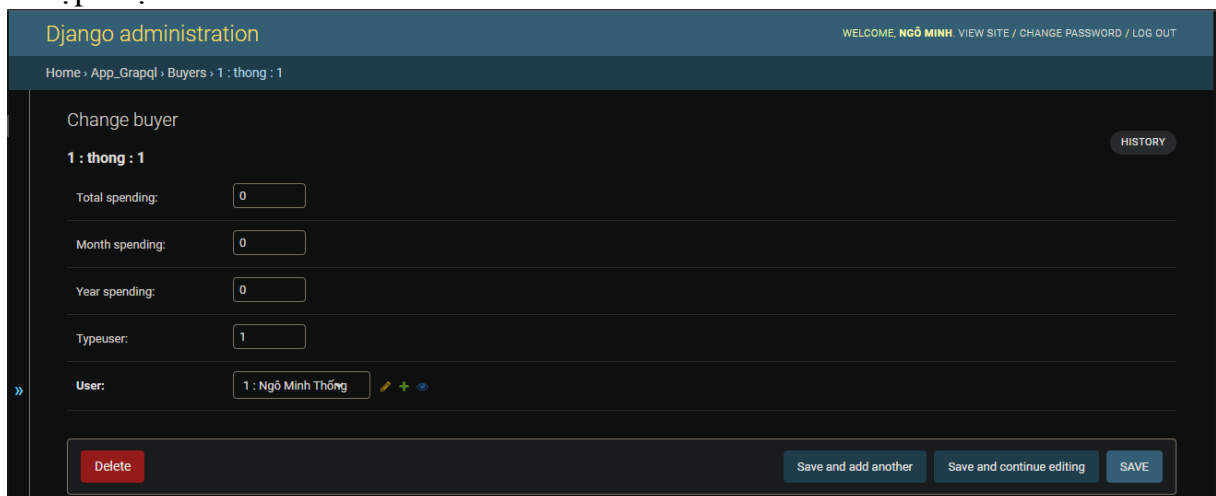
Hình 32 giao diện danh sách các đối tượng người bán

+ Thêm:



Hình 33 giao diện thêm đối tượng người bán

+ Cập nhật

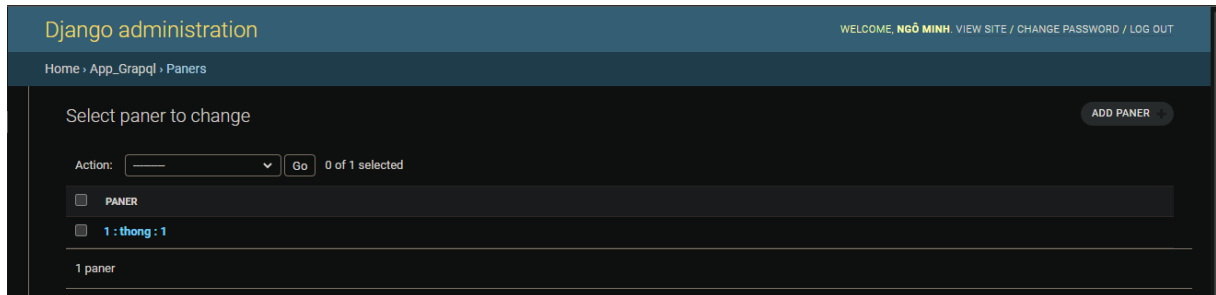


Hình 34 giao diện cập nhật đối tượng người bán

- admin.site.register(Paner)

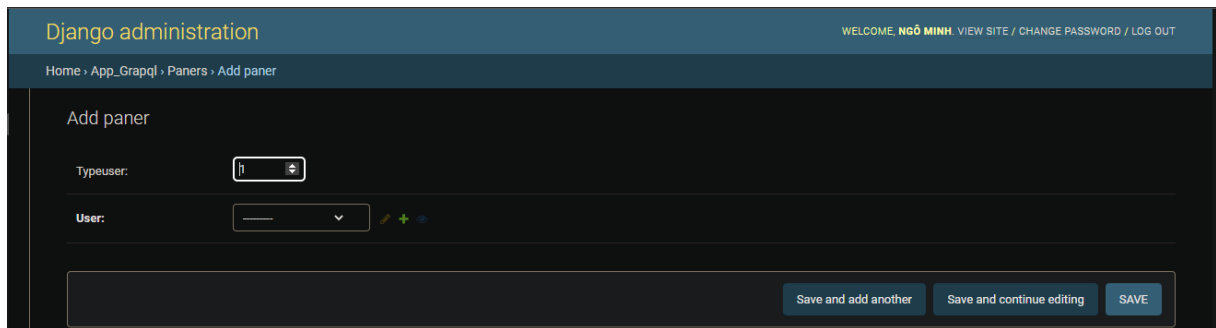
Đăng ký đối tượng Paner với trình quản lý admin trong django

+ Xem danh sách:



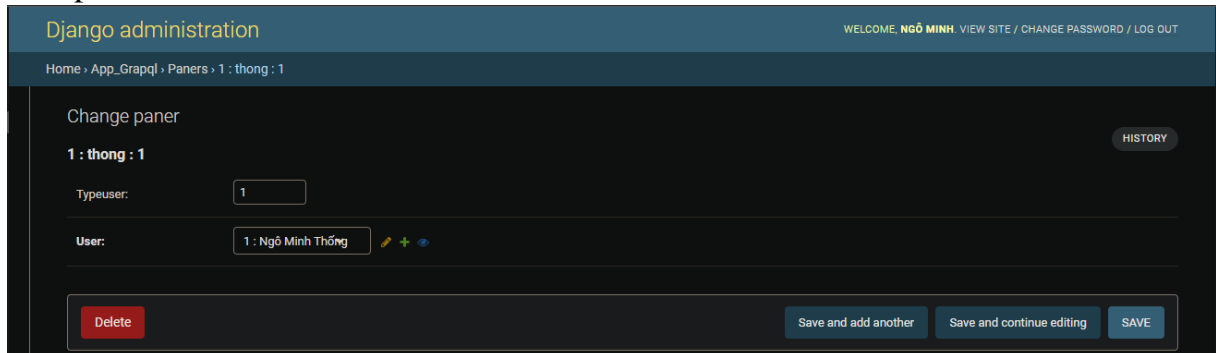
Hình 35 giao diện danh sách các đối tượng đối tác

+ Thêm:



Hình 36 giao diện thêm đối tượng đối tác

+ Cập nhật

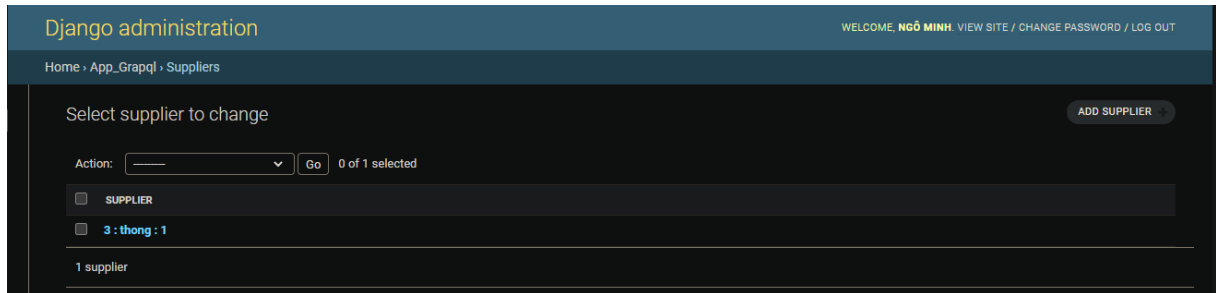


Hình 37 giao diện cập nhật đối tượng đối tác

- admin.site.register(Supplier)

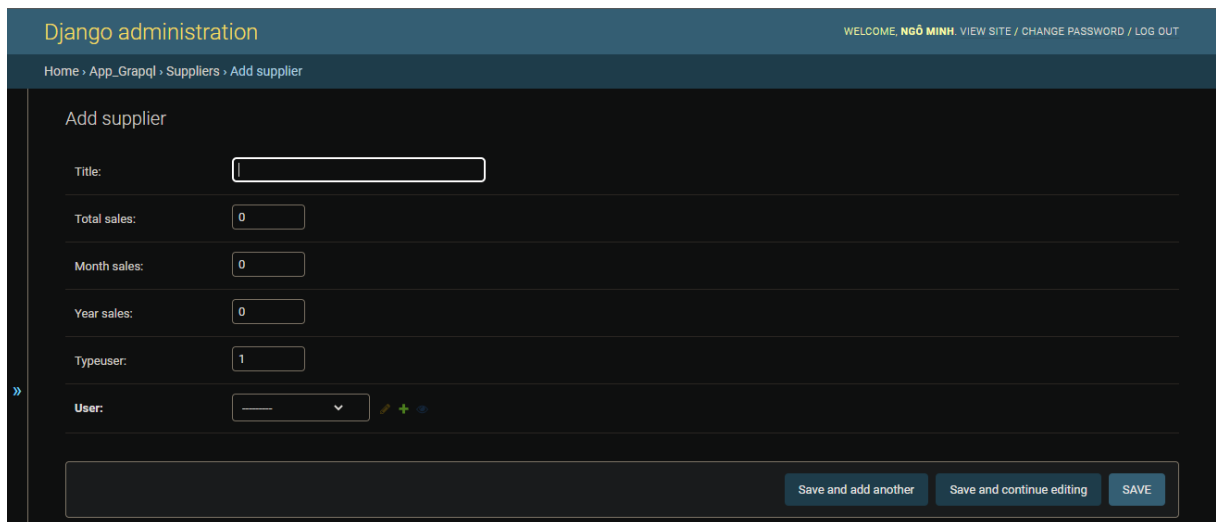
Đăng ký đối tượng Supplier với trình quản lý admin trong django

+ Xem danh sách:



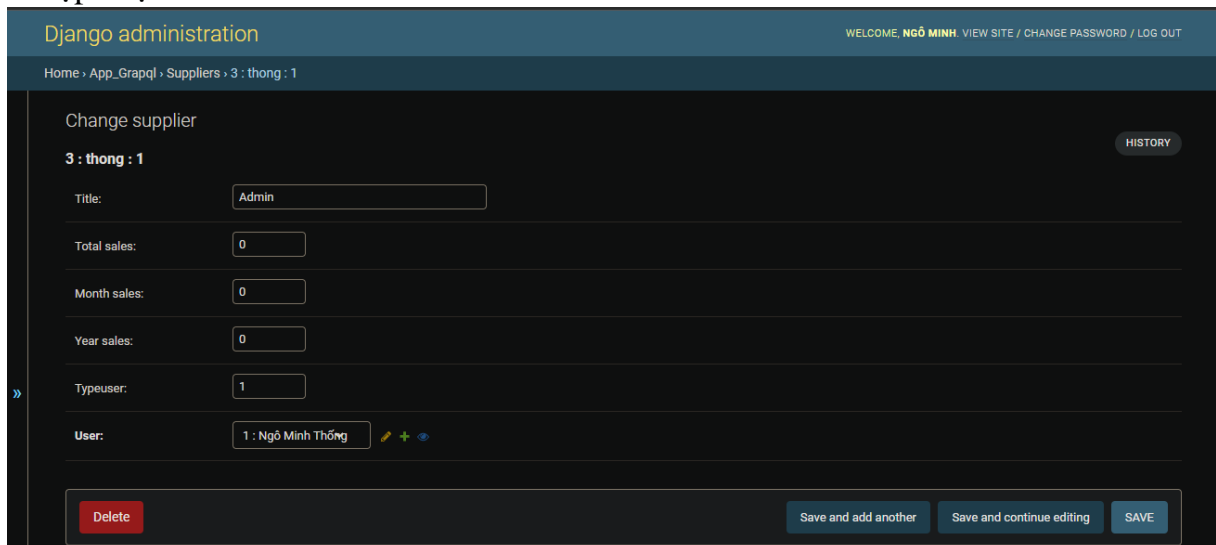
Hình 38 giao diện danh sách các đối tượng nhà cung cấp

+ Thêm:



Hình 39 giao diện thêm đối tượng nhà cung cấp

+ Cập nhật

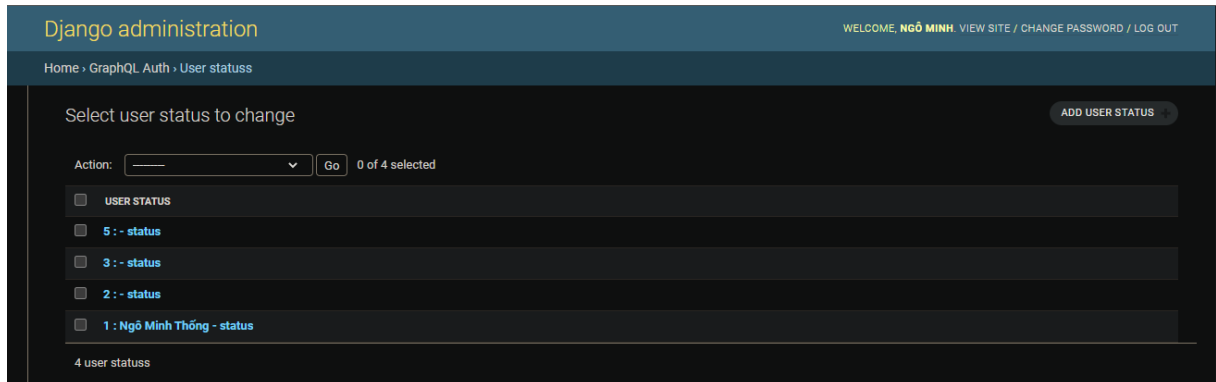


Hình 40 giao diện cập nhật đối tượng nhà cung cấp

- admin.site.register(UserStatus)

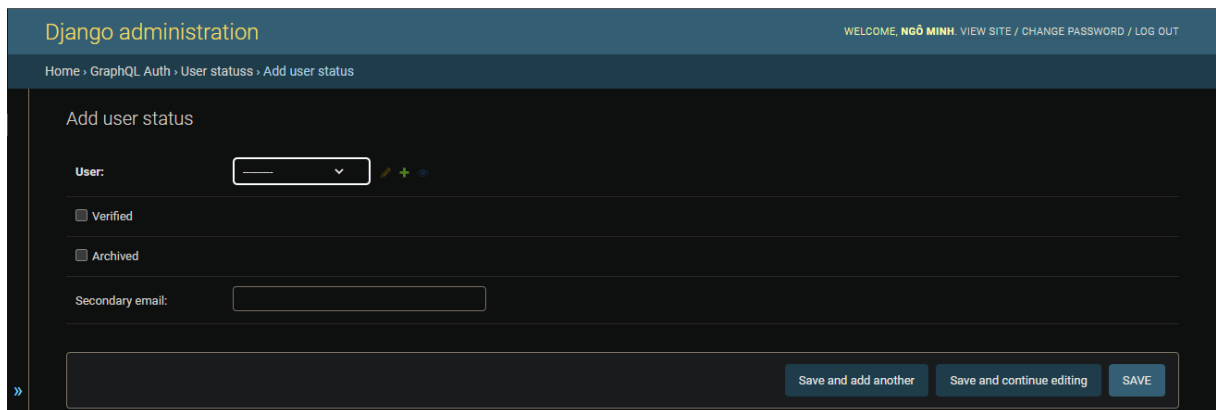
Đăng ký đối tượng UserStatus với trình quản lý admin trong django

+ Xem danh sách:



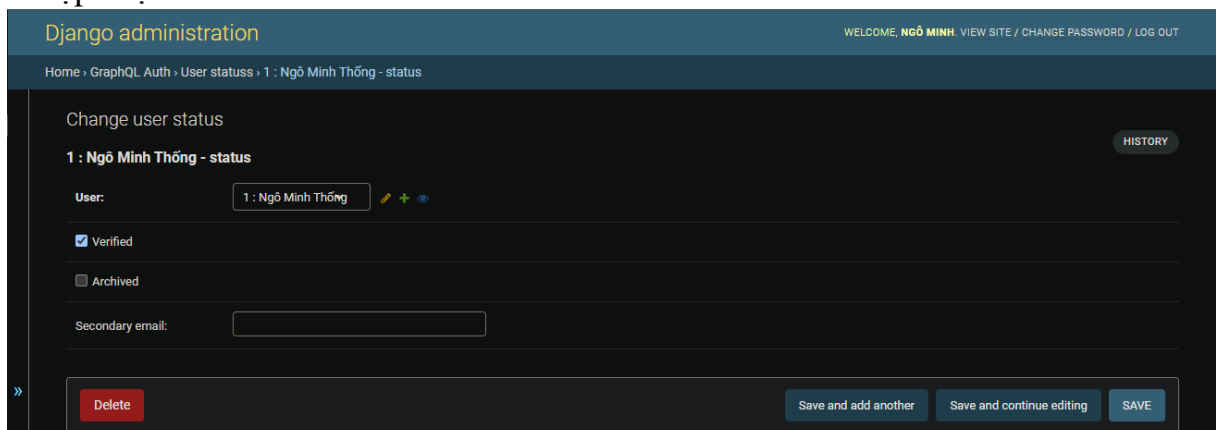
Hình 41 giao diện danh sách các đối tượng trạng thái người dùng

+ Thêm:



Hình 42 giao diện thêm đối tượng trạng thái người dùng

+ Cập nhật

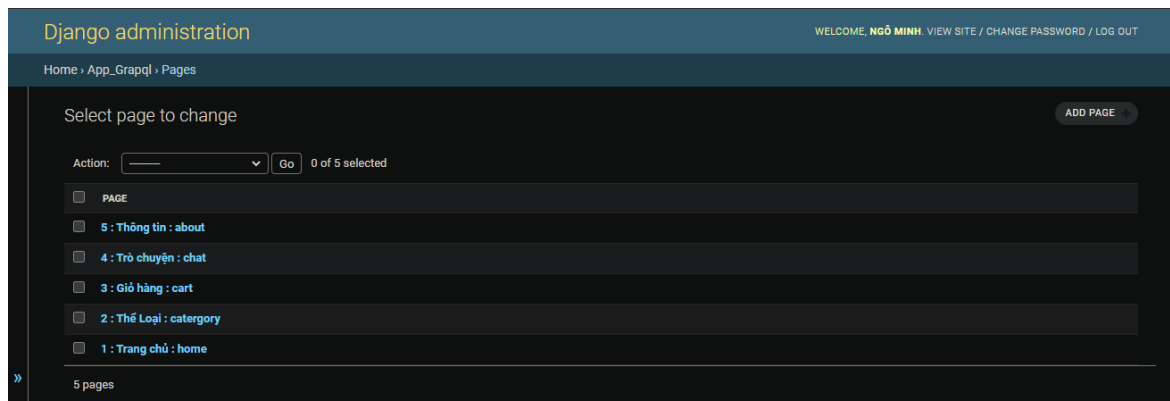


Hình 43 giao diện cập nhật đối tượng trạng thái người dùng

- admin.site.register(Page)

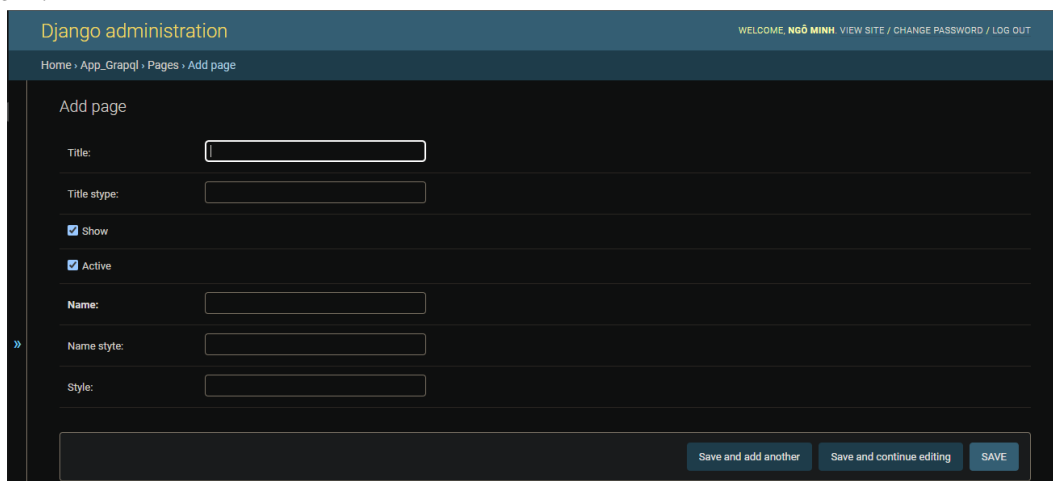
Đăng ký đối tượng Page với trình quản lý admin trong django

+ Xem danh sách:



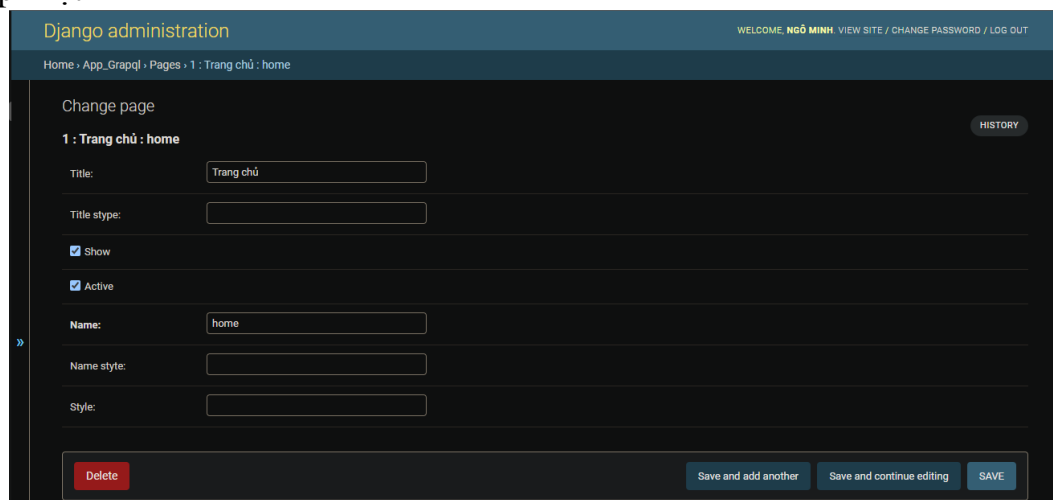
Hình 44 giao diện danh sách các đối tượng trang hiển thị

+ Thêm:



Hình 45 giao diện thêm đối tượng trang hiển thị

+ Cập nhật

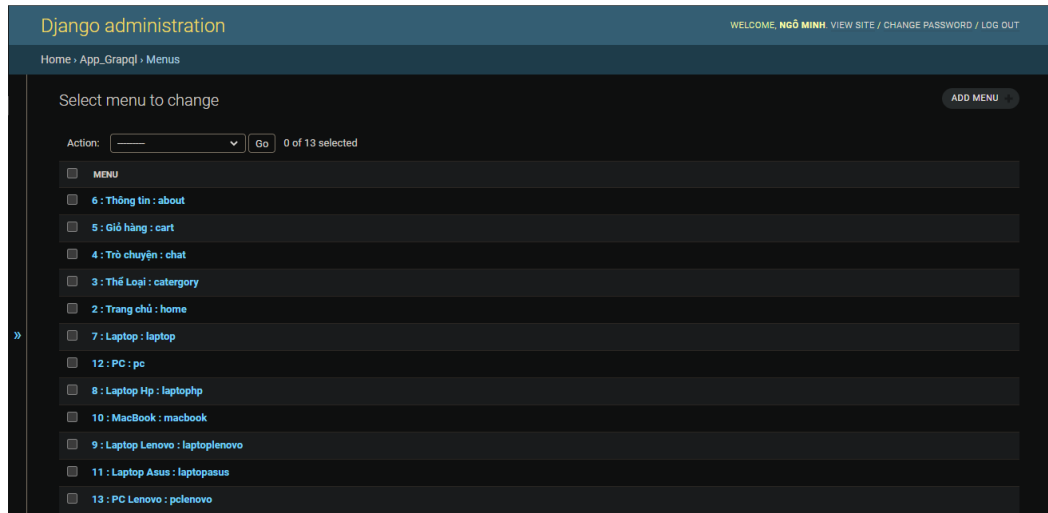


Hình 46 giao diện cập nhật đối tượng trang hiển thị

- admin.site.register(Menu)

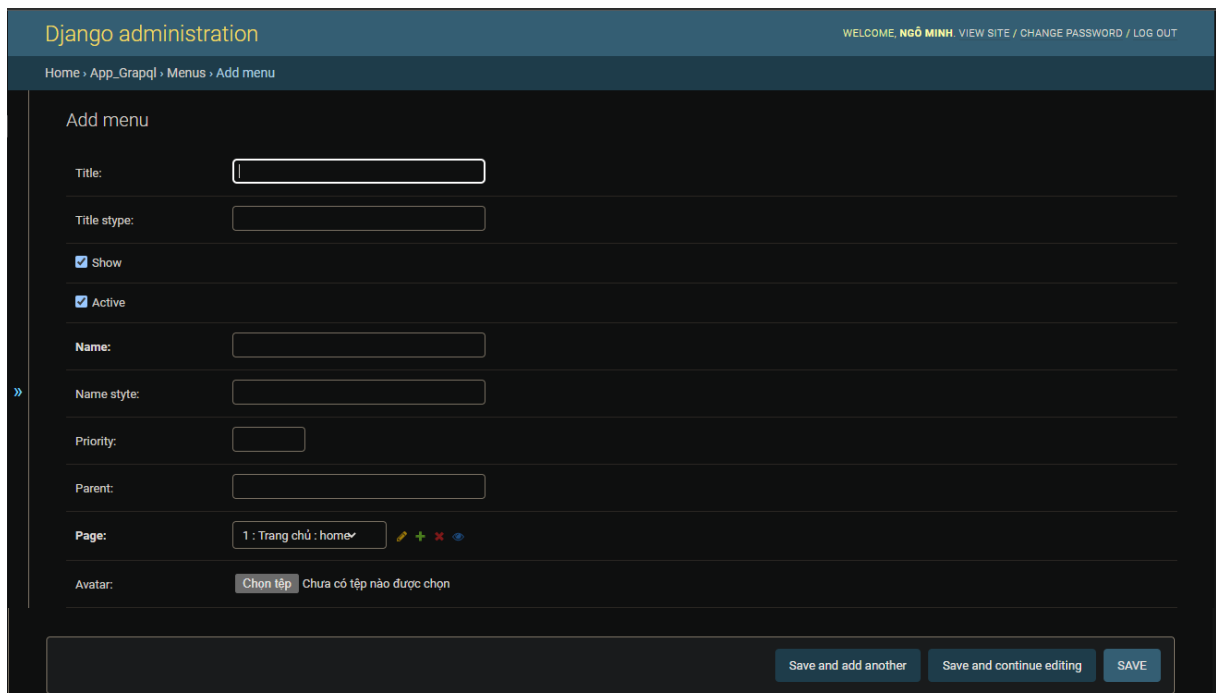
Đăng ký đối tượng Menu với trình quản lý admin trong django

+ Xem danh sách:



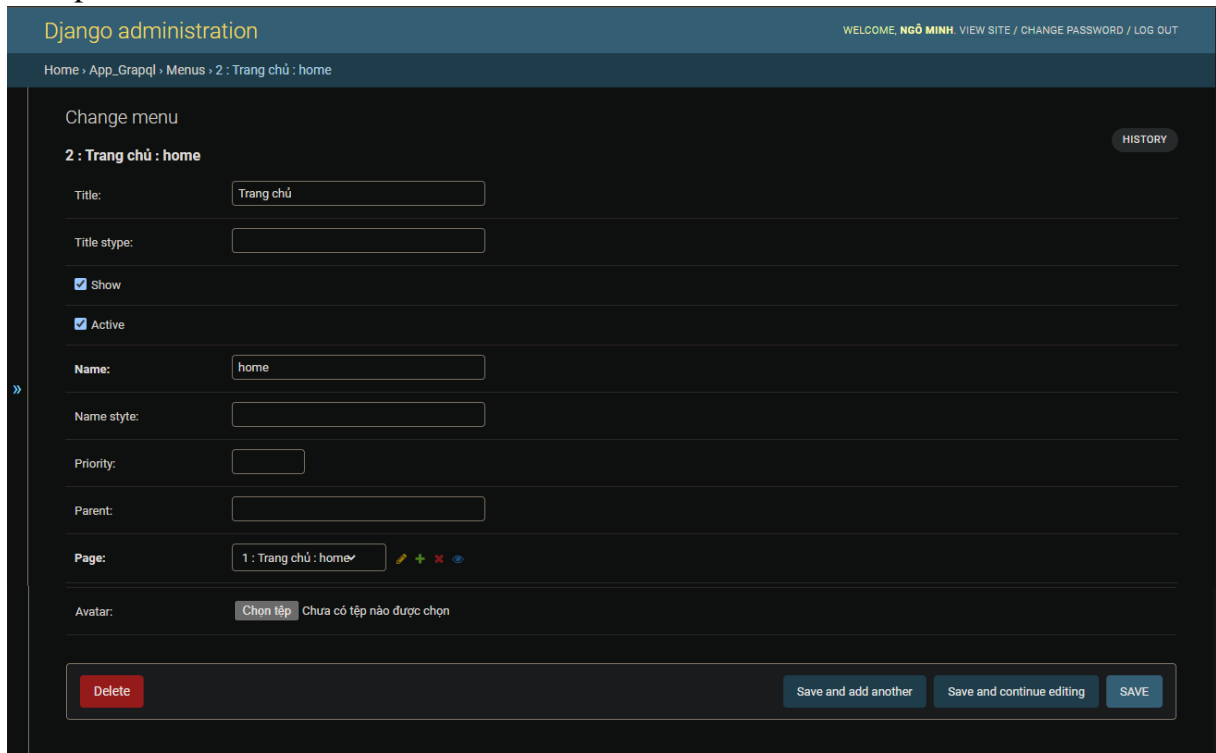
Hình 47 giao diện danh sách các đối tượng menu

+ Thêm:



Hình 48 giao diện thêm đối tượng menu

+ Cập nhật

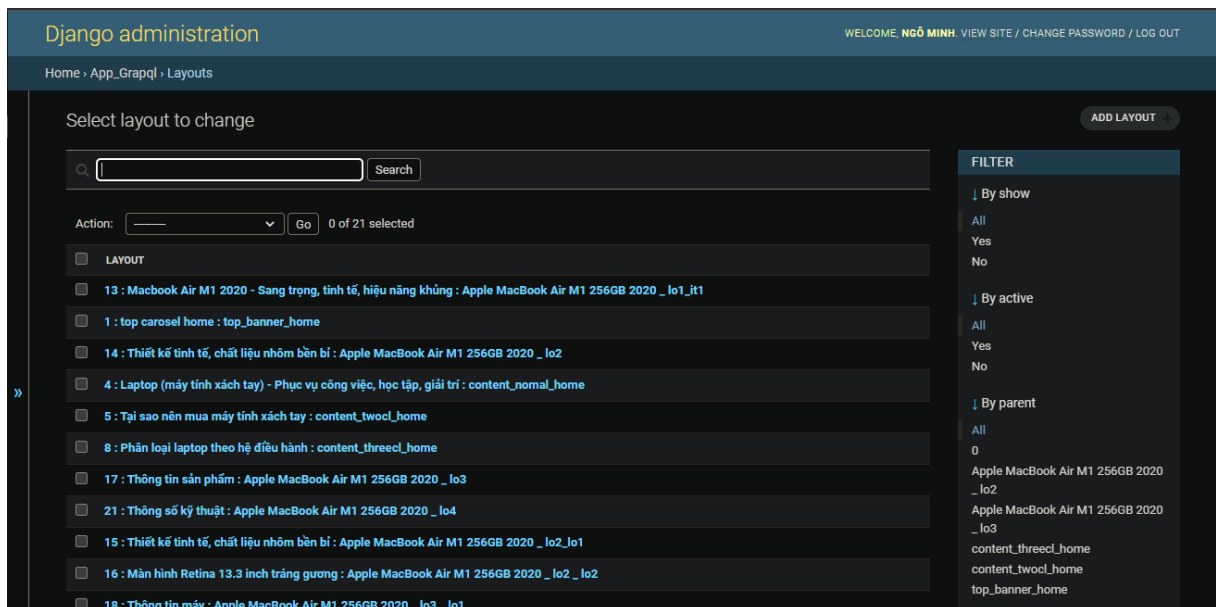


Hình 49 giao diện cập nhật đối tượng menu

- admin.site.register(Layout,layoutAdmin)

Đăng ký đối tượng Layout với trình quản lý admin trong django

+ Xem danh sách:



Hình 50 giao diện danh sách các đối tượng thành phần trang

+ Thêm:

Django administration WELCOME, NGÔ MINH / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > App_Grapql > Layouts > Add layout

Add layout

Title:

Title style:

Show

Active

Name:

Name style:

Priority:

Dest:

User: ✎ + ✖ ⌂

Dest style:

Style:

Parent:

Category: ✎ + ✖ ⌂

Background: Chưa có tệp nào được chọn

Interval:

Phần layouting

LAYOUT_IMAGES

Layout_img: #1

Title:

Title style:

Show

Active

Name:

Name style:

Avatar: Chưa có tệp nào được chọn


Dest:

Priority:

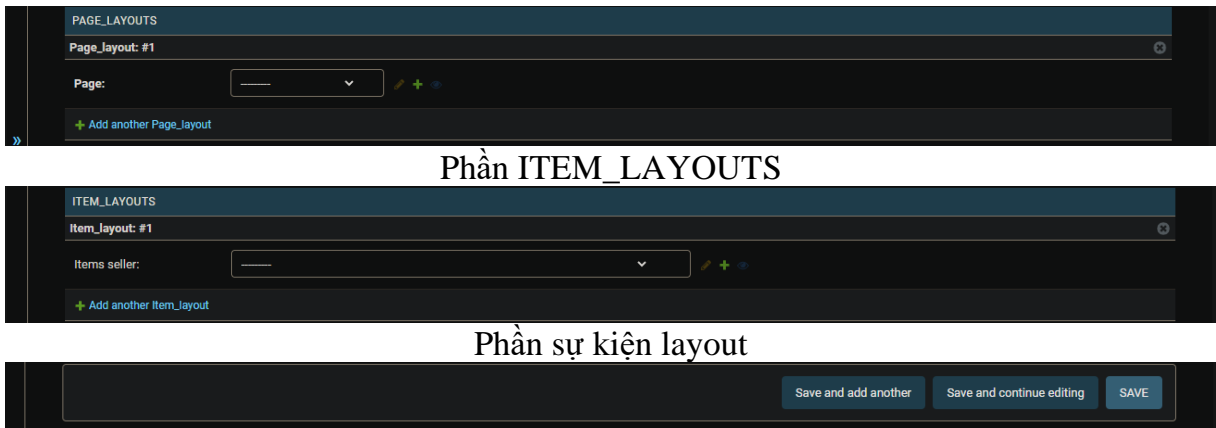
Priority:

Dest style:

Style:

Img read: 

Phần PAGE_LAYOUTS



Phần ITEM_LAYOUTS

Phần sự kiện layout

Hình 51 giao diện thêm đối tượng thành phần trang

+ Cập nhật

Django administration WELCOME NGÔ MINH VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > App_Grapql > Layouts > 10 : Laptop Macbook OS : content_threecl_home_lo2

Change layout

[HISTORY](#)

10 : Laptop Macbook OS : content_threecl_home_lo2

Title:

Title style:

Show

Active

Name:

Name style:

Priority:

Dest:

```
<p>Thông tin chung : "Ông hoàng" máy tính xách tay trong nhiều năm qua được nhiều người ưa chuộng chính là Apple. Họ đã vươn lên vị trí số 1 trong lòng người tiêu dùng bởi tính bền bỉ, hoàn thiện về phần cứng và hệ điều hành, thiết kế mỏng nhẹ, sang trọng. Bên cạnh đó công nghệ trên máy tính xách tay của Apple dẫn đầu các hãng khác, về bàn phím, chuột, chất lượng âm thanh, các cổng kết nối, độ phân giải màn hình. Hệ điều hành Mac OS phải nói là rất tuyệt, cho hiệu năng xử lý cao, ổn định, mượt mà, đặc biệt là tính bảo mật thông tin. Hỗ trợ tất cả công việc về văn phòng, thiết kế, lập trình.</p>
```

User: [+](#) [-](#) [x](#) [e](#)

Dest style:

Style:

Parent:

Category: [+](#) [-](#) [x](#) [e](#)

Background: Chưa có tệp nào được chọn

Interval:

LAYOUT_IMAGES

LAYOUT_img: 5 : Macbook 2016, Macbook 2017, Macbook Pro 2016, Macbook Pro 2017, Macbook Air 2016, Macbook Air 2017. : content_threecl_home_lo2_it1 [Delete](#)

Title:

Title style:

Show

Active

Name:

Name style:

Avatar: Currently: static/upload/Layout_img/2023/03/macbook1.jpg
Change: Chưa có tệp nào được chọn

Dest:

```
<p>hidden</p>
```

Dest:

Priority:

Dest style:

Style:

Img read:

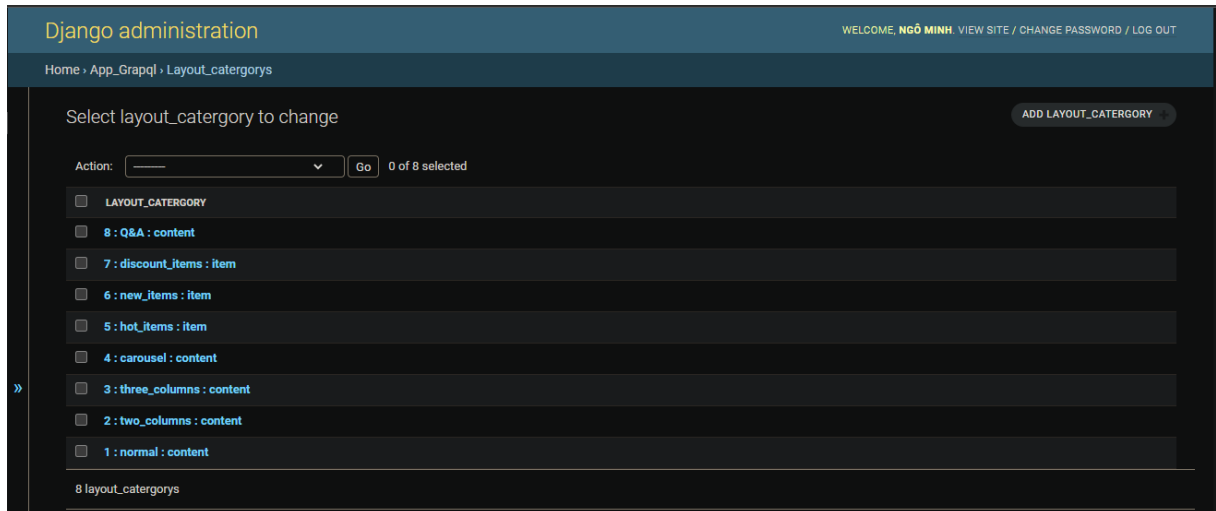
The image shows a dark-themed user interface for editing a page. At the top, there are several input fields: 'Dest' with the value '<p>hidden</p>', 'Priority' with '1', 'Dest style' with 'hidden', and 'Style' with 'lg.px-12'. Below these is a preview area labeled 'Img read:' showing a white background with the text 'The Evolution of the MacBook' and two laptop images. The bottom section contains a 'PAGE_LAYOUTS' section with 'Page_layout: 10 : Trang chủ : Laptop Macbook OS' and a 'Page:' dropdown set to '1 : Trang chủ : home'. Below that is an 'ITEM_LAYOUTS' section with 'Item_layout: #1' and an 'Items seller:' dropdown. At the very bottom, there is a bar with a red 'Delete' button, and three blue buttons: 'Save and add another', 'Save and continue editing', and 'SAVE'.

Hình 52 giao diện thêm đối tượng thành phần trang

- admin.site.register(Layout_category)

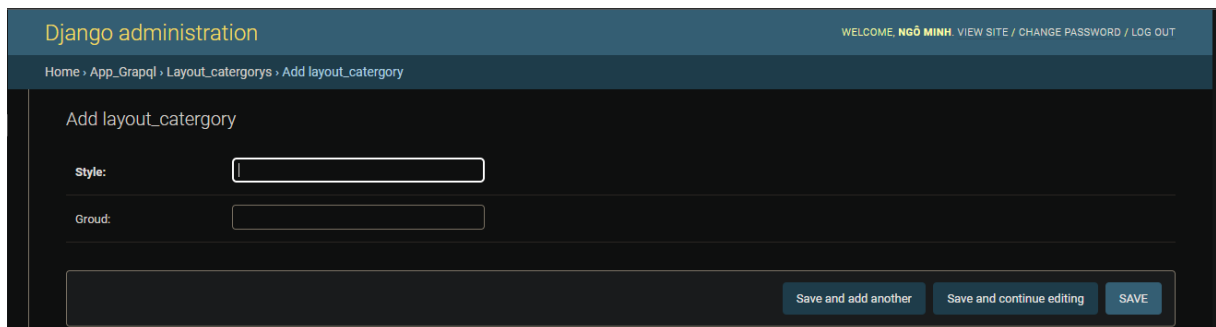
Đăng ký đối tượng Layout_category với trình quản lý admin trong django

+ Xem danh sách:



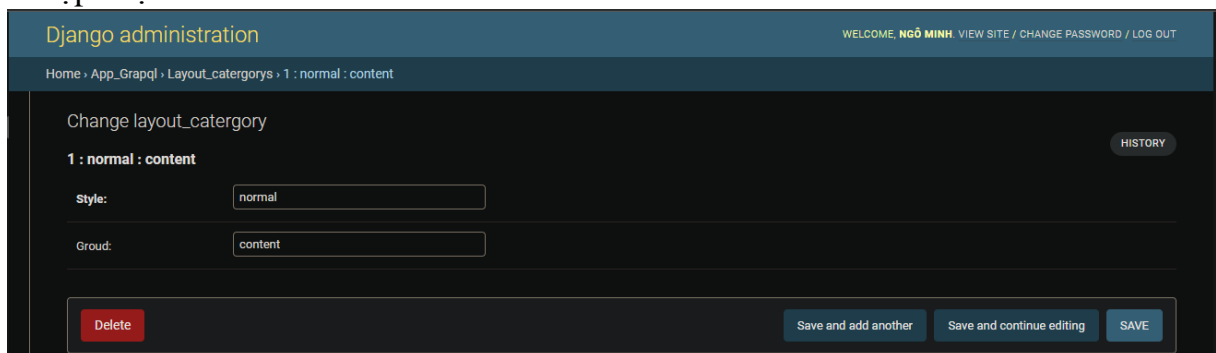
Hình 53 giao diện danh sách các đối tượng thể loại hiển thị thành phần trang

+ Thêm:



Hình 54 giao diện thêm đối tượng thể loại hiển thị thành phần trang

+ Cập nhật

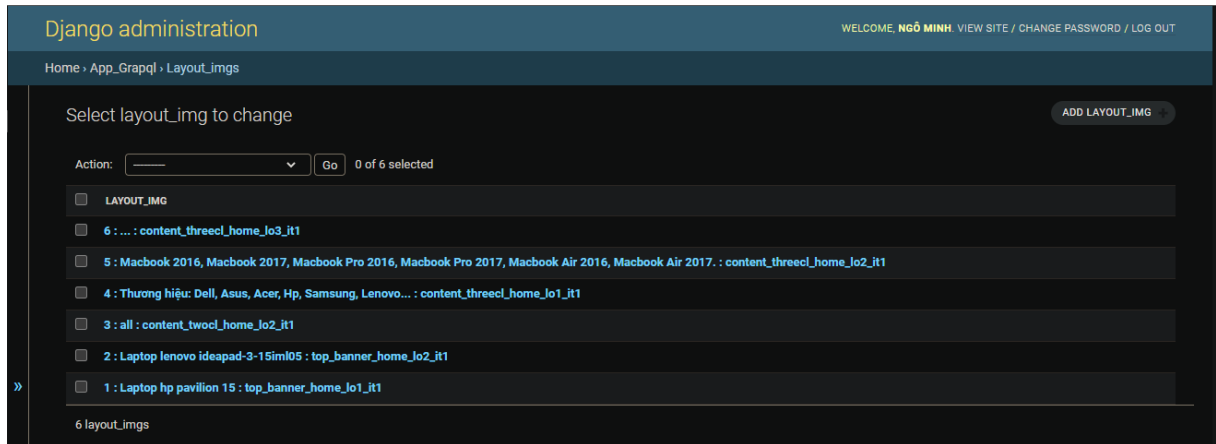


Hình 55 giao diện cập đối tượng thể loại hiển thị thành phần trang

- admin.site.register(Layout_img)

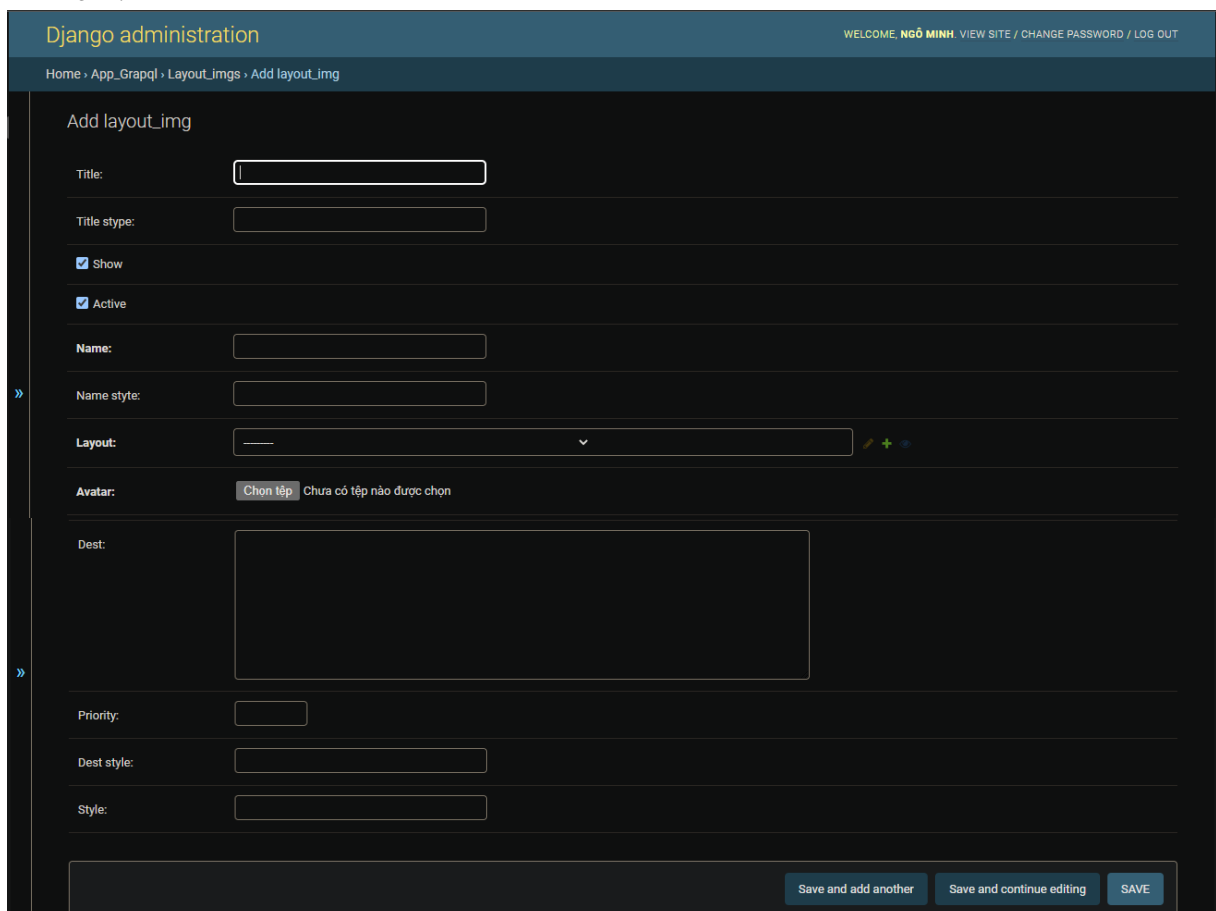
Đăng ký đối tượng Layout_img với trình quản lý admin trong django

+ Xem danh sách:



Hình 56 giao diện danh sách các đối tượng thành phần con trong thành phần trang

+ Thêm:



Hình 57 giao diện thêm đối tượng thành phần con trong thành phần trang

+ Cập nhật

The screenshot shows the Django administration interface for updating a `layout_img` object. The page title is "Change layout_img" and the breadcrumb is "Home > App_Grapql > Layout_imgs > 1 : Laptop hp pavilion 15 : top_banner_home_lo1_it1". The form contains the following fields and options:

- Title:** Laptop hp pavilion 15
- Title style:** top-1 lg:top-5 lg:bottom-[80px] absolute text-
- Show
- Active
- Name:** top_banner_home_lo1_it1
- Name style:** (empty)
- Layout:** 2 : Laptop hp : top_banner_home_lo1
- Avatar:** Currently: static/upload/Layout_img/2023/03/10050199-laptop-hp-pavilion-15-eg0513tu-3-1125g4-46m12_rv3D.JLX.jpg. Change: Chưa có tệp nào được chọn
- Dest:** <p>Sản phẩm đang hot</p>
- Priority:** 3
- Dest style:** text-left text-red-500 lg:ml-4
- Style:** w-1/3 right-2 rounded absolute

Buttons at the bottom: Delete, Save and add another, Save and continue editing, SAVE.

Hình 58 giao diện cập nhật đối tượng thành phần con trong thành phần trang

- `admin.site.register(Page_layout)`

Đăng ký đối tượng `Page_layout` với trình quản lý admin trong django

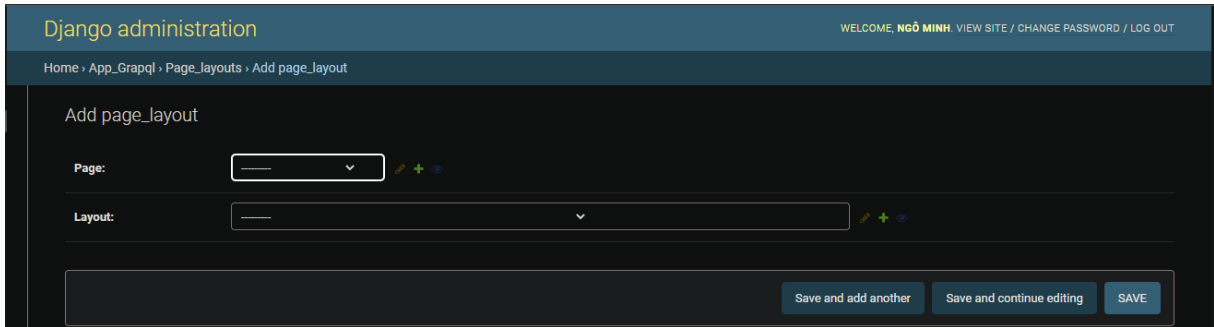
+ Xem danh sách:

The screenshot shows the Django administration interface for viewing a list of `page_layout` objects. The page title is "Select page_layout to change" and the breadcrumb is "Home > App_Grapql > Page_layouts". The interface includes an "Action:" dropdown menu, a "Go" button, and a selection count "0 of 12 selected". The list of objects is as follows:

Object ID	Object Name
PAGE_LAYOUT	PAGE_LAYOUT
16	Trang chủ : test
11	Trang chủ : Laptop Chrome OS
10	Trang chủ : Laptop Macbook OS
9	Trang chủ : Laptop Windows
8	Trang chủ : Phân loại laptop theo hệ điều hành
7	Trang chủ : Các loại máy tính laptop phổ biến
6	Trang chủ : Lợi ích khi sử dụng laptop là gì?
5	Trang chủ : Tại sao nên mua máy tính xách tay
4	Trang chủ : Laptop (máy tính xách tay) - Phục vụ công việc, học tập, giải trí
3	Trang chủ : Laptop Lenovo
2	Trang chủ : Laptop hp
1	Trang chủ : top carousel home

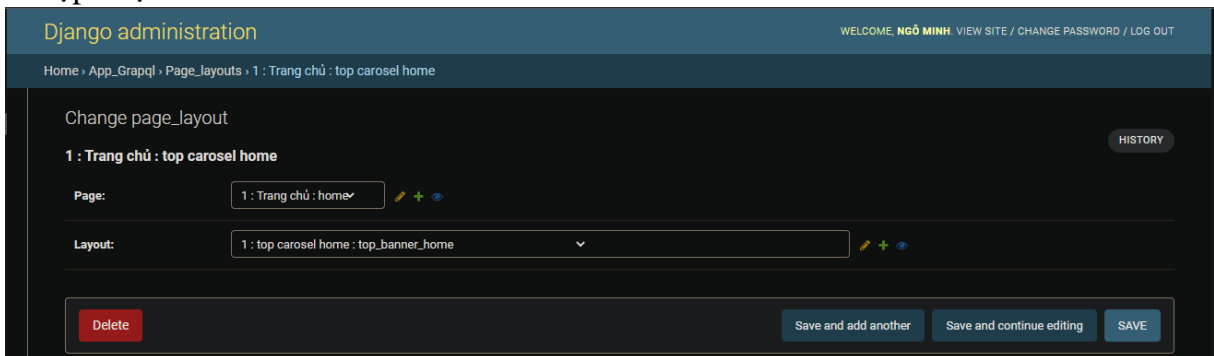
Hình 59 giao diện danh sách các đối tượng quy định các thành phần trong trang

+ Thêm:



Hình 60 giao diện thêm đối tượng quy định các thành phần trong trang

+ Cập nhật

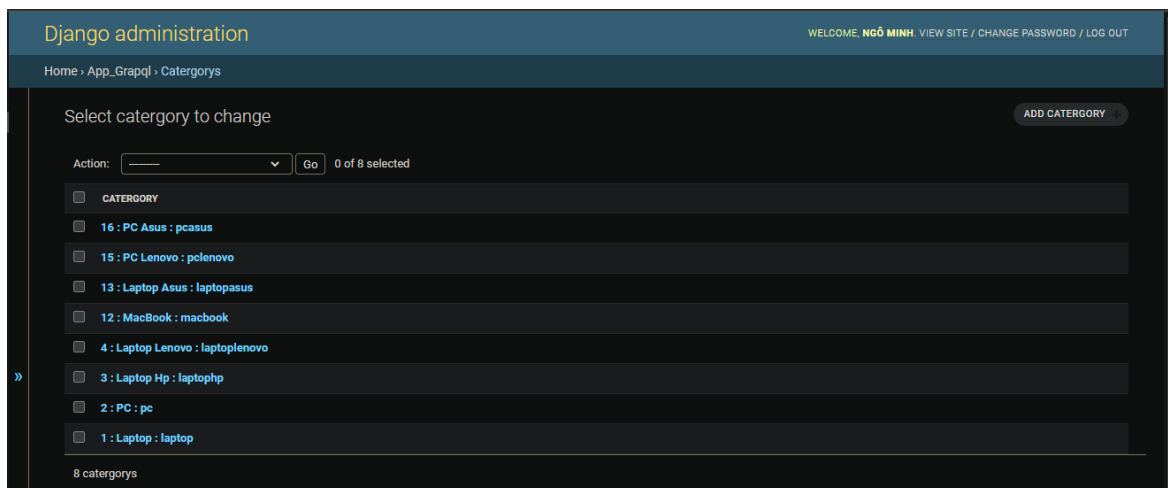


Hình 61 giao diện cập nhật đối tượng quy định các thành phần trong trang

- admin.site.register(Category)

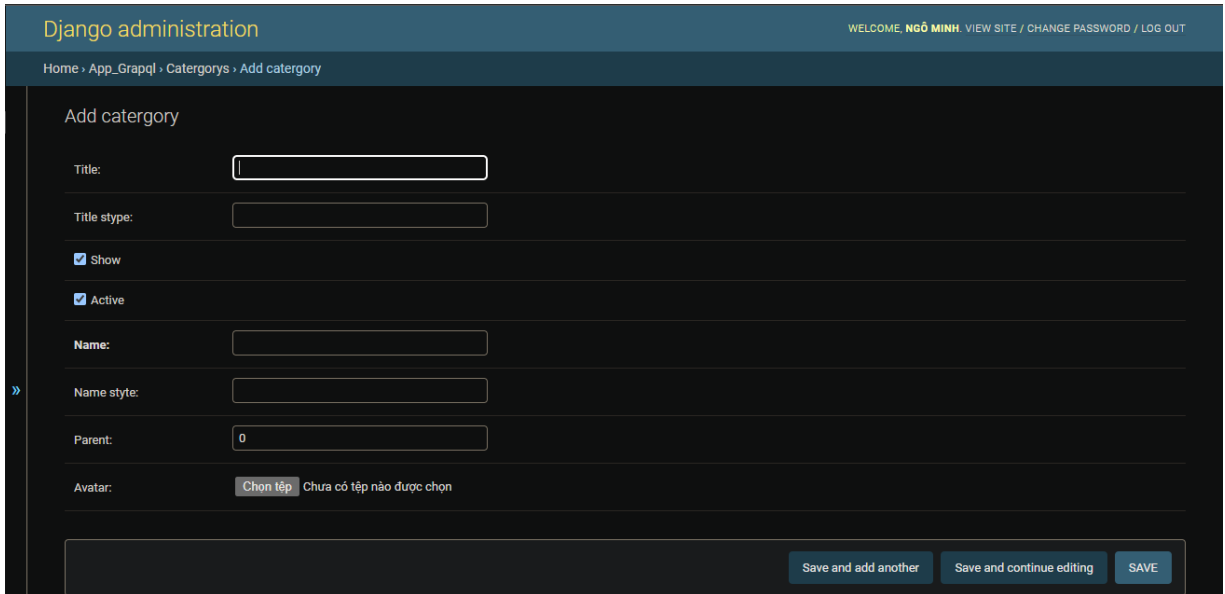
Đăng ký đối tượng Category với trình quản lý admin trong django

+ Xem danh sách:



Hình 62 giao diện danh sách các đối tượng thể loại sản phẩm

+ Thêm:



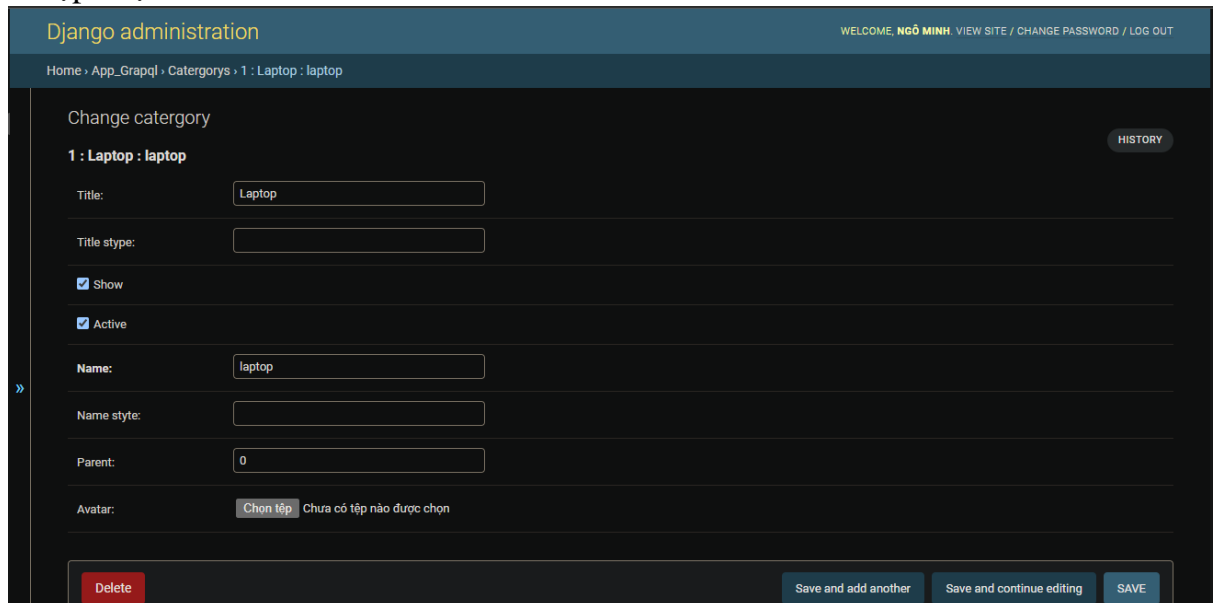
The screenshot shows the Django administration interface for adding a new category. The page title is "Django administration" and the user is logged in as "NGÔ MINH". The breadcrumb trail is "Home > App_Grapql > Categories > Add category". The form is titled "Add category" and contains the following fields and options:

- Title:
- Title stype:
- Show
- Active
- Name:
- Name styte:
- Parent:
- Avatar: Chưa có tệp nào được chọn

At the bottom of the form, there are three buttons: "Save and add another", "Save and continue editing", and "SAVE".

Hình 63 giao diện thêm đối tượng thể loại sản phẩm

+ Cập nhật



The screenshot shows the Django administration interface for updating an existing category. The page title is "Django administration" and the user is logged in as "NGÔ MINH". The breadcrumb trail is "Home > App_Grapql > Categories > 1 : Laptop : laptop". The form is titled "Change category" and includes a "HISTORY" button. The form is for the category "1 : Laptop : laptop" and contains the following fields and options:

- Title:
- Title stype:
- Show
- Active
- Name:
- Name styte:
- Parent:
- Avatar: Chưa có tệp nào được chọn

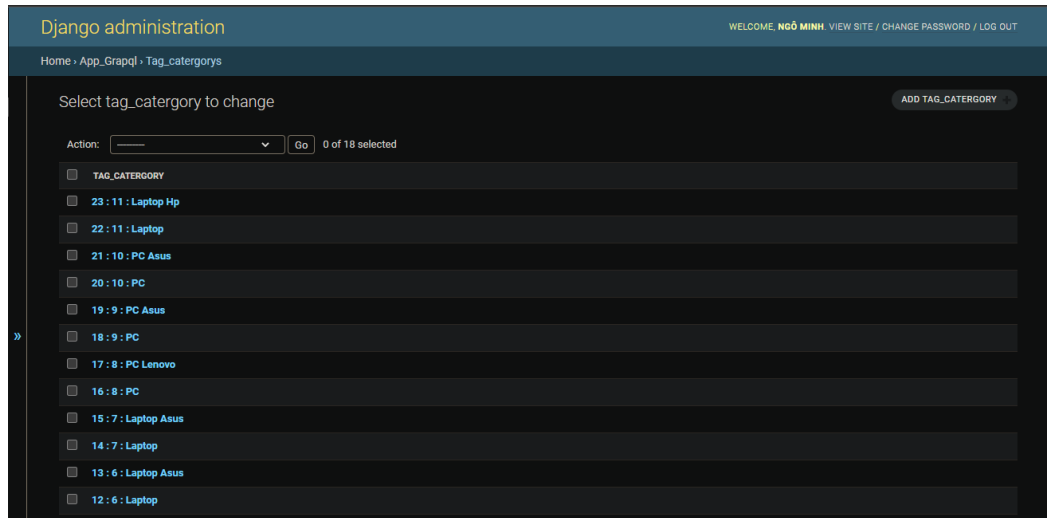
At the bottom of the form, there are four buttons: "Delete", "Save and add another", "Save and continue editing", and "SAVE".

Hình 64 giao diện cập nhật đối tượng thể loại sản phẩm

- admin.site.register(Tag_catergory)

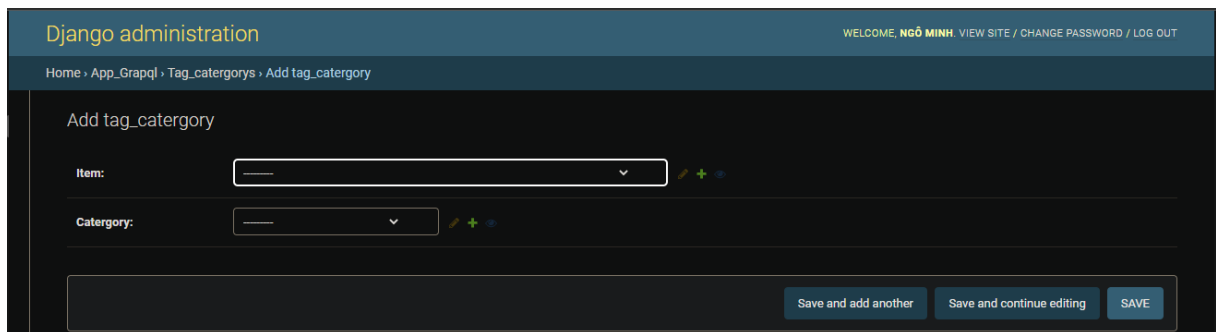
Đăng ký đối tượng Tag_catergory với trình quản lý admin trong django

+ Xem danh sách:



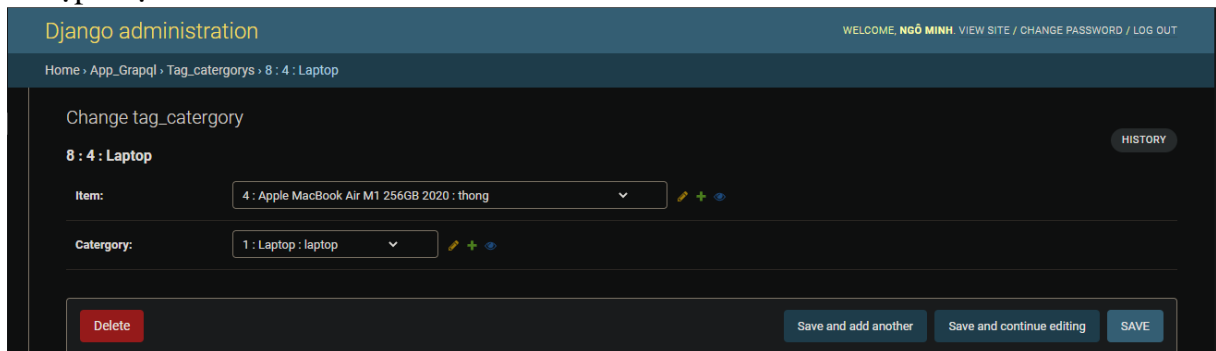
Hình 65 giao diện danh sách các đối tượng thẻ thể loại

+ Thêm:



Hình 66 giao diện thêm đối tượng thẻ thể loại

+ Cập nhật

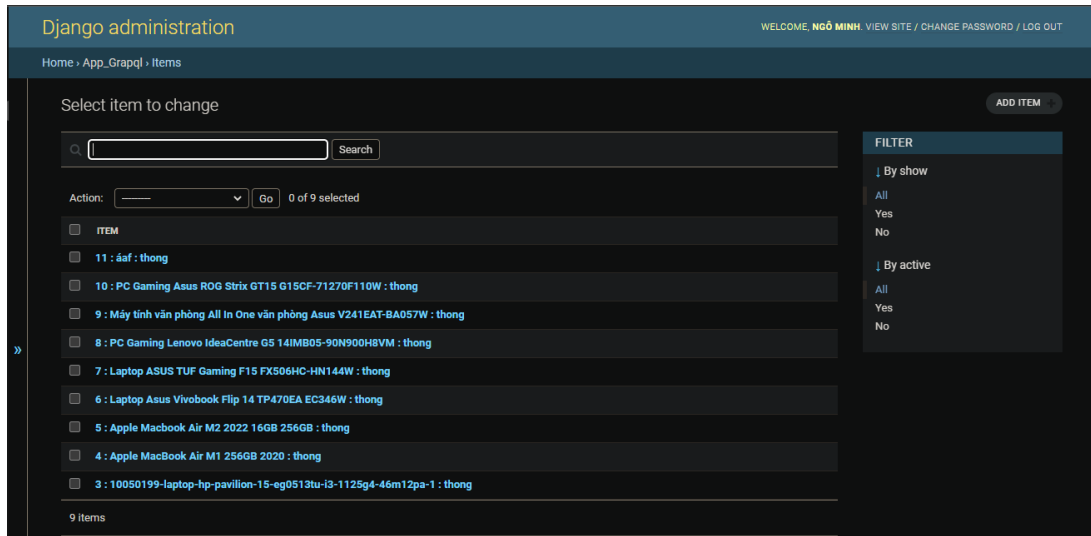


Hình 67 giao diện cập nhật đối tượng thẻ thể loại

- admin.site.register(Item,itemAdmin)

Đăng ký đối tượng Item với trình quản lý admin trong django

+ Xem danh sách:



Hình 68 giao diện danh sách các đối tượng sản phẩm

+ Thêm:

The screenshot shows the Django administration interface for adding a new item. The page title is "Django administration" and the user is logged in as "NGO MINH". The breadcrumb trail is "Home > App_Grapql > Items > Add Item". The form is titled "Add item" and contains the following fields:

- Title:
- Title style:
- Show
- Active
- Name:
- Name style:
- Avatar: Chưa có tệp nào được chọn
- Price:
- Price promotion:
- Number:
- Dest:
- Dest style:
- Supplier:
- Staatua:
- Img read:
- Tag_category: #1
- Category:

At the bottom of the form, there is a link "+ Add another Tag_category" and three buttons: "Save and add another", "Save and continue editing", and "SAVE".

Hình 69 giao diện thêm đối tượng sản phẩm

+ Cập nhật

Django administration WELCOME NGÔ MINH | VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > App_Grapql > Items > 4 : Apple MacBook Air M1 256GB 2020 : thông

Change item

4 : Apple MacBook Air M1 256GB 2020 : thông HISTORY

Title:

Title type:

Show

Active

Name:

Name style:

Avatar: Currently: static/upload/product/2023/03/air_m2.webp
Change: Chưa có tệp nào được chọn

Price:

Price promotion:

Number:

Dest:


```
<div>
<ul>
<li>Ph&grave; hợp cho lập tr&grave;nh vi&circ;nh thiết kế đồ họa 2D, đ&grave; văn
ph&grave;ng</li>
<li>Hiệu năng vượt trội - C&grave; mọi t&grave; vụ từ word, excel đến chỉnh sửa ảnh tr&grave;
c&grave; phần mềm như AI, PTS</li>
<li>Đ&grave; nhiệm mượt m&grave; - Ram 8GB cho ph&grave; vừa mở tr&grave;nh duyệt để tra cứu
th&grave;ng tin, vừa l&grave; việc tr&grave;nh phần mềm</li>
<li>Trang bị SSD 256GB - Cho thời gian khởi động nhanh ch&grave;ng, tối ưu ho&grave; thời gian
load ứng dụng</li>

```

Dest style:

Supplier:

Status:

Img read: 

TAG_CATERGORYS

Tag_category: 8 : 4 : Laptop

Category:

Tag_category: 9 : 4 : MacBook

Category:

Tag_category: #3

Category:

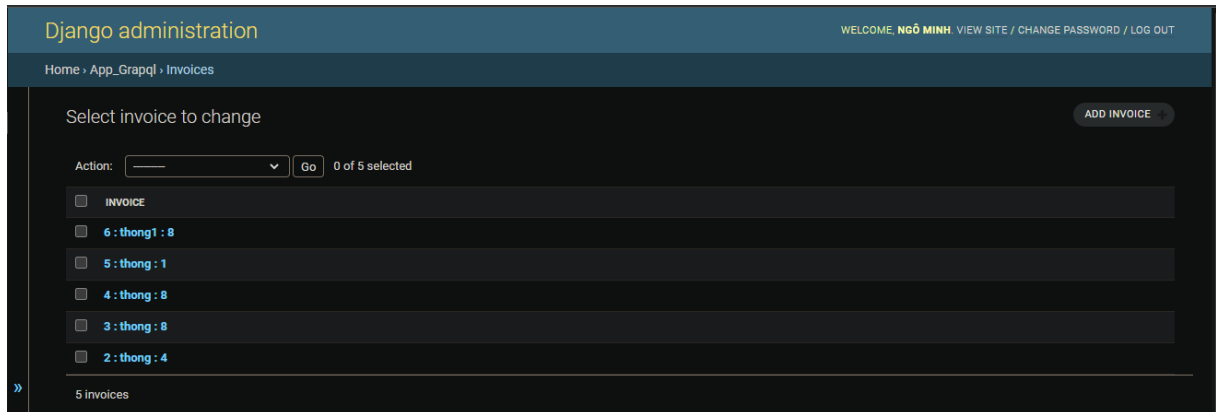
[+ Add another Tag_category](#)

Hình 70 giao diện cập nhật đối tượng sản phẩm

- admin.site.register(Invoice)

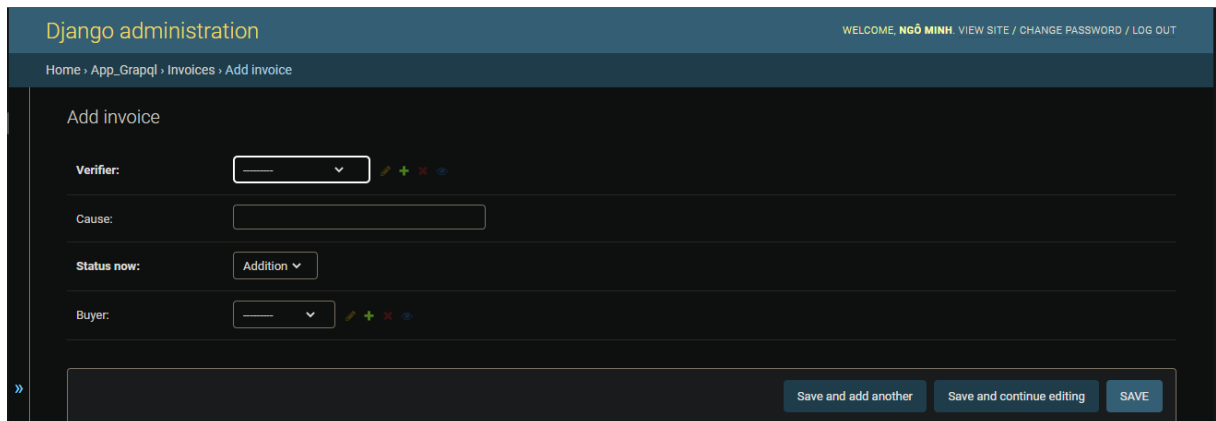
Đăng ký đối tượng Invoice với trình quản lý admin trong django

+ Xem danh sách:



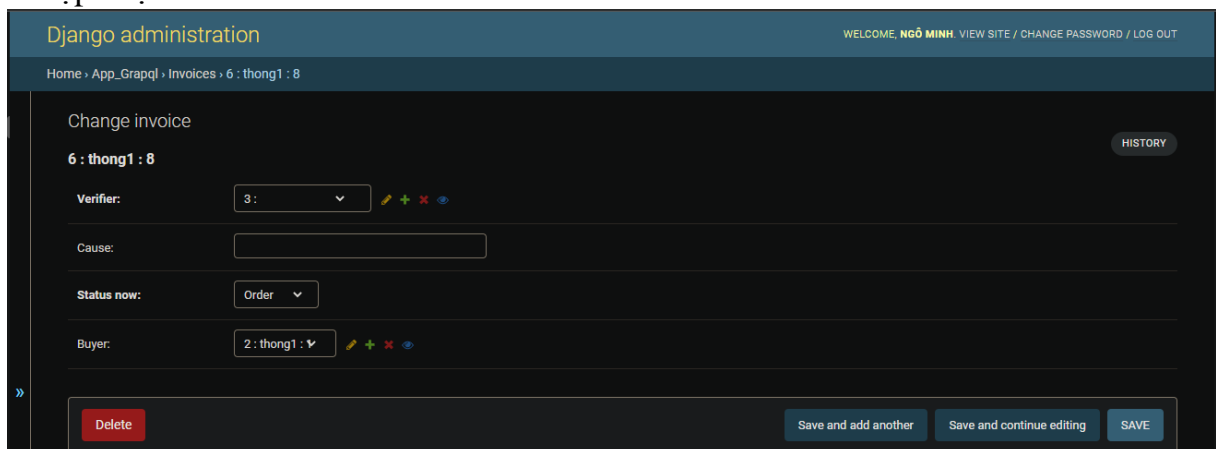
Hình 71 giao diện danh sách các đối tượng sản phẩm

+ Thêm:



Hình 72 giao diện thêm đối tượng sản phẩm

+ Cập nhật

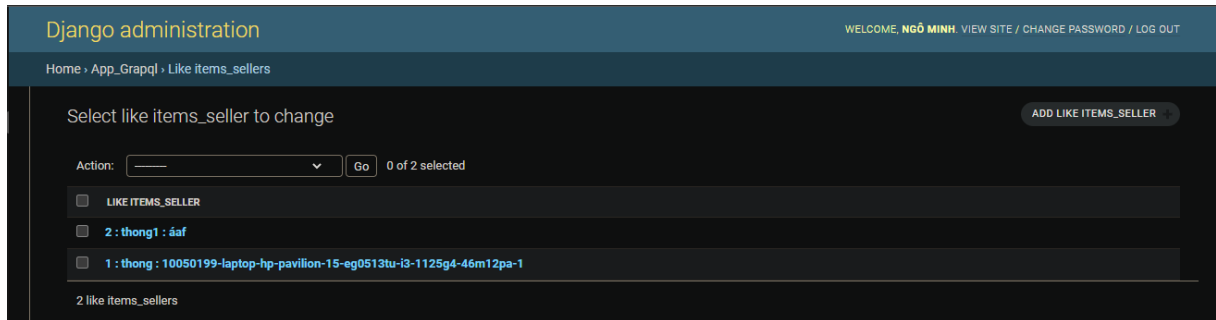


Hình 73 giao diện cập nhật đối tượng sản phẩm

- admin.site.register(LikeItems_seller)

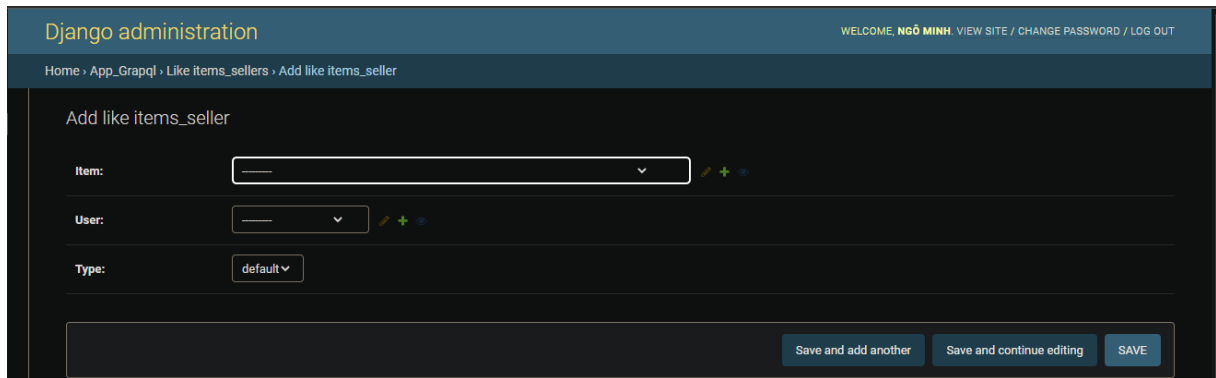
Đăng ký đối tượng LikeItems_seller với trình quản lý admin trong django

+ Xem danh sách:



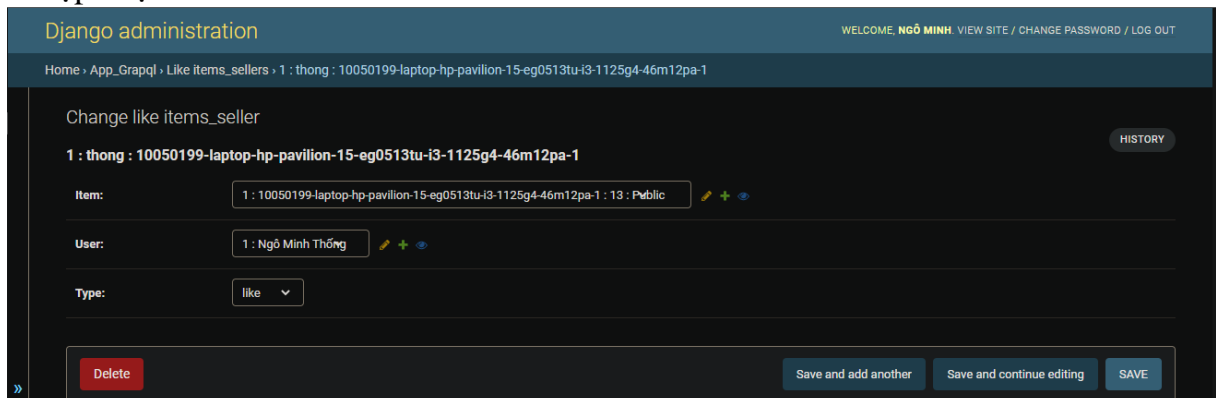
Hình 74 giao diện danh sách các đối tượng thích các sản phẩm

+ Thêm:



Hình 75 giao diện thêm đối tượng thích các sản phẩm

+ Cập nhật

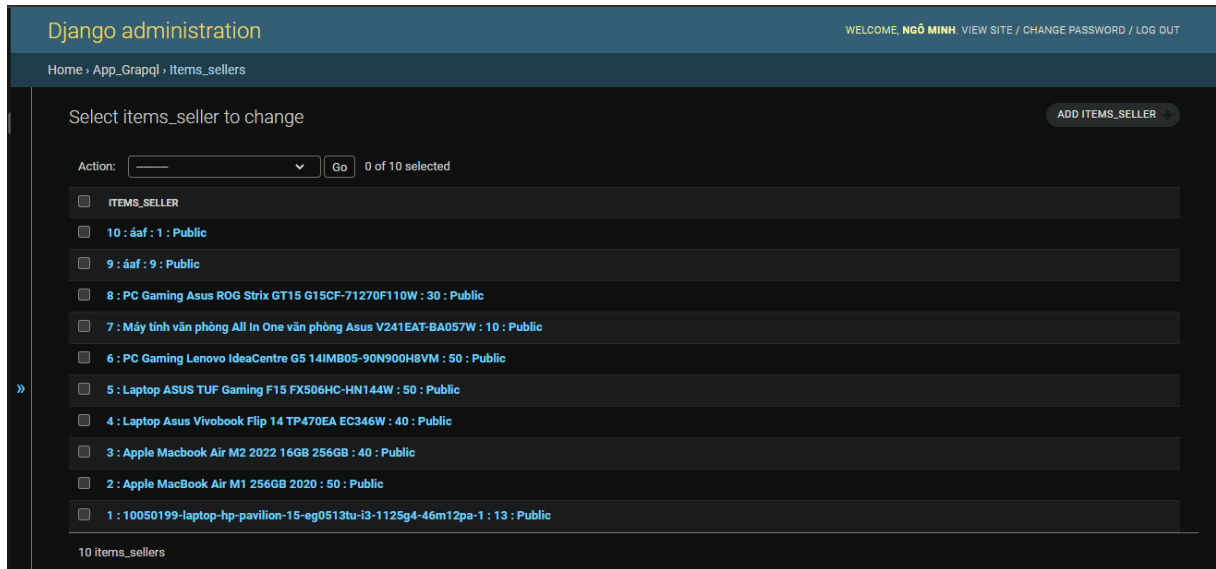


Hình 76 giao diện cập nhật đối tượng thích các sản phẩm

- admin.site.register(Items_seller)

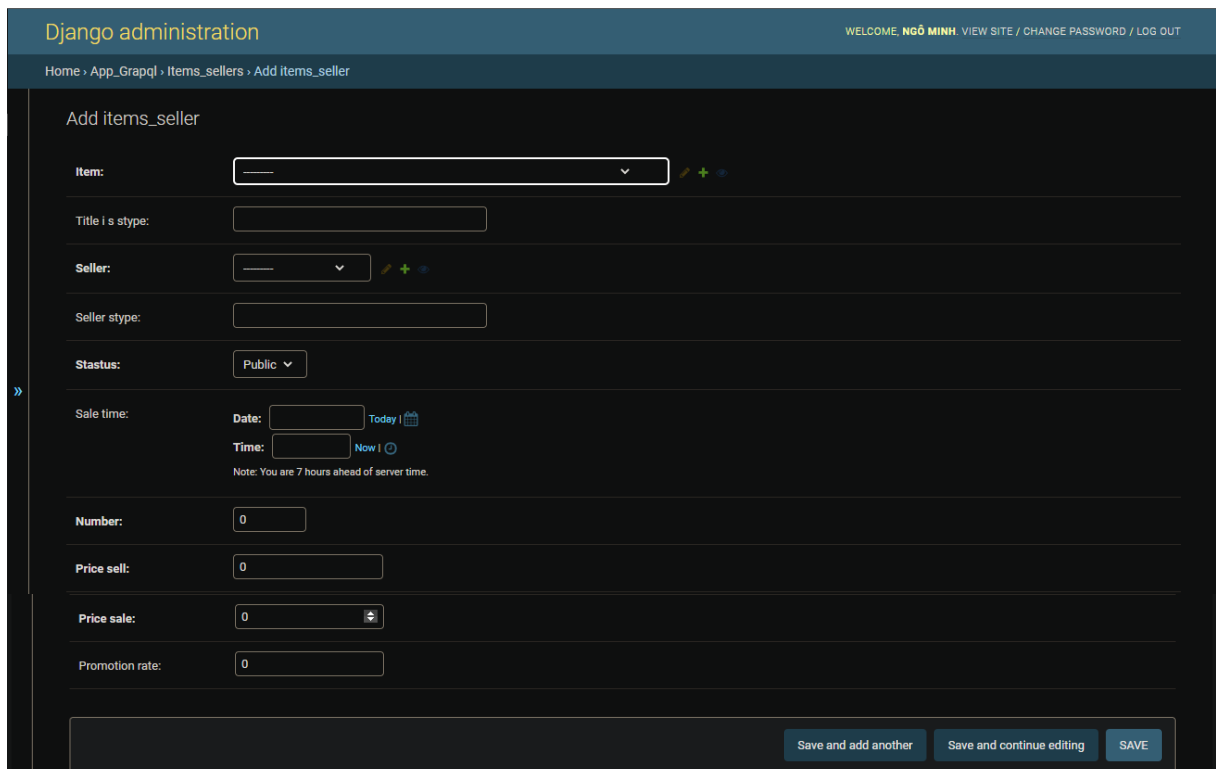
Đăng ký đối tượng Items_seller với trình quản lý admin trong django

+ Xem danh sách:



Hình 77 giao diện danh sách các đối tượng sản phẩm bán ra

+ Thêm:



Hình 78 giao diện thêm đối tượng sản phẩm bán ra

+ Cập nhật

The screenshot shows the Django administration interface for updating an item seller. The page title is "Change items_seller" and the breadcrumb is "Home > App_Grapql > Items_sellers > 1 : 10050199-laptop-hp-pavilion-15-eg0513tu-i3-1125g4-46m12pa-1 : 13 : Public". The form contains the following fields:

- Item: 3 : 10050199-laptop-hp-pavilion-15-eg0513tu-i3-1125g4-46m12pa-1 : thong
- Title is stype: (empty)
- Seller: 1 : Ngô Minh Thống
- Seller stype: (empty)
- Status: Public
- Sale time: Date: (empty) Today | Time: (empty) Now | Note: You are 7 hours ahead of server time.
- Number: 13
- Price sell: 13000000,0
- Price sale: 0,0
- Promotion rate: 0,0

At the bottom, there are buttons for "Delete", "Save and add another", "Save and continue editing", and "SAVE".

Hình 79 giao diện cập nhật đối tượng sản phẩm bán ra

- admin.site.register(Item_layout)

Đăng ký đối tượng Item_layout với trình quản lý admin trong django

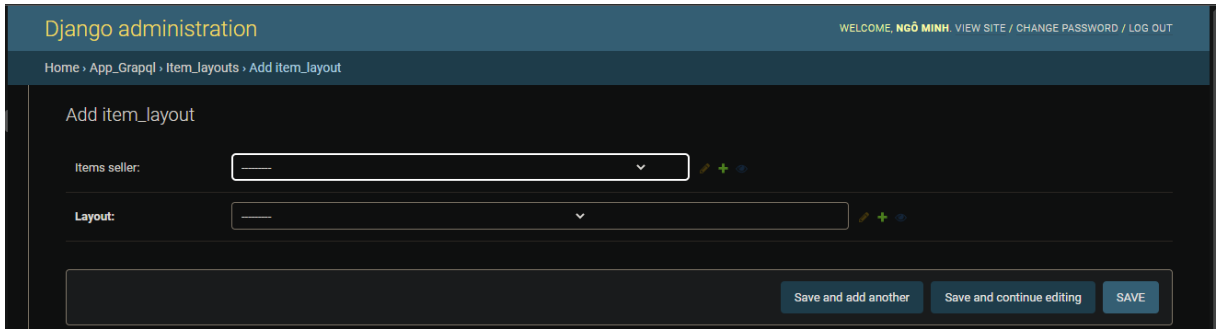
+ Xem danh sách:

The screenshot shows the Django administration interface for viewing a list of item layouts. The page title is "Select item_layout to change" and the breadcrumb is "Home > App_Grapql > Item_layouts". The interface includes an "ADD ITEM_LAYOUT" button and an "Action:" dropdown menu with a "Go" button and "0 of 12 selected" text. The list of item layouts is as follows:

- ITEM_LAYOUT
- 14 : Laptop Lenovo
- 13 : Laptop hp
- 12 : top carousel home
- 10 : Thông số kỹ thuật
- 9 : Chế độ bảo hành
- 8 : Phụ kiện
- 7 : Thông tin máy
- 6 : Thông tin sản phẩm
- 5 : Màn hình Retina 13.3 inch tráng gương
- 4 : Thiết kế tinh tế, chất liệu nhôm bền bỉ
- 3 : Thiết kế tinh tế, chất liệu nhôm bền bỉ
- 2 : Macbook Air M1 2020 - Sang trọng, tinh tế, hiệu năng khủng

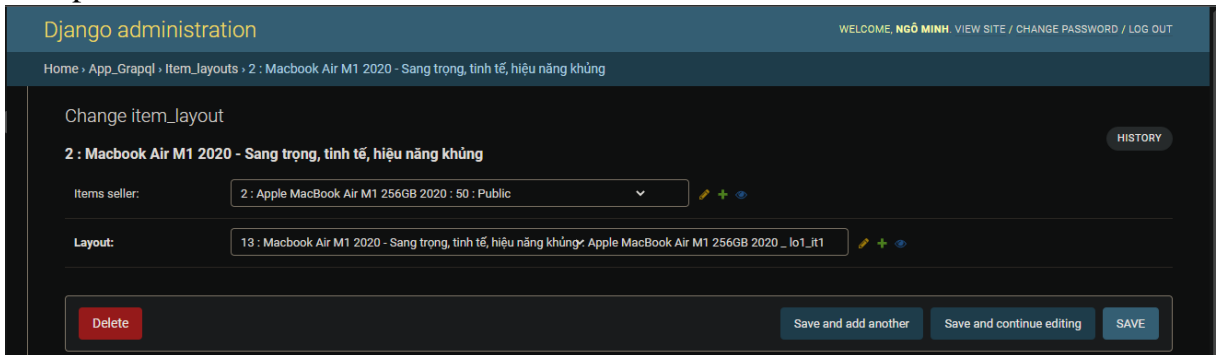
Hình 80 giao diện danh sách các đối tượng thành phần trang theo sản phẩm

+ Thêm:



Hình 81 giao diện thêm đối tượng thành phần trang theo sản phẩm

+ Cập nhật

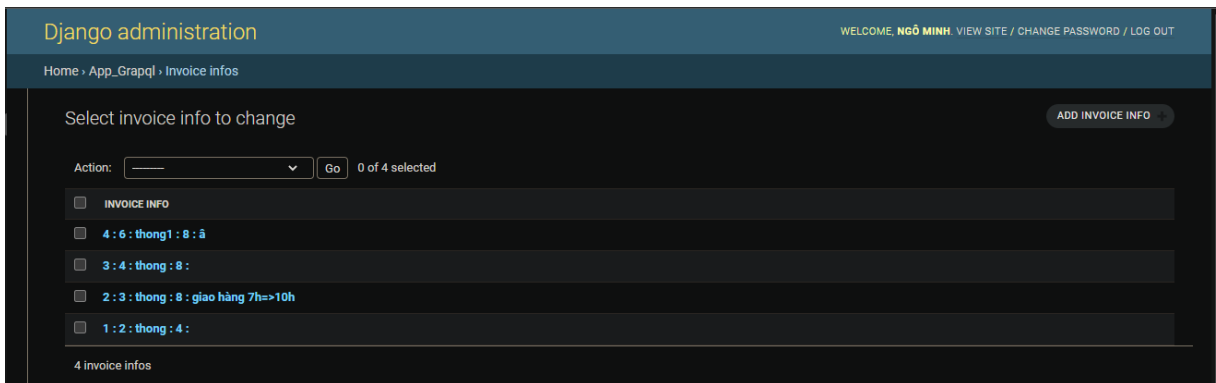


Hình 82 giao diện cập nhật đối tượng thành phần trang theo sản phẩm

- admin.site.register(InvoiceInfo)

Đăng ký đối tượng InvoiceInfo với trình quản lý admin trong django

+ Xem danh sách:



Hình 83 giao diện danh sách các đối tượng thông tin giao hàng

+ Thêm:

The screenshot shows the Django administration interface for adding a new invoice item. The breadcrumb trail is 'Home > App_Grapql > Invoice Infos > Add invoice info'. The form includes fields for 'Invoice' (a dropdown menu), 'Fullname', 'Address', 'Phone' (with a '0' in the input), 'Email', and 'Cause'. At the bottom right, there are three buttons: 'Save and add another', 'Save and continue editing', and 'SAVE'.

Hình 84 giao diện thêm đối tượng thông tin giao hàng

+ Cập nhật

The screenshot shows the Django administration interface for updating an existing invoice item. The breadcrumb trail is 'Home > App_Grapql > Invoice Infos > 2 : 3 : thông : 8 : giao hàng 7h=>10h'. The form title is 'Change invoice info'. The 'Invoice' dropdown is set to '3 : thông : 8'. The other fields are filled with: 'Fullname: Ngô Minh Thống', 'Address: Bà Rịa Vũng Tàu', 'Phone: 9618112', 'Email: petomo2003@gmail.com', and 'Cause: giao hàng 7h=>10h'. A 'HISTORY' button is visible in the top right. At the bottom, there is a 'Delete' button and 'Save and add another', 'Save and continue editing', and 'SAVE' buttons.

Hình 85 giao diện cập nhật đối tượng thông tin giao hàng

- admin.site.register(Invoice_item)

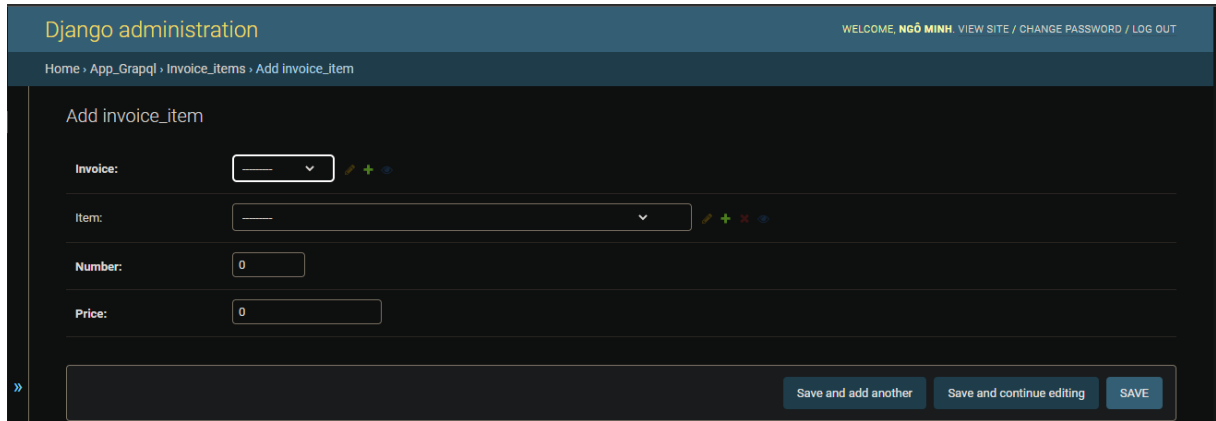
Đăng ký đối tượng Invoice_item với trình quản lý admin trong django

+ Xem danh sách:

The screenshot shows the Django administration interface for selecting an invoice item to edit. The breadcrumb trail is 'Home > App_Grapql > Invoice_items'. The page title is 'Select invoice_item to change'. There is an 'ADD INVOICE_ITEM' button in the top right. Below the title, there is an 'Action:' dropdown menu and a 'Go' button, with '0 of 9 selected' indicated. A list of 9 invoice items is displayed, each with a checkbox and a label. The labels include 'INVOICE_ITEM', '18 : 6 : áaf', '17 : 6 : áaf', '16 : 5 : PC Gaming Lenovo IdeaCentre G5 14IMB05-90N900H8VM', '15 : 5 : Apple MacBook Air M2 2022 16GB 256GB', '14 : 5 : Apple MacBook Air M1 256GB 2020', '13 : 5 : 10050199-laptop-hp-pavillon-15-eg0513tu-i3-1125g4-46m12pa-1', '11 : 4 : 10050199-laptop-hp-pavillon-15-eg0513tu-i3-1125g4-46m12pa-1', '10 : 3 : 10050199-laptop-hp-pavillon-15-eg0513tu-i3-1125g4-46m12pa-1', and '6 : 2 : 10050199-laptop-hp-pavillon-15-eg0513tu-i3-1125g4-46m12pa-1'. At the bottom, it says '9 invoice_items'.

Hình 86 giao diện danh sách các đối tượng sản phẩm trong hóa đơn

+ Thêm:



The screenshot shows the Django administration interface for adding a new invoice item. The page title is 'Django administration' and the user is 'NGÔ MINH'. The breadcrumb trail is 'Home > App_Grapql > Invoice_items > Add invoice_item'. The form fields are: 'Invoice:' (a dropdown menu), 'Item:' (a dropdown menu), 'Number:' (a text input with '0'), and 'Price:' (a text input with '0'). At the bottom right, there are three buttons: 'Save and add another', 'Save and continue editing', and 'SAVE'.

Hình 87 giao diện thêm đối tượng sản phẩm trong hóa đơn

+ Cập nhật



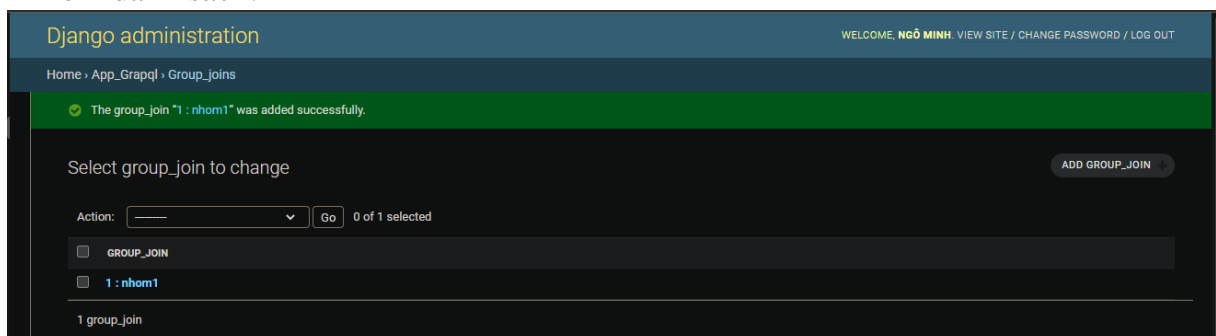
The screenshot shows the Django administration interface for updating an existing invoice item. The page title is 'Django administration' and the user is 'NGÔ MINH'. The breadcrumb trail is 'Home > App_Grapql > Invoice_items > 6 : 2 : 10050199-laptop-hp-pavilion-15-eg0513tu-i3-1125g4-46m12pa-1'. The form fields are: 'Invoice:' (a dropdown menu with '2 : thong : 4'), 'Item:' (a dropdown menu with '1 : 10050199-laptop-hp-pavilion-15-eg0513tu-i3-1125g4-46m12pa-1 : 13 : Public'), 'Number:' (a text input with '1'), and 'Price:' (a text input with '13000000,0'). At the bottom left, there is a red 'Delete' button. At the bottom right, there are three buttons: 'Delete', 'Save and add another', 'Save and continue editing', and 'SAVE'.

Hình 88 giao diện cập nhật đối tượng sản phẩm trong hóa đơn

- admin.site.register(Group_join)

Đăng ký đối tượng Group_join với trình quản lý admin trong django

+ Xem danh sách:



The screenshot shows the Django administration interface for the 'Group_joins' list view. The page title is 'Django administration' and the user is 'NGÔ MINH'. The breadcrumb trail is 'Home > App_Grapql > Group_joins'. A green success message at the top says 'The group_join "1 : nhóm1" was added successfully.'. Below the message, there is a 'Select group_join to change' section with an 'ADD GROUP_JOIN' button. The 'Action:' dropdown is set to 'Go' and '0 of 1 selected'. The table below has two rows: 'GROUP_JOIN' and '1 : nhóm1'. At the bottom, it says '1 group_join'.

Hình 89 giao diện danh sách các đối tượng nhóm người dùng có thể tham gia

+ Thêm:

The screenshot shows the Django administration interface for adding a new group_join. The page title is 'Django administration' and the user is logged in as 'NGÔ MINH'. The breadcrumb trail is 'Home > App_Grapql > Group_joins > Add group_join'. The form has two input fields: 'Name:' and 'Dest:'. At the bottom, there are three buttons: 'Save and add another', 'Save and continue editing', and 'SAVE'.

Hình 90 giao diện thêm đối tượng nhóm người dùng có thể tham gia

+ Cập nhật

The screenshot shows the Django administration interface for updating an existing group_join. The page title is 'Django administration' and the user is logged in as 'NGÔ MINH'. The breadcrumb trail is 'Home > App_Grapql > Group_joins > 1 : nhóm1'. The form shows the current values: 'Name: nhóm1' and 'Dest: nhóm chat'. There is a 'HISTORY' button on the right. At the bottom, there are four buttons: 'Delete', 'Save and add another', 'Save and continue editing', and 'SAVE'.

Hình 91 giao diện cập nhật đối tượng nhóm người dùng có thể tham gia

- admin.site.register(Chat)

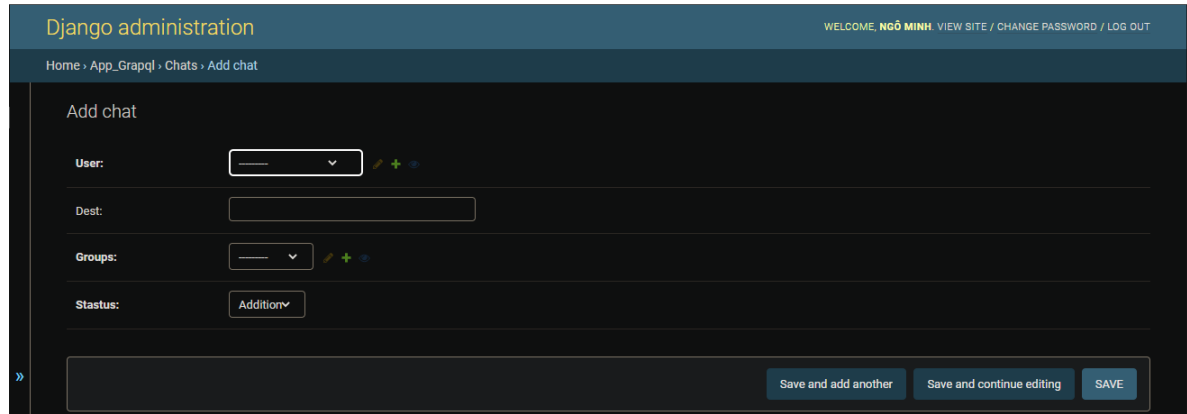
Đăng ký đối tượng Chat với trình quản lý admin trong django

+ Xem danh sách:

The screenshot shows the Django administration interface for the 'Chats' list view. The page title is 'Django administration' and the user is logged in as 'NGÔ MINH'. The breadcrumb trail is 'Home > App_Grapql > Chats'. A green success message at the top reads: 'The chat "2 : Ngô Minh Thống : 1 : nhóm1 : 2023-04-27 07:09:52.336482+00:00 : 1" was added successfully.'. Below the message, there is a 'Select chat to change' section with an 'ADD CHAT' button. The 'Action:' dropdown is set to '---' and 'Go' is next to it, with '0 of 1 selected' below. A table lists the chat objects, with the first row selected: '2 : Ngô Minh Thống : 1 : nhóm1 : 2023-04-27 07:09:52.336482+00:00 : 1'. At the bottom, it says '1 chat'.

Hình 92 giao diện danh sách các đối tượng hội thoại

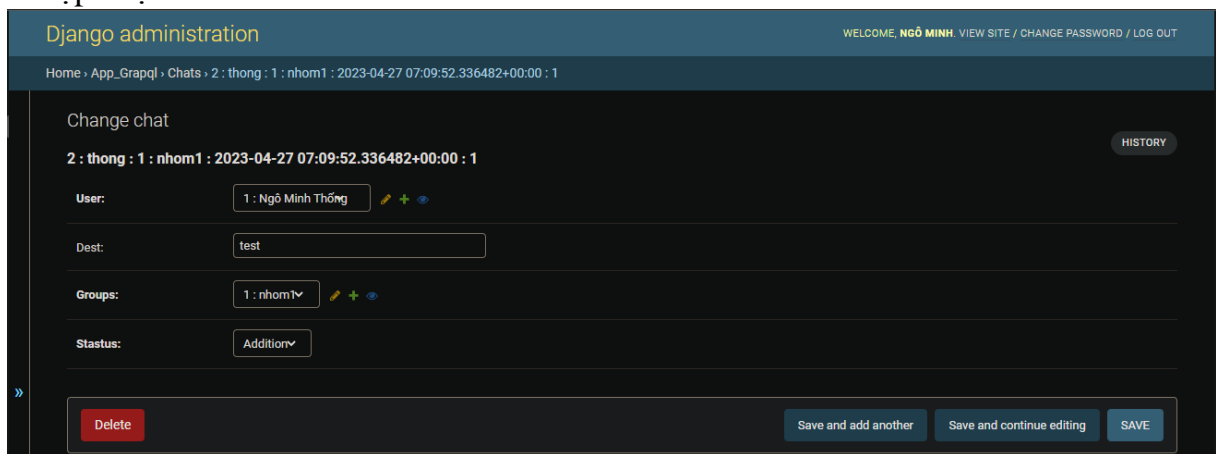
+ Thêm:



The screenshot shows the Django administration interface for adding a chat object. The breadcrumb trail is 'Home > App_Grapql > Chats > Add chat'. The form includes fields for 'User' (a dropdown menu), 'Dest' (a text input), 'Groups' (a dropdown menu), and 'Status' (a dropdown menu set to 'Addition'). At the bottom, there are three buttons: 'Save and add another', 'Save and continue editing', and 'SAVE'.

Hình 93 giao diện thêm đối tượng hội thoại

+ Cập nhật



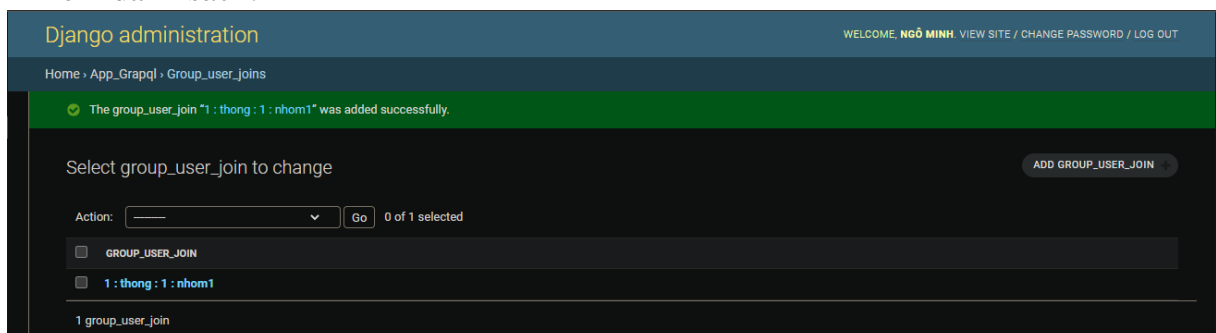
The screenshot shows the Django administration interface for updating a chat object. The breadcrumb trail is 'Home > App_Grapql > Chats > 2 : thong : 1 : nhom1 : 2023-04-27 07:09:52.336482+00:00 : 1'. The form includes fields for 'User' (set to '1 : Ngô Minh Thống'), 'Dest' (set to 'test'), 'Groups' (set to '1 : nhom1'), and 'Status' (set to 'Addition'). A 'Delete' button is visible on the left, and 'Save and add another', 'Save and continue editing', and 'SAVE' buttons are on the right.

Hình 94 giao diện cập nhật đối tượng hội thoại

- admin.site.register(Group_user_join)

Đăng ký đối tượng Group_user_join với trình quản lý admin trong django

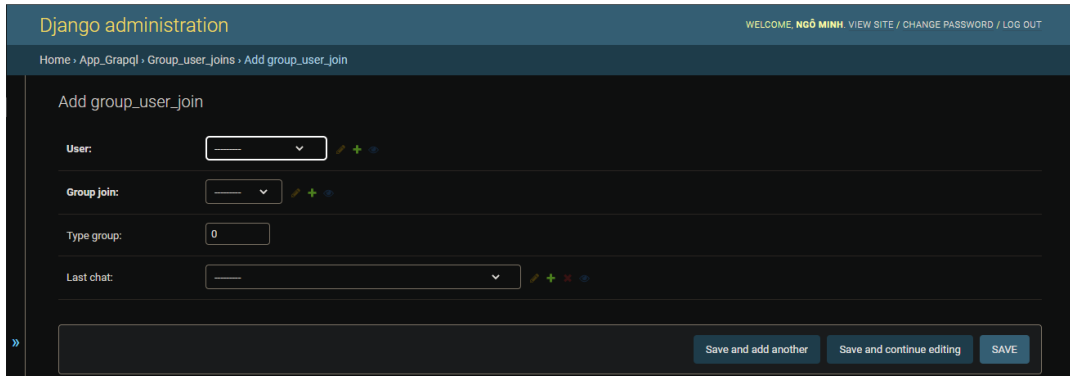
+ Xem danh sách:



The screenshot shows the Django administration interface for the 'Group_user_joins' list view. The breadcrumb trail is 'Home > App_Grapql > Group_user_joins'. A green success message at the top states: 'The group_user_join "1 : thong : 1 : nhom1" was added successfully.' Below the message, there is a 'Select group_user_join to change' section with an 'ADD GROUP_USER_JOIN' button. The 'Action:' dropdown is set to 'Go' and shows '0 of 1 selected'. A table lists the objects, with one row selected: '1 : thong : 1 : nhom1'. The bottom of the table shows '1 group_user_join'.

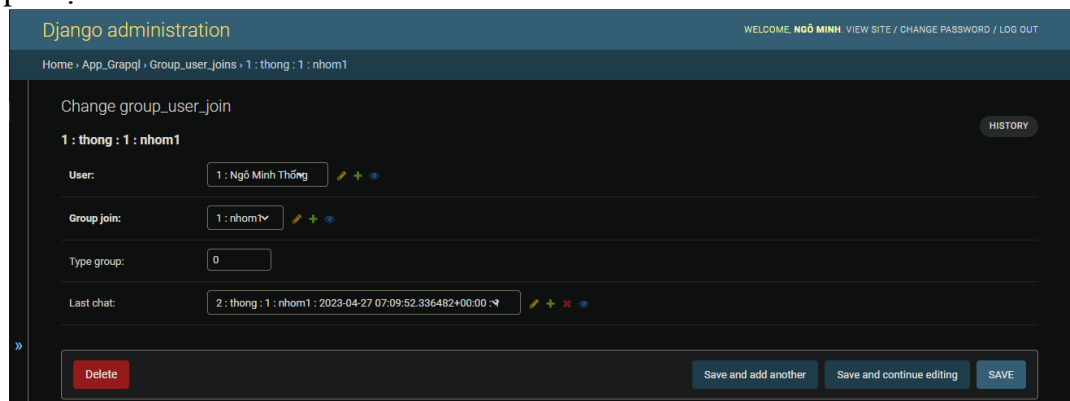
Hình 95 giao diện danh sách các đối tượng thành viên tham gia nhóm người dùng

+ Thêm:



Hình 96 giao diện thêm đối tượng thành viên tham gia nhóm người dùng

+ Cập nhật

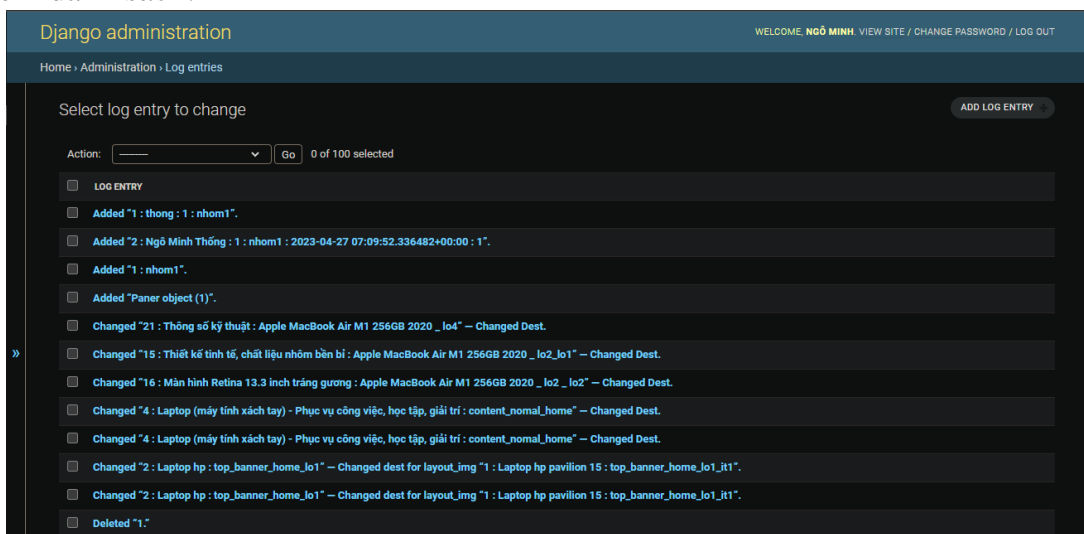


Hình 97 giao diện cập nhật đối tượng thành viên tham gia nhóm người dùng

- admin.site.register(LogEntry)

Đăng ký đối tượng LogEntry với trình quản lý admin trong django

+ Xem danh sách:



Hình 98 giao diện danh sách các đối tượng lịch sử thao tác trên giao diện admin server API

+ Thêm:

The screenshot shows the Django administration interface for adding a log entry. The page title is "Django administration" and the user is logged in as "NGÔ MINH". The breadcrumb trail is "Home > Administration > Log entries > Add log entry". The form fields are:

- User: A dropdown menu.
- Content type: A dropdown menu.
- Object id: A large text area.
- Object repr: A text input field.
- Action flag: A dropdown menu.
- Change message: A large text area.

At the bottom right, there are three buttons: "Save and add another", "Save and continue editing", and "SAVE".

Hình 99 giao diện thêm đối tượng lịch sử thao tác trên giao diện admin server API

+ Cập nhật

The screenshot shows the Django administration interface for updating a log entry. The page title is "Django administration" and the user is logged in as "NGÔ MINH". The breadcrumb trail is "Home > Administration > Log entries > Added '1 : thong : 1 : nhom1'". The form fields are:

- User: A dropdown menu with the value "1 : Ngô Minh Thống".
- Content type: A dropdown menu with the value "app_grapql | group_user_login".
- Object id: A large text area with the value "1".
- Object repr: A text input field with the value "1 : thong : 1 : nhom1".
- Action flag: A dropdown menu with the value "Addition".
- Change message: A large text area with the value "[{"added": {}}]".

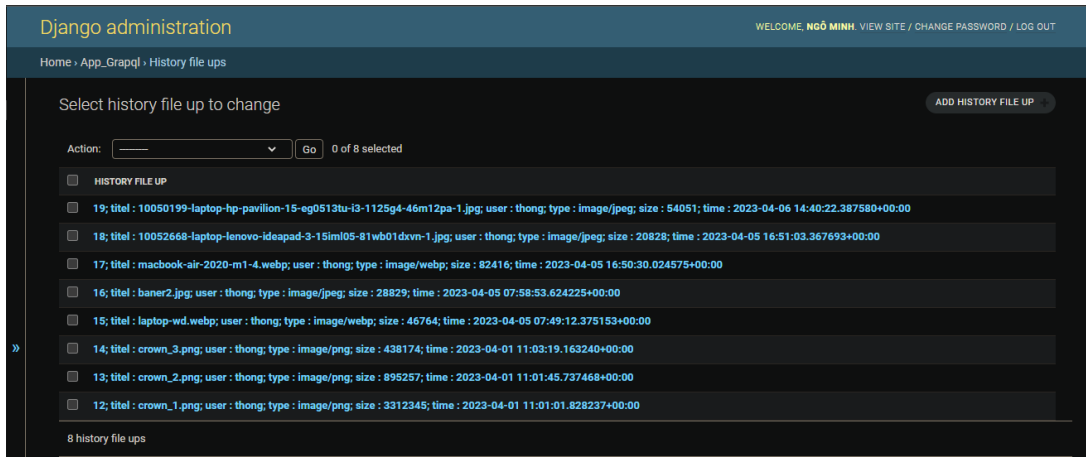
At the bottom left, there is a "Delete" button. At the bottom right, there are three buttons: "Save and add another", "Save and continue editing", and "SAVE".

Hình 100 giao diện cập nhật đối tượng lịch sử thao tác trên giao diện admin server API

- admin.site.register(HistoryFileUp)

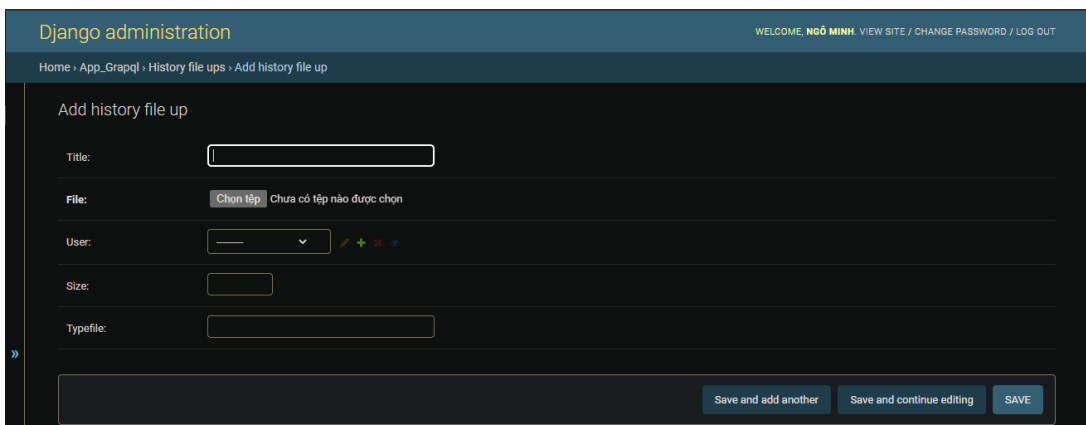
Đăng ký đối tượng HistoryFileUp với trình quản lý admin trong django

+ Xem danh sách:



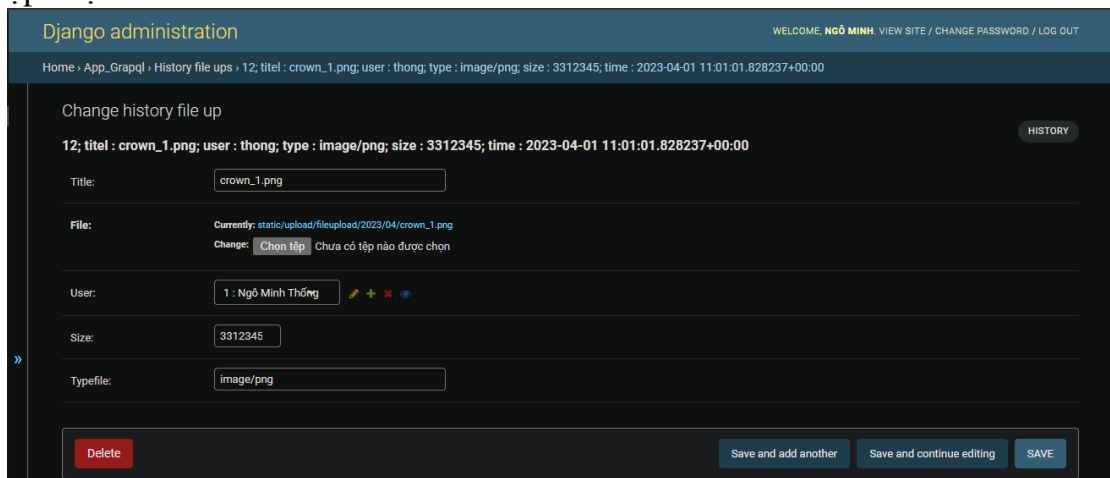
Hình 101 giao diện danh sách các đối tượng file đã tải lưu trên server lưu trữ

+ Thêm:



Hình 102 giao diện thêm đối tượng file đã tải lưu trên server lưu trữ

+ Cập nhật

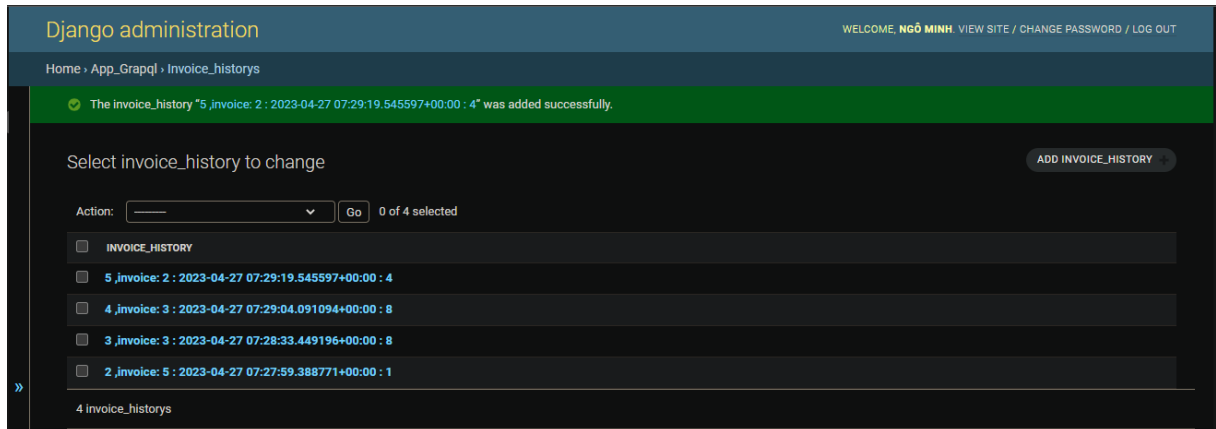


Hình 103 giao diện cập nhật đối tượng file đã tải lưu trên server lưu trữ

- admin.site.register(Invoice_history)

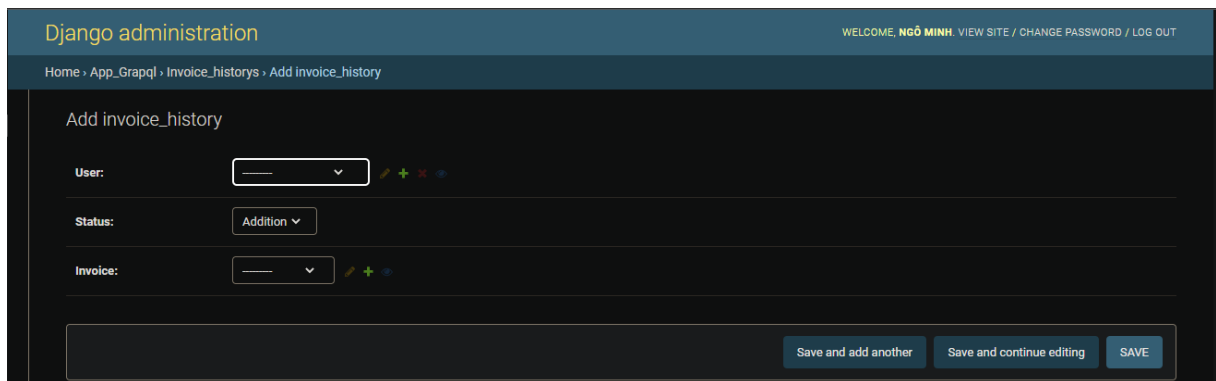
Đăng ký đối tượng Invoice_history với trình quản lý admin trong django

+ Xem danh sách:



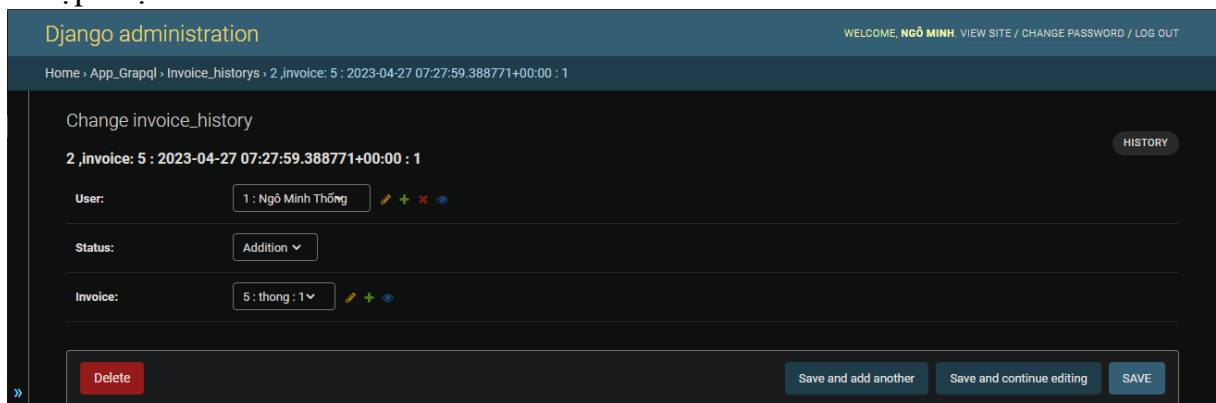
Hình 104 giao diện danh sách các đối tượng lịch sử hóa đơn

+ Thêm:



Hình 105 giao diện thêm đối tượng lịch sử hóa đơn

+ Cập nhật

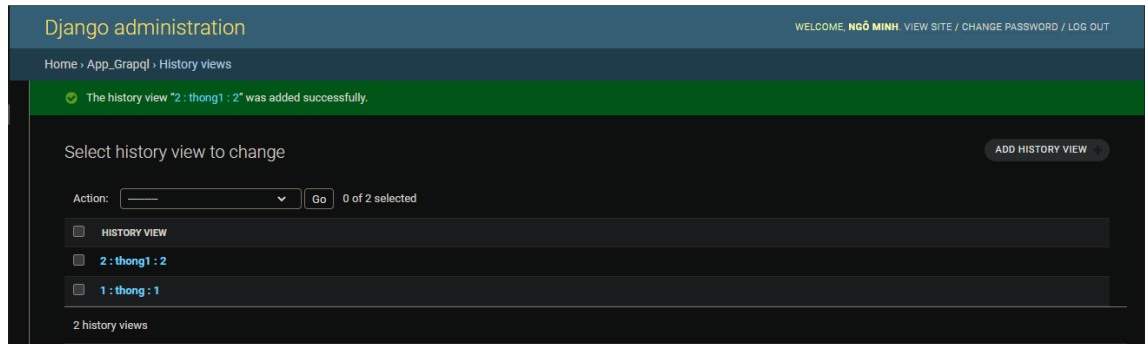


Hình 106 giao diện cập nhật đối tượng lịch sử hóa đơn

- admin.site.register(HistoryView)

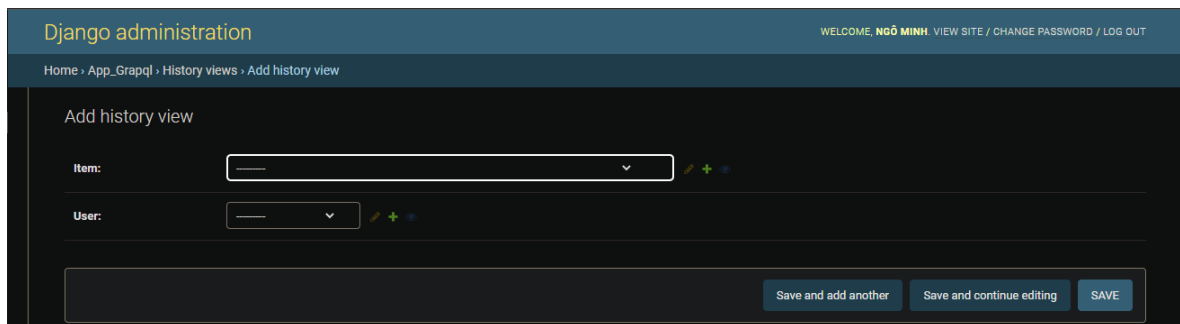
Đăng ký đối tượng HistoryView với trình quản lý admin trong django

+ Xem danh sách:



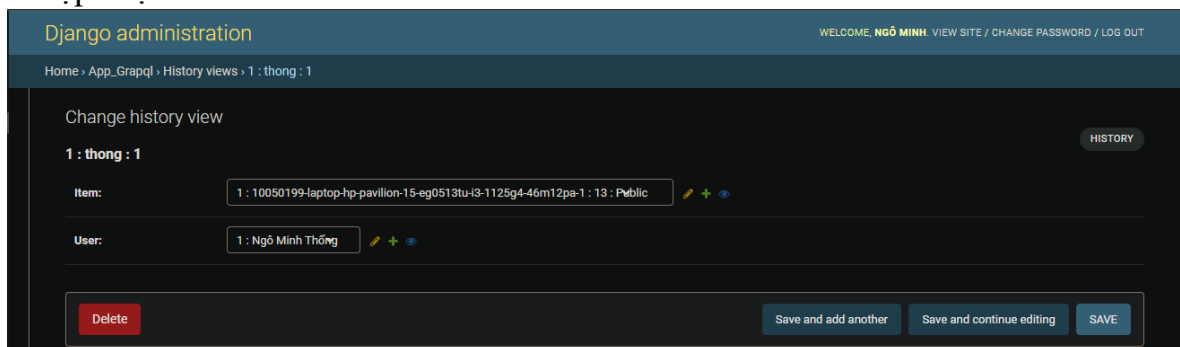
Hình 107 giao diện danh sách các đối tượng lịch sử xem của khách hàng

+ Thêm:



Hình 108 giao diện thêm đối tượng lịch sử xem của khách hàng

+ Cập nhật

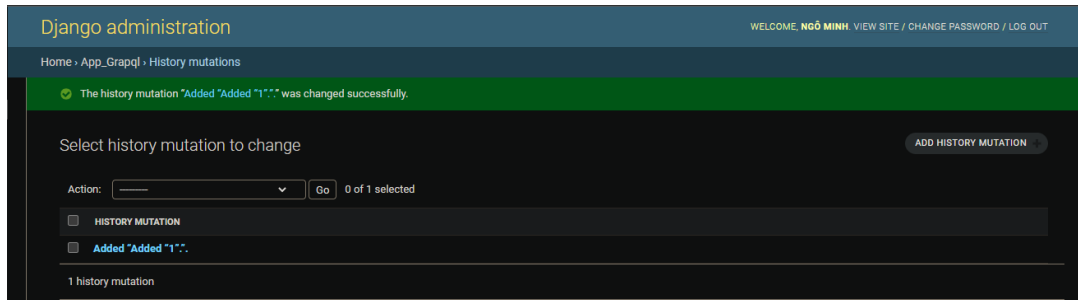


Hình 109 giao diện cập nhật đối tượng lịch sử xem của khách hàng

- admin.site.register(HistoryMutation) Added "1". [{"added": {}}]

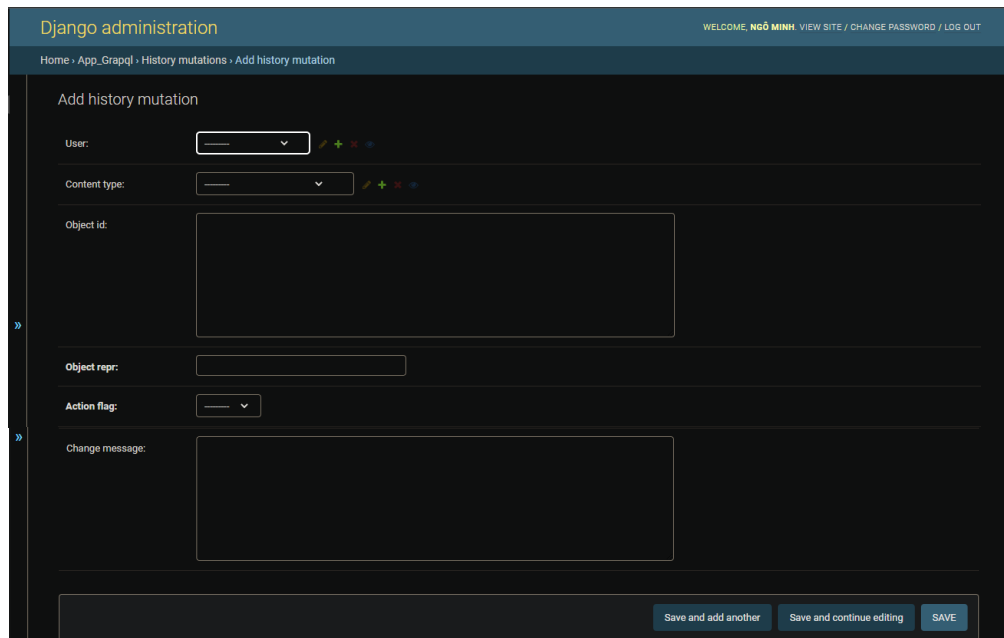
Đăng ký đối tượng HistoryMutation với trình quản lý admin trong django

+ Xem danh sách:



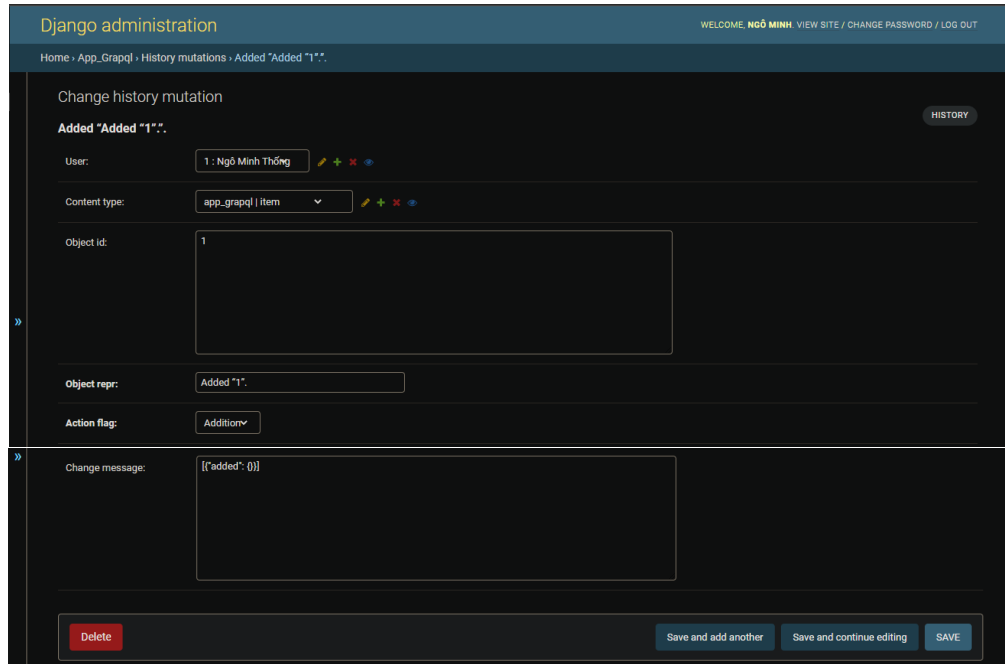
Hình 110 giao diện danh sách các lịch sử hành động của server supadmin

+ Thêm:



Hình 111 giao diện thêm đối tượng lịch sử hành động của server supadmin

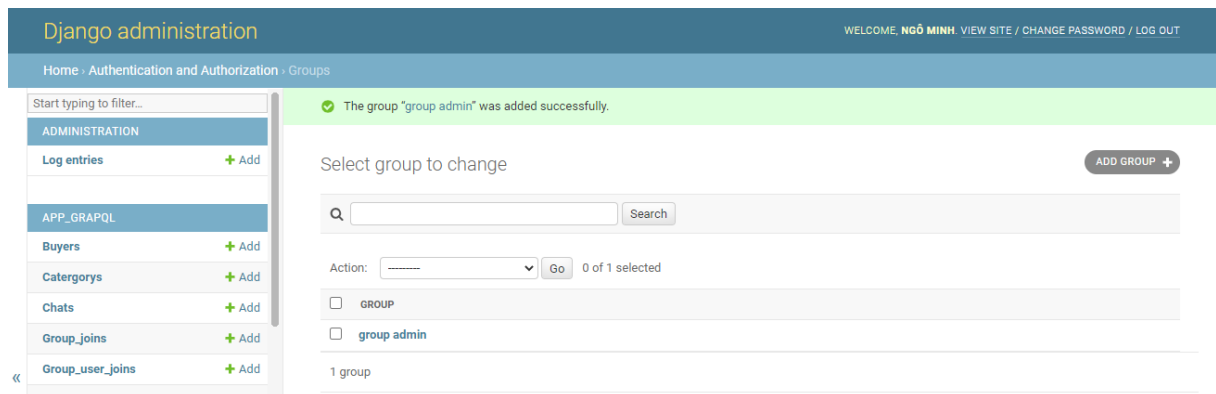
+ Cập nhật



Hình 112 giao diện cập nhật đối tượng lịch sử hành động của server supadmin

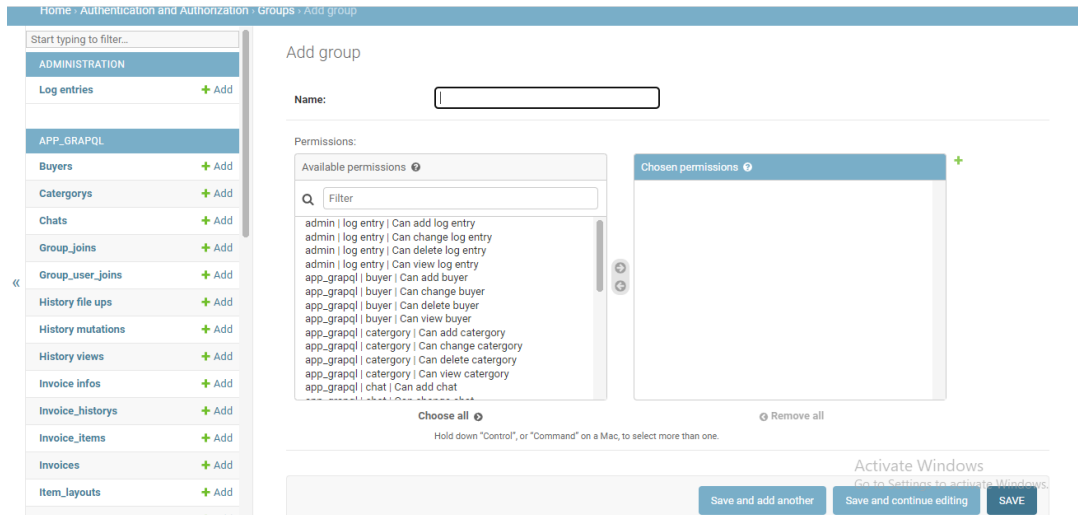
Giao diện nhóm phân quyền user trên giao diện admin server api

+ Xem danh sách:



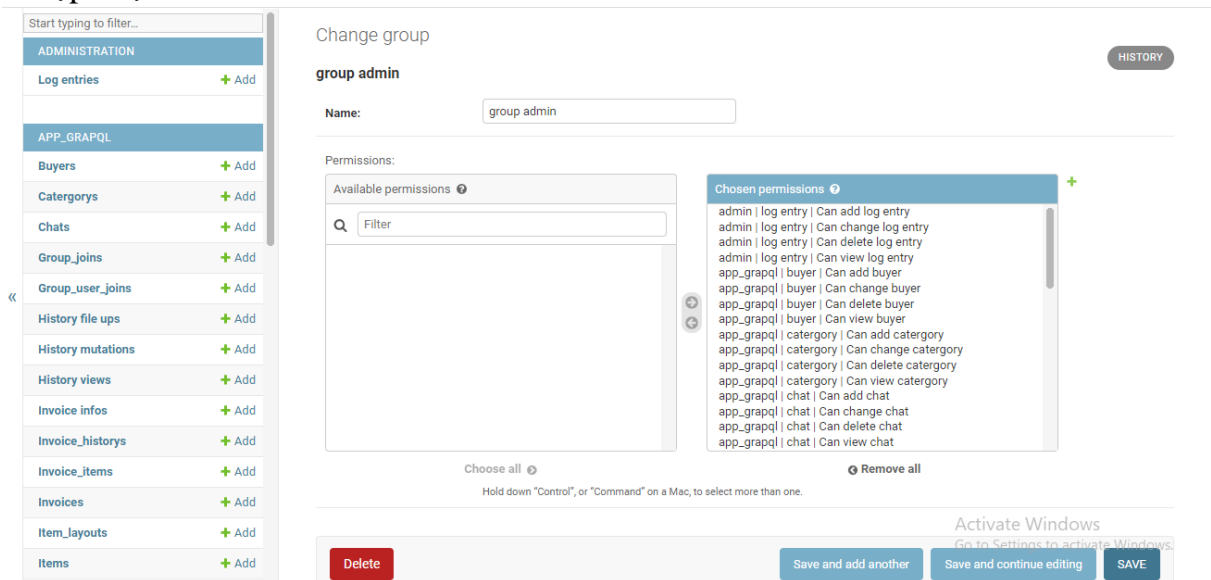
Hình 113 giao diện danh sách các đối tượng nhóm phân quyền người dùng

+ Thêm:



Hình 114 giao diện thêm đối tượng nhóm phân quyền người dùng

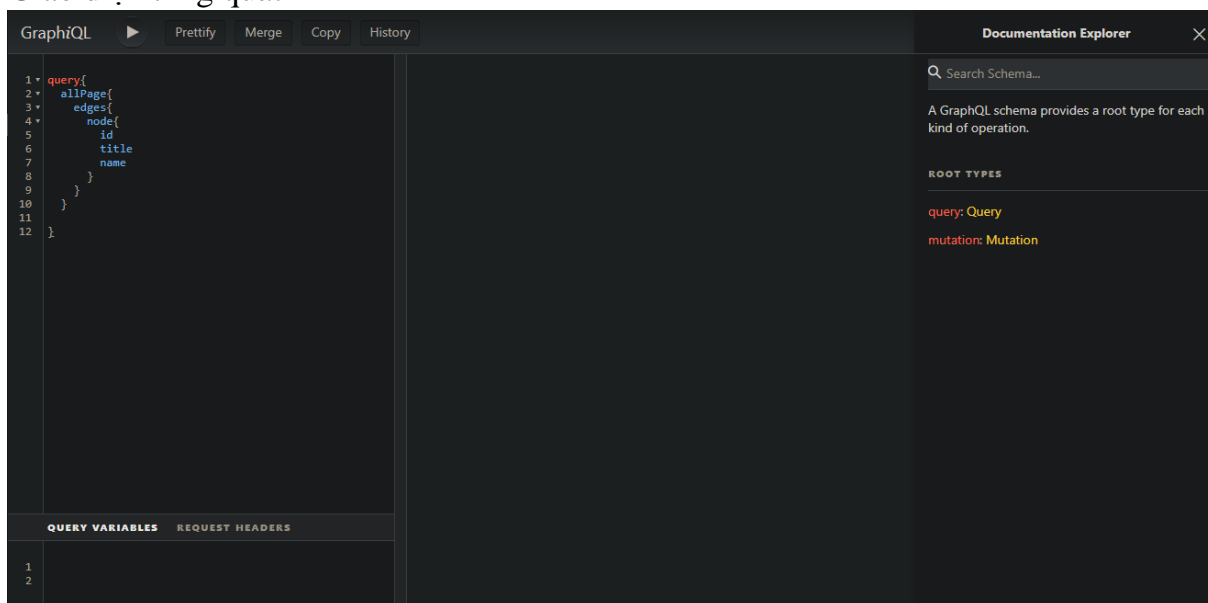
+ Cập nhật



Hình 115 giao diện cập nhật đối tượng nhóm phân quyền người dùng

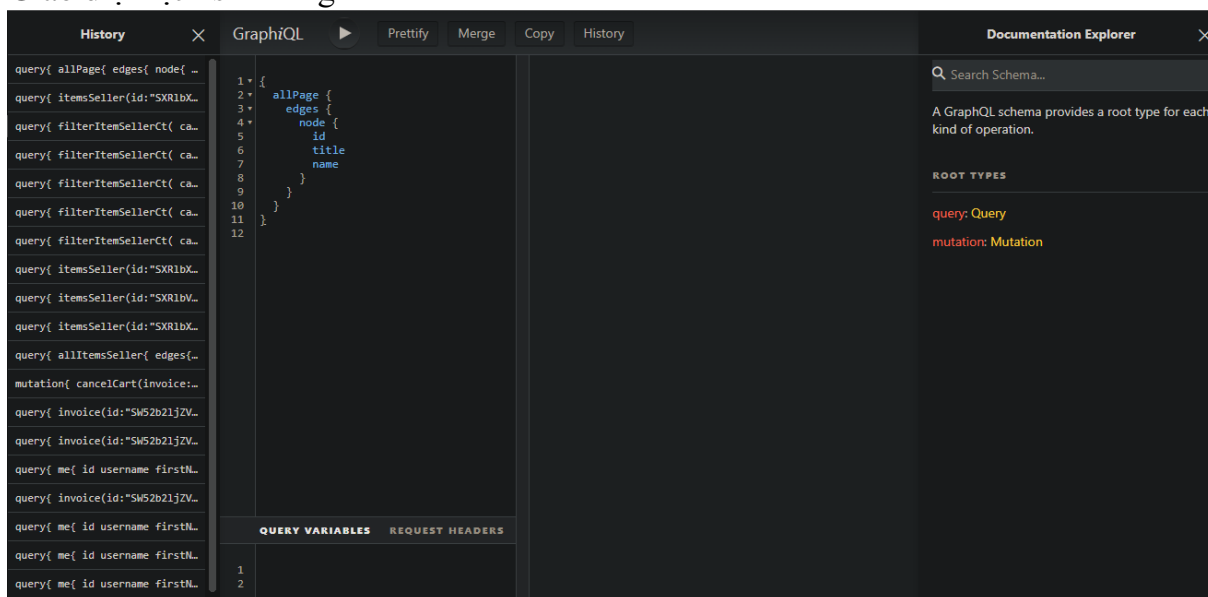
❖ Giao diện tài liệu hỗ trợ API graphql

Giao diện tổng quát



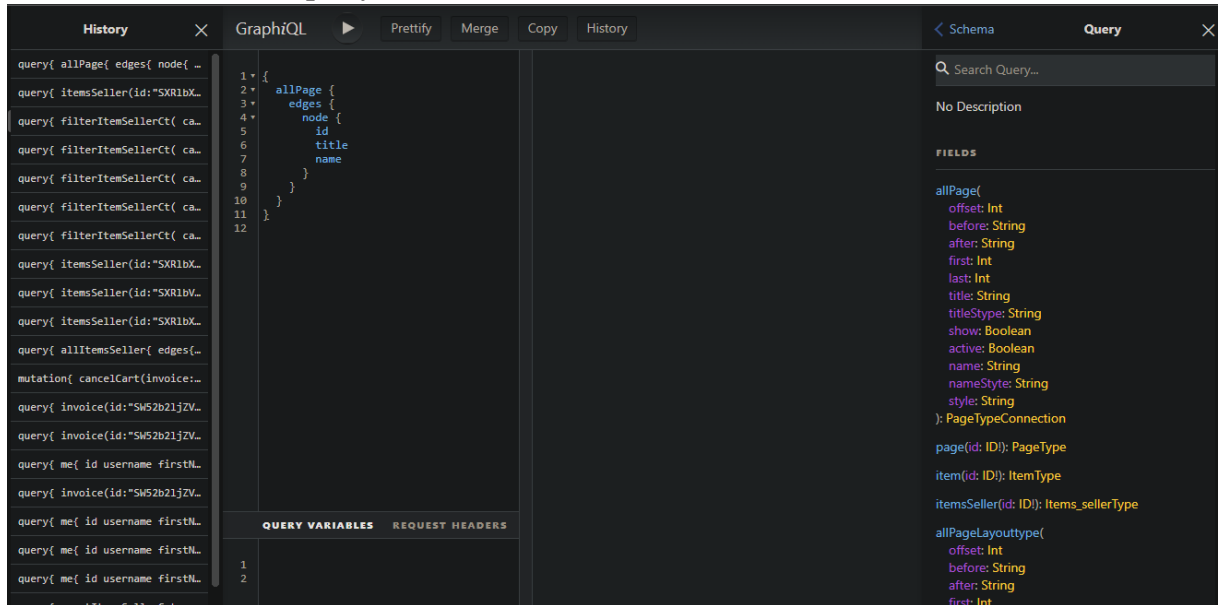
Hình 116 giao diện hướng dẫn sử dụng api graphql

Giao diện lịch sử tương tác



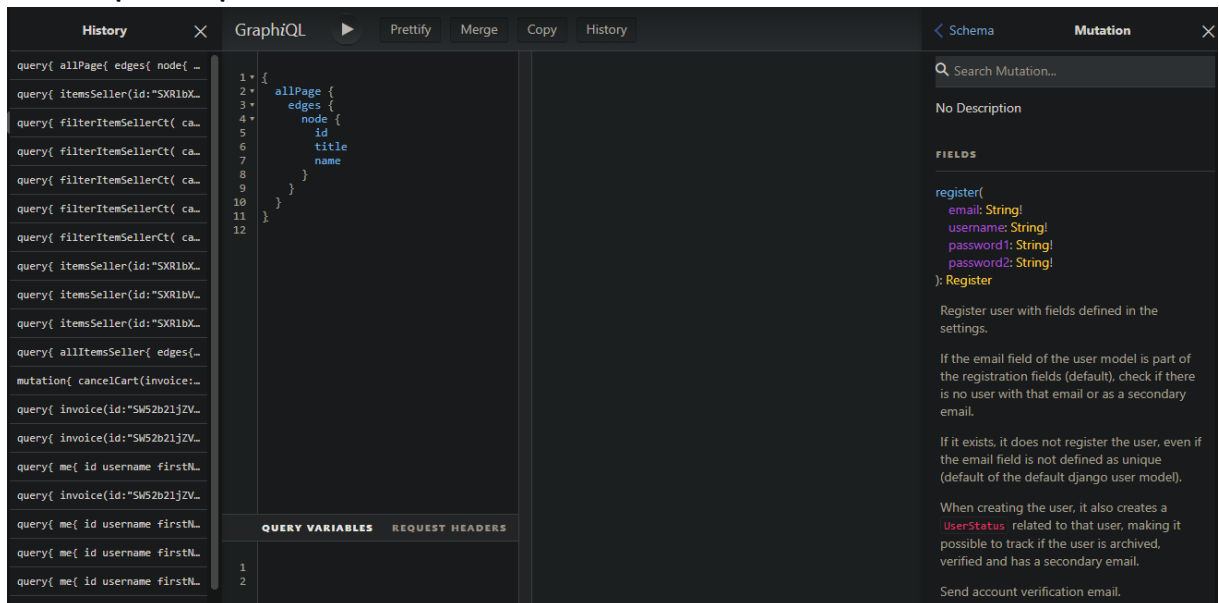
Hình 117 giao diện xem lại lịch sử hướng dẫn sử dụng api graphql

Giao diện tài liệu về query



Hình 118 giao diện hướng dẫn sử dụng query

Giao diện tài liệu về mutation



Hình 119 giao diện hướng dẫn sử dụng mutation

Giao diện từng type(đối tượng) trong api

- Type trong query
- + allPage:

Lọc lấy tất cả các đối tượng Page thỏa mãn điều kiện trong input

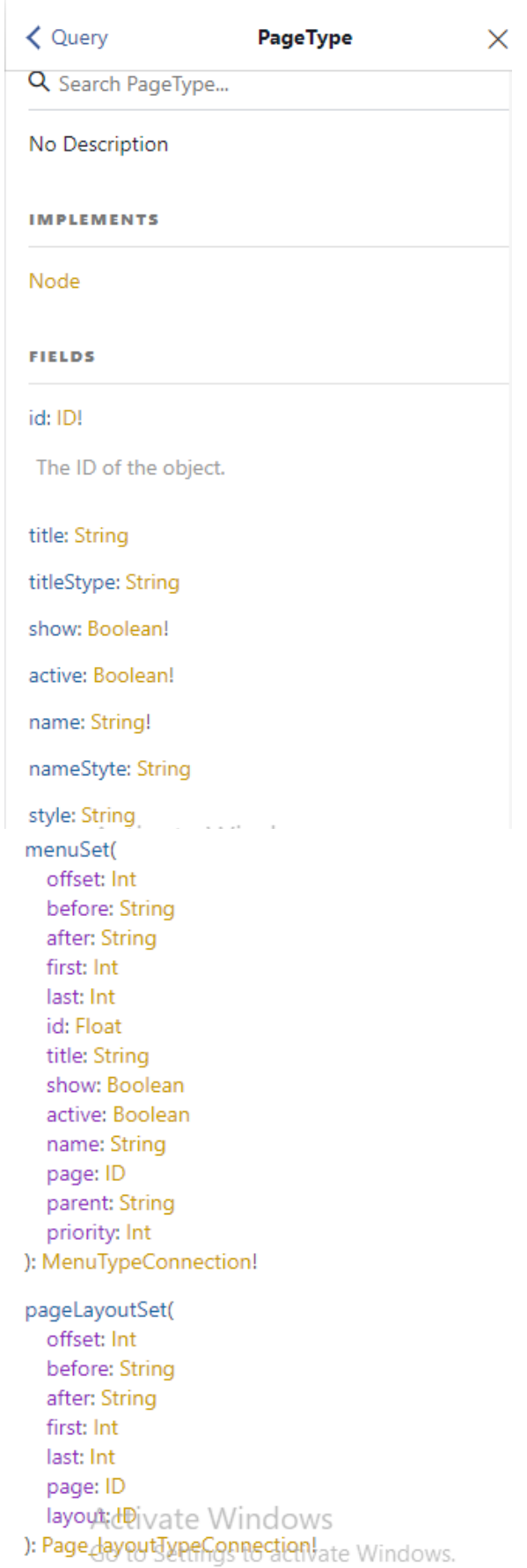
input

```
allPage(  
  offset: Int  
  before: String  
  after: String  
  first: Int  
  last: Int  
  title: String  
  titleStyle: String  
  show: Boolean  
  active: Boolean  
  name: String  
  nameStyle: String  
  style: String  
): PageTypeConnection
```

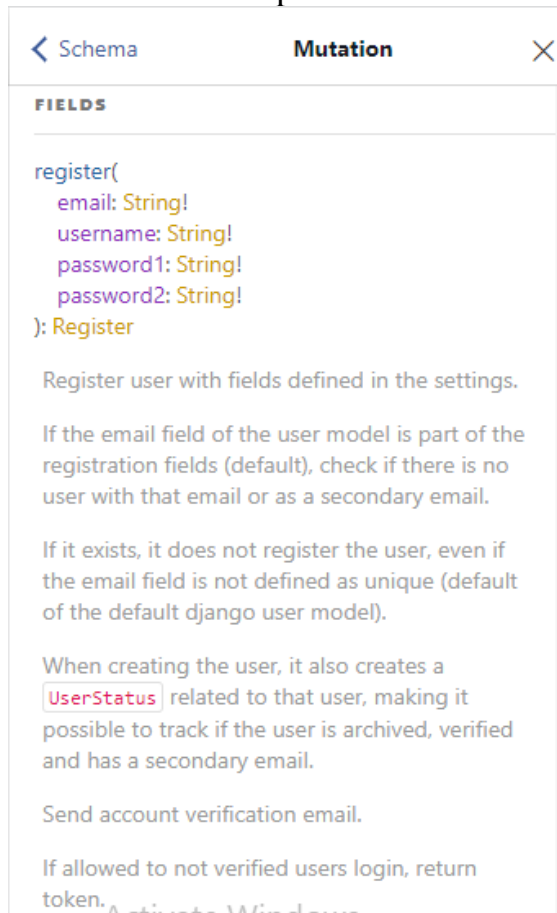
output

Type trong output

+ page:

input	Output
<pre>page(id: ID!): PageType</pre>	 <p>The screenshot shows a GraphQL IDE window titled "Query" and "PageType". It displays the definition of the PageType type, including its fields and connections. The fields listed are: id: ID!, title: String, titleStyle: String, show: Boolean!, active: Boolean!, name: String!, nameStyle: String, style: String, menuSet (with fields: offset: Int, before: String, after: String, first: Int, last: Int, id: Float, title: String, show: Boolean, active: Boolean, name: String, page: ID, parent: String, priority: Int), and pageLayoutSet (with fields: offset: Int, before: String, after: String, first: Int, last: Int, page: ID, layout: ID). The output is color-coded to match the input query.</p>

- Type trong mutation input



Schema Mutation

FIELDS

```
register(
  email: String!
  username: String!
  password1: String!
  password2: String!
): Register
```

Register user with fields defined in the settings.

If the email field of the user model is part of the registration fields (default), check if there is no user with that email or as a secondary email.

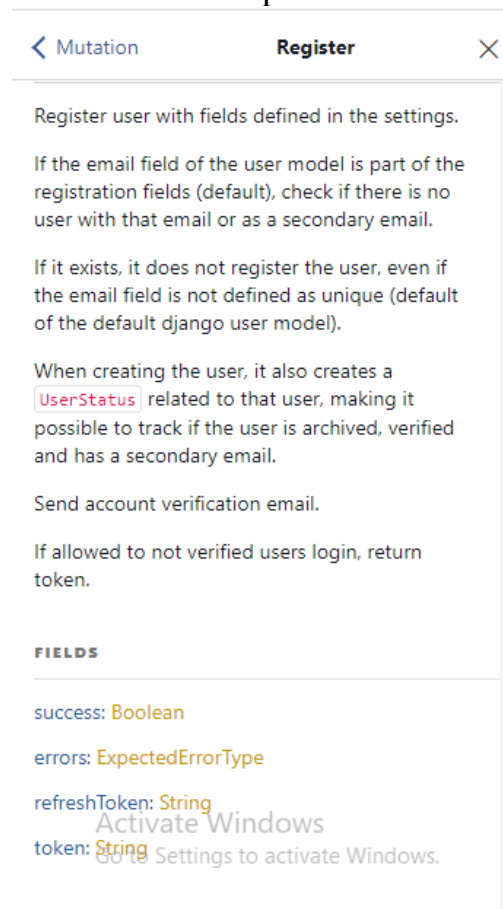
If it exists, it does not register the user, even if the email field is not defined as unique (default of the default django user model).

When creating the user, it also creates a `UserStatus` related to that user, making it possible to track if the user is archived, verified and has a secondary email.

Send account verification email.

If allowed to not verified users login, return token.

output



Mutation Register

Register user with fields defined in the settings.

If the email field of the user model is part of the registration fields (default), check if there is no user with that email or as a secondary email.

If it exists, it does not register the user, even if the email field is not defined as unique (default of the default django user model).

When creating the user, it also creates a `UserStatus` related to that user, making it possible to track if the user is archived, verified and has a secondary email.

Send account verification email.

If allowed to not verified users login, return token.

FIELDS

```
success: Boolean
errors: ExpectedErrorType
refreshToken: String
token: String
```

2) Server Client(Flask):

- Các bộ thư viện sử dụng:

Thư viện kèm theo khi cài đặt Flask

- bidict==0.22.1
- certifi==2022.12.7
- charset-normalizer==3.0.1
- click==8.1.3
- colorama==0.4.6
- Flask==2.2.3
- idna==3.4
- itsdangerous==2.1.2
- Jinja2==3.1.2
- MarkupSafe==2.1.2
- python-engineio==4.3.4
- requests==2.28.2
- urllib3==1.26.14
- Werkzeug==2.2.3

Thư viện cài đặt thêm

❖ Flask-SocketIO==5.3.2

Thư viện hỗ trợ tương tác thời gian thực trong flask tức là các dữ liệu được truyền tải và hiển thị ngay lập tức khi có sự thay đổi, một framework phổ biến trong việc xây dựng ứng dụng web Python.

Các chức năng chính:

- Cung cấp một giao thức truyền tải dữ liệu theo thời gian thực (real-time) giữa máy chủ và trình duyệt của khách hàng (client).
- Cho phép phát hiện sự kiện (event) trên máy chủ và truyền tải chúng đến client để xử lý.
- Hỗ trợ tạo phòng (room) để các client cùng tham gia và giao tiếp với nhau.
- Cung cấp các tiện ích để kiểm soát kết nối giữa client và máy chủ.
- Hỗ trợ các kiểu dữ liệu phổ biến, bao gồm JSON và MessagePack.
- Có thể sử dụng cùng với các thư viện JavaScript như Socket.IO hoặc jQuery để phát triển các ứng dụng web hiệu quả hơn.

❖ localStoragePy==0.2.3

Thư viện localStoragePy là một thư viện Python nhằm hỗ trợ lưu trữ dữ liệu trên máy. Thư viện cho phép lưu trữ dữ liệu cục bộ trên máy của người dùng, do đó cho phép các ứng dụng web lưu trữ dữ liệu của người dùng một cách dễ dàng.

Các chức năng của thư viện localStoragePy bao gồm:

- Cho phép lưu trữ và truy xuất các giá trị dữ liệu trong localStorage của người dùng, bao gồm cả các đối tượng Python.
- Hỗ trợ các phương thức để thêm, sửa đổi và xóa các giá trị dữ liệu.
- Hỗ trợ đọc và ghi các giá trị dữ liệu dưới dạng JSON, giúp dữ liệu được truyền tải một cách dễ dàng.
- Cho phép cấu hình các tham số tùy chỉnh để điều khiển việc lưu trữ dữ liệu, bao gồm cả thời gian sống của dữ liệu và tên của kho lưu trữ.

b) Hướng dẫn cài đặt:

- ❖ Cài biến môi trường : python -m venv venv
- ❖ Khởi động môi trường : venv\Scripts\activate
- ❖ Di chuyển tới thư mục root : growpro_flash
- ❖ Cài đặt các bộ thư viện: pip install -r requirements.txt
- ❖ Đồng bộ các bộ thư viện : pip freeze
- ❖ Vào file growpro_flash/settings.py cấu hình thông tin pj:

```
SECRET_KEY="8c7dbd270cb98e83f9d8d57fb8a2ab7bac9d755asf41sg51"
url_backend=f"https://thong1993.pythonanywhere.com/"
endpoint = f"{url_backend}graphql"
#app
app_settings={
    'host':'127.0.0.1',
    'port':4000,
    'debug':True,
}
#background
item_background=40
#item
number_item=9
item_category=3
item_category_len=13
item_tilte_len=33
```

Cấu hình thông tin địa chỉ truyền và nhận dữ liệu:

```
url_backend=f"https://thong1993.pythonanywhere.com/"
endpoint = f"{url_backend}graphql"
```

Cấu hình thông tin hiển thị

```
#background
item_background=40

#item
number_item=9
item_category=3
item_category_len=13
item_tilte_len=33
```

c) Cấu trúc thư mục và chức năng:

	<p><u>__pycache__</u> : thư mục do hệ thống tạo quản lý các file sẽ được sử dụng</p> <hr/> <p>GraphQL: thư mục quản lý các api truy vấn tài nguyên</p> <hr/> <p>Page: thư mục quản lý các file code py chính</p> <hr/> <p>Static: thư mục quản lý các file khác: js, css, image,...</p> <hr/> <p>Templates: thư mục quản lý các file html</p> <hr/> <p>Apps.py : file quản lý và chỉ định các đường dẫn</p> <hr/> <p>Export_function_socket.py: file quản lý các hàm sử lý sự kiện từ socket.io</p> <hr/> <p>Export_function_urls.py: file quản lý các hàm sử lý sự kiện từ flask</p> <hr/> <p>Models.py: file quản lý database</p> <hr/> <p>requirements.txt: file liệt kê danh sách các thư viện sử dụng có trong venv</p> <hr/> <p>settings.py: file quản lý các thiết lập chung của chương trình</p> <hr/> <p>venv: thư mục chứa các thư viện/thiết lập để chạy chương trình.</p>
--	---

- GraphQL:

	<p><u>__pycache__</u> : thư mục do hệ thống tạo quản lý các file sẽ được sử dụng</p> <hr/> <p>Thư mục control: quản lý các file chứa hàm quản lý tài nguyên</p> <hr/> <p>Thư mục query: quản lý các file chứa các câu lệnh thực thi</p> <hr/> <p>File cart.py: file chứa các câu lệnh liên quan với giỏ hàng</p> <hr/> <p>File invoice.py: file chứa các câu lệnh liên quan với hóa đơn</p> <hr/> <p>File item.py: file chứa các câu lệnh liên quan với sản phẩm</p> <hr/> <p>File menu.py: file chứa các câu lệnh liên quan với menu</p> <hr/> <p>File page.py: file chứa các câu lệnh liên quan với các thành phần trong trang</p> <hr/> <p>File user.py: file chứa các câu lệnh liên quan với người dùng</p> <hr/> <p>File settings.py: chứa các cấu hình chung của các hàm.</p>
--	---

- Page:

<ul style="list-style-type: none">▼ page<ul style="list-style-type: none">> __pycache__▼ chat<ul style="list-style-type: none">> __pycache__chat.py▼ item<ul style="list-style-type: none">> __pycache__cart.pycategory.pyinvoice.pyitem.py▼ layout<ul style="list-style-type: none">> __pycache__header.pypage.py▼ user<ul style="list-style-type: none">> __pycache__changepass.pyextend.pyinformation.pylogin.pyregister.pyverify.py	<p><u>__pycache__</u> : thư mục do hệ thống tạo quản lý các file sẽ được sử dụng</p> <hr/> <p>Thư mục chat quản lý các file chứa các hàm liên quan về hội thoại</p> <hr/> <p>File chat.py chứa các hàm liên quan về hội thoại</p> <hr/> <p>Thư mục item: quản lý các file chứa các hàm liên quan về sản phẩm</p> <hr/> <p>File cart.py: chứa các hàm liên quan về giỏ hàng</p> <hr/> <p>File category.py: chứa các hàm liên quan về thể loại sản phẩm</p> <hr/> <p>File invoice.py: chứa các hàm liên quan về hóa đơn</p> <hr/> <p>File item.py: chứa các hàm liên quan về sản phẩm</p> <hr/> <p>Thư mục user: quản lý các file chứa các hàm liên quan về người dùng</p> <hr/> <p>File changepass.py: chứa các hàm liên quan về thay đổi mật khẩu</p> <hr/> <p>File extend.py: chứa các hàm khác</p> <hr/> <p>File information.py: chứa các hàm về thông tin người dùng</p> <hr/> <p>File login.py: chứa các hàm về đăng nhập</p> <hr/> <p>File register.py: chứa các hàm về đăng ký</p> <hr/> <p>File verify.py: chứa các hàm về xác nhận</p>
---	--

- Static:

<ul style="list-style-type: none">▼ static<ul style="list-style-type: none">▼ css<ul style="list-style-type: none"># font-awesome.min.css# font-googleapis.css# main.css# responsive.css# tailwind.index.min.css▼ fonts<ul style="list-style-type: none">fontawesome-webfont.ttffontawesome-webfont.wofffontawesome-webfont.woff2▼ img<ul style="list-style-type: none">▼ major<ul style="list-style-type: none">find-out-more-bt.pngmain_bg.jpgmain_bg.png▼ js<ul style="list-style-type: none">index.min.js.mapmain.jssocket.4.0.1.io.jssocket.io.js.maptailwind.index.min.jstailwindcss.js	<p>Thư mục css quản lý các file css</p> <hr/> <p>Thư mục fonts quản lý các file font</p> <hr/> <p>Thư mục img: quản lý các file ảnh</p> <hr/> <p>Thư mục js: chứa các file/thư viện js</p>
---	--

- Templates:

<ul style="list-style-type: none">▼ templates<ul style="list-style-type: none">▼ carts<ul style="list-style-type: none">> macros<ul style="list-style-type: none">◇ cart.html> chats▼ invoice<ul style="list-style-type: none">▼ macros<ul style="list-style-type: none">◇ mr_item.html◇ mr_iteminfo.html◇ mr_items.html▼ response<ul style="list-style-type: none">◇ item.html◇ iteminfo.html◇ historyinvoice.html▼ items<ul style="list-style-type: none">> macros<ul style="list-style-type: none">◇ category.html◇ item.html▼ layouts<ul style="list-style-type: none">> macros<ul style="list-style-type: none">◇ page.html> users<ul style="list-style-type: none">◇ base.html	<p>File base.html: là một file template cơ bản chứa các phần chung của một trang web. Như header, footer, menu, thanh điều hướng và mã JavaScript hoặc CSS chung.</p> <hr/> <p>Thư mục macros: quản lý các file chứa các đoạn mã macros, dùng để định nghĩa các thành phần/khối lặp lại một cách dễ dàng.</p> <hr/> <p>Thư mục response: quản lý các file chứa các đoạn mã dùng để tạo các kiểu trả về</p> <hr/> <p>Thư mục carts: quản lý các file chứa các đoạn mã liên quan tới giỏ hàng</p> <hr/> <p>Thư mục chats: quản lý các file chứa các đoạn mã liên quan tới hội thoại</p> <hr/> <p>Thư mục invoice: quản lý các file chứa các đoạn mã liên quan tới hóa đơn</p> <hr/> <p>Thư mục items: quản lý các file chứa các đoạn mã liên quan tới sản phẩm</p> <hr/> <p>Thư mục layouts: quản lý các file chứa các đoạn mã liên quan tới thành phần trang</p> <hr/> <p>Thư mục users: quản lý các file chứa các đoạn mã liên quan tới người dùng</p>
--	--

d) Các giao diện người dùng trên laptop/pc:

Trang chủ

The screenshot shows a website interface for laptops. At the top, there is a navigation bar with links: Trang chủ, Thể Loại, Trò chuyện, Giỏ hàng, Thông tin, and a Login button. Below the navigation is a banner for 'Laptop hp pavilion 15' with the model number 'Laptop-hp-pavilion-15-eg0513tu-i3-1125g4-46m12pa-1' and a 'Sản phẩm đang hot' tag. A central image shows the laptop. Below the banner is a section titled 'Laptop (máy tính xách tay) - Phục vụ công việc, học tập, giải trí'. The text discusses the evolution of laptops in 2019, their benefits for work and study, and mentions various brands like Dell, Asus, Acer, HP, and Lenovo. It also lists benefits of laptops such as portability, ease of use, and durability. The page is divided into three columns: 'Laptop Windows', 'Laptop Macbook OS', and 'Laptop Chrome OS', each providing information about the operating system and listing popular models. The footer contains a copyright notice: 'Copyright ©2023 by Thống'.

Trang chủ

Thể Loại

Trò chuyện

Giỏ hàng

Thông tin

Login

Laptop hp pavilion 15

Laptop-hp-pavilion-15-eg0513tu-i3-1125g4-46m12pa-1

Sản phẩm đang hot

Laptop hp

Laptop (máy tính xách tay) - Phục vụ công việc, học tập, giải trí

Năm 2019, thị trường laptop trở nên khó khăn hơn vì sự phát triển về cấu hình và các công nghệ trên các thiết bị di động khác, đặc biệt là điện thoại thông minh. Nhưng không vì thế mà máy tính xách tay (hay còn gọi là laptop) bị lãng quên, công việc vẫn phòng hiện giờ vẫn ưu tiên laptop hơn các thiết bị khác. Trong bối cảnh này, các hãng máy tính, hãng công nghệ đã tung ra nhiều sản phẩm tối ưu đáp ứng đa dạng người dùng hơn. Hiện nay, xu hướng sử dụng thiết bị điện tử thường tập trung vào dòng sản phẩm có mức giá hợp lý, sở hữu thiết kế siêu di động, mỏng nhẹ, thời lượng pin tốt, chế độ ưu đãi và bảo hành.

Giờ đây, những chiếc laptop 2 trong 1 có thể xoay, gấp và tách rời, trở thành máy tính bảng hoặc sử dụng máy tính xách tay như thông thường, đáp ứng nhu cầu học tập và giải trí. Ngoài ra, màn hình cảm ứng giúp thao tác người dùng giống hệt như trên máy tính bảng. Bên cạnh đó, người sử dụng có thể chọn những chiếc ultrabook, có cấu hình mạnh hơn, mỏng hơn, độ phân giải lớn, nhỏ gọn có khả năng di động cao, phù hợp với doanh nhân.

Activate Windows
Go to Settings to activate Windows.

Tại sao nên mua máy tính xách tay

Lợi ích khi sử dụng laptop là gì?

- Laptop được trang bị hệ điều hành tối ưu, kèm theo đó là cấu hình đa dạng, ở nhiều mức độ khác nhau giúp người dùng có thể xử lý mọi công việc.
- Laptop được trang bị bàn phím, chuột và các cú pháp tiện dụng mang lại sự thuận lợi giúp công việc được hoàn thiện một cách tốt hơn so với các thiết bị khác. Một số dòng laptop cao cấp có kèm theo màn cảm ứng tiên nghi.
- Laptop có đầy đủ các chức năng của một máy tính bàn và tính di động cao
- Dung lượng pin lớn mang đến sự tiện lợi. Đặc biệt là đối với người thường xuyên phải ra ngoài làm việc.
- Khả năng lưu trữ dữ liệu trên laptop nhiều hơn.
- Sửa chữa và nâng cấp đơn giản, nhanh chóng.

Các loại máy tính laptop phổ biến

Phân loại laptop theo hệ điều hành

Laptop Windows

Thông tin chung : Ở hệ điều hành này, có khá nhiều mẫu để lựa chọn, từ phân khúc phổ thông tới cao cấp. Trong top 10 máy tính xách tay bán chạy nhất có hai thương hiệu lớn là Dell và Asus, sau đó đến HP và Lenovo cho chất lượng âm thanh tốt hoặc bền bỉ với máy trạm. Máy tính xách tay chạy hệ điều hành Windows hỗ trợ cho công việc và giải trí khá tốt, vì người dùng đã quen với hệ điều hành này từ hàng chục năm nay. Windows 10 cho phép tối ưu hóa trong việc xử lý, tìm kiếm, khởi động nhanh, ứng dụng universal dùng chung cho cả Windows Phone.

Thương hiệu: Dell, Asus, Acer, Hp, Samsung, Lenovo...

Laptop Macbook OS

Thông tin chung : "Ông hoàng" máy tính xách tay trong nhiều năm qua được nhiều người ưa chuộng chính là Apple. Họ đã vươn lên vị trí số 1 trong lòng người tiêu dùng bởi tính bền bỉ, hoàn thiện về phần cứng và hệ điều hành, thiết kế mỏng nhẹ, sang trọng. Bên cạnh đó công nghệ trên máy tính xách tay của Apple dẫn đầu các hãng khác, về bàn phím, chuột, chất lượng âm thanh, các cổng kết nối, độ phân giải màn hình. Hệ điều hành Mac OS phải nói là rất tuyệt, cho hiệu năng xử lý cao, ổn định, mượt mà, đặc biệt là tính bảo mật thông tin. Hỗ trợ tất cả các công việc về văn phòng, thiết kế, lập trình.

Macbook 2016, Macbook 2017, Macbook Pro 2016, Macbook Pro 2017, Macbook Air 2016, Macbook Air 2017.

Laptop Chrome OS

Thông tin chung : Ở thị trường này, máy tính xách tay Samsung đạt được nhiều thành công về bước đi đầu trong việc thiết kế chiếc máy tính xách tay chạy hệ điều hành mới. Một chiếc ultrabook có thiết kế cao cấp, chạy hệ điều hành Chrome OS, hỗ trợ các ứng dụng có sẵn cơ bản do Google xây dựng, không lo mất dữ liệu với Google Drive. Bên cạnh đó, Chromebook có giá hợp lý hơn các dòng máy tính xách tay khác.

...

Activate Windows
Go to Settings to activate Windows.

Copyright ©2023 by Thống

Hình 120 giao diện trang chủ

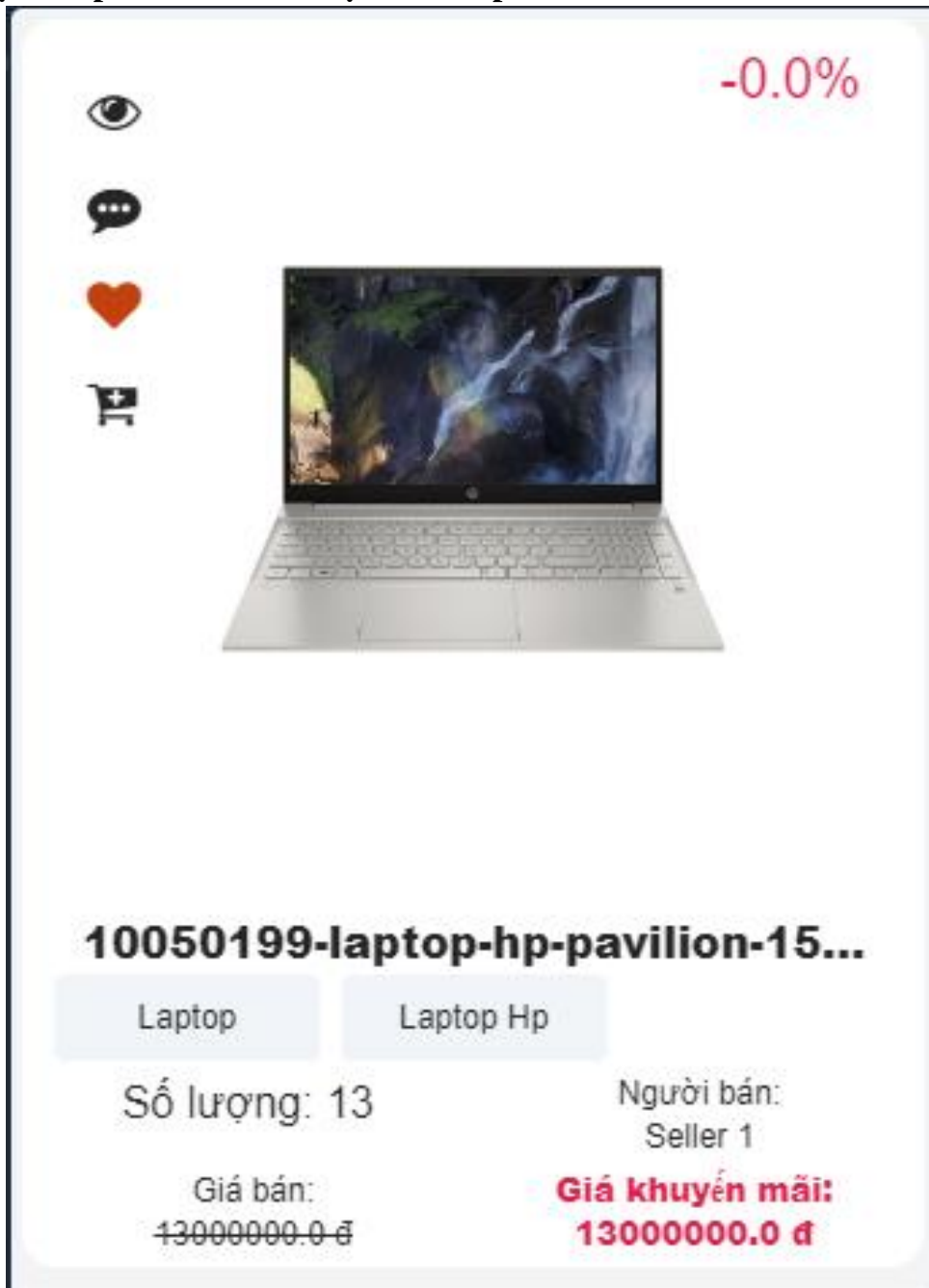
Thẻ loại

The screenshot displays a website interface for a 'Thẻ Loại' (Category) page. The top navigation bar includes 'Trang chủ', 'Thẻ Loại', 'Trò chuyện', 'Giỏ hàng', 'Thông tin', and a user profile icon labeled 'thong'. A search bar is positioned below the navigation. The main content area is a grid of product listings, each featuring an image, a price, a discount percentage, and detailed specifications. The products include various laptop models (HP Pavilion, Apple MacBook Air M1 and M2, ASUS Vivobook Flip, ASUS TUF Gaming) and desktop PCs (Lenovo IdeaCentre, ASUS ROG Strix). The bottom of the page shows a pagination control with page numbers 1 and 2, and a 'Copyright ©2023 by Thống' notice.

Product Name	Discount	Price	Quantity
10050199-laptop-hp-pavilion-15...	-0.0%	13000000.0 đ	13
Apple MacBook Air M1 256GB 2020	-33.5%	29990000.0 đ	50
Apple MacBook Air M2 2022 16GB...	-16.0%	39990000.0 đ	40
Laptop ASUS Vivobook Flip 14 T...	-31.0%	16500000.0 đ	40
Laptop ASUS TUF Gaming F15 FX5...	-25.0%	29990000.0 đ	50
PC Gaming Lenovo IdeaCentre G5...	3.0%	22990000.0 đ	50
Máy tính văn phòng All In One ...	-13.0%	46990000.0 đ	10
PC Gaming ASUS ROG Strix GT15 ...	13.0%	46990000.0 đ	30
áaf	-50.0%	42442442.0 đ	9

Hình 121 giao diện trang thẻ loại

Giao diện sản phẩm khi trỏ chuột vào sản phẩm



Hình 122 giao diện sản phẩm khi trỏ chuột

- Thanh menu sản phẩm sẽ hiện ra
- Thanh menu hỗ trợ phép :
- + Xem chi tiết sản phẩm
- + Trò chuyện với người bán
- + Thích sản phẩm
- + Thêm sản phẩm vào giỏ hàng

Sản phẩm

Trang chủ Thể Loại Trò chuyện Giỏ hàng Thông tin Login

Growpro



Apple M1 | 8 nhân GPU
8GB | 256GB | 13.3" 2K

Apple MacBook Air M1 256GB 2020

Laptop MacBook

Người bán: Seller 1 NCC: Admin
Giá tiền: ~~28990000.0~~ Số lượng: 50

Giá khuyến mãi: **19290000.0** [Đặt hàng](#)

Giới thiệu:
Phù hợp cho lập trình viên, thiết kế đồ họa 2D, dân văn phòng
Hiệu năng vượt trội - Cân mọi tác vụ từ word, excel đến chỉnh sửa ảnh trên các phần mềm như AI, PTS
Đa nhiệm mượt mà - Ram 8GB cho phép vừa mở trình duyệt để tra cứu thông tin, vừa làm việc trên phần mềm
Trang bị SSD 256GB - Cho thời gian khởi động nhanh chóng, tối ưu hoá thời gian load ứng dụng
Chất lượng hình ảnh sắc nét - Màn hình Retina cao cấp cùng công nghệ TrueTone cân bằng màu sắc
Thiết kế sang trọng - Nặng chỉ 1.29KG, độ dày 16.1mm. Tiện lợi mang theo mọi nơi

Macbook Air M1 2020 - Sang trọng, tinh tế, hiệu năng khủng
Macbook Air M1 là dòng sản phẩm có thiết kế mỏng nhẹ, sang trọng và tinh tế cùng với đó là giá thành phải chăng nên Macbook Air đã thu hút được đông đảo người dùng yêu thích và lựa chọn. Đây cũng là một trong những phiên bản Macbook Air mới nhất mà khách hàng không thể bỏ qua khi đến với Cellphones. Dưới đây là chi tiết về thiết kế, cấu hình của máy.

Thiết kế tinh tế, chất liệu nhôm bền bỉ
Macbook Air M1 là máy sản xuất toàn thủ thiết kế với những đường nét đơn nhưng vô cùng sang trọng. Máy có độ mỏng nhẹ chỉ 1,29kg và các cạnh tròn viền khiến thiết bị trở nên đẹp hơn và cao cấp hơn.
Vỏ nhôm bên ngoài mang đến sự bền bỉ vượt trội theo thời gian.

Thông số kỹ thuật

Vị xử lý & đồ họa
Loại card đồ họa: GPU 7 nhân, 16 nhân Neural Engine
Loại CPU: 8 nhân với 4 nhân hiệu năng cao và 4 nhân tiết kiệm điện

RAM & Ổ cứng
Dung lượng RAM: 8GB
Loại RAM: LPDDR4
Dung lượng Ổ cứng: 256GB
Loại ổ cứng: SSD

Thông số khác
Chất liệu: Vỏ kim loại

Màn hình
Màn hình cảm ứng: Không
Chất liệu tấm nền: Tấm nền IPS
Độ phân giải: Retina
Kích thước màn hình: 13.3 inches
Công nghệ màn hình: Độ sáng 400 nits, Hỗ trợ dải màu P3, True-Tone
Độ phân giải màn hình: 2560 x 1600 pixels (2K)

Công nghệ âm thanh
Công nghệ âm thanh: 3 microphones, headphones, loa kép (2 kênh)

Giao tiếp & kết nối
Khe đọc thẻ nhớ: Không
Hệ điều hành: macOS Big Sur
Wi-Fi: 802.11ax Wi-Fi 6 wireless networking
Bluetooth: Bluetooth 5.0

Tiện ích khác
Tính năng đặc biệt: Ổ cứng SSD, Viên màn hình siêu mỏng, Wi-Fi 6, Bảo mật vân tay

Pin & công nghệ sạc
Pin: 49.9-watt lithium-polymer, củ sạc công suất 30W

Thiết kế & Trọng lượng
Kích thước: 1.61 cm x 30.41 cm x 21.24 cm
Trọng lượng: 1.29 kg

Thông số kỹ thuật
Cổng giao tiếp: 2 cổng Thunderbolt / USB 4

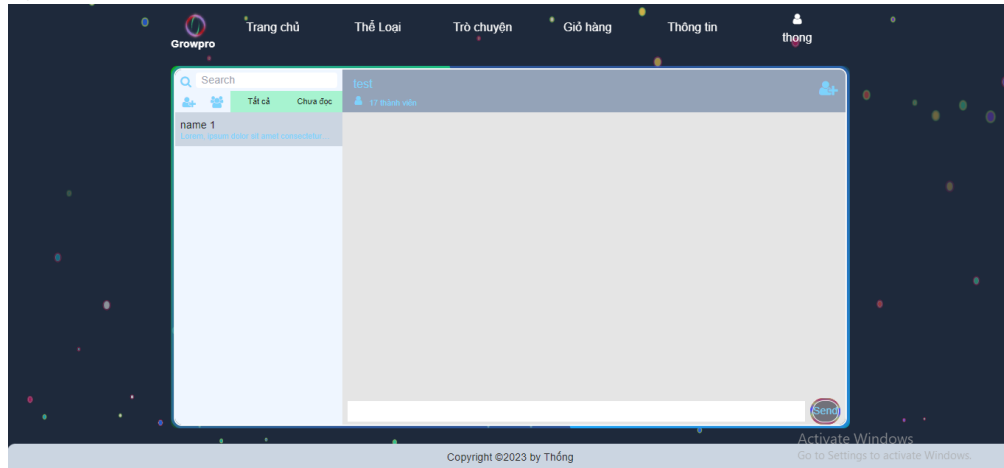
Tiện ích
Tính năng khác: Cảm biến vân tay Touch ID, 720p FaceTime HD camera

Activate Windows
Go to Settings to activate Windows.

Copyright ©2023 by Thống

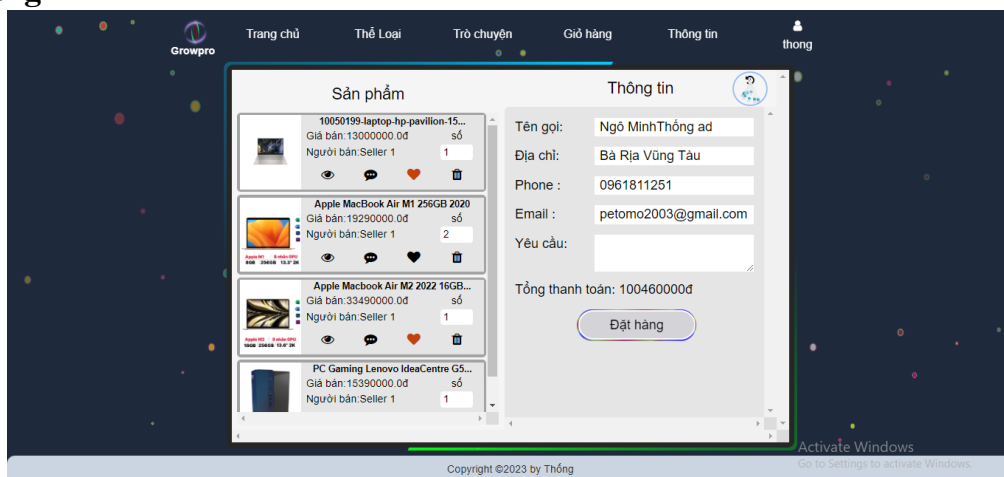
Hình 123 giao diện sản phẩm khi xem chi tiết

Trò chuyện



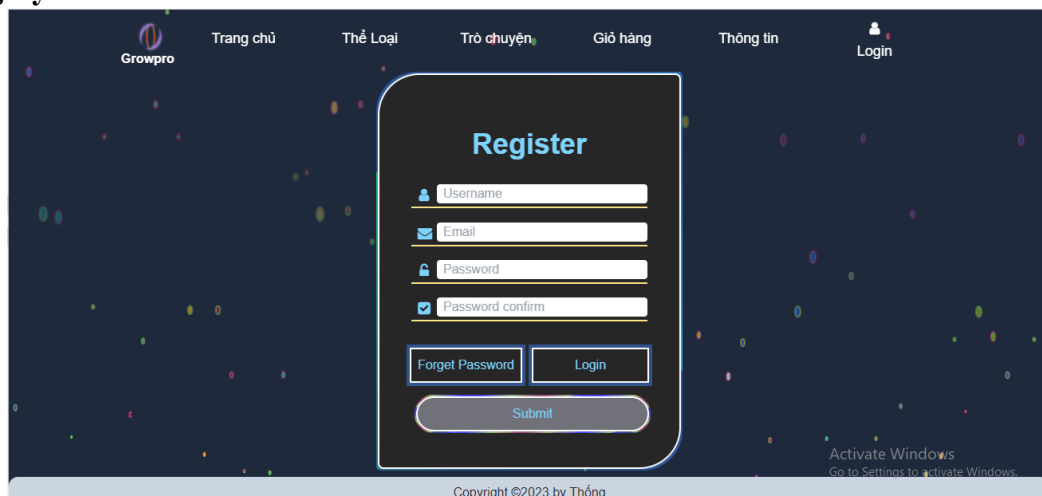
Hình 124 giao diện trang trò chuyện

Giỏ hàng



Hình 125 giao diện trang giỏ hàng

Đăng ký



Hình 126 giao diện trang đăng ký

Đăng nhập(username)

The screenshot shows the Growpro website's login interface. At the top, there is a navigation bar with the Growpro logo and menu items: Trang chủ, Thẻ Loại, Trò chuyện, Giỏ hàng, Thông tin, and Login. The main content area features a dark-themed login form with the title "Login". It includes two input fields: "Username" and "Password". Below these fields are two buttons: "Login by Email" and "Register". A large "Submit" button is positioned at the bottom of the form. The footer contains the text "Copyright ©2023 by Thống" and a Windows activation notice: "Activate Windows Go to Settings to activate Windows."

Hình 127 giao diện trang đăng nhập với username

Đăng nhập(email)

The screenshot shows the Growpro website's login interface for email authentication. The layout is identical to the previous page, but the input fields are labeled "Email" and "Password". The buttons below the fields are "Forget Password" and "Register". The "Submit" button remains at the bottom. The footer text is the same: "Copyright ©2023 by Thống" and "Activate Windows Go to Settings to activate Windows."

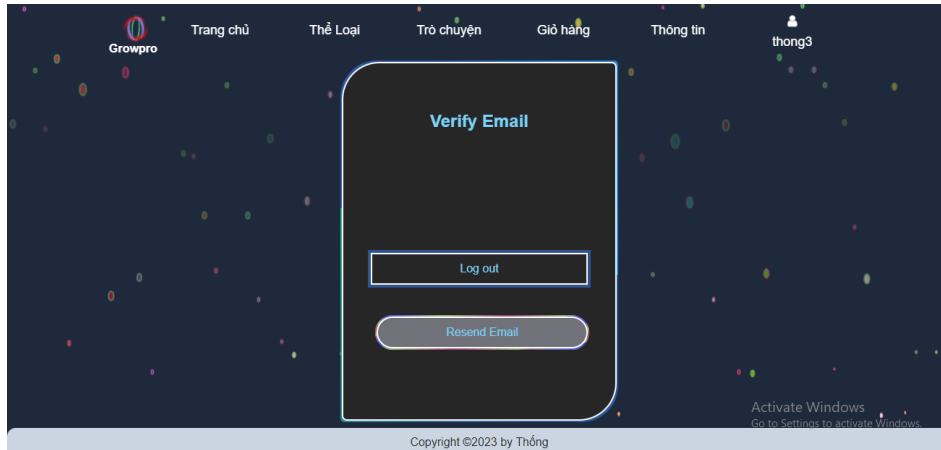
Hình 128 giao diện trang đăng nhập với email

Lấy lại mật khẩu

The screenshot shows the Growpro website's password reset interface. The navigation bar and footer are consistent with the previous pages. The main content area features a dark-themed form titled "Reset Pass". It has a single input field labeled "Email". Below the field are two buttons: "Login By Email" and "Register". A large "Send" button is located at the bottom of the form. The footer text is "Copyright ©2023 by Thống" and "Activate Windows Go to Settings to activate Windows."

Hình 129 giao diện trang lấy lại mật khẩu

Xác thực email



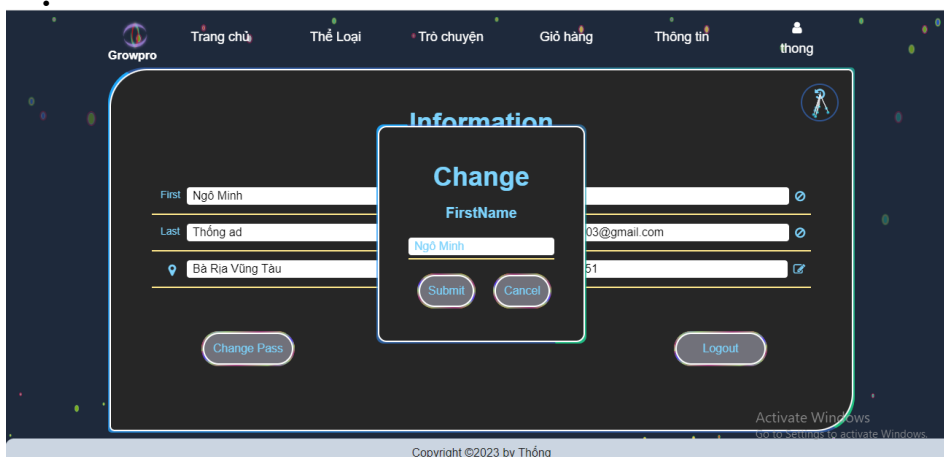
Hình 130 giao diện trang xác thực email

Thông tin người dùng



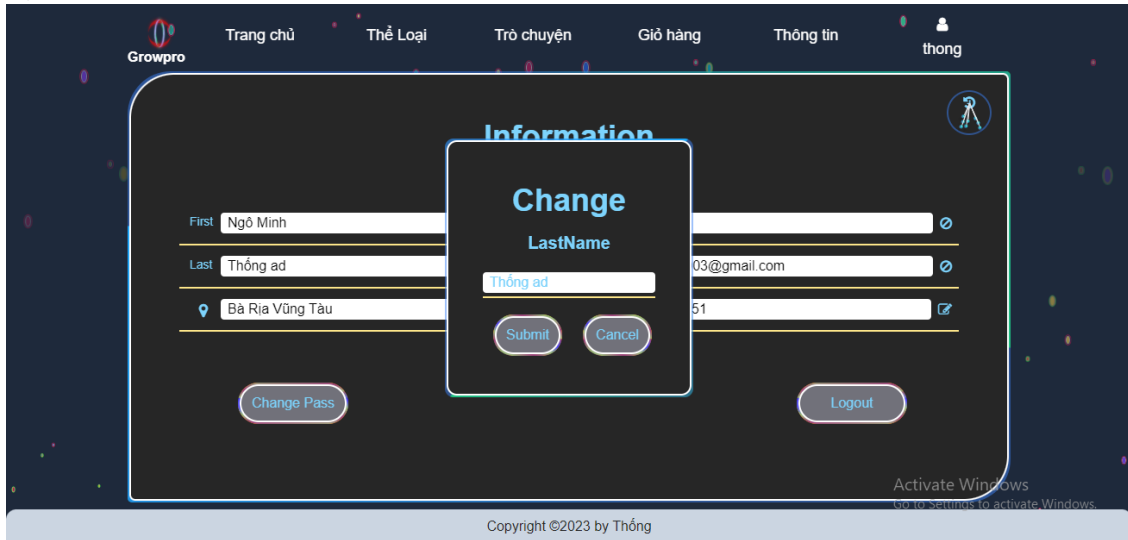
Hình 131 giao diện trang thông tin người dùng

Thay đổi họ



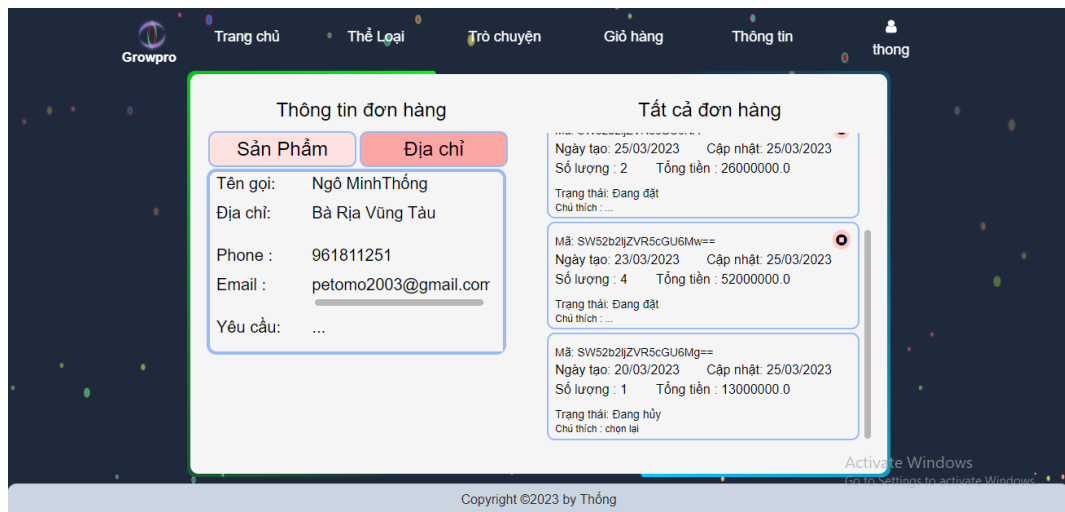
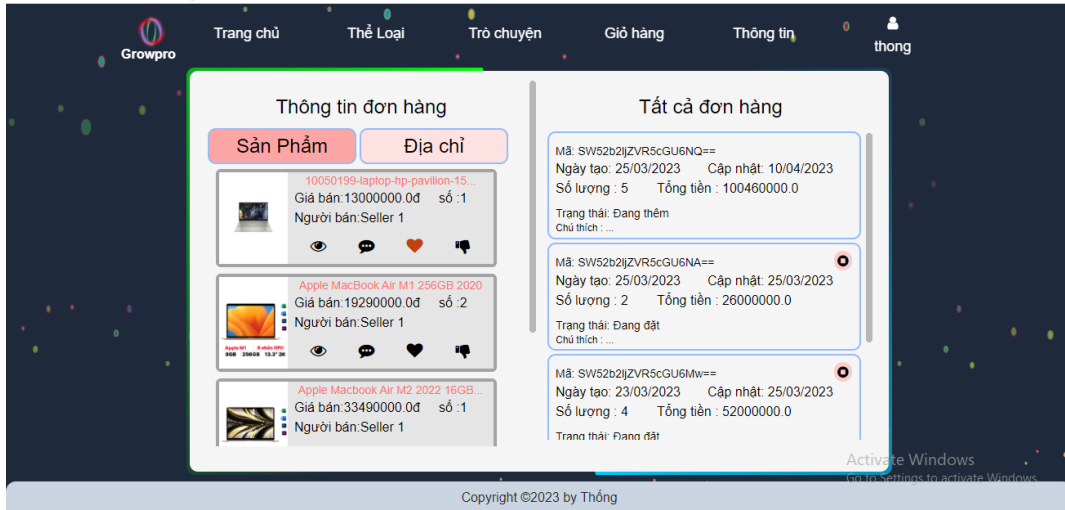
Hình 132 giao diện thay đổi họ

Thay tên



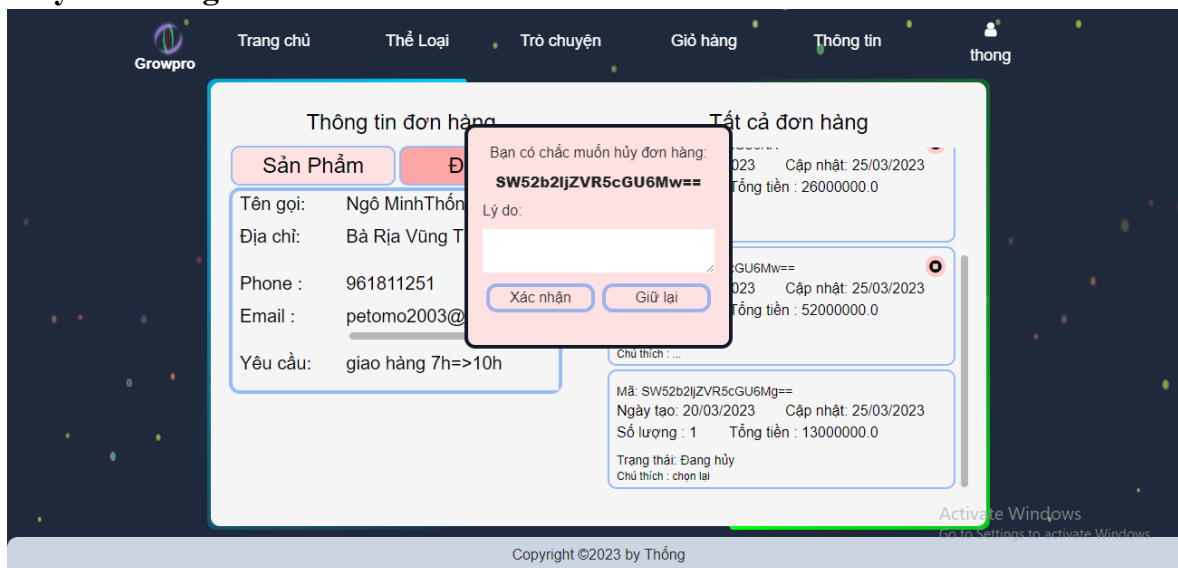
Hình 133 giao diện thay đổi tên

Lịch sử mua hàng



Hình 134 giao diện trang lịch sử mua hàng – sản phẩm

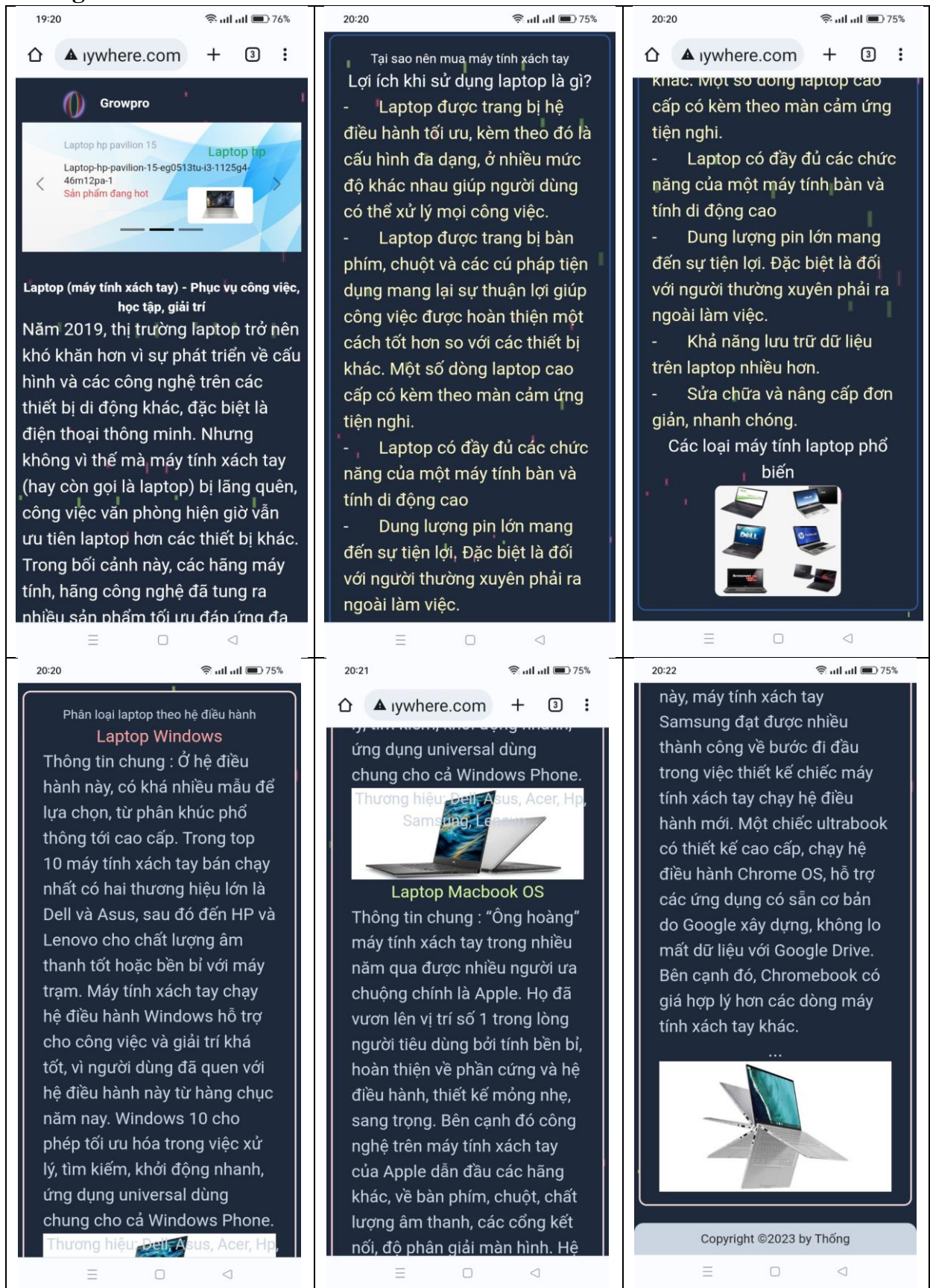
Hủy đơn hàng



Hình 135 giao diện yêu cầu hủy đơn hàng

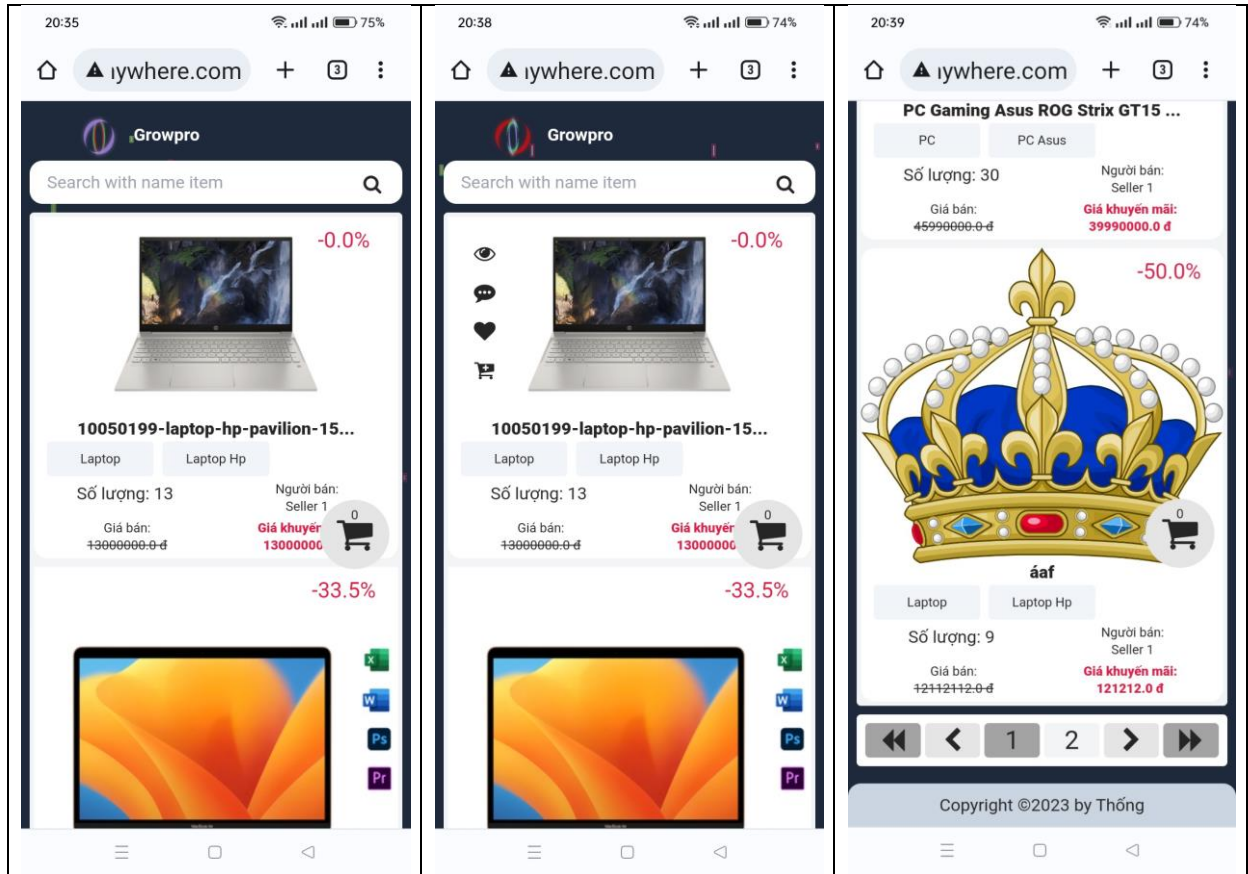
e) Các giao diện người dùng trên điện thoại:

Trang chủ



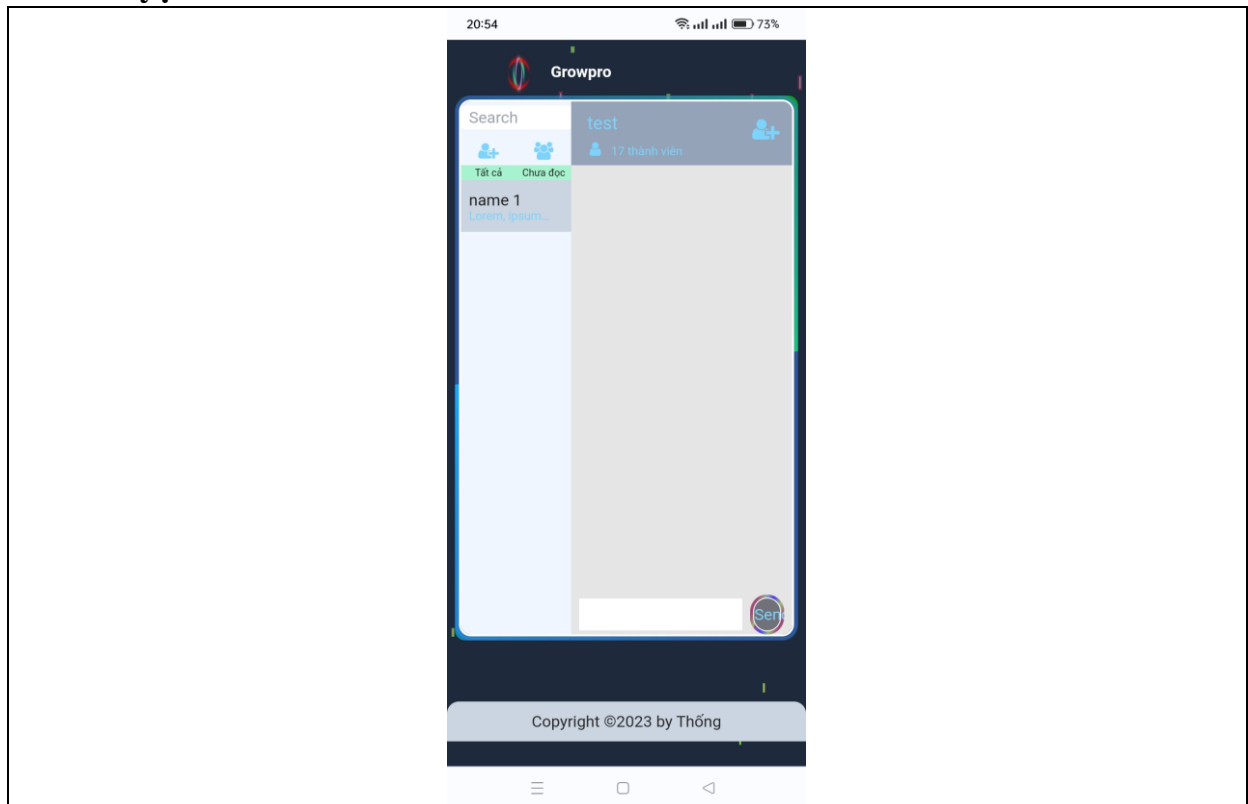
Hình 136 giao diện trang chủ trên điện thoại

Thẻ loại



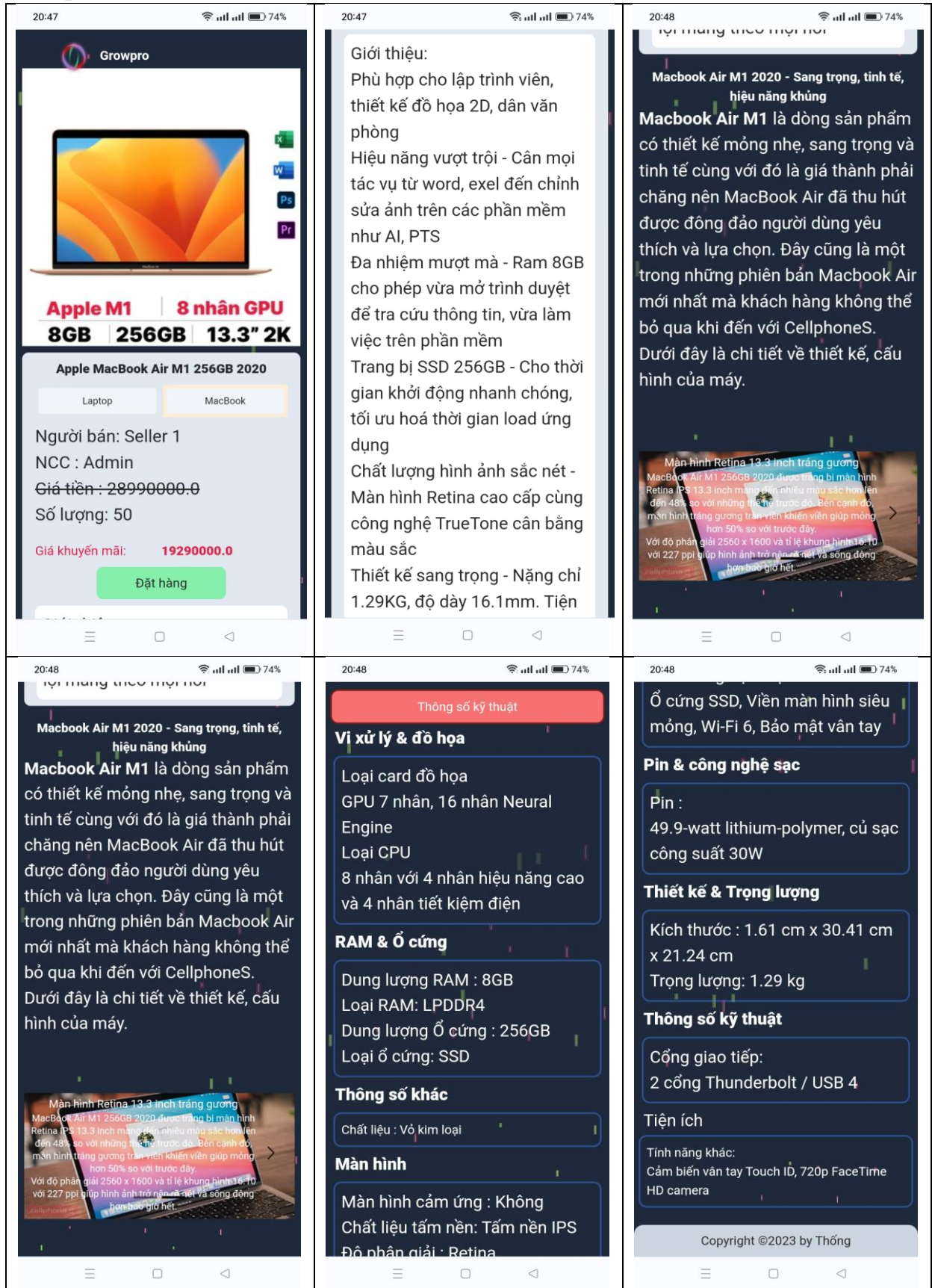
Hình 137 giao diện trang thẻ loại trên điện thoại

Trò chuyện



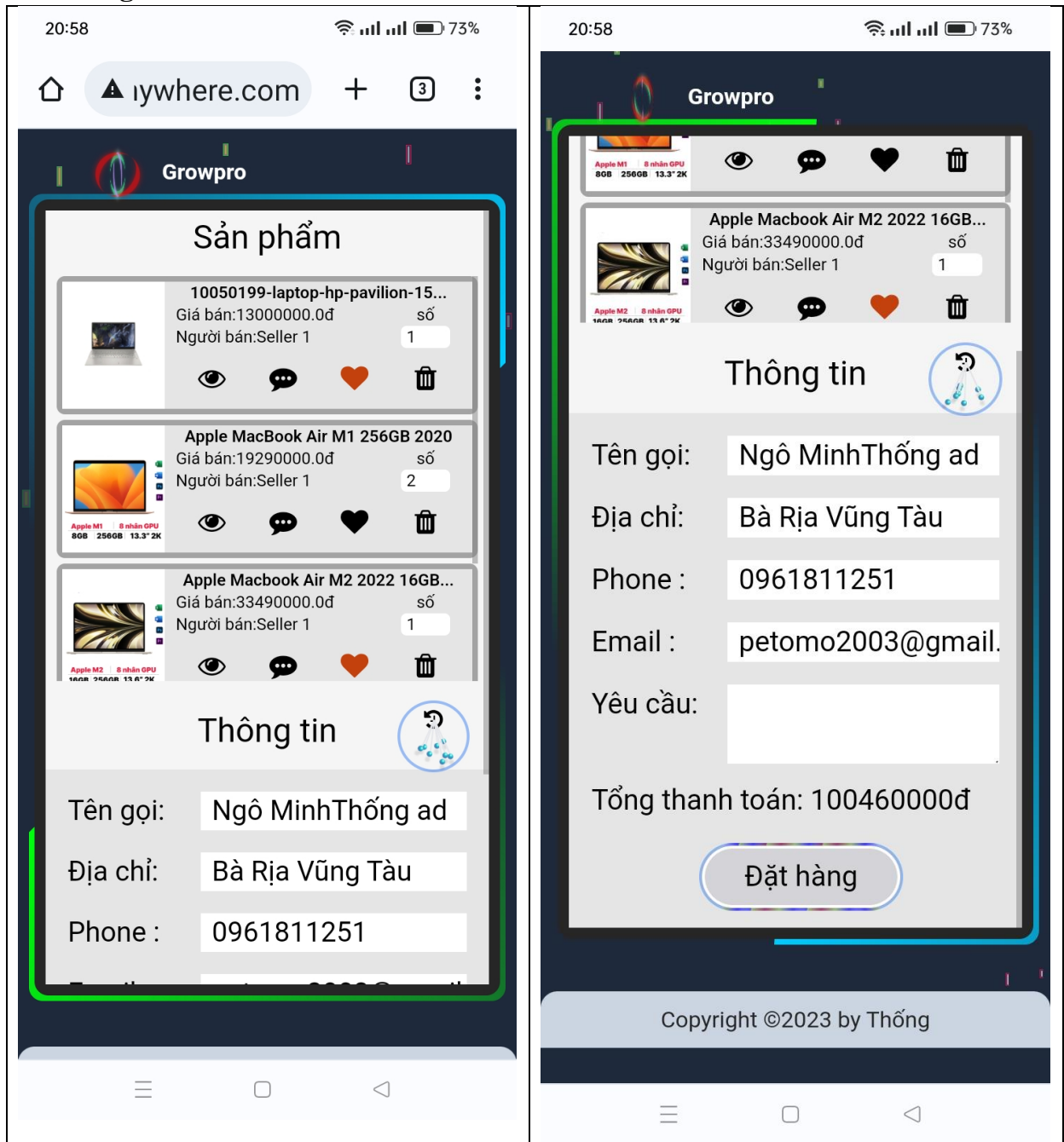
Hình 138 giao diện trang trò chuyện trên điện thoại

Sản phẩm



Hình 139 giao diện trang chủ trên điện thoại


Giỏ hàng



Hình 140 giao diện trang giỏ hàng trên điện thoại

Đăng ký

21:03 📶 🔋 73%


 **Growpro**

Register

Password confirm

[Forget Password](#) [Login](#)


Copyright ©2023 by Thống



Hình 141 giao diện trang đăng ký trên điện thoại

Đăng nhập

21:06 📶 🔋 73%

**Growpro**

Login

Username

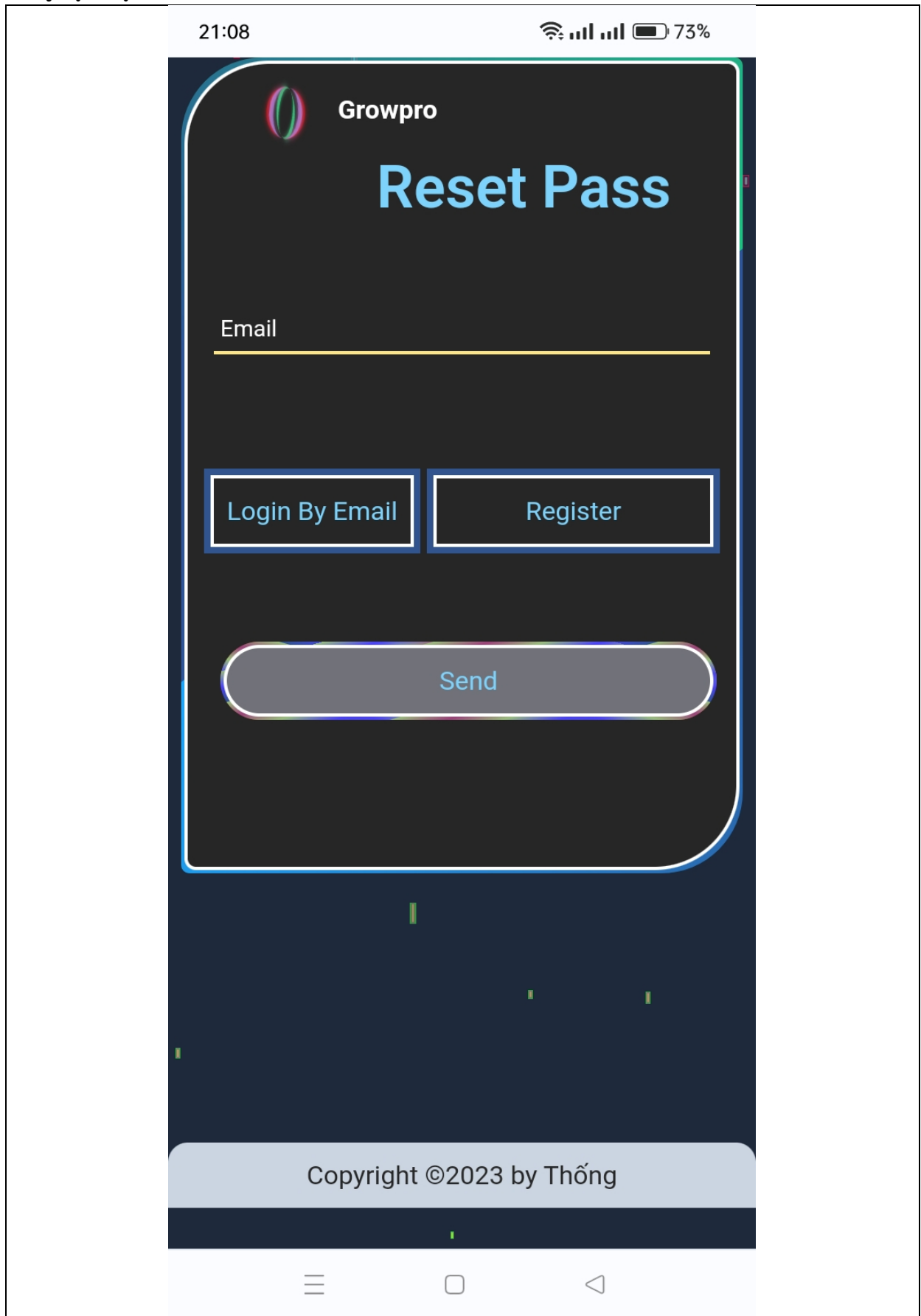
Password

Copyright ©2023 by Thống

☰ ○ ◀

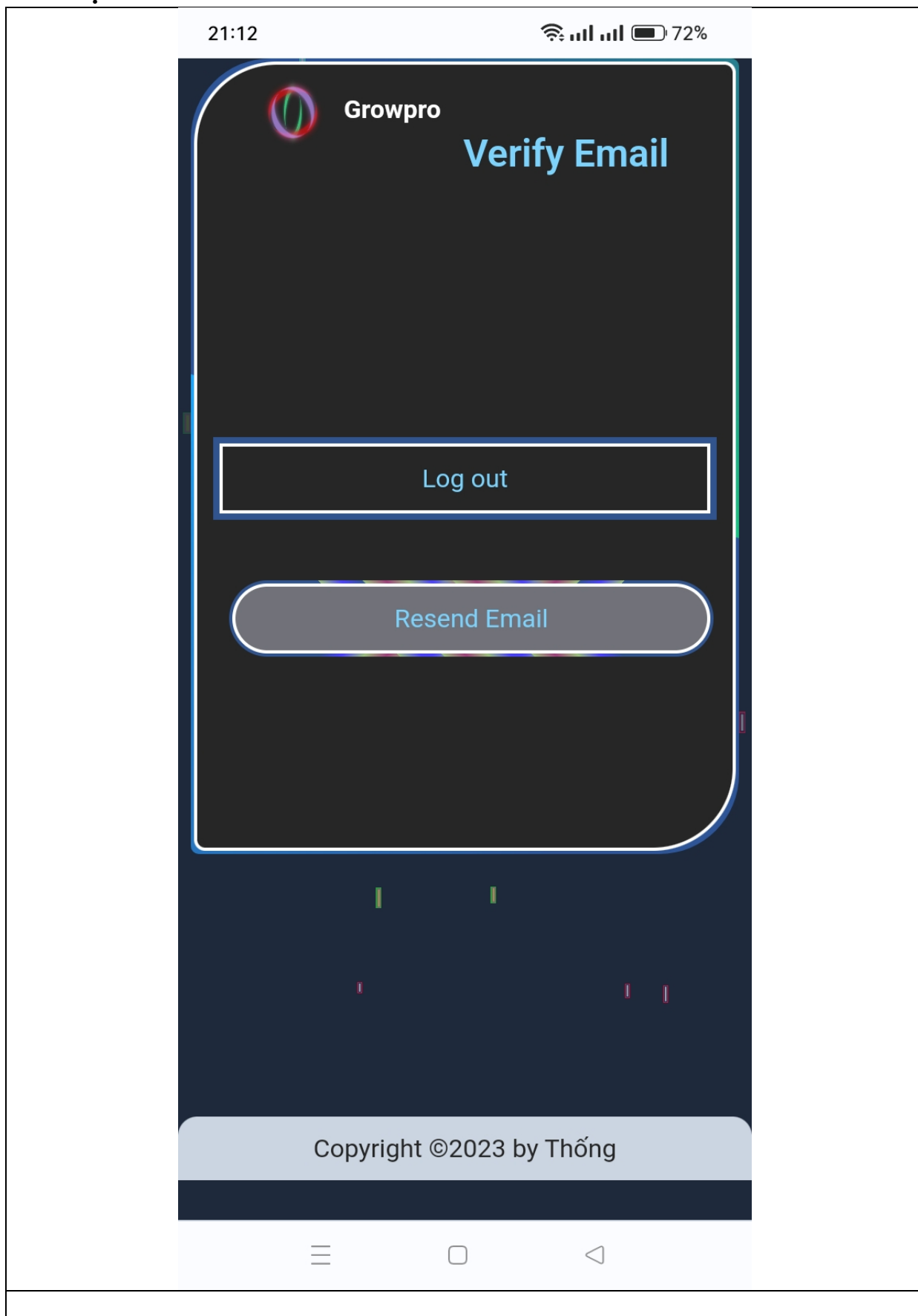
Hình 142 giao diện trang đăng nhập trên điện thoại

Lấy lại mật khẩu



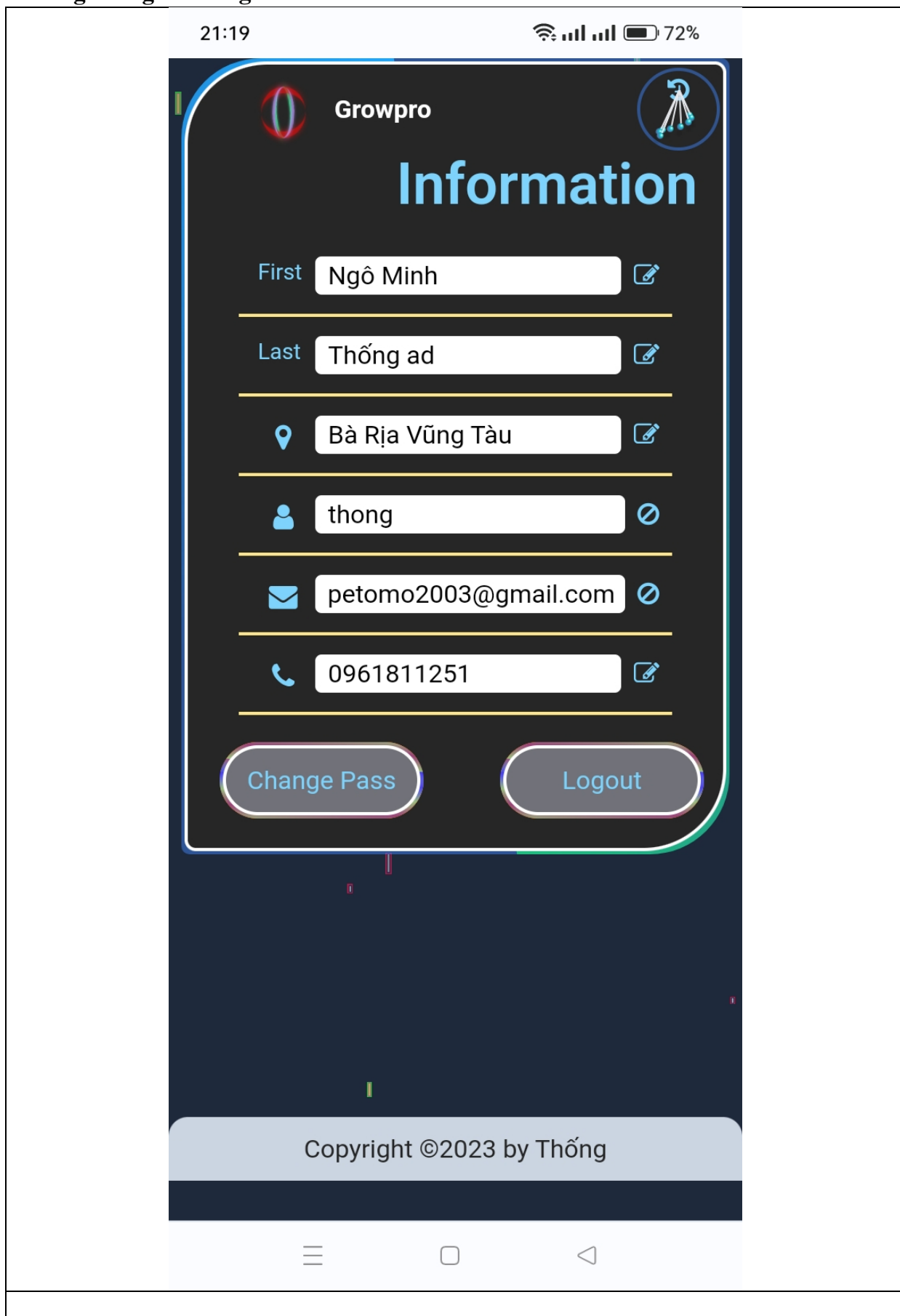
Hình 143 giao diện trang quên mật khẩu trên điện thoại

Xác thực email



Hình 144 giao diện trang xác thực email trên điện thoại

Thông tin người dùng



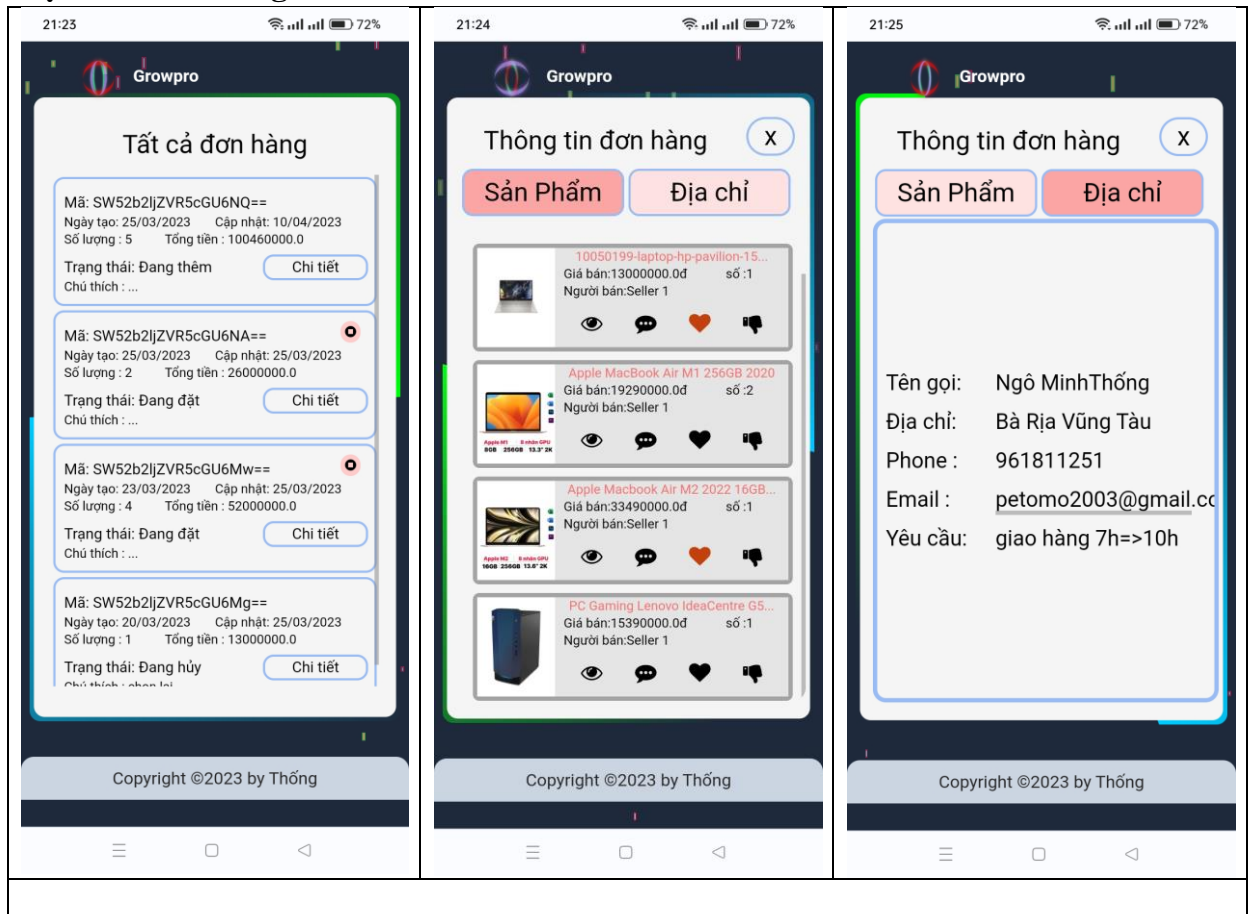
Hình 145 giao diện trang thông tin người dùng trên điện thoại

Thay đổi thông tin



Hình 146 giao diện thay đổi thông tin người dùng trên điện thoại

Lịch sử mua hàng



Hình 147 giao diện trang lịch sử mua hàng trên điện thoại

3) Server Admin(Flask)

a) Các thư viện sử dụng

Tương đồng các bộ thư viện trong server Client:

b) Hướng dẫn cài đặt

- ❖ Cài biến môi trường : python -m venv venv
- ❖ Khởi động môi trường : venv\Scripts\activate
- ❖ Di chuyển tới thư mục root : growpro_sv
- ❖ Cài đặt các bộ thư viện: pip install -r requirements.txt
- ❖ Đóng bộ các bộ thư viện : pip freeze
- ❖ Vào file growpro_sv/settings.py cấu hình thông tin pj:

```
SECRET_KEY="8c7dbd270cb98e83f9d8d57fb8a2ab7bac9d755asf41sg51"

#url_backend=f"http://127.0.0.1:8000/"
url_backend=f"https://thong1993.pythonanywhere.com/"
endpoint = f"{url_backend}graphql"
#app
app_settings={
    'host':'127.0.0.1',
    'port':5000,
    'debug':True,
}

#background
item_backgroud=40

#item
number_item=9
item_catergory=3
item_catergory_len=13
item_tilte_len=33
```

Cấu hình thông tin địa chỉ truyền và nhận dữ liệu:

```
url_backend=f"https://thong1993.pythonanywhere.com/"
endpoint = f"{url_backend}graphql"
```

Cấu hình thông tin hiển thị

```
#background
item_backgroud=40
#item
number_item=9
item_catergory=3
item_catergory_len=13
item_tilte_len=33
```

c) Cấu trúc thư mục và chức năng:

<pre> v SUPADMIN v growpro_sv > __pycache__ > graphql > page > static > templates apps.py export_function_socket.py export_function_urls.py models.py requirements.txt settings.py > venv </pre>	<p><code>__pycache__</code>: thư mục do hệ thống tạo quản lý các file sẽ được sử dụng</p> <p>GraphQL: thư mục quản lý các api truy vấn tài nguyên</p> <p>Page: thư mục quản lý các file code py chính</p> <p>Static: thư mục quản lý các file khác: js, css, image,...</p> <p>Templates: thư mục quản lý các file html</p> <p>Apps.py : file quản lý và chỉ định các đường dẫn</p> <p>Export_function_socket.py: file quản lý các hàm sử lý sự kiện từ socket.io</p> <p>Export_function_urls.py: file quản lý các hàm sử lý sự kiện từ flask</p> <p>Models.py: file quản lý database</p> <p>requirements.txt: file liệt kê danh sách các thư viện sử dụng có trong venv</p> <p>settings.py: file quản lý các thiết lập chung của chương trình</p> <p>venv: thư mục chứa các thư viện/thiết lập để chạy chương trình.</p>
--	--

- GraphQL:

<pre> v graphql > __pycache__ v control > __pycache__ cart.py invoice.py item.py layout.py menu.py page.py user.py v query > __pycache__ cart.py invoice.py item.py layout.py menu.py page.py user.py settings.py </pre>	<p><code>__pycache__</code>: thư mục do hệ thống tạo quản lý các file sẽ được sử dụng</p> <p>Thư mục control: quản lý các file chứa hàm quản lý tài nguyên</p> <p>Thư mục query: quản lý các file chứa các câu lệnh thực thi</p> <p>File cart.py: file chứa các câu lệnh liên quan với giỏ hàng</p> <p>File layout.py: file chứa các câu lệnh liên quan với thành phần trang</p> <p>File invoice.py: file chứa các câu lệnh liên quan với hóa đơn</p> <p>File item.py: file chứa các câu lệnh liên quan với sản phẩm</p> <p>File menu.py: file chứa các câu lệnh liên quan với menu</p> <p>File page.py: file chứa các câu lệnh liên quan với các thành phần trong trang</p> <p>File user.py: file chứa các câu lệnh liên quan với người dùng</p> <p>File settings.py: chứa các cấu hình chung của các hàm.</p>
---	---

- Page:

<ul style="list-style-type: none">▼ page> __pycache__▼ chat> __pycache__🔗 chat.py▼ item> __pycache__🔗 item.py▼ layout> __pycache__🔗 header.py🔗 layout.py🔗 page.py▼ user> __pycache__🔗 changepass.py🔗 extend.py🔗 information.py🔗 login.py🔗 register.py🔗 verify.py	<p>__pycache__: thư mục do hệ thống tạo quản lý các file sẽ được sử dụng</p> <p>Thư mục chat quản lý các file chứa các hàm liên quan về hội thoại</p> <p>Thư mục item: quản lý các file chứa các hàm liên quan về sản phẩm</p> <p>Thư mục user: quản lý các file chứa các hàm liên quan về người dùng</p> <p>File chat.py chứa các hàm liên quan về hội thoại</p> <p>File item.py: chứa các hàm liên quan về sản phẩm</p> <p>File layout.py: chứa các hàm liên quan về thành phần trang</p> <p>File header.py: chứa các hàm liên quan về menu</p> <p>File page.py: chứa các hàm liên quan về trang</p> <p>File changepass.py: chứa các hàm liên quan về thay đổi mật khẩu</p> <p>File extend.py: chứa các hàm khác</p> <p>File information.py: chứa các hàm về thông tin người dùng</p> <p>File login.py: chứa các hàm về đăng nhập</p> <p>File register.py: chứa các hàm về đăng ký</p> <p>File verify.py: chứa các hàm về xác nhận</p>
---	--

- Static:

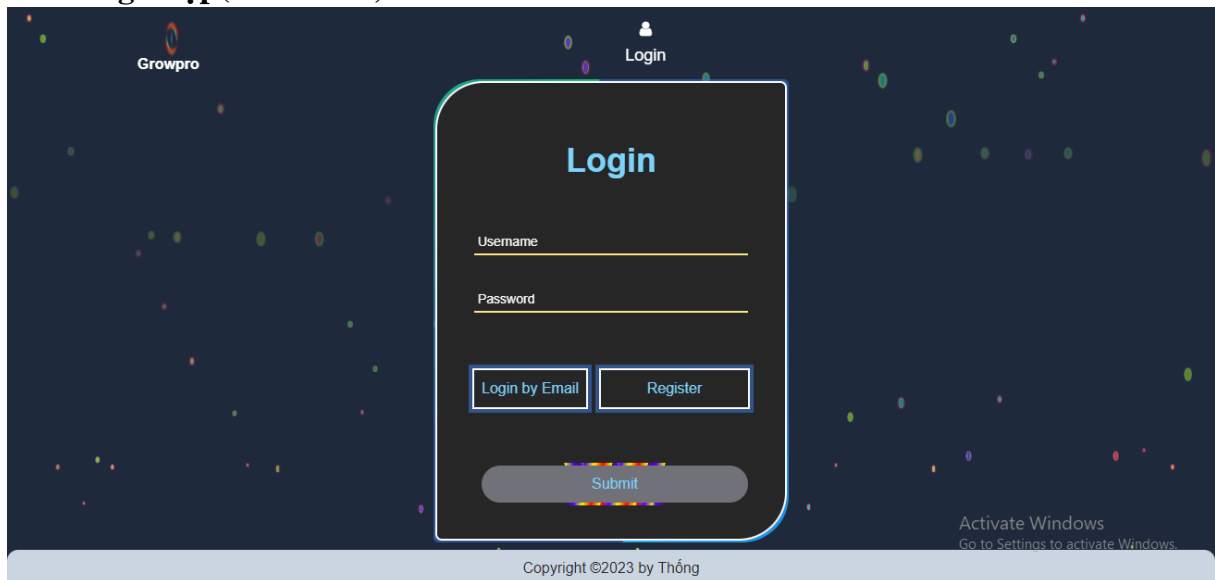
<ul style="list-style-type: none">▼ static▼ css# font-awesome.min.css# font-googleapis.css# main.css# responsive.css# tailwind.index.min.css▼ fonts🔗 fontawesome-webfont.ttf🔗 fontawesome-webfont.woff🔗 fontawesome-webfont.woff2▼ img▼ major🔗 find-out-more-bt.png🔗 main_bg.jpg🔗 main_bg.png▼ jsJS index.min.js.mapJS main.jsJS socket.4.0.1.io.jsJS socket.io.js.mapJS tailwind.index.min.jsJS tailwindcss.js	<p>Thư mục css quản lý các file css</p> <p>Thư mục fonts quản lý các file font</p> <p>Thư mục img: quản lý các file ảnh</p> <p>Thư mục js: chứa các file/thư viện js</p>
---	--

- Templates:

<pre> └─ templates └─ layouts ├── macros ├── layout.html └─ layoutupdate.html └─ pages ├── macros └─ page.html └─ users ├── macros ├── changePass.html ├── getpass.html ├── information.html ├── login_email.html ├── login.html ├── register.html ├── resetpass.html ├── verify.html └─ base.html </pre>	<p>File base.html: là một file template cơ bản chứa các phần chung của một trang web. Như header, footer, menu, thanh điều hướng và mã JavaScript hoặc CSS chung.</p> <p>Thư mục macros: quản lý các file chứa các đoạn mã macros, dùng để định nghĩa các thành phần/khối lặp lại một cách dễ dàng.</p> <p>Thư mục pages: quản lý các file chứa các đoạn mã liên quan tới trang</p> <p>Thư mục layouts: quản lý các file chứa các đoạn mã liên quan tới thành phần trang</p> <p>Thư mục users: quản lý các file chứa các đoạn mã liên quan tới người dùng</p>
---	---

d) Các giao diện người dùng:

Đăng nhập(username)



Hình 148 giao diện trang đăng nhập với username

Thông tin người dùng

Growpro Trang chủ Layout Đơn hàng Sản phẩm Trò chuyện thong

Information

First: Ngô Minh

Last: Thông ad

Email: petomo2003@gmail.com

Phone: 0961811251

Change Pass Logout

Activate Windows
Go to Settings to activate Windows.

Copyright ©2023 by Thông

Hình 149 giao diện trang thông tin người dùng

Danh sách layout

Growpro Trang chủ Layout Đơn hàng Sản phẩm Trò chuyện thong

danh sách các layout

user: ... show: True active: True

Category: style: carousel

name: top_banner_home1 id: TGF5b3V0VHlwZTox

title: top carousel home titleStyle: text-center

parent: 0 page: home priority: 1 interval: 6000

item

Style:

dest: Đây là top banner

destStyle:

background: cập nhật Xóa Layout

Add Layout

Activate Windows
Go to Settings to activate Windows.

Hình 150 giao diện trang danh sách layout

Thêm layout

Growpro Trang chủ Layout Đơn hàng Sản phẩm Trò chuyện thong

New Layout

user: thong show: True active: True priority: 1 interval: 5000

Page: None ItemSeller: None

Category: name:

title: titleStyle: Nan

Style: Nan destStyle: Nan

dest:

background: clear

Chọn tệp Chưa có tệp nào được chọn

Lưu Hủy

Activate Windows
Go to Settings to activate Windows.

Hình 151 giao diện thêm layout

Cập nhật layout

Growpro Trang chủ Layout Đơn hàng Sản phẩm Trò chuyện thông

Xem trước

Laptop hp pavilion 15

Laptop-hp-pavilion-15-eg0513tu-i3-1125g4-46m12pa-1

Sản phẩm đang hot

Laptop hp



Chỉnh sửa

Layout chính

user: ... show: True active: True

Category: style: carousel group: content

name: top_banner_home1 id: TGF5b3V0VHlwZTox

title: top carosel home titleStyle: text-center

parent: 0 page: home priority: 1 interval: 6000

item

Style: dest: <p class="w-full">Đây là top banner</p>

destStyle: background:

Activate Windows

Layouts con

user: thong show: True active: True

Category: style: carousel group: content

name: test id: TGF5b3V0VHlwZToyNA==

title: test titleStyle: Nan

parent: top_banner_home1 page: home priority: 1 interval: 5000

item

Style: Nan dest: destStyle: Nan background:



Items

Thêm Item

user: ... show: True active: True

Category:
 style: carousel group: content
 name: top_banner_home_lo1 id: TGF5b3V0VHwZToy
 title: Laptop hp titleStyle: text-right text-green-600 text-right -mt-3 -mr-2
 parent: top_banner_home1 page: home priority: 3 interval: 5000

item
 ...
 Style:
 dest:
 <p>Laptop hp pavilion 15-eg0513tu-3-1125g4-46m12pa-1</p>
 destStyle: text-left text-stone-900 relative lg.my-4
 background:



Items

Item

title: Laptop hp pavilion 15 show: True active: True
 titleStyle: top-1 lg:top-5 lg:bottom-[-80px] absolute text-slate-400
 name: top_banner_home_lo1_it1 id:TGF5b3V0X2lZ1R5cGU6MQ==
 Style: priority: 3
 dest:
 <p>Sản phẩm đang hot</p>
 destStyle: text-left text-red-500 lg.ml-4
 avatar:

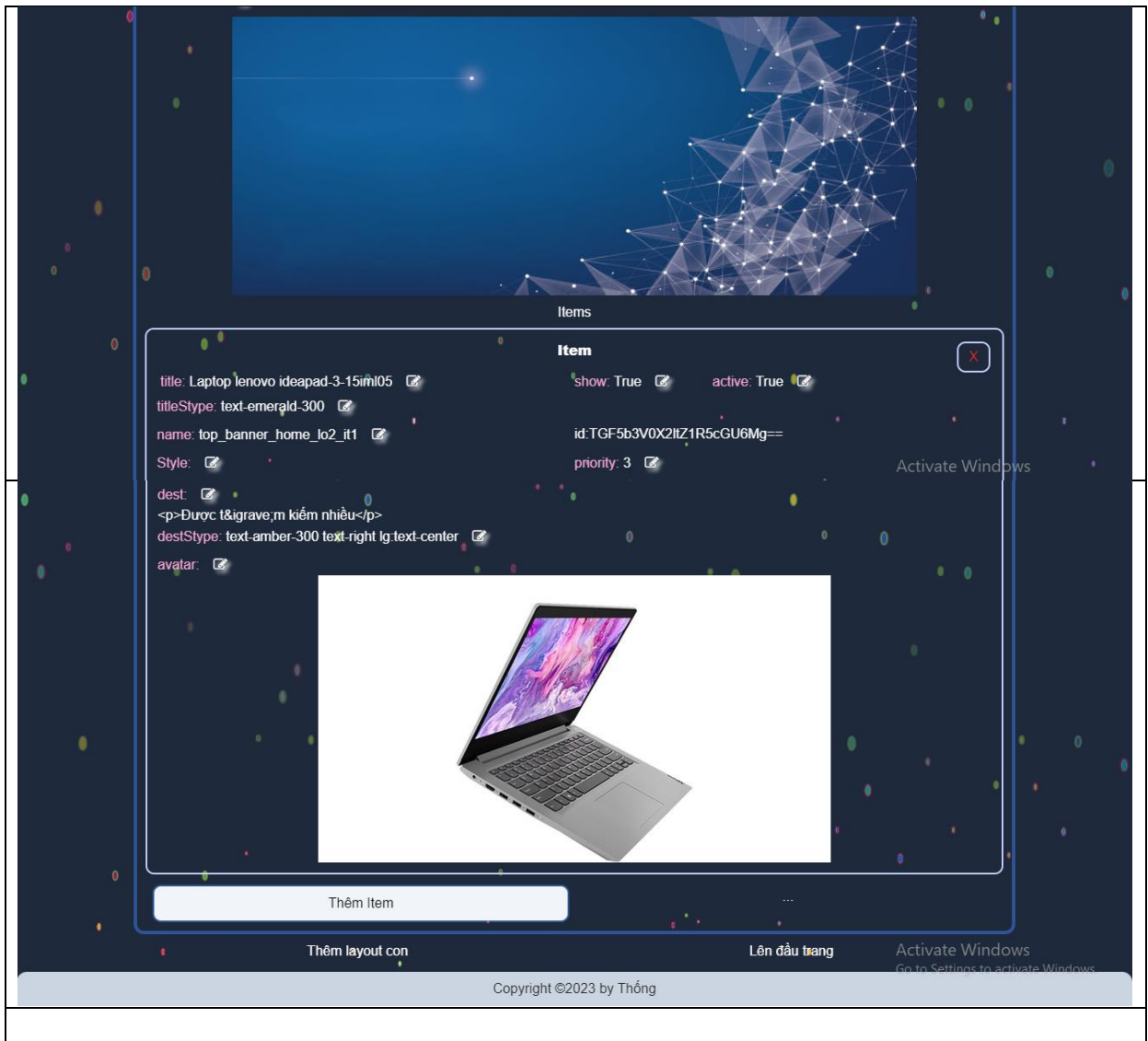


Thêm Item

user: ... show: True active: True

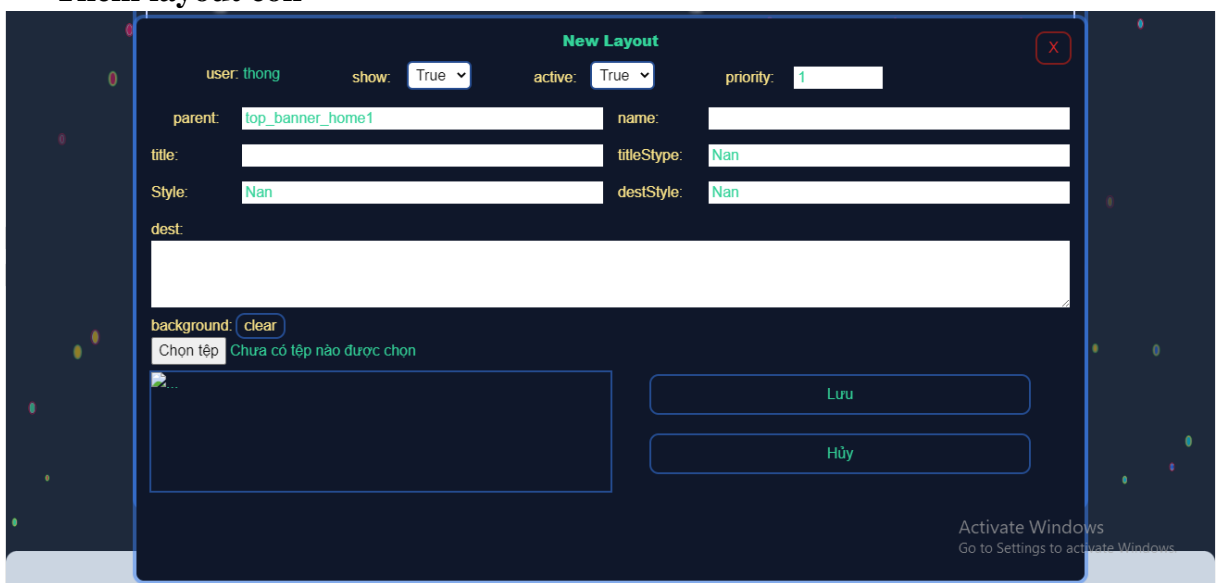
Category:
 style: carousel group: content
 name: top_banner_home_lo2 id: TGF5b3V0VHwZToz
 title: Laptop Lenovo titleStyle: text-left text-orange-500
 parent: top_banner_home1 page: home priority: 3 interval: 5000

item
 ...
 Style:
 dest:
 <p>Sản Phẩm được tìm kiếm nhiều nhất</p> Xem thêm
 destStyle: text-left text-white
 background:



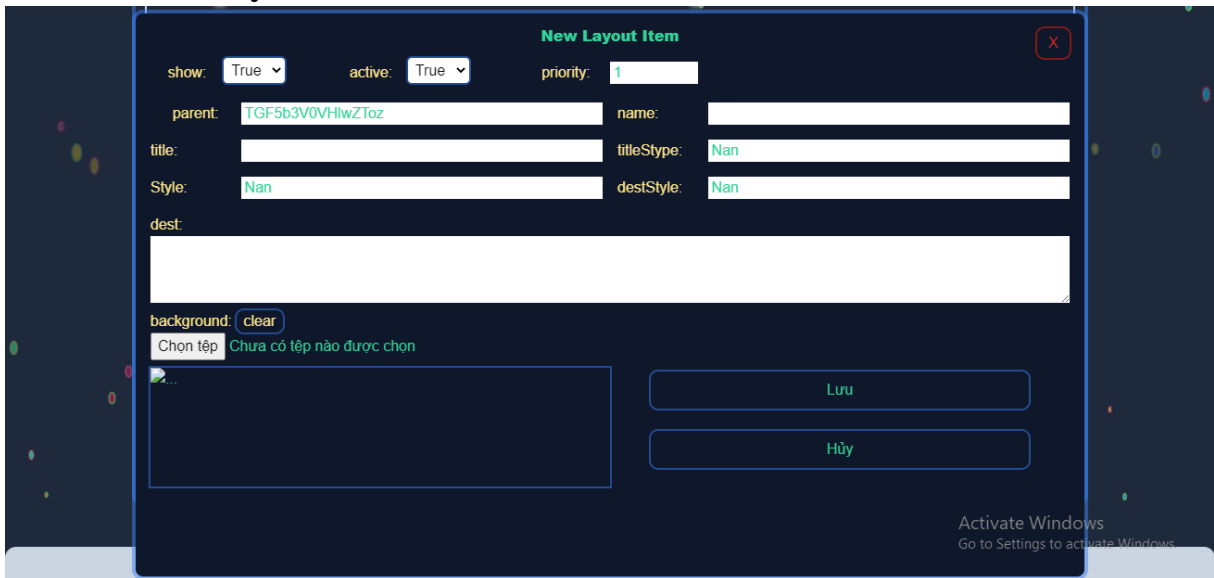
Hình 152 giao diện cập nhật layout

Thêm layout con



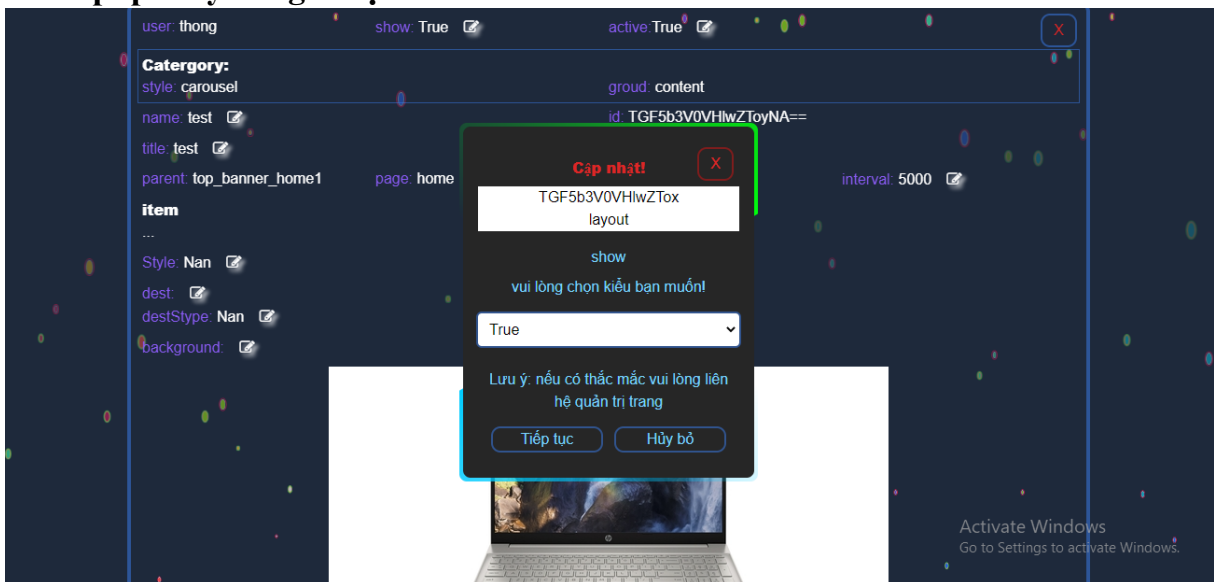
Hình 153 giao diện thêm layout con

Thêm item layout



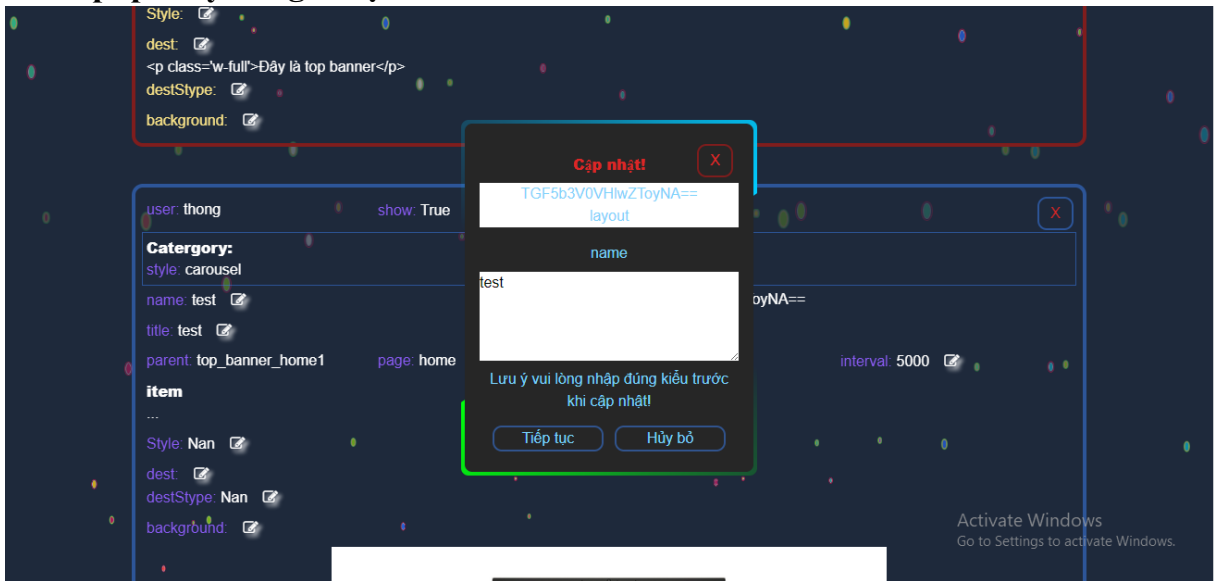
Hình 154 giao diện thêm item layout

Popup thay đổi giá trị kiểu combobox



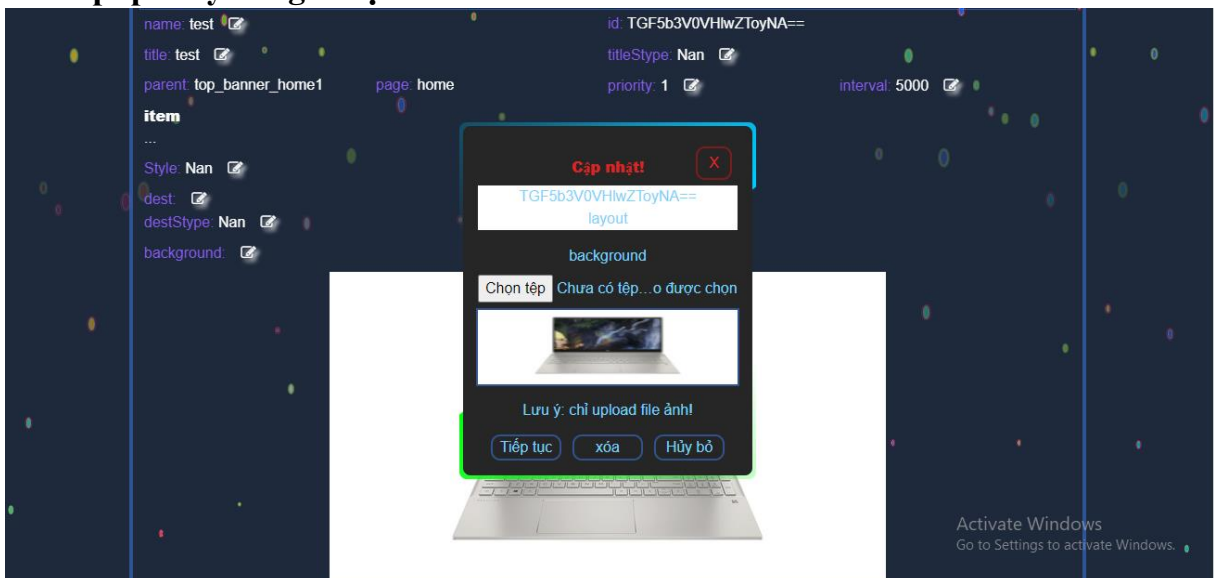
Hình 155 giao diện popup thay đổi giá trị kiểu combobox

Popup thay đổi giá trị kiểu text



Hình 156 giao diện popup thay đổi giá trị kiểu text

Popup thay đổi giá trị kiểu file



Hình 157 giao diện popup thay đổi giá trị kiểu file

V. KẾT LUẬN

1) Kết quả đạt được

- Xây dựng được một hệ thống đơn giản từ các kiến thức và kỹ năng được học. Nó được xem như bộ khung để về sau mở rộng nếu phù hợp
- Nâng cao khả năng tìm hiểu các bộ thư viện mã nguồn mở để ứng dụng
- Nâng cao khả năng phân tích hệ thống
- Hiểu rõ hơn về các thuật toán giải quyết vấn đề
- Biết thêm về các cách sử dụng kết hợp nhiều loại ngôn ngữ cùng lúc để giải quyết vấn đề
- Hiểu thêm về các luồng hoạt động của hệ thống khi vận hành

2) Hướng phát triển

- Hoàn thiện thêm các chức năng chưa được xây dựng/chưa hoàn thiện
- Thêm các giao diện mẫu để người dùng có thể dễ sử dụng
- Trực quan hóa các lựa chọn khi sử dụng
- Mở rộng thêm về các thể loại trang có thể ứng dụng hệ thống
- Tối ưu hóa phần bảo mật và tính an toàn dữ liệu
- Đa ngôn ngữ cho hệ thống

VI. TÀI LIỆU THAM KHẢO

- <https://topdev.vn/blog/session-la-gi-cookie-la-gi/>
- <https://stackoverflow.com/>
- <https://jinja.palletsprojects.com/en/3.1.x/templates/>
- <https://www.djangoproject.com/>
- <https://tailwindcss.com/>
- <https://www.w3schools.com/>
- <https://pypi.org/>
- <https://flask-socketio.readthedocs.io/en/latest/#>
- <https://flask.palletsprojects.com/en/2.3.x/>
- <https://docs.graphene-python.org/projects/django/en/latest/>

Em xin chân thành cảm ơn quý thầy cô và các bạn đã xem đồ án này.