

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



BARIA VUNGTAU
UNIVERSITY
CAP SAINT JACQUES

BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

**ĐỀ TÀI: PHÁT TRIỂN ỨNG DỤNG CHIA SẺ VIDEO
NGẮN - SHORTVIBE**

Trình độ đào tạo	: Đại học
Ngành	: Công nghệ thông tin
Chuyên ngành	: Công nghệ thông tin
Khóa học	: 2019-2023
Lớp	: DH19CT
Sinh viên thực hiện	: Lê Nguyễn Đăng Khoa
Mã số sinh viên	19034103
GVHD	: TS. Phan Ngọc Hoàng

VŨNG TÀU, NĂM 2023

LỜI CẢM ƠN

Trước tiên, em cảm thấy rất may mắn khi được học tập và làm việc trong môi trường giáo dục của trường. Em không thể nào quên được sự giúp đỡ của các thầy cô trong khoa và các bộ phận hành chính của nhà trường. Sự tận tâm và tâm huyết của các giảng viên không chỉ giúp em tiếp thu được kiến thức chuyên môn mà còn giúp em phát triển các kỹ năng cần thiết để có thể thực hiện thành được án tốt nghiệp.

Trong quá trình làm đồ án, em đã được giảng viên hướng dẫn đưa ra các phương pháp giải quyết vấn đề hiệu quả, cung cấp thông tin cần thiết và tạo điều kiện để em có thể phát triển ý tưởng và thực hiện các bước tiếp theo. Sự động viên và giúp đỡ của thầy đã giúp em vượt qua được những khó khăn trong quá trình thực hiện đồ án.

Bên cạnh đó, em cũng rất biết ơn Ban Giám hiệu trường Đại học Bà Rịa – Vũng Tàu và các bộ phận hành chính của nhà trường đã tạo điều kiện tốt nhất để các giảng viên và sinh viên có thể phát triển tối đa khả năng của mình.

Một lần nữa, em xin chân thành cảm ơn thầy Phan Ngọc Hoàng, là giảng viên trực tiếp hướng dẫn em, các thầy cô trong khoa Kỹ thuật – Công nghệ, Ban Giám hiệu trường Đại học Bà Rịa – Vũng Tàu. Em hi vọng rằng những giá trị mà em đã học được trong suốt thời gian học tập tại trường sẽ trở thành tài sản quý báu để em phát triển sự nghiệp và đóng góp cho xã hội trong tương lai.

Em xin chân thành cảm ơn!

LỜI CAM ĐOAN

Tôi xin cam đoan rằng những gì tôi đã thực hiện trong đồ án tốt nghiệp của mình hoàn toàn là công sức của bản thân mình, không sao chép từ bất kỳ nguồn nào khác. Tôi đã thu thập các thông tin và dữ liệu từ những nguồn hợp pháp và luôn đảm bảo các thông tin này được trích dẫn đầy đủ và chính xác. Tôi không có bất kỳ hành vi nào vi phạm quy định về đạo đức học thuật, bản quyền, hoặc pháp luật.

Nếu có bất kỳ sự không chính xác hay vi phạm quy định nào trong đồ án của tôi, tôi 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 01 tháng 05 năm 2023

Sinh viên thực hiện

Lê Nguyễn Đăng Khoa

NHẬN XÉT CỦA GIÁNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Vũng Tàu, ngày ... tháng ... năm 20

Người hướng dẫn

TS. Phan Ngọc Hoàng

NHẬN XÉT CỦA GIÁNG VIÊN PHẢN BIỆN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

MỤC LỤC

DANH MỤC SƠ ĐỒ.....	10
DANH MỤC HÌNH.....	11
LỜI NÓI ĐẦU.....	14
CHƯƠNG 1 TỔNG QUAN.....	15
1.1 Lý do chọn đề tài.....	15
1.2 Mục tiêu và chức năng của hệ thống.....	16
1.2.1 Mục tiêu.....	16
1.2.2 chức năng.....	16
1.3 Mô tả bài toán.....	16
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT.....	18
2.1 Cấu trúc của chương trình.....	18
2.2 Giới thiệu ngôn ngữ lập trình Python.....	18
2.1.1 Tổng quát.....	18
2.1.2 Lịch sử phát triển của Python.....	18
2.1.3 Ưu và nhược điểm của ngôn ngữ Python.....	22
2.1.4 Những ứng dụng của ngôn ngữ Python.....	23
2.2 Giới thiệu Framework Django.....	24
2.2.1 Tổng quát.....	24
2.2.1 Giới thiệu extension Channels.....	25
2.2.2 Giới thiệu Redis.....	26
2.2.3 Giới thiệu Jinja2.....	27
2.2.4 Giới thiệu về MySQL.....	27
2.2.5 Giới thiệu sơ quát các ngôn ngữ hỗ trợ giao diện.....	28
2.2.6 Một số thư viện khác.....	29
CHƯƠNG 3 PHÂN TÍCH HỆ THỐNG.....	30
3.1 Sơ đồ tổng quát.....	30
3.1.1 Chức năng của user.....	30
3.1.2 Chức năng của admin.....	39
3.2 Thiết kế hệ thống.....	41
3.2.1 Sơ đồ cơ sở dữ liệu.....	41
3.2.2 Chi tiết cơ sở dữ liệu:.....	42

CHƯƠNG 4 XÂY DỰNG ỨNG DỤNG	50
4.1 Giao diện giới thiệu	50
4.2 Giao diện đăng ký	50
4.3 Giao diện đăng nhập.....	51
4.4 Giao diện lấy lại mật khẩu.....	52
4.5 Giao diện email hướng dẫn lấy lại mật khẩu	53
4.6 Giao diện đặt lại mật khẩu.....	53
4.7 Giao diện trang chủ	54
4.8 Giao diện chi tiết bài đăng.....	55
4.9 Giao diện trang thông báo	57
4.10 Giao diện trang tìm kiếm bài đăng.....	58
4.11 Giao diện danh sách hội thoại riêng tư và hội thoại nhóm.....	59
4.12 Giao diện phòng hội thoại	61
4.13 Giao diện danh sách bài đăng của những người dùng đang theo dõi.....	63
4.14 Giao diện danh sách những bài đăng mà người dùng đã yêu thích.....	64
4.15 Giao diện danh sách bài đăng người dùng đã thêm vào xem sau	65
4.16 Giao diện danh sách bài đăng của bản thân người dùng.....	66
4.17 Giao diện quản lý danh sách người dùng hạn chế.....	66
4.18 Giao diện trang cá nhân.....	68
4.19 Giao diện cập nhật thông tin người dùng	72
4.20 Giao diện khi đăng xuất	74
4.21 Giao diện người dùng quản trị	75
CHƯƠNG 5 KẾT LUẬN	82
5.1 Kết quả đạt được	82
5.2 Hướng phát triển	82
TÀI LIỆU THAM KHẢO	84
PHỤ LỤC	85
Cấu trúc thư mục	85
Thư mục Root	85
Thư mục myproject.....	85
Thư mục blog.....	86
Thư mục chat	87
Thư mục friend	88

Thư mục notification	89
Thư mục users.....	89

DANH MỤC BẢNG

Bảng 3.1 auth_user	42
Bảng 3.2 auth_permission	42
Bảng 3.3 django_content_type	43
Bảng 3.4 auth_group_permission	43
Bảng 3.5 auth_user_user_permission	43
Bảng 3.6 auth_group.....	43
Bảng 3.7 user_blacklist.....	44
Bảng 3.8 auth_user_groups	44
Bảng 3.9 django_admin_log.....	44
Bảng 3.10 chat_chatgroup	45
Bảng 3.11 chat_roomgroup	45
Bảng 3.12 chat_chat	45
Bảng 3.13 chat_roommempermit	46
Bảng 3.14 chat_roomgroup_members.....	46
Bảng 3.15 blog_post	46
Bảng 3.16 blog_post_save	46
Bảng 3.17 friend_friendrequest	47
Bảng 3.18 users_profile.....	47
Bảng 3.19 user_profile_following.....	47
Bảng 3.20 friend_friendlist.....	48
Bảng 3.21 friend_friendlist_friends.....	48
Bảng 3.22 blog_comment.....	48
Bảng 3.23 blog_post_likes	48
Bảng 3.24 blog_comment_likes	49
Bảng 3.25 chat_room.....	49
Bảng 3.26 notification_notification.....	49

DANH MỤC SƠ ĐỒ

Sơ đồ 3.1 Sơ đồ use-case chức năng của user	30
Sơ đồ 3.2 Sơ đồ use-case tổng quát chức năng quản lý bài đăng.....	31
Sơ đồ 3.3 Sơ đồ use-case tổng quát chức năng quản lý quan hệ người dùng.....	32
Sơ đồ 3.4 Sơ đồ use-case tổng quát chức năng quản lý hội thoại	34
Sơ đồ 3.5 Sơ đồ use-case tổng quát chức năng quản lý người dùng	36
Sơ đồ 3.6 Sơ đồ use-case tổng quát chức năng quản lý tương tác bài đăng.....	38
Sơ đồ 3.7 Sơ đồ use-case tổng quát các chức năng của admin.....	39
Sơ đồ 3.8 Sơ đồ cơ sở dữ liệu.....	41

DANH MỤC HÌNH

Hình 2.1 Biểu tượng django	24
Hình 2.2 Biểu tượng của phần mở rộng Django-Channels	25
Hình 2.3 Biểu tượng của Redis.....	26
Hình 2.4 Biểu tượng của Jinja 2	27
Hình 2.5 Biểu tượng của MySQL.....	27
Hình 2.6 Biểu tượng của bộ 3 HTML, CSS và Javascript	28
Hình 4.7 Giao diện giới thiệu	50
Hình 4.8 Giao diện đăng ký.....	51
Hình 4.9 Giao diện đăng nhập	52
Hình 4.10 Giao diện gửi yêu cầu đặt lại mật khẩu	52
Hình 4.11 Giao diện gửi yêu cầu cấp mật khẩu thành công	52
Hình 4.12 Giao diện email hướng dẫn đặt lại mật khẩu	53
Hình 4.13 Giao diện đặt lại mật khẩu	54
Hình 4.14 Giao diện sau khi đặt lại mật khẩu thành công.....	54
Hình 4.15 Giao diện trang chủ.....	55
Hình 4.16 Giao diện trang chi tiết bài đăng với video tỉ lệ dọc.....	56
Hình 4.17 Giao diện trang chi tiết bài đăng với video tỉ lệ ngang.....	56
Hình 4.18 Giao diện cập nhật tiêu đề bài đăng.....	57
Hình 4.19 Giao diện xóa bài đăng	57
Hình 4.20 Giao diện icon thông báo.....	57
Hình 4.21 Giao diện danh sách thông báo	58
Hình 4.22 Giao diện thanh tìm kiếm yêu cầu phải có giá trị để có thể thao tác	59
Hình 4.23 Giao diện trang tìm kiếm hiển thị bài đăng theo nội dung tìm kiếm	59
Hình 4.24 Giao diện danh sách hội thoại riêng tư và hội thoại nhóm	60
Hình 4.25 Giao diện cửa sổ tạo nhóm hội thoại.	60
Hình 4.26 Giao diện phòng hội thoại riêng tư	61
Hình 4.27 Giao diện phòng hội thoại nhóm với thành viên là trưởng nhóm.....	62
Hình 4.28 Giao diện của sổ mời thành viên ra khỏi nhóm của thành viên trưởng nhóm	62

Hình 4.29	Giao diện phòng hội thoại với người dùng là thành viên	63
Hình 4.30	Giao diện thông báo người dùng đã tham gia nhóm hội thoại thành công .	63
Hình 4.31	Lối tắt dẫn đến danh sách bài đăng của người user đang theo dõi	63
Hình 4.32	Giao diện danh sách bài đăng của những người dùng mà user theo dõi	64
Hình 4.33	Lối tắt dẫn đến danh sách các bài đăng người dùng đã yêu thích	64
Hình 4.34	Giao diện danh sách những bài đăng mà người dùng đã yêu thích.	65
Hình 4.35	Lối tắt đến danh sách các bài đăng đã thêm vào xem sau.	65
Hình 4.36	Danh sách bài đăng người dùng đã thêm vào xem sau.....	66
Hình 4.37	Giao diện danh sách bài đăng của bản thân người dùng.....	66
Hình 4.38	Lối tắt đến trang quản lý người dùng hạn chế	66
Hình 4.39	Giao diện trang quản lý người dùng hạn chế.....	67
Hình 4.40	Xác nhận đưa người dùng vào danh sách hạn chế.....	67
Hình 4.41	Người dùng đã được đưa vào danh sách hạn chế	68
Hình 4.42	Lối tắt đến trang cá nhân của người dùng	68
Hình 4.43	Giao diện danh sách bạn bè của bản thân người dùng.	69
Hình 4.44	Giao diện danh sách những người đang theo dõi người dùng.....	69
Hình 4.45	Giao diện danh sách những người mà người dùng đang theo dõi.....	69
Hình 4.46	Giao diện trang cá nhân của người dùng khác.	70
Hình 4.47	Giao diện trang cá nhân của người dùng khác khi đã gửi lời mời và theo dõi.	70
Hình 4.48	giao diện bên phía người dùng nhận được lời mời kết bạn.	71
Hình 4.49	Giao diện trang cá nhân của người dùng khác khi cả 2 là bạn bè	71
Hình 4.50	lối tắt đến trang quản lý thông tin bản thân người dùng.....	72
Hình 4.51	Giao diện quản lý thông tin của người dùng.	72
Hình 4.52	Giao diện cửa sổ điều chỉnh kích thước ảnh đại diện.	73
Hình 4.53	Thông báo khi thông tin người dùng được cập nhật thành công.	73
Hình 4.54	Lối tắt đến chức năng đăng xuất.....	74
Hình 4.55	Giao diện xác nhận đã đăng xuất.....	74
Hình 4.56	Giao diện đăng nhập trang quản trị	75
Hình 4.57	Giao diện danh sách các đối tượng có thể quản lý từ trang quản trị	76
Hình 4.58	Giao diện danh sách các bài đăng được liệt kê trong trang quản trị.....	76

Hình 4.59	Giao diện các bài đăng đã được chọn để thực thi trong trang quản trị.....	77
Hình 4.60	giao diện thông tin của bài đăng từ trang quản trị.....	78
Hình 4.61	Thông báo cập nhật bài đăng thành công từ trang quản trị	78
Hình 4.62	Giao diện tạo mới bài đăng từ trang quản trị.....	79
Hình 4.63	Thông báo thêm bài đăng thành công từ trang quản trị.....	79
Hình 4.64	Kiểm tra bài đăng vừa thêm từ trang quản trị trên giao diện người dùng ...	80
Hình 4.65	Tiến hành xóa bài đăng từ trang quản trị.....	80
Hình 4.66	Thông báo xóa bài đăng thành công từ trang quản trị	81

LỜI NÓI ĐẦU

Trong suốt nhiều năm qua, mạng xã hội đã trở thành một phần quan trọng của cuộc sống đương đại. Từ khi xuất hiện vào những năm đầu của thế kỷ 21, mạng xã hội đã trải qua một quá trình phát triển đáng kinh ngạc, từ những trang web đơn giản cho phép người dùng kết nối với nhau đến các nền tảng phức tạp và đa dạng hơn cho phép chia sẻ nội dung, tạo ra các cộng đồng trực tuyến và thậm chí thực hiện các giao dịch thương mại.

Các mạng xã hội đầu tiên, như Friendster và MySpace, chỉ cho phép người dùng kết nối với những người quen của mình thông qua một mạng lưới đơn giản. Tuy nhiên, với sự ra đời của Facebook vào năm 2004, mạng xã hội đã trở thành một hiện tượng toàn cầu. Facebook cho phép người dùng kết nối với bất kỳ ai trên thế giới và chia sẻ các thông tin cá nhân, hình ảnh và video của họ. Đến năm 2022, Facebook đã trở thành một trong những nền tảng truyền thông xã hội lớn nhất trên thế giới với hơn 2,8 tỷ người dùng đăng ký.

Từ Facebook, các mạng xã hội khác như Instagram, Twitter và Snapchat cũng đã xuất hiện. Tuy nhiên, xu hướng mạng xã hội mới nhất và nổi bật nhất trong những năm gần đây là video ngắn. Với sự trỗi dậy của TikTok vào năm 2016, các nền tảng video ngắn đã trở nên rất phổ biến. TikTok cho phép người dùng tạo ra các video ngắn và phát trực tiếp cho hàng triệu người xem trên toàn cầu. Điều này đã mở ra một lĩnh vực mới cho các nền tảng video ngắn khác như Instagram Reels và YouTube Shorts.

Điều đó đã dẫn đến sự ra đời của đề tài “Phát triển ứng dụng chia sẻ video ngắn - ShortVibe”. Để không bỏ lỡ xu hướng ShortVibe ra đời là một nền tảng chia sẻ video ngắn, cho phép người dùng đăng tải những video ngắn độc đáo và sáng tạo để chia sẻ với người dùng khác. ShortVibe đã được thiết kế để cung cấp trải nghiệm giải trí độc đáo và đầy thú vị.

CHƯƠNG 1 TỔNG QUAN

1.1 Lý do chọn đề tài

Hiện nay, việc phát triển ứng dụng mạng xã hội chia sẻ video ngắn đang trở thành xu hướng đáng chú ý trong lĩnh vực công nghệ thông tin. Với sự phát triển không ngừng của kết nối internet và smartphone, người dùng có thể dễ dàng chia sẻ những đoạn video ngắn, ghi lại những khoảnh khắc cuộc sống một cách nhanh chóng và tiện lợi.

Một trong những lý do chính đằng sau sự phát triển nhanh chóng của ứng dụng mạng xã hội chia sẻ video ngắn là sự thay đổi trong cách thức mà con người tiêu thụ thông tin. Trong thời đại của tốc độ và tính ngắn gọn, người dùng càng có xu hướng tìm kiếm những nội dung ngắn gọn, dễ tiêu thụ và mang tính giải trí cao. Việc chia sẻ video ngắn trên mạng xã hội cung cấp cho người dùng một trải nghiệm trực quan, hấp dẫn và thu hút sự chú ý của người xem một cách nhanh chóng.

Bắt kịp xu hướng, các ông lớn như Facebook, Youtube, Bytedance,... cũng đã phát triển những công cụ đem lại cho người dùng những nội dung với thời lượng ngắn và mang tính giải trí cao. Không chỉ dừng lại đó, những ứng dụng mạng xã hội chia sẻ video ngắn cũng được xem như một kênh quảng cáo hiệu quả. Do tính chất giải trí, trực quan và dễ tiếp cận của nội dung video ngắn, các doanh nghiệp và nhà quảng cáo đang chuyển hướng sử dụng ứng dụng này để tiếp cận khách hàng một cách hiệu quả hơn. Việc quảng cáo trên ứng dụng mạng xã hội chia sẻ video ngắn sẽ giúp doanh nghiệp tiếp cận được một lượng lớn khách hàng tiềm năng một cách nhanh chóng và tiết kiệm chi phí.

Từ những xu hướng và lợi ích trên, ứng dụng chia sẻ video ngắn “ShortVibe” cũng sẽ là một giải pháp để đáp ứng nhu cầu tiêu thụ nội dung ngắn và sự phát triển của mạng xã hội hiện nay.

1.2 Mục tiêu và chức năng của hệ thống

1.2.1 Mục tiêu

Quản lý nội dung chia sẻ, cung cấp không gian trực tuyến cho người dùng thỏa sức thể hiện bản thân và kết nối với cộng đồng, các mục tiêu bao gồm:

- Trao đổi thông tin giữa người dùng.
- Quản lý nội dung chia sẻ.
- Xem thông tin cá nhân.
- Quản lý nâng cao cho người dùng quản trị.
- Tiếp cận nội dung được chia sẻ.
- ...

Trong thời đại công nghệ, con người đã trải qua những thay đổi đáng kể về cách tiêu thụ giải trí và nội dung, cũng như cách tương tác và chia sẻ thông tin với nhau. Công nghệ đã tác động mạnh mẽ đến lối sống và thói quen của con người, khiến cho việc tiếp cận, tương tác và chia sẻ thông tin trở nên nhanh chóng và dễ dàng hơn bao giờ hết. Các nền tảng mạng xã hội và ứng dụng giải trí trực tuyến đã trở thành điểm đến quen thuộc cho nhu cầu giải trí và tương tác của con người.

1.2.2 chức năng

- ✓ Quản lý bài đăng
- ✓ Quản lý quan hệ người dùng
- ✓ Quản lý thông báo
- ✓ Quản lý hội thoại
- ✓ Quản lý bình luận
- ✓ Quản lý tương tác bài đăng
- ✓ Quản lý người dùng

1.3 Mô tả bài toán

- Mục đích của hệ thống là cung cấp nền tảng trực tuyến cho người dùng giao lưu và chia sẻ thông tin.
 - Quản lý bài đăng
 - Quản lý bài đăng
 - Quản lý quan hệ người dùng

- Quản lý bạn bè: Gửi lời mời, theo dõi, xác nhận, hủy kết bạn,...
- Quản lý danh sách hạn chế: chặn người dùng, bỏ chặn người dùng.
- Quản lý thông báo
 - Quản lý thông báo
- Quản lý hội thoại
 - Quản lý hội thoại nhóm
 - Quản lý hội thoại riêng tư
- Quản lý tương tác bài đăng
 - Quản lý yêu thích
 - Quản lý yêu thích bình luận
 - Quản lý bình luận
 - Quản lý danh sách xem sau
- Quản lý người dùng
 - Quản lý thông tin cá nhân
 - Quản lý tài khoản

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

2.1 Cấu trúc của chương trình

Ứng dụng “ShortVibe” được phát triển trên nền tảng web, và được sử dụng các công nghệ:

- Ngôn ngữ lập trình chính: Python
- Các ngôn ngữ lập trình hỗ trợ: HTML, JavaScript, CSS.
- Framework: Django
- Cơ sở dữ liệu: MySQL
- Phần mềm hỗ trợ lập trình: Visual Studio Code (VS Code).

2.2 Giới thiệu ngôn ngữ lập trình Python

2.1.1 Tổng quát

Python là một ngôn ngữ lập trình đa năng và phổ biến, được phát triển bởi Guido van Rossum từ năm 1991. Với cú pháp dễ đọc và cấu trúc rõ ràng, Python được sử dụng trong nhiều lĩnh vực khác nhau, từ phát triển ứng dụng web, khoa học dữ liệu và trí tuệ nhân tạo đến game và ứng dụng máy tính. Python có một cộng đồng phát triển đông đảo và phong phú, với hàng nghìn các thư viện và công cụ hỗ trợ, giúp cho việc phát triển và thực hiện các tác vụ lập trình trở nên dễ dàng và hiệu quả. Với những ưu điểm đó, Python đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất hiện nay.

2.1.2 Lịch sử phát triển của Python

Có thể chia sự phát triển của ngôn ngữ Python thành các giai đoạn sau:

- Python 0.9.x: là phiên bản đầu tiên của Python, được tạo ra vào tháng 2 năm 1991 bởi Guido van Rossum tại Centrum Wiskunde & Informatica (CWI) ở Amsterdam, Hà Lan. Phiên bản này chỉ chạy trên các hệ thống Unix và được phát hành miễn phí.
 - Python 0.9.0 chỉ có khoảng 3000 dòng mã và chỉ hỗ trợ một số ít tính năng, bao gồm số nguyên, số thực, chuỗi, hàm, điều kiện và vòng lặp.
 - Tên "Python" được đặt theo tên của chương trình truyền hình "Monty Python's Flying Circus", một chương trình hài kịch ưa thích của Guido van Rossum.

- Python 0.9.0 được phát hành dưới giấy phép GPL (General Public License), cho phép bất kỳ ai sử dụng, phân phối và thay đổi phiên bản này một cách tự do.
 - Python 0.9.0 không có bất kỳ công cụ định dạng hoặc trình soạn thảo mã nguồn nào, vì vậy người dùng phải sử dụng các công cụ bên ngoài để phát triển và chạy mã.
 - Python 0.9.0 chỉ có thể chạy trên các hệ thống Unix và không hỗ trợ các nền tảng khác như Windows hoặc Mac.
 - Mặc dù Python 0.9.0 đã được thay thế bởi các phiên bản mới hơn, nhưng nó vẫn là một phần quan trọng trong lịch sử phát triển của Python. Sự ra đời của phiên bản này đã mở ra một hướng đi mới trong lĩnh vực phát triển phần mềm, giúp cho Python trở thành một trong những ngôn ngữ lập trình phổ biến nhất hiện nay.
- Python 1.x: Được phát hành vào năm 1994 và được phát triển tiếp từ phiên bản đầu tiên. Dưới đây là một số thông tin chi tiết về phiên bản này:
- Python 1.x đã thêm nhiều tính năng mới như hỗ trợ đối tượng, ngoại lệ, module và package.
 - Phiên bản đầu tiên của Python 1.x là 1.0, được phát hành vào tháng 1 năm 1994. Phiên bản này hỗ trợ nền tảng Unix, Macintosh và Windows.
 - Python 1.4, được phát hành vào năm 1996, đã đưa ra nhiều tính năng mới, bao gồm hỗ trợ Unicode, cải tiến bộ nhớ đệm và thêm một số loại đối tượng mới.
 - Phiên bản cuối cùng của Python 1.x là 1.6, được phát hành vào tháng 9 năm 2000. Phiên bản này cung cấp nhiều cải tiến và tính năng mới, bao gồm module 'collections', khả năng lập lịch (scheduling) và khả năng truyền giá trị mặc định cho hàm.
 - Python 1.x không tương thích với Python 2.x về cú pháp và cách thực hiện. Tuy nhiên, nó vẫn được sử dụng rộng rãi trong các dự án lớn và trong các trường đại học.

- Python 2.x: Được phát hành vào năm 2000 và tiếp tục được phát triển cho đến khi nó chính thức ngừng được hỗ trợ vào năm 2020.
 - Python 2.x đã thêm nhiều tính năng mới như hỗ trợ các gói (packages) lớn hơn, hỗ trợ cho các loại dữ liệu mới, cải thiện hiệu suất, và cải tiến về khả năng xử lý lỗi.
 - Phiên bản đầu tiên của Python 2.x là 2.0, được phát hành vào tháng 10 năm 2000. Phiên bản này có tính năng đáng chú ý là hỗ trợ cho hệ thống Unicode, cho phép người dùng xử lý các ký tự và bộ ký tự không chỉ giới hạn trong các bộ mã ASCII.
 - Python 2.2, được phát hành vào năm 2001, đã đưa ra tính năng new-style classes, cải tiến về xử lý lỗi, và hỗ trợ cho các phương thức có thể gọi được (callable methods).
 - Python 2.5, được phát hành vào năm 2006, đã đưa ra nhiều cải tiến về cú pháp, khả năng xử lý lỗi và tăng tốc độ.
 - Phiên bản cuối cùng của Python 2.x là Python 2.7, được phát hành vào năm 2010. Phiên bản này có các tính năng mới như hỗ trợ cho các biểu thức generator, cải tiến hiệu suất và khả năng xử lý lỗi.
 - Một trong những sự kiện đáng chú ý của Python 2.x là khi Guido van Rossum, người sáng lập Python, đã quyết định giới hạn phát triển của phiên bản này để tập trung vào việc phát triển Python 3.x. Tuy nhiên, Python 2.x vẫn được sử dụng rộng rãi cho đến khi chính thức ngừng được hỗ trợ vào năm 2020.
 - Python 2.x có cú pháp rất giống với Python 1.x, vì vậy nó được xem là một phiên bản cải tiến và nâng cấp của phiên bản trước đó. Nó đã giúp xây dựng cộng đồng người dùng Python phát triển mạnh mẽ hơn, đặc biệt là trong các dự án lớn và doanh nghiệp.
 - Tuy nhiên, Python 2.x cũng có những hạn chế như không hỗ trợ đầy đủ các tính năng của hệ thống

- Python 3.x được phát hành vào năm 2008, là phiên bản lớn nhất và được coi là bản phát triển tiếp theo của Python. Dưới đây là một số thông tin chi tiết về phiên bản này:
- Python 3.x có nhiều cải tiến đáng kể so với phiên bản 2.x. Điều này bao gồm cải tiến về cú pháp, tính năng, xử lý lỗi và hiệu suất. Python 3.x cũng cải thiện hỗ trợ Unicode và thêm nhiều khả năng xử lý chuỗi.
 - Python 3.0, phiên bản đầu tiên của Python 3.x, được phát hành vào năm 2008. Phiên bản này có sự thay đổi cú pháp đáng kể, bao gồm việc thay đổi cú pháp print thành một hàm, sử dụng dấu hai chấm để thay thế dấu hai chấm đơn và sử dụng dấu gạch chân để thay thế dấu gạch ngang.
 - Python 3.1, được phát hành vào năm 2009, đã đưa ra nhiều cải tiến về tính năng, xử lý lỗi và khả năng tương thích ngược. Phiên bản này cũng có thêm một số tính năng mới như hỗ trợ cho định dạng chuỗi mới và cải tiến về khả năng nhập dữ liệu.
 - Python 3.3, được phát hành vào năm 2012, cải tiến hiệu suất và tính năng hỗ trợ cho phép sử dụng dấu chấm thay cho dấu gạch chân.
 - Phiên bản Python 3.4, được phát hành vào năm 2014, cải thiện hiệu suất, thêm tính năng hỗ trợ cho phép sử dụng asyncio và cải tiến về khả năng xử lý chuỗi.
 - Python 3.5, được phát hành vào năm 2015, đã thêm một số tính năng mới như khả năng async/await, tăng cường hỗ trợ cho định dạng chuỗi mới, và cải tiến hiệu suất.
 - Một trong những cải tiến quan trọng của Python 3.x là việc hỗ trợ đầy đủ Unicode, giúp cho việc xử lý các ký tự và bộ ký tự không phụ thuộc vào bộ mã ASCII. Điều này rất hữu ích trong việc phát triển các ứng dụng toàn cầu.
 - Tuy nhiên, do những thay đổi cú pháp và tính năng, Python 3.x không tương thích với các ứng dụng và thư viện được viết cho Python 2.x một cách trực tiếp. Điều này đòi hỏi các lập trình viên phải thay đổi mã của họ để tương thích với Python 3.x. Để hỗ trợ quá trình chuyển đổi này, Python cung cấp

một công cụ có tên là 2to3, giúp tự động chuyển đổi các mã Python 2.x sang Python 3.x.

- Vào tháng 12 năm 2020, phiên bản Python 3.9 được phát hành. Phiên bản này có nhiều tính năng mới như hỗ trợ cho PEG-based parser, cải tiến về performance, cải tiến về tính năng debugging và khả năng tương thích ngược với phiên bản cũ.
- Hiện nay, Python 3.x được coi là phiên bản chính thức và được khuyến khích sử dụng cho các dự án phát triển mới. Nó có sẵn trên nhiều hệ điều hành và được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm khoa học dữ liệu, trí tuệ nhân tạo, phát triển web và hệ thống.
- Một số lý do nên sử dụng Python 3.x bao gồm tính năng mạnh mẽ, cú pháp đơn giản và dễ đọc, khả năng tương thích ngược tốt hơn so với phiên bản cũ, khả năng xử lý chuỗi Unicode tốt hơn, và sự phát triển đang tiếp tục với cộng đồng lớn và nhiều hỗ trợ từ nhà phát triển Python.

2.1.3 Ưu và nhược điểm của ngôn ngữ Python

➤ Ưu điểm của Python:

- Cú pháp đơn giản, dễ đọc và dễ hiểu: Python sử dụng cú pháp rất đơn giản, giúp người mới bắt đầu có thể nhanh chóng tiếp cận và viết code.
- Tính linh hoạt và đa năng: Python có thể sử dụng cho nhiều mục đích, từ phát triển web đến xử lý dữ liệu và trí tuệ nhân tạo.
- Hỗ trợ đa nền tảng: Python có thể chạy trên nhiều hệ điều hành khác nhau, bao gồm Windows, macOS và Linux.
- Cộng đồng lớn và hỗ trợ mạnh mẽ: Python có một cộng đồng lớn và nhiều tài liệu học tập và hỗ trợ từ các nhà phát triển trên toàn thế giới.
- Có nhiều thư viện hỗ trợ: Python có nhiều thư viện hỗ trợ mạnh mẽ, bao gồm NumPy, Pandas, SciPy và TensorFlow, giúp xử lý dữ liệu và phân tích dữ liệu dễ dàng hơn.

➤ Nhược điểm của Python:

- Tốc độ chậm hơn so với một số ngôn ngữ khác: Python có tốc độ thực thi chậm hơn một số ngôn ngữ khác như C++ và Java.
- Cấu trúc dữ liệu không linh hoạt: Python có cấu trúc dữ liệu cơ bản nhưng không phù hợp với một số bài toán đặc thù.
- Khả năng tương thích ngược có hạn: Sự thay đổi trong các phiên bản của Python có thể làm cho các chương trình không tương thích với phiên bản mới hơn.

2.1.4 Những ứng dụng của ngôn ngữ Python

- Phát triển web: Python được sử dụng để phát triển các trang web và ứng dụng web, thông qua các framework như Django và Flask.
- Khoa học dữ liệu: Python là ngôn ngữ phổ biến trong lĩnh vực khoa học dữ liệu và machine learning, được sử dụng để xử lý và phân tích dữ liệu, xây dựng các mô hình dự đoán và học máy.
- Automating tasks: Python là một ngôn ngữ lập trình thông dịch, được sử dụng rộng rãi trong việc tự động hóa các tác vụ thông qua các thư viện như Selenium và BeautifulSoup.
- Game development: Python được sử dụng để phát triển trò chơi, thông qua các framework như Pygame.
- Robotics: Python được sử dụng để lập trình các robot thông qua các thư viện như ROS (Robot Operating System).
- Internet of Things (IoT): Python là một trong những ngôn ngữ được sử dụng phổ biến trong lĩnh vực IoT, để kết nối và điều khiển các thiết bị thông qua các thư viện như MQTT và ZMQ.
- Desktop applications: Python cũng được sử dụng để phát triển các ứng dụng desktop, thông qua các framework như PyQt và wxPython.

Tổng quát lại, Python được sử dụng rộng rãi trong các lĩnh vực như phát triển web, khoa học dữ liệu, tự động hóa các tác vụ, phát triển trò chơi, lập trình robot, IoT và phát triển ứng dụng desktop. Python cũng được đánh giá cao về tính dễ học, độ linh

hoạt, sức mạnh và hiệu suất của nó, và có sự hỗ trợ đông đảo từ cộng đồng lập trình viên trên toàn thế giới.

2.2 Giới thiệu Framework Django

2.2.1 Tổng quát



Hình 2.1 Biểu tượng django

Django là một framework phát triển ứng dụng web theo mô hình Model-View-Controller (MVC), được viết bằng ngôn ngữ lập trình Python. Django cung cấp nhiều tính năng và công cụ để phát triển các ứng dụng web chuyên nghiệp, bao gồm hệ thống xử lý URL, ORM (Object-Relational Mapping) để tương tác với cơ sở dữ liệu, hệ thống giao diện quản trị, hệ thống bảo mật và quản lý session. Django cũng có cộng đồng lớn và sự hỗ trợ mạnh mẽ từ những nhà phát triển trên toàn thế giới. Với những tính năng và công cụ đó, Django là lựa chọn hàng đầu cho việc phát triển các ứng dụng web Python chuyên nghiệp và ổn định.

- Django được phát triển bởi Django Software Foundation, một tổ chức phi lợi nhuận.
- Django đã được sử dụng bởi nhiều công ty và tổ chức nổi tiếng như Instagram, Mozilla, Pinterest và NASA.
- Django cung cấp cách tiếp cận tổng thể để phát triển ứng dụng web, giúp giảm thiểu việc lặp lại và nâng cao năng suất.
- Django có nhiều tính năng tích hợp sẵn như quản lý tài khoản người dùng, quản lý session, phân quyền truy cập, tạo giao diện quản trị, ORM tương tác với cơ sở dữ liệu và nhiều tính năng khác.

- Django cung cấp hệ thống template đơn giản và dễ sử dụng để tạo giao diện người dùng.
- Django đã phát triển từ phiên bản đầu tiên vào năm 2005 và hiện đang có phiên bản mới nhất là Django 4.2.1.
- Django được cộng đồng Python đánh giá cao về tính ổn định, bảo mật và sự hỗ trợ mạnh mẽ.

2.2.1 Giới thiệu extension Channels



Hình 2.2 Biểu tượng của phần mở rộng Django-Channels

Là một extension (phần mở rộng) của Django cho phép phát triển ứng dụng web realtime (thời gian thực) và các ứng dụng có tính năng xử lý dữ liệu theo thời gian thực. Channels cung cấp cơ chế cho phép Django xử lý các kết nối WebSocket và các kết nối HTTP ở chế độ streaming, giúp cho việc phát triển ứng dụng realtime trở nên dễ dàng hơn.

Channels cũng cho phép tích hợp với các server message queue như Redis hoặc RabbitMQ để xử lý các công việc bất đồng bộ (asynchronous tasks) hoặc các tác vụ xử lý dữ liệu khác. Channels cho phép phát triển các ứng dụng web có khả năng xử lý các tác vụ phức tạp theo thời gian thực, như chat trực tuyến, giao tiếp giữa các ứng dụng hoặc các ứng dụng đa người chơi.

Channels được tích hợp sẵn vào Django từ phiên bản 2.0 trở đi, và đã trở thành một công cụ quan trọng cho việc phát triển các ứng dụng web realtime trên nền tảng Django.

2.2.2 Giới thiệu Redis



Hình 2.3 Biểu tượng của Redis

Redis là một hệ thống cơ sở dữ liệu NoSQL (không quan hệ) được sử dụng rộng rãi cho các ứng dụng có tính năng realtime, bất đồng bộ và khả năng mở rộng. Redis cung cấp các tính năng đa dạng như bộ nhớ đệm, key-value store và các thao tác phức tạp như Sorted Set hay List.

Khi kết hợp với Channels trong Django, Redis có thể được sử dụng làm message broker để xử lý các công việc bất đồng bộ, như gửi email, tạo báo cáo, xử lý tác vụ nặng... và cho phép các worker (những tiến trình hoạt động độc lập) xử lý các công việc này.

Channels cung cấp các giao diện trừu tượng hóa để sử dụng Redis trong ứng dụng Django một cách dễ dàng. Khi sử dụng Redis với Channels, các message được gửi qua Redis sẽ được chuyển đến các channel tương ứng, và các worker có thể đăng ký để lắng nghe các channel này và xử lý các message được gửi đến. Các message broker như Redis cho phép xử lý các công việc bất đồng bộ một cách hiệu quả hơn so với các phương pháp truyền thống.

Vì vậy, kết hợp Channels với Redis sẽ giúp cho việc phát triển các ứng dụng Django với tính năng realtime và bất đồng bộ trở nên dễ dàng hơn và hiệu quả hơn.

2.2.3 Giới thiệu Jinja2



Hình 2.4 Biểu tượng của Jinja 2

Jinja2 là một thư viện mẫu (template) mã nguồn mở được viết bằng Python, cho phép các lập trình viên tạo ra các mẫu HTML, XML, JSON, CSS và văn bản khác để hiển thị dữ liệu động. Jinja2 có cú pháp đơn giản và dễ đọc, và có thể được tích hợp dễ dàng với các framework web như Flask và Django.

Jinja2 hỗ trợ các tính năng như biểu thức điều kiện, vòng lặp, kế thừa, các khối dữ liệu, các bộ lọc (filter),... Nó cũng hỗ trợ các tính năng nâng cao như viết mã Python trong các mẫu, tùy chỉnh hành vi cú pháp và khả năng mở rộng để sử dụng các chức năng tùy chỉnh.

Jinja2 được sử dụng rộng rãi trong các dự án web để tạo ra các trang web động và các email mẫu, và là một trong những thư viện mẫu phổ biến nhất trong cộng đồng Python.

2.2.4 Giới thiệu về MySQL



Hình 2.5 Biểu tượng của MySQL

MySQL là hệ thống quản trị cơ sở dữ liệu mã nguồn mở (open-source) phổ biến nhất trên thế giới. Nó được thiết kế để lưu trữ và quản lý dữ liệu, và cung cấp một cách

thức dễ dàng để truy vấn, sắp xếp, tìm kiếm, thêm, sửa và xóa các bản ghi trong cơ sở dữ liệu. MySQL được sử dụng rộng rãi trong các ứng dụng web, ứng dụng doanh nghiệp và các ứng dụng máy chủ khác. Nó được hỗ trợ trên nhiều hệ điều hành, bao gồm Windows, macOS và các bản phân phối Linux khác nhau.

2.2.5 Giới thiệu sơ quát các ngôn ngữ hỗ trợ giao diện



Hình 2.6 Biểu tượng của bộ 3 HTML, CSS và Javascript

HTML, CSS và JavaScript là ba công nghệ chính trong việc tạo ra các trang web động và tương tác trên web.

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu cấu trúc cho các trang web. Nó được sử dụng để tạo ra các phần tử và định dạng văn bản, hình ảnh, đường link và các thành phần khác của trang web.

CSS (Cascading Style Sheets) là ngôn ngữ định dạng để định dạng và thiết kế cho các phần tử HTML. CSS cho phép bạn định dạng các phần tử HTML như kích thước, màu sắc, vị trí và hình dạng.

JavaScript là ngôn ngữ lập trình để tạo ra các tính năng tương tác trên web. Nó được sử dụng để thêm các hiệu ứng động, xử lý sự kiện và kiểm soát các thành phần của trang web. JavaScript cũng được sử dụng để giao tiếp với các dịch vụ web bên ngoài và các ứng dụng khác.

Với sự kết hợp của HTML, CSS và JavaScript, người phát triển web có thể tạo ra các trang web đẹp mắt và tương tác, đáp ứng các yêu cầu của người dùng và cung cấp trải nghiệm web tốt hơn.

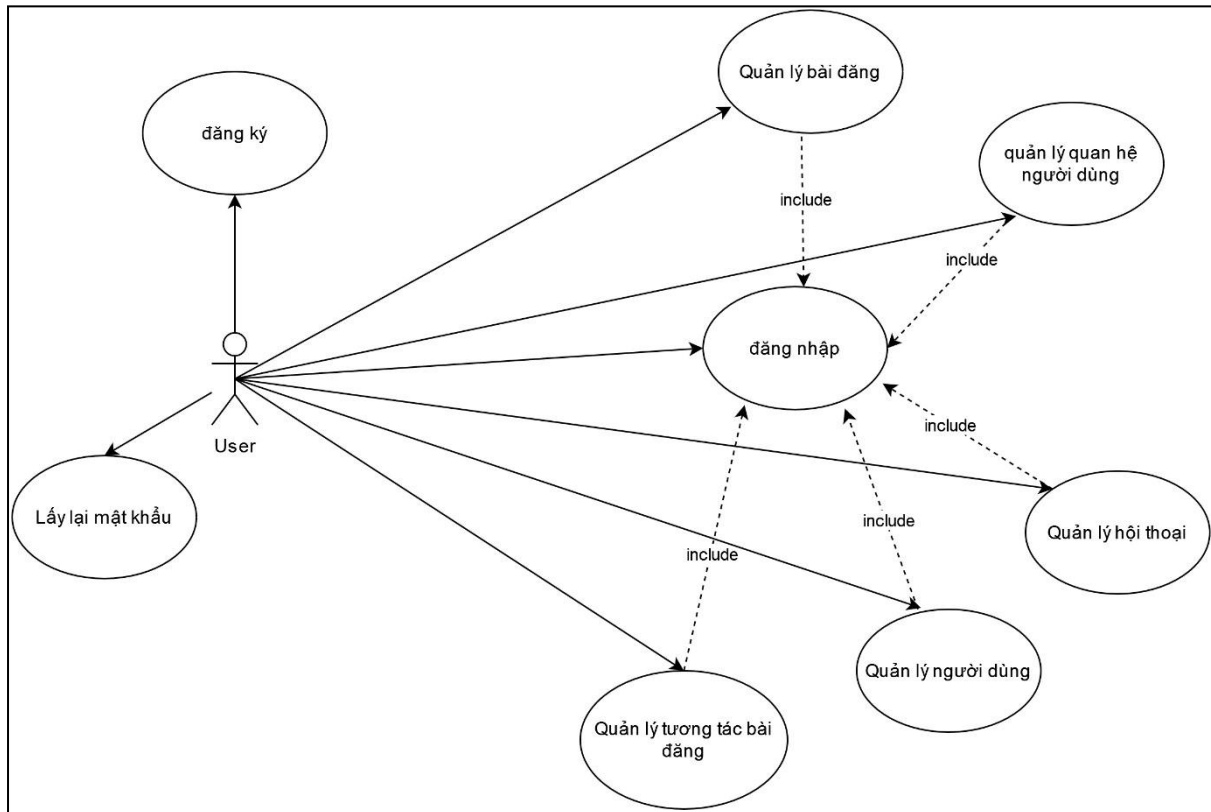
2.2.6 Một số thư viện khác

- Django-ckeditor: Cung cấp trình soạn thảo văn bản mở rộng cho Django.
- Pillow: Là thư viện hỗ trợ xử lý hình ảnh trong Python.
- PyJWT: Là thư viện quản lý token (xác thực người dùng).

CHƯƠNG 3 PHÂN TÍCH HỆ THỐNG

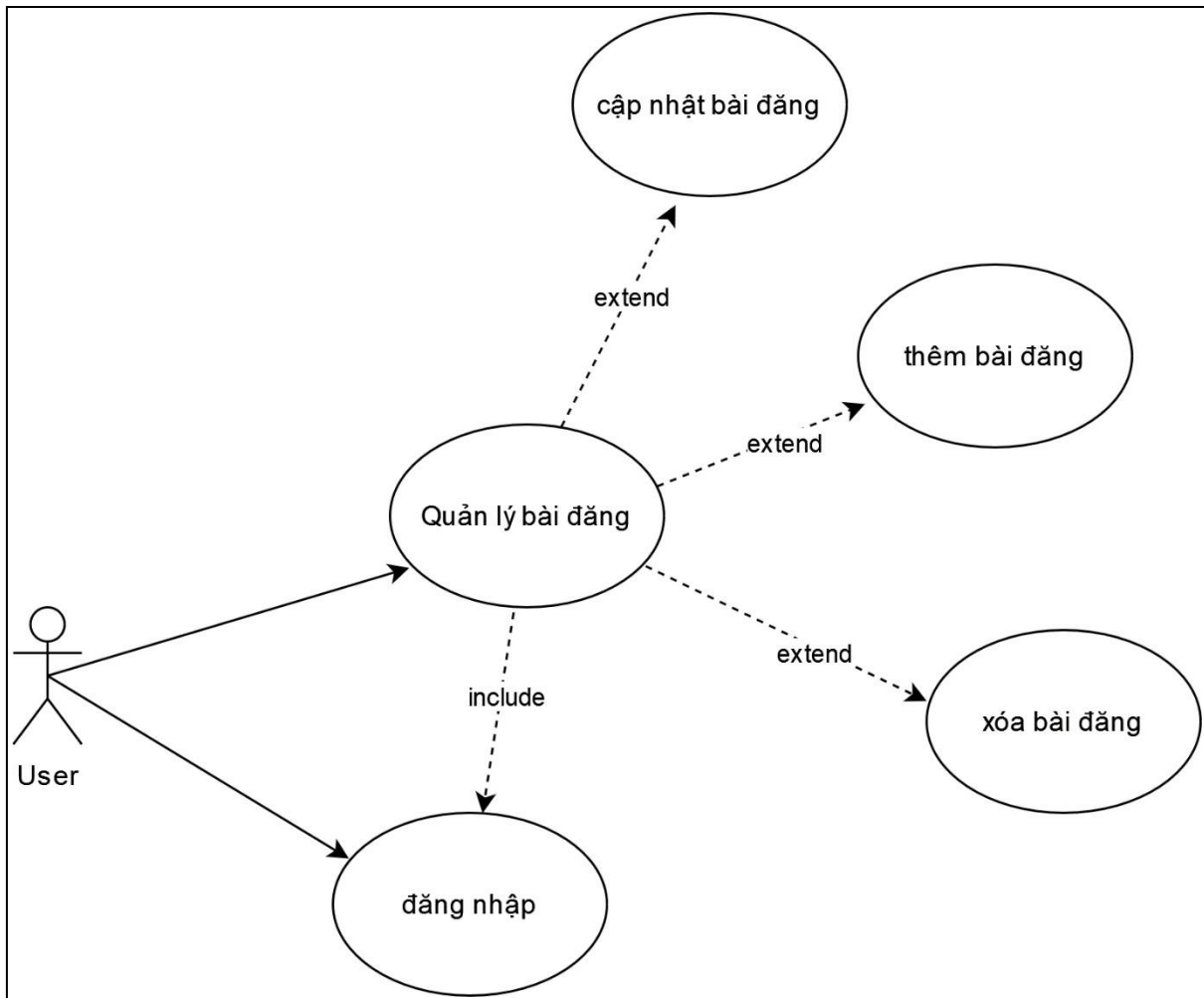
3.1 Sơ đồ tổng quát

3.1.1 Chức năng của user



Sơ đồ 3.1 Sơ đồ use-case chức năng của user

- Chức năng đăng ký:
 - Cho phép người dùng đăng ký tài khoản.
- Chức năng đăng nhập:
 - Cho phép người dùng đăng nhập.
- Chức năng lấy lại mật khẩu:
 - Cho phép người dùng lấy lại mật khẩu thông qua email đã đăng ký.
- Chức năng quản lý bài đăng:
 - Bao gồm tạo, cập nhật và xóa bài đăng

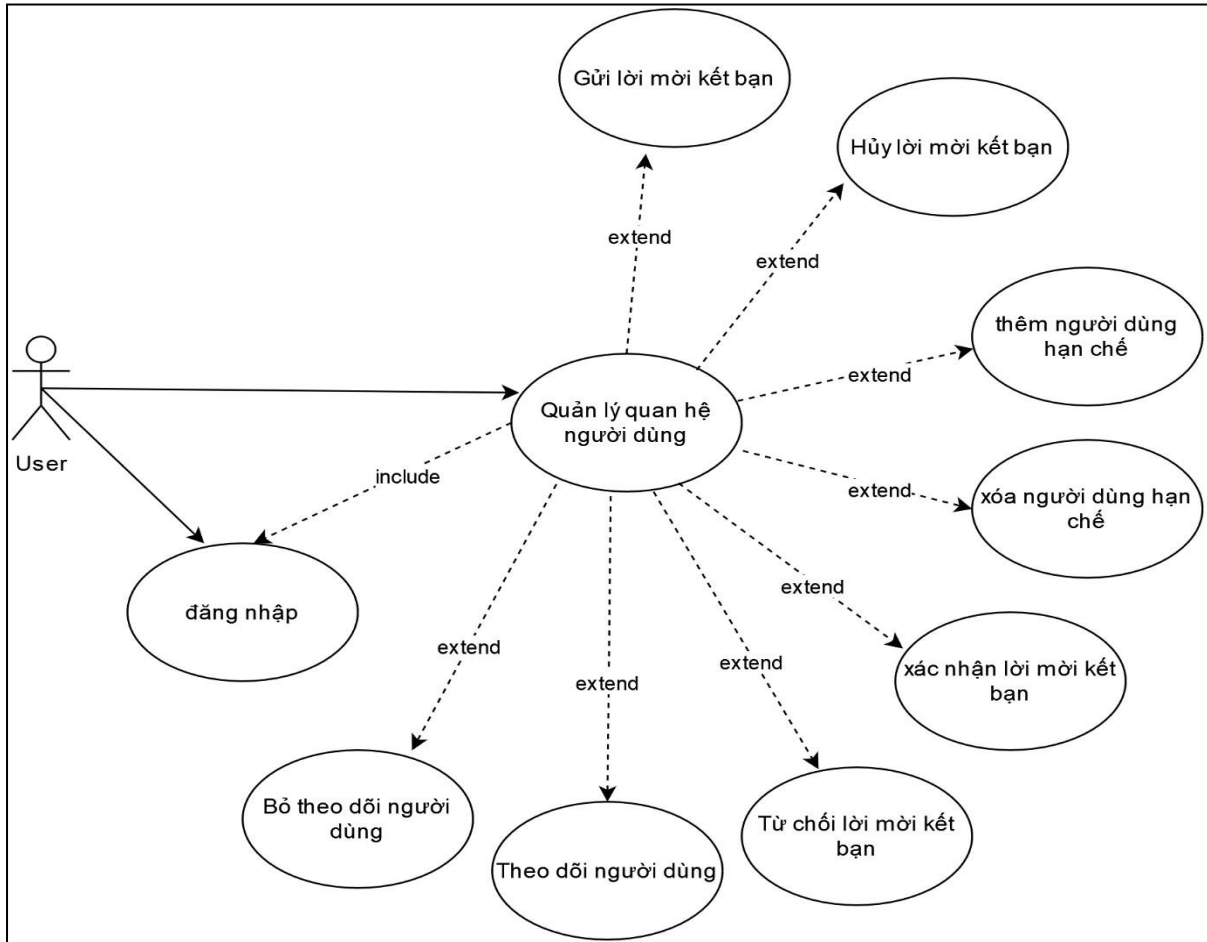


Sơ đồ 3.2 Sơ đồ use-case tổng quát chức năng quản lý bài đăng

- Yêu cầu người dùng phải đăng nhập từ trước.
- Tạo bài đăng: Cho phép người dùng tạo bài đăng với các thuộc tính là tiêu đề và file video, file video được ràng buộc với kích thước dưới 10MB và phải là định dạng video.
- Cập nhật bài đăng: Cho phép người dùng thay đổi tiêu đề của bài đăng của bản thân.
- Xóa bài đăng: Cho phép người dùng xóa bài đăng của bản thân.

➤ Chức năng quản lý quan hệ người dùng:

Bao gồm gửi lời mời, hủy lời mời, thêm người dùng hạn chế, xóa người dùng hạn chế, xác nhận lời mời kết bạn, từ chối lời mời kết bạn, theo dõi người dùng và bỏ theo dõi người dùng.



Sơ đồ 3.3 Sơ đồ use-case tổng quát chức năng quản lý quan hệ người dùng

- Yêu cầu người dùng phải đăng nhập từ trước.
- Gửi lời mời kết bạn: Cho phép người dùng gửi lời mời kết bạn đến người dùng khác.
- Hủy lời mời kết bạn: Cho phép người dùng hủy lời mời kết bạn đã gửi.
- Thêm người dùng hạn chế: Cho phép người dùng thêm người dùng khác vào danh sách hạn chế, để hạn chế người dùng đó xem những bài đăng của bản thân, người dùng bị chặn cũng sẽ không thể xem được trang cá nhân hay chat với người đã chặn họ, việc đưa vào danh sách hạn chế này

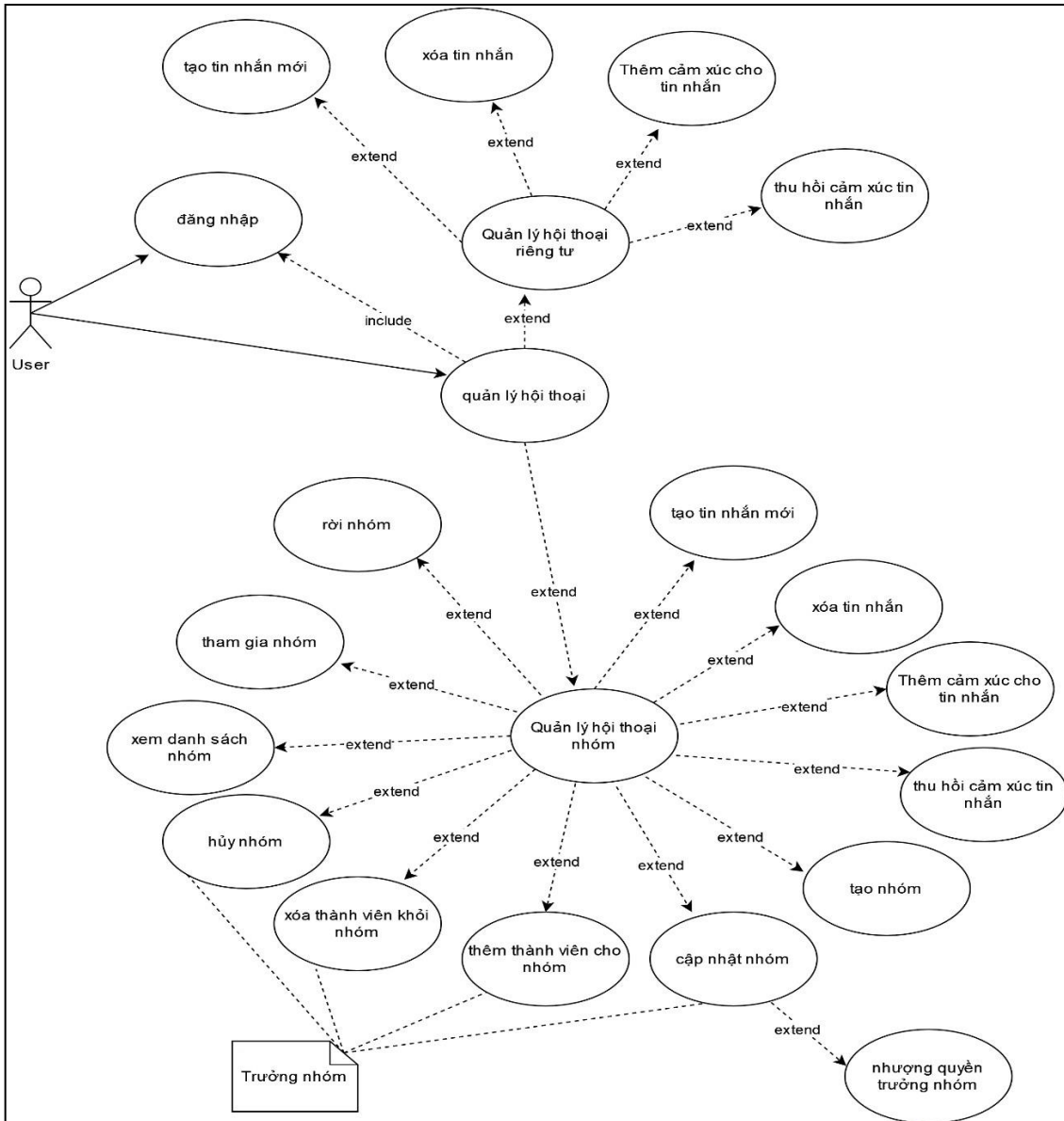
sẽ khiến cho người chặn và người bị chặn không thể nào tìm thấy bất kỳ dấu vết nào về nhau.

- Xóa người dùng hạn chế: Cho phép người dùng xóa người dùng khác khỏi danh sách hạn chế.
- Xác nhận lời mời kết bạn: Cho phép người dùng xác nhận lời mời kết bạn của người dùng khác để trở thành bạn bè, để có thể chat được với nhau.
- Từ chối lời mời kết bạn: Cho phép người dùng từ chối lời mời kết bạn của người dùng khác
- Theo dõi người dùng: Cho phép người dùng theo dõi người dùng khác và xem được những bài đăng của người đó.
- Bỏ theo dõi người dùng: Cho phép người dùng bỏ theo dõi người dùng khác.

➤ Chức năng quản lý hội thoại

Bao gồm quản lý hội thoại riêng tư và quản lý hội thoại nhóm.

- Quản lý hội thoại riêng tư: tạo tin nhắn mới, xóa tin nhắn, thêm cảm xúc cho tin nhắn và thu hồi cảm xúc cho tin nhắn.
- Quản lý hội thoại nhóm: tạo tin nhắn mới, xóa tin nhắn, thêm cảm xúc cho tin nhắn, thu hồi cảm xúc cho tin nhắn, tạo nhóm, cập nhật nhóm, thêm thành viên cho nhóm, xóa thành viên khỏi nhóm, hủy nhóm, xem danh sách nhóm, tham gia nhóm và rời nhóm.



Sơ đồ 3.4 Sơ đồ use-case tổng quát chức năng quản lý hội thoại

- Yêu cầu người dùng phải đăng nhập từ trước.
- Quản lý hội thoại riêng tư:
 - Tạo tin nhắn mới: Cho phép người dùng tạo tin nhắn mới, người dùng còn lại trong đoạn hội thoại sẽ xem được tin nhắn mới.
 - Xóa tin nhắn: Cho phép người dùng xóa tin nhắn đã gửi.
 - Thêm cảm xúc cho tin nhắn: Cho phép người dùng thêm các cảm xúc cho tin nhắn ví dụ như “yêu thích”, “giận dữ”,...

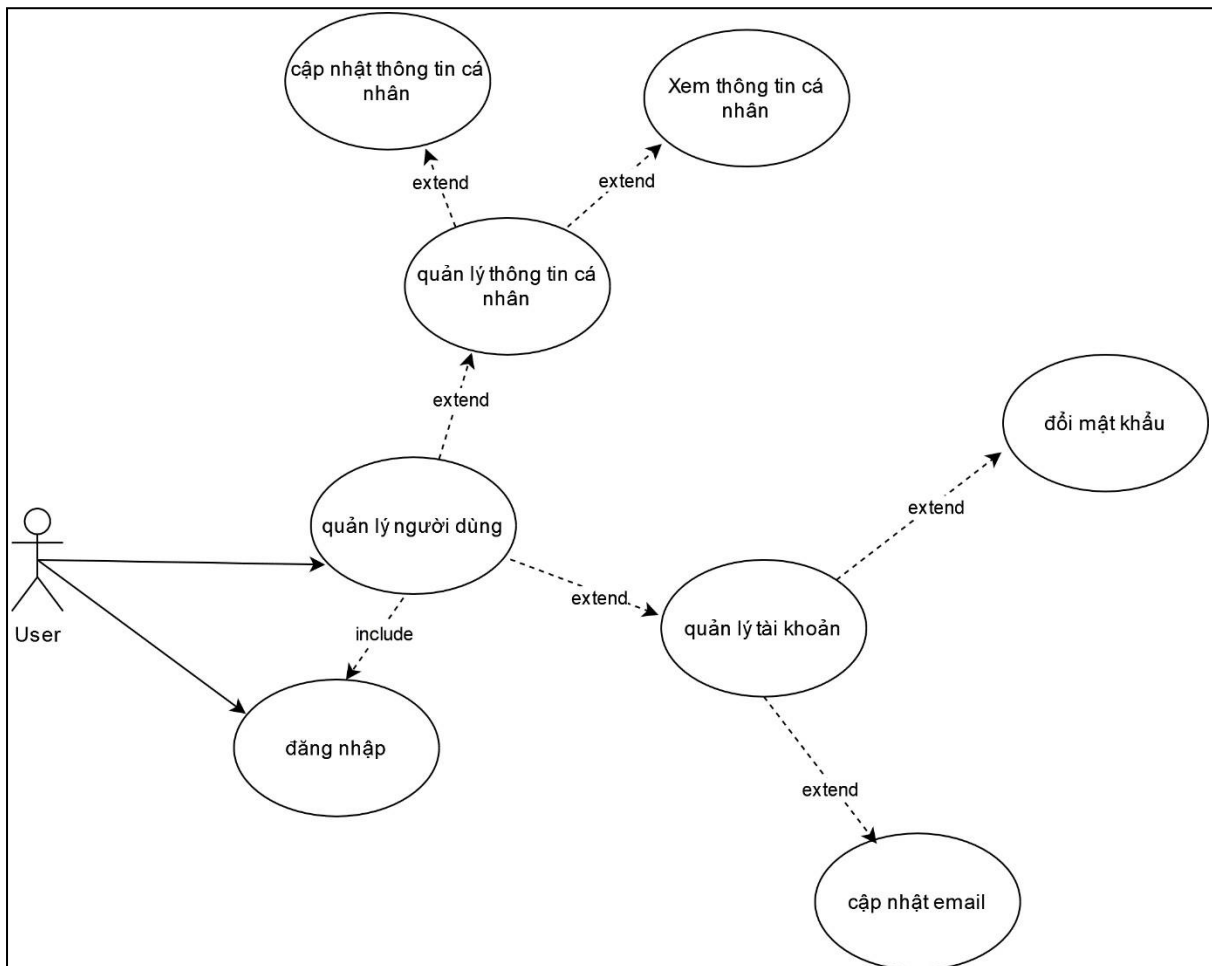
- Thu hồi cảm xúc tin nhắn: Cho Phép người dùng thu hồi các cảm xúc đã thêm.
 - Lưu ý: cuộc hội thoại riêng tư có 1 ràng buộc là 2 người trong cuộc hội thoại phải là bạn bè của nhau, phòng hội thoại riêng tư được tạo tự động khi 2 người trở thành bạn bè của nhau.
- Quản lý hội thoại nhóm
- Tạo tin nhắn mới: Cho phép người dùng tạo tin nhắn mới, những người dùng trong nhóm hội thoại sẽ xem được tin nhắn mới.
 - Xóa tin nhắn: Cho phép người dùng xóa tin nhắn đã gửi.
 - Thêm cảm xúc cho tin nhắn: Cho phép người dùng thêm các cảm xúc cho tin nhắn ví dụ như “yêu thích”, “giận dữ”,...
 - Thu hồi cảm xúc tin nhắn: Cho Phép người dùng thu hồi các cảm xúc đã thêm.
 - Tạo nhóm: Cho phép người dùng tạo nhóm hội thoại mới với trưởng nhóm chính là người tạo, với các thuộc tính là tên nhóm, hình ảnh và thành viên.
 - Cập nhật nhóm: Cho phép người dùng cập nhật nhóm, nhượng quyền trưởng nhóm cho người khác.(chỉ có trưởng nhóm mới có thể thực hiện chức năng này)
 - Thêm thành viên cho nhóm: Cho phép người dùng thêm thành viên cho nhóm hội thoại sau khi nhóm đã được tạo. (chỉ có trưởng nhóm mới có thể thực hiện chức năng này)
 - Xóa thành viên khỏi nhóm: Cho phép người dùng xóa thành viên khỏi nhóm. (chỉ có trưởng nhóm mới có thể thực hiện chức năng này)
 - Hủy nhóm: Cho phép người dùng xóa nhóm hội thoại, mọi tin nhắn trong nhóm hội thoại sẽ được xóa theo. (chỉ có trưởng nhóm mới có thể thực hiện chức năng này)
 - Xem danh sách nhóm hội thoại: cho phép người dùng xem danh sách những nhóm hội thoại đã tham gia.

- Tham gia nhóm: Chức năng cho phép người dùng tham gia vào nhóm hội thoại đã được tạo thông qua 1 đường dẫn được người chia sẻ.
- Rời nhóm: chức năng cho phép người dùng rời khỏi nhóm hội thoại đã tham gia.

➤ Chức năng quản lý người dùng

Bao gồm quản lý thông tin cá nhân và quản lý tài khoản.

- Quản lý thông tin cá nhân: Xem và cập nhật thông tin cá nhân.
- Quản lý tài khoản: Cập nhật Email, đổi mật khẩu.

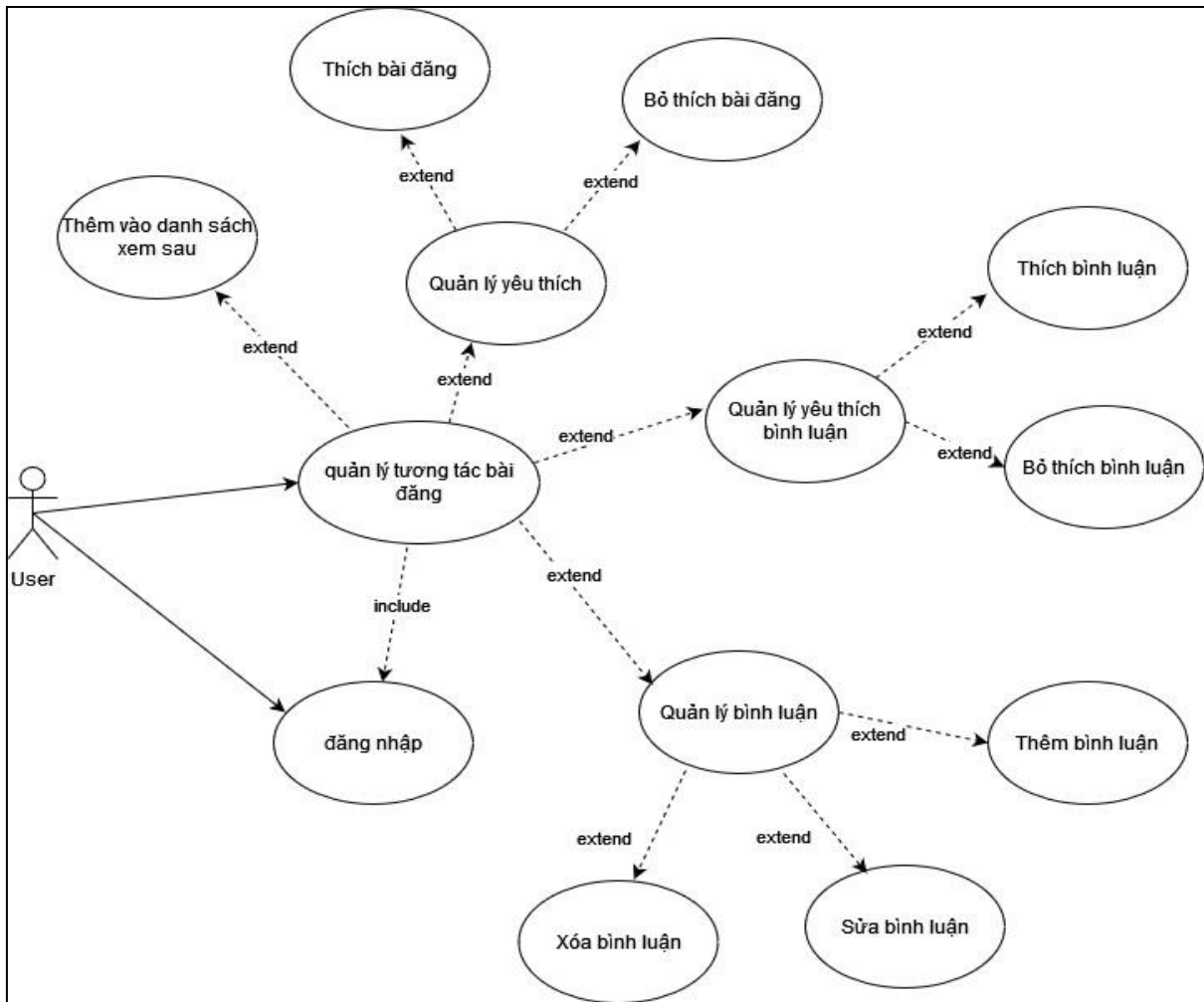


Sơ đồ 3.5 Sơ đồ use-case tổng quát chức năng quản lý người dùng.

- Yêu cầu người dùng phải đăng nhập từ trước
- Quản lý thông tin cá nhân
 - Xem thông tin cá nhân: Cho phép người dùng xem thông tin của bản thân.
 - Cập nhật thông tin cá nhân: Cho phép người dùng cập nhật thông tin của bản thân.
- Quản lý tài khoản
 - Đổi mật khẩu: Cho phép người dùng thay đổi mật khẩu tài khoản.
 - Cập nhật Email: Cho phép người dùng cập nhật Email mới.
- Chức năng quản lý tương tác bài đăng

Bao gồm quản lý yêu thích bài đăng, quản lý yêu thích bình luận và quản lý bình luận.

- Quản lý yêu thích bài đăng: Yêu thích bài đăng và bỏ thích bài đăng.
- Quản lý yêu thích bình luận: Yêu thích bình luận và bỏ yêu thích bình luận.
- Quản lý bình luận: Thêm bình luận, sửa và xóa bình luận.
- Quản lý danh sách xem sau.

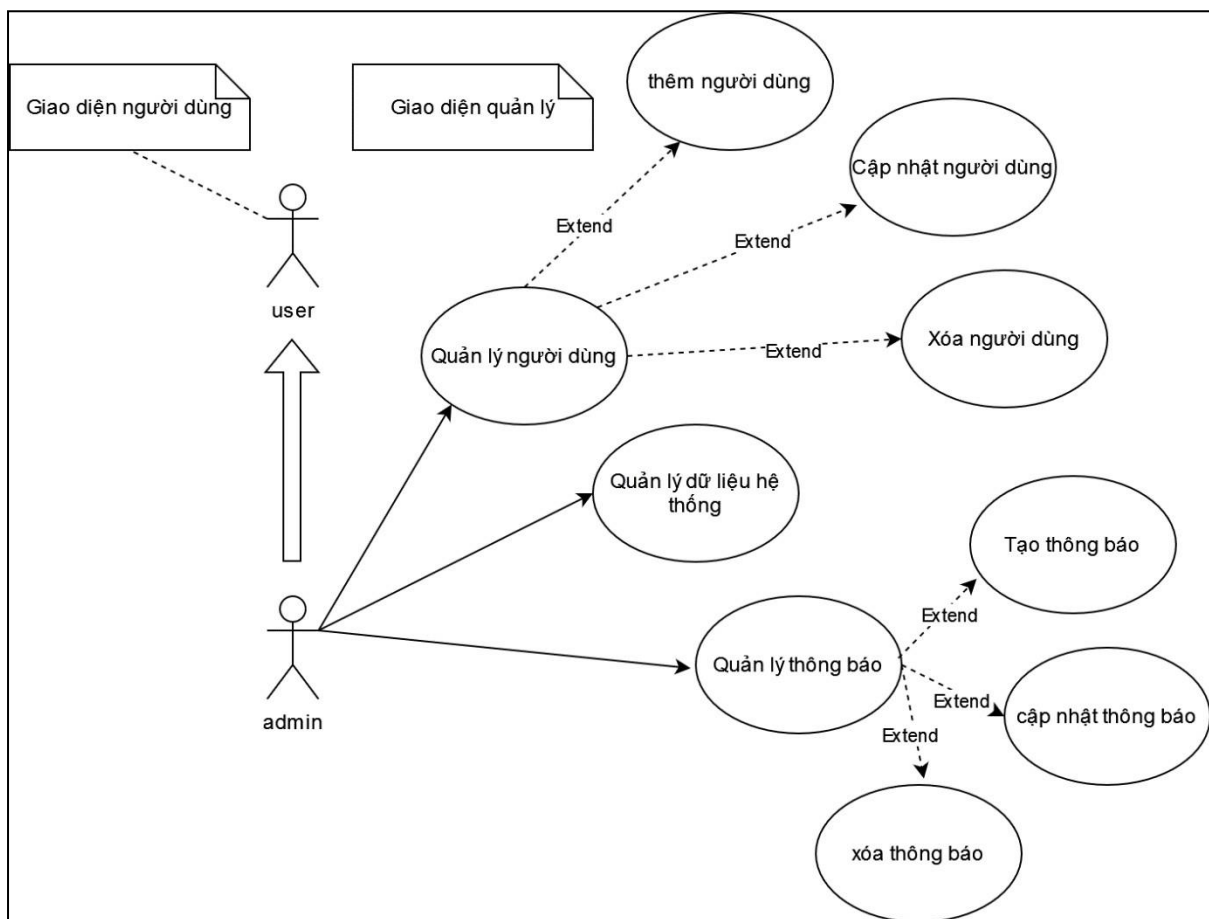


Sơ đồ 3.6 Sơ đồ use-case tổng quát chức năng quản lý tương tác bài đăng

- Yêu cầu người dùng phải đăng nhập từ trước.
- Quản lý yêu thích:
 - Thích bài đăng: Cho phép người dùng thích bài đăng.
 - Bỏ thích bài đăng: Cho phép người dùng bỏ thích bài đăng.
- Quản lý yêu thích bình luận:
 - Thích bình luận: Cho phép người dùng thích bình luận.
 - Bỏ thích bình luận: Cho phép người dùng bỏ thích bình luận.
- Quản lý bình luận:

- Thêm bình luận: Cho phép người dùng thêm bình luận mới cho bài đăng.
 - Sửa bình luận: Cho phép người dùng sửa bình luận của bản thân.
 - Xóa bình luận: Cho phép người dùng xóa bình luận của bản thân.
- Thêm bài đăng vào danh sách xem sau:
 - Thêm vào danh sách xem sau: Cho phép người dùng lưu bài đăng vài mục danh sách xem sau của bản thân.

3.1.2 Chức năng của admin



Sơ đồ 3.7 Sơ đồ use-case tổng quát các chức năng của admin

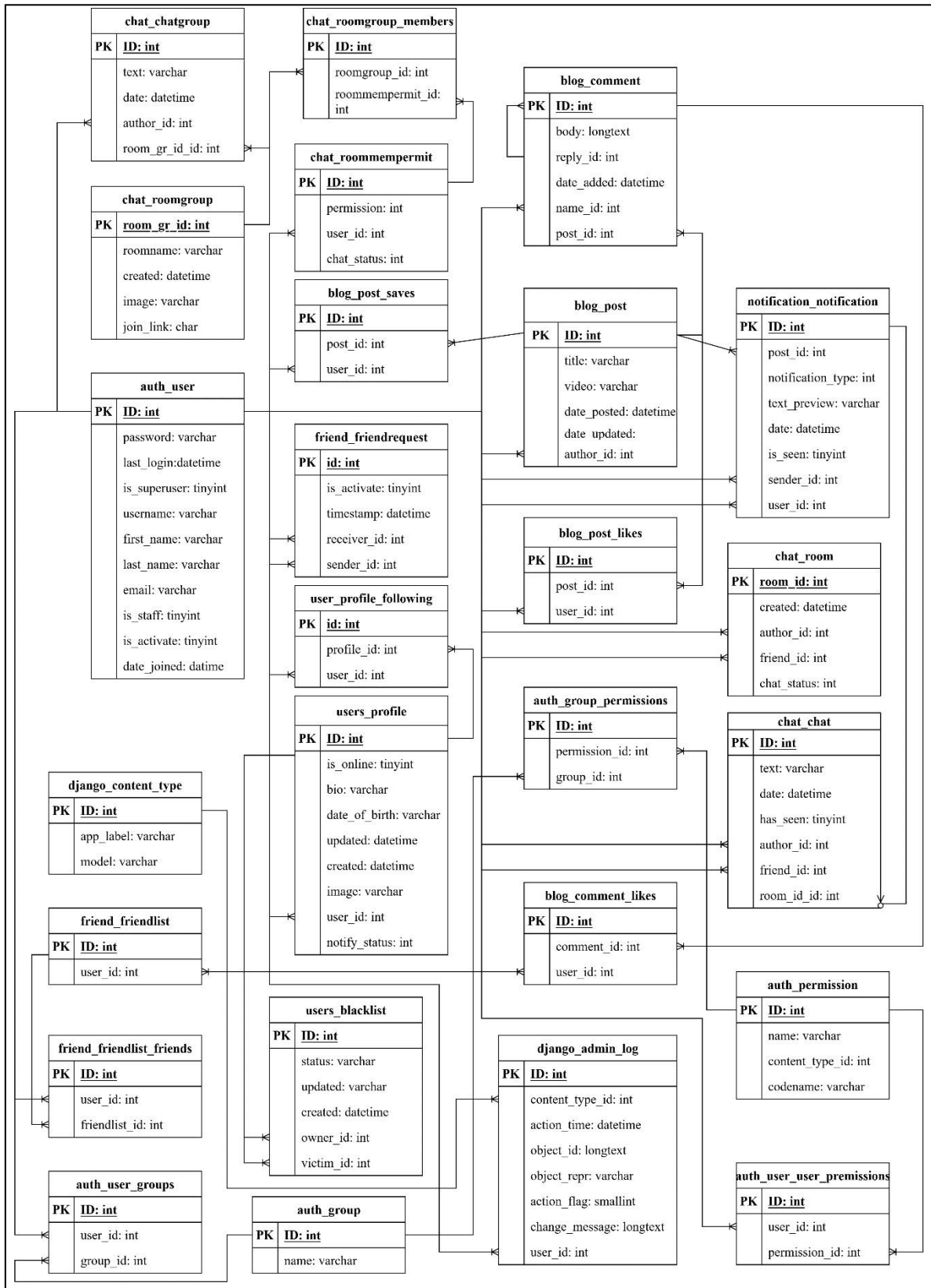
- Yêu cầu người dùng phải đăng nhập từ trước.
- Bao gồm tất cả các chức năng của người dùng, khi sử dụng với giao diện người dùng.

➤ Khi sử dụng với giao diện quản trị:

- Quản lý người dùng: thêm người dùng, cập nhật người dùng và xóa người dùng.
 - Thêm người dùng: Cho phép người dùng quản trị thêm người dùng mới với các thông tin cơ bản như username, tài khoản, mật khẩu, ...
 - Cập nhật người dùng: Cho phép người dùng quản trị cập nhật thông tin người dùng, bao gồm cả thông tin về tài khoản như email, cấp mật khẩu mới, cũng có thể phân quyền cho người dùng.
 - Xóa người dùng: Người quản trị có thể xóa tài khoản người dùng.
- Quản lý dữ liệu hệ thống: Người dùng quản trị có quyền truy cập và quản lý toàn bộ hệ thống, bao gồm cả cơ sở dữ liệu. Điều này đồng nghĩa người dùng quản trị có khả năng thay đổi tất cả thông tin liên quan đến người dùng, bài đăng, dữ liệu của hệ thống,...
- Quản lý thông báo:
 - Tạo thông báo mới: Người quản trị có thể tạo thông báo mới.
 - Cập nhật thông báo: Người quản trị có thể cập nhật nội dung cho thông báo.
 - Xóa thông báo: Người quản trị có thể xóa thông báo.
- *Lưu ý: Thông báo được tạo từ trang của người quản trị thường sẽ là những thông báo đặc biệt, không thường xuyên xuất hiện. Mặc định, thông báo sẽ được tạo ra khi người dùng tương tác với 1 vài chức năng đã được thiết lập sẵn, vd: “khi người dùng A yêu thích bài viết của người dùng B thì người dùng B sẽ nhận được thông báo”, “khi người dùng A gửi lời mời kết bạn đến người dùng B thì người dùng B sẽ nhận được thông báo với nội dung tương ứng”,...

3.2 Thiết kế hệ thống

3.2.1 Sơ đồ cơ sở dữ liệu



Sơ đồ 3.8 Sơ đồ cơ sở dữ liệu

3.2.2 Chi tiết cơ sở dữ liệu:

➤ Bảng auth_user: Lưu trữ thông tin của user

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	Int	Khóa chính
2	Password	Varchar	Mật khẩu của người dùng
3	Is_superuser	Tinyint	Có phải người dùng quản trị hay không
4	username	varchar	Tên đăng nhập
5	First_name	Varchar	Họ người dùng
6	Last_name	Varchar	Tên người dùng
7	Email	Varchar	Địa chỉ Email
8	If_staff	Tinyint	Có phải nhân viên hay không
9	Is_activate	Tinyint	Trạng thái hoạt động
10	Date_joined	Datetime	Ngày đăng ký
11	Last_login	Datetime	Thời gian đăng nhập gần nhất

Bảng 3.1 auth_user

➤ Bảng auth_permission: Quyền người dùng

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	ID	Int	Khóa chính
2	Name	Varchar	Tên hiển thị
3	Content_type_id	Int	Đối tượng tham chiếu trong hệ thống
4	Codename	Varchar	Tên chức năng

Bảng 3.2 auth_permission

- Bảng `django_content_type`: Bảng tham chiếu các đối tượng trên bảng trong cơ sở dữ liệu trên toàn bộ hệ thống.

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	App_label	Varchar	Tên của app
3	Model	Varchar	Tên của đối tượng

Bảng 3.3 django_content_type

- Bảng `auth_group_permission`: Phân quyền nhóm người dùng:

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	Group_id	Int	Id Nhóm người dùng
3	Permission_id	Int	Id của quyền

Bảng 3.4 auth_group_permission

- Bảng `auth_user_user_permission`: Bảng phân quyền người dùng cá nhân.

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	User_id	Int	Id của Người dùng
3	Permission_id	Int	Id của quyền

Bảng 3.5 auth_user_user_permission

- Bảng `auth_group`: Bảng nhóm người dùng để phân quyền

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	Name	Varchar	Tên nhóm

Bảng 3.6 auth_group

➤ Bảng `user_blacklist`: Bảng danh sách người dùng hạn chế

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	Status	Varchar	Trạng thái
3	Updated	Datetime	Thời gian được cập nhật
4	Created	Datetime	Thời gian tạo
5	Owner_id	Int	Id người hạn chế
6	Victim_id	Int	Id người bị hạn chế

Bảng 3.7 user_blacklist

➤ Bảng `auth_user_groups`: Người dùng tham gia nhóm được phân quyền

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	User_id	Int	Id người dùng tham gia
3	Group_id	Int	Id nhóm phân quyền

Bảng 3.8 auth_user_groups

➤ Bảng `django_admin_log`: Lịch sử hành động trên giao diện admin

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	Content_type_id	Int	Id loại thao tác
3	Action_time	Datetime	Thời gian thao tác
4	Object_id	Longtext	Id đối tượng tương tác
5	Object_repr	Varchar	Đối tượng tương tác
6	Action_flag	Smallint	Hành động
7	Change_message	Longtext	Nội dung thay đổi
8	User_id	Int	Id người dùng tương tác

Bảng 3.9 django_admin_log

➤ Bảng chat_chatgroup: Tin nhắn của hội thoại nhóm

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	Text	Varchar	Nội dung tin nhắn
3	Date	Datetime	Thời gian tin nhắn được tạo
4	Author_id	Int	Id người tạo tin nhắn
5	Room_gr_id_id	Int	Id phòng hội thoại nhóm

Bảng 3.10 chat_chatgroup

➤ Bảng chat_roomgroup: Phòng hội thoại nhóm

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Room_gr_id	int	Khóa chính
2	Roomname	Varchar	Tên phòng
3	Create	Datetime	Ngày tạo phòng
4	Image	Varchar	Đường dẫn đến file hình ảnh
5	Join_link	Char	Đường dẫn để tham gia nhóm

Bảng 3.11 chat_roomgroup

➤ Bảng chat_chat: Tin nhắn hội thoại riêng tư

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	Text	Varchar	Nội dung tin nhắn
3	Date	Datetime	Thời gian tạo tin nhắn
4	Has_seen	Tinyint	Tin nhắn đã được xem hay chưa
5	Author_id	Int	Id người tạo tin nhắn
6	Friend_id	Int	Id người nhận tin nhắn
7	Room_id_id	Int	Id phòng hội thoại riêng tư

Bảng 3.12 chat_chat

➤ Bảng chat_roommempermit: Quyền của người dùng trong hội thoại nhóm

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	Permission	Int	Quyền trong hội thoại nhóm
3	User_id	Int	Id người dùng được phân quyền
4	Chat_status	Int	Trạng thái tin nhắn

Bảng 3.13 chat_roommempermit

➤ Bảng chat_roomgroup_members: Bảng quy định người dùng thuộc nhóm hội thoại nào

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	Roomgroup_id	Int	Id của nhóm hội thoại
3	Roommempermit_id	Int	Id của

Bảng 3.14 chat_roomgroup_members

➤ Bảng blog_post: Lưu trữ toàn bộ bài đăng

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	Title	Varchar	Tiêu đề bài đăng
3	Video	Varchar	Đường dẫn đến tệp video
4	Date_posted	Datetime	Ngày đăng bài
5	Date_update	Datetime	Ngày cập nhật bài đăng
6	Author_id	Int	Id tác giả

Bảng 3.15 blog_post

➤ Bảng blog_post_save: Lưu trữ bài đăng mà người dùng thêm vào xem sau.

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	Post_id	Int	Id bài đăng mà người dùng lưu trữ
3	User_id	Int	Id người dùng lưu trữ bài đăng

Bảng 3.16 blog_post_save

➤ Bảng friend_friendrequest: Lời mời kết bạn

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	Is_activate	Tinyint	Lời mời có đang gửi hay không
3	Timestamp	Datetime	Thời gian tạo lời mời
4	Receiver_id	Int	Id người nhận lời mời
5	Sender_id	Int	Id người gửi lời mời

Bảng 3.17 friend_friendrequest

➤ Bảng users_profile: Thông tin trang cá nhân của người dùng

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	Is_online	Tinyint	Trạng thái hoạt động
3	Bio	Varchar	Tiểu sử người dùng
4	Date_of_birth	Varchar	Ngày tháng năm sinh
5	Update	Datetime	Ngày cập nhật thông tin trang cá nhân
6	Created	Datetime	Ngày tạo thông tin trang cá nhân
7	Image	Varchar	Đường dẫn đến ảnh đại diện
8	User_id	Int	Id người dùng
9	Notify_status	Int	Trạng thái thông báo

Bảng 3.18 users_profile

➤ Bảng user_profile_following: Danh sách trang cá nhân mà người dùng theo dõi

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	Profile_id	Int	Id trang cá nhân mà người dùng theo dõi
3	User_id	Int	Id người dùng theo dõi

Bảng 3.19 user_profile_following

- Bảng friend_friendlist: Bảng người dùng được xây dựng cho mỗi quan hệ bạn bè.

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	User_id	Int	Id người dùng

Bảng 3.20 friend_friendlist

- Bảng friend_friendlist_friends: Quy định mối quan hệ bạn bè

STT	Thuộc tính	Kiểu dữ liệu	Thuộc tính
1	Id	Int	Khóa chính
2	Friendlist_id	Int	Id bạn bè của người dùng
3	User_id	Int	Id người dùng

Bảng 3.21 friend_friendlist_friends

- Bảng blog_comment: Lưu trữ bình luận bài đăng

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	Body	Longtext	Nội dung bình luận
3	Date_added	Datetime	Ngày tạo bình luận
4	Name_id	Int	Id người tạo bình luận
5	Post_id	Int	Id bài đăng của bình luận
6	Reply_id	Int	Id của bình luận được trả lời

Bảng 3.22 blog_comment

- Bảng blog_post_likes: Lưu trữ lượt yêu thích của bài đăng

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	Post_id	Int	Id bài đăng
3	User_id	Int	Id người dùng yêu thích bài đăng

Bảng 3.23 blog_post_likes

➤ Bảng `blog_comment_likes`: Lưu trữ lượt yêu thích của bình luận

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	Comment_id	Int	Id bình luận
3	User_id	Int	Id người dùng yêu thích bình luận

Bảng 3.24 blog_comment_likes

➤ Bảng `chat_room`: Phòng hội thoại riêng tư

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Room_id	Int	Khóa chính
2	Created	Datetime	Ngày tạo phòng
3	Author_id	Int	Id người thứ 1 của hội thoại
4	Friend_id	Int	Id người thứ 2 của hội thoại
5	Chat_status	Int	Trạng thái chat

Bảng 3.25 chat_room

➤ Bảng `notification_notification`: Lưu trữ thông báo gửi đến người dùng

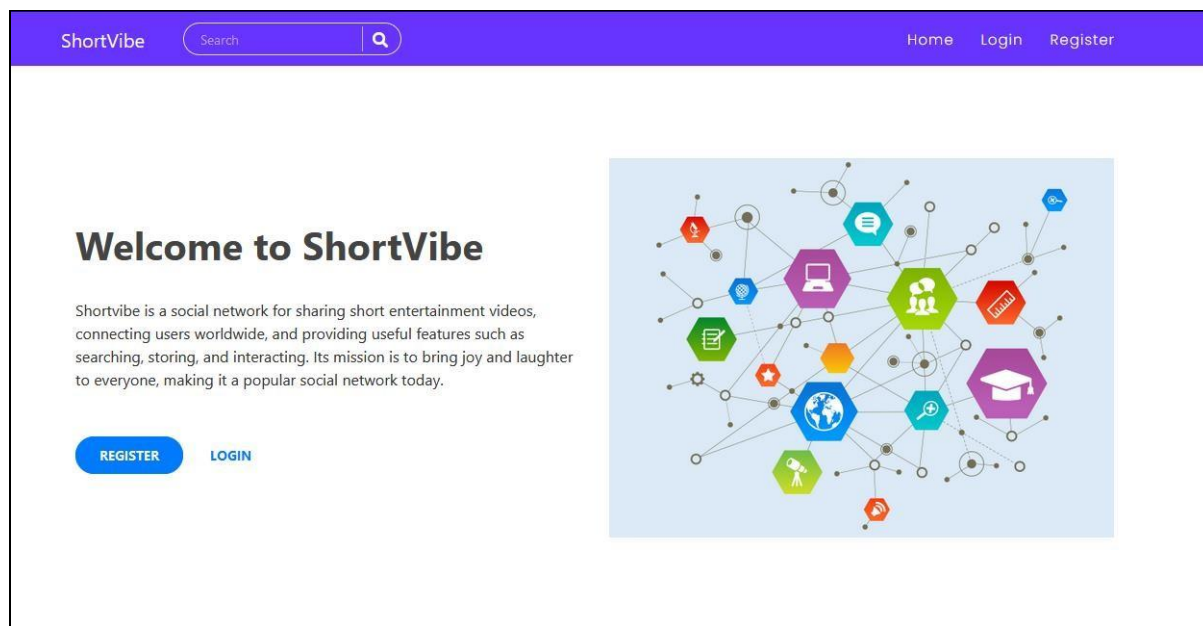
STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Id	Int	Khóa chính
2	Post_id	Int	Id bài đăng
3	Notification_type	Int	Loại thông báo
4	Text_preview	Varchar	Nội dung xem trước của thông báo
5	Date	Datetime	Ngày tạo thông báo
6	Is_seen	Tinyint	Thông báo đã được xem hay chưa
7	Sender_id	Int	Id người gửi thông báo
8	User_id	Int	Id người nhận thông báo

Bảng 3.26 notification_notification

CHƯƠNG 4 XÂY DỰNG ỨNG DỤNG

4.1 Giao diện giới thiệu

Khi người dùng truy cập vào trang web thì mặc định sẽ là giao diện giới thiệu, mô tả khái quát về trang web. Từ màn hình giới thiệu này có thể truy cập đến màn hình đăng nhập hoặc màn hình đăng ký thông qua các lối tắt tương ứng.



Hình 4.7 Giao diện giới thiệu

4.2 Giao diện đăng ký

Ở màn hình đăng ký có chứa các dòng để người dùng có thể nhập thông tin đăng ký, việc đăng ký sẽ kèm theo các ràng buộc về dữ liệu nhất định mà người đăng ký cần phải đọc kỹ để có thể thao tác thành công, ngoài ra người dùng còn phải xác nhận Captcha để có thể hoàn tất quá trình đăng ký.

Lưu ý, email ở khi đăng ký sẽ được dùng để lấy lại mật khẩu trong trường hợp người dùng quên mật khẩu.

ShortVibe Search Home Login Register

Join Today

First name

Last name

Username*

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

Email*

Password*

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation*

Enter the same password as before, for verification.

I'm not a robot

reCAPTCHA
Privacy - Terms

Sign Up

Hình 4.8 Giao diện đăng ký

4.3 Giao diện đăng nhập

Ở giao diện đăng nhập, người dùng sẽ phải nhập tên đăng nhập và mật khẩu để thực hiện quá trình đăng nhập, người dùng có thể tích chọn “remember me” để lưu lại thông tin đăng nhập. Ngoài ra, trong trường hợp người dùng quên mật khẩu, có thể di chuyển đến trang lấy lại mật khẩu qua lối tắt “Forgot password ?”

ShortVibe Search Home Login Register

Login

Username*

Password*

Remember Me

[Login](#) [Forgot Password?](#)

Need an Account? [Sign Up Now](#)

Hình 4.9 Giao diện đăng nhập

4.4 Giao diện lấy lại mật khẩu

Ở màn hình này, người dùng sẽ nhập địa chỉ email đã khai báo khi đăng ký tài khoản để yêu cầu đặt lại mật khẩu mới, sau khi yêu cầu thì sẽ có 1 email được gửi đến hộp thư của người dùng, kèm theo hướng dẫn đặt lại mật khẩu mới.

ShortVibe Search Home Login Register

Reset Password

Email*

[Request Password Reset](#)

Hình 4.10 Giao diện gửi yêu cầu đặt lại mật khẩu

Sau khi gửi yêu cầu thành công.

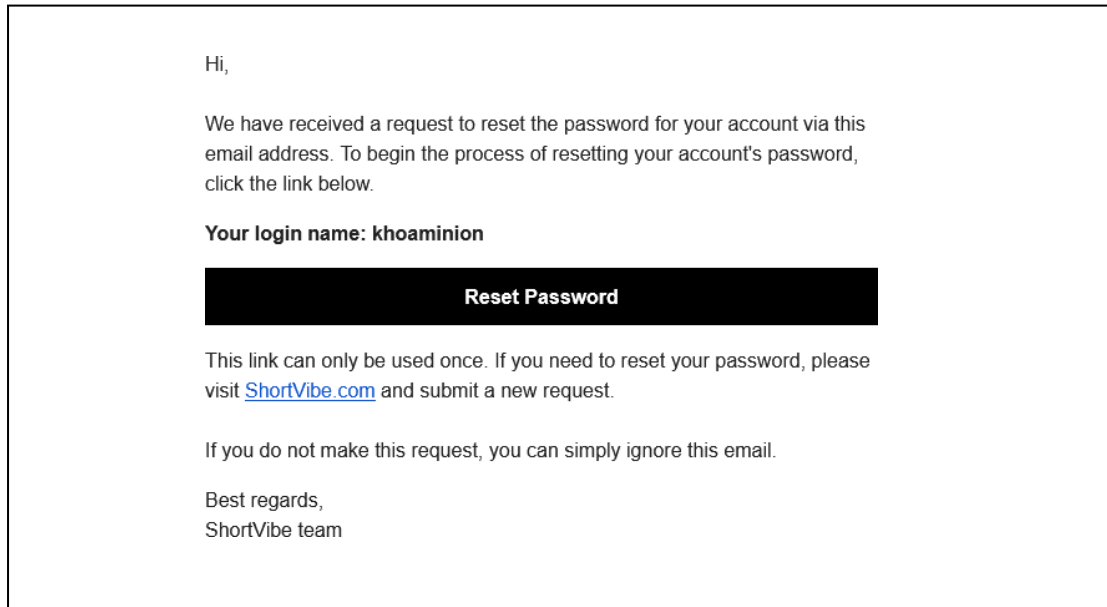
ShortVibe Search Home Login Register

An Email has been sent with instruction to reset your password

Hình 4.11 Giao diện gửi yêu cầu cấp mật khẩu thành công

4.5 Giao diện email hướng dẫn lấy lại mật khẩu

Email được gửi đến sẽ kèm 1 đường dẫn, đường dẫn sẽ dẫn đến trang để người dùng có thể đặt lại mật khẩu, ngoài ra, trong email còn gửi cả tên đăng nhập hàng trường hợp người dùng có thể sẽ quên.



Hình 4.12 Giao diện email hướng dẫn đặt lại mật khẩu

4.6 Giao diện đặt lại mật khẩu

Sau khi nhấn vào đường dẫn từ email thì đây chính là trang đặt lại mật khẩu mà người dùng sẽ được dẫn đến, tại đây người dùng sẽ nhập mật khẩu mới 2 lần để tiến hành quá trình đặt lại mật khẩu, việc đặt lại mật khẩu cũng sẽ có các ràng buộc tương ứng như khi tạo tài khoản.

ShortVibe Search Home Login Register

Reset Password

New password*

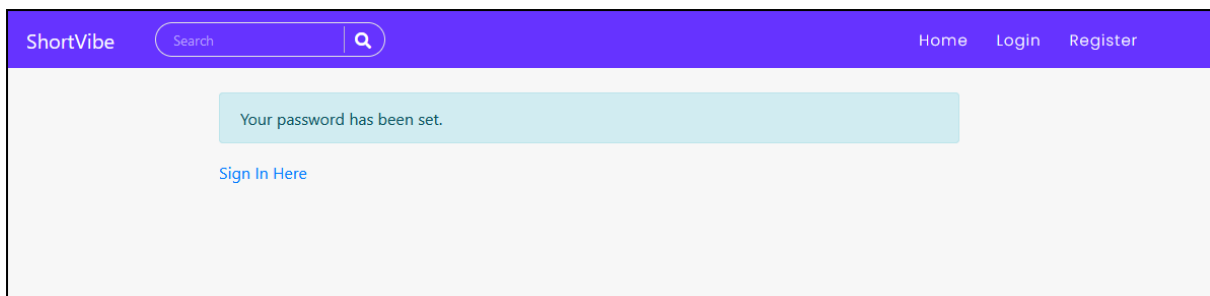
- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

New password confirmation*

Reset Password

Hình 4.13 Giao diện đặt lại mật khẩu

Sau khi đặt lại mật khẩu thành công.



Hình 4.14 Giao diện sau khi đặt lại mật khẩu thành công

4.7 Giao diện trang chủ

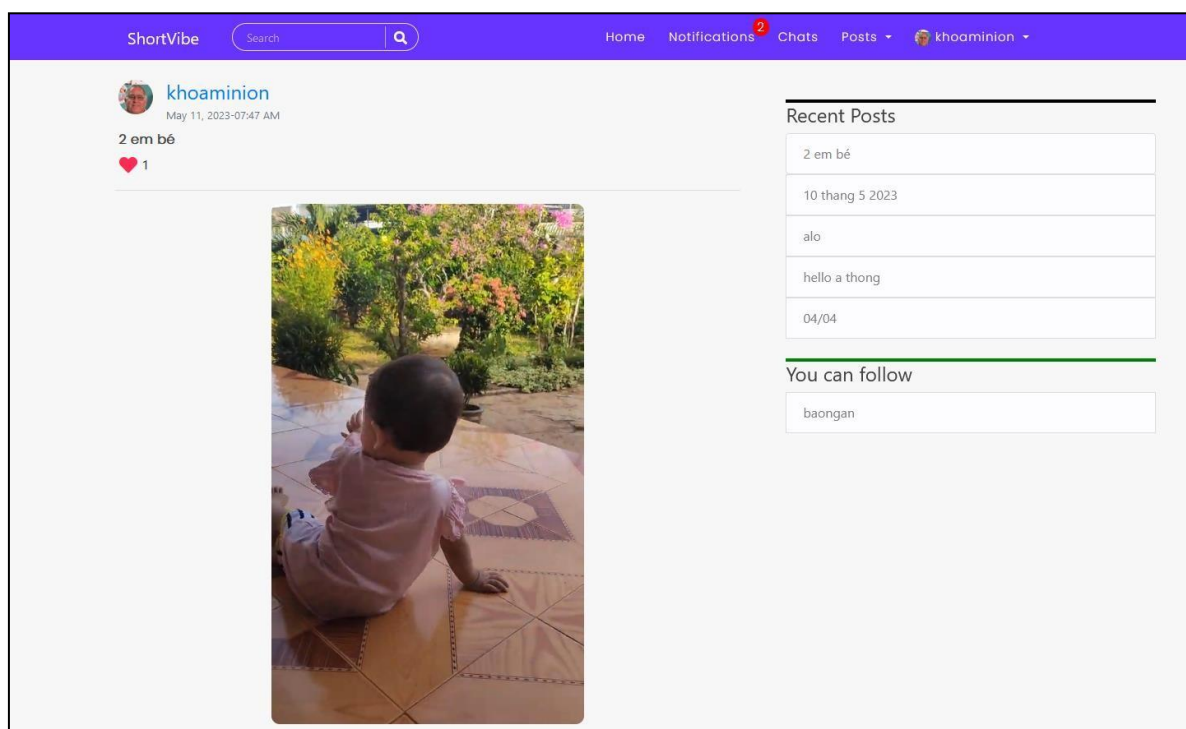
Sau khi đăng nhập thành công thì người dùng sẽ được chuyển đến trang chủ, ở đây hiển thị toàn bộ bài đăng của hệ thống với thứ tự sắp xếp theo thời gian tạo của bài đăng, phía trên cùng là logo của trang web, thanh tìm kiếm và các shortcut dẫn đến các chức năng khác.

Phía chính giữa là danh sách các bài đăng, các bài đăng sẽ được phát tự động khi người dùng cuộn chuột đến, và cũng sẽ tự động dừng khi bị che khuất. Các bài đăng sẽ được tải thêm mỗi khi người dùng cuộn đến gần cuối trang.

Mỗi bài đăng sẽ có các thông tin như, tác giả, tiêu đề bài đăng, tổng số lượt yêu thích, và cuối cùng là nội dung video.

Khi người dùng nhấn vào biểu tượng trái tim của bài đăng, thì sẽ thay đổi trạng thái yêu thích hoặc bỏ yêu thích bài đăng, và tương ứng tổng số lượt yêu thích cũng sẽ tăng hoặc giảm tùy theo trạng thái của biểu tượng. Nếu nhấn vào video đang phát thì video sẽ được dừng và ngược lại, nếu người dùng nhấn vào tiêu đề của bài đăng thì sẽ được chuyển đến trang chi tiết của bài đăng đó.

Phía bên phải sẽ là phần các bài đăng mới nhất vừa được đăng tải, ngay dưới đó là phần những người dùng ngẫu nhiên khác, mà người dùng có thể theo dõi hoặc kết bạn.



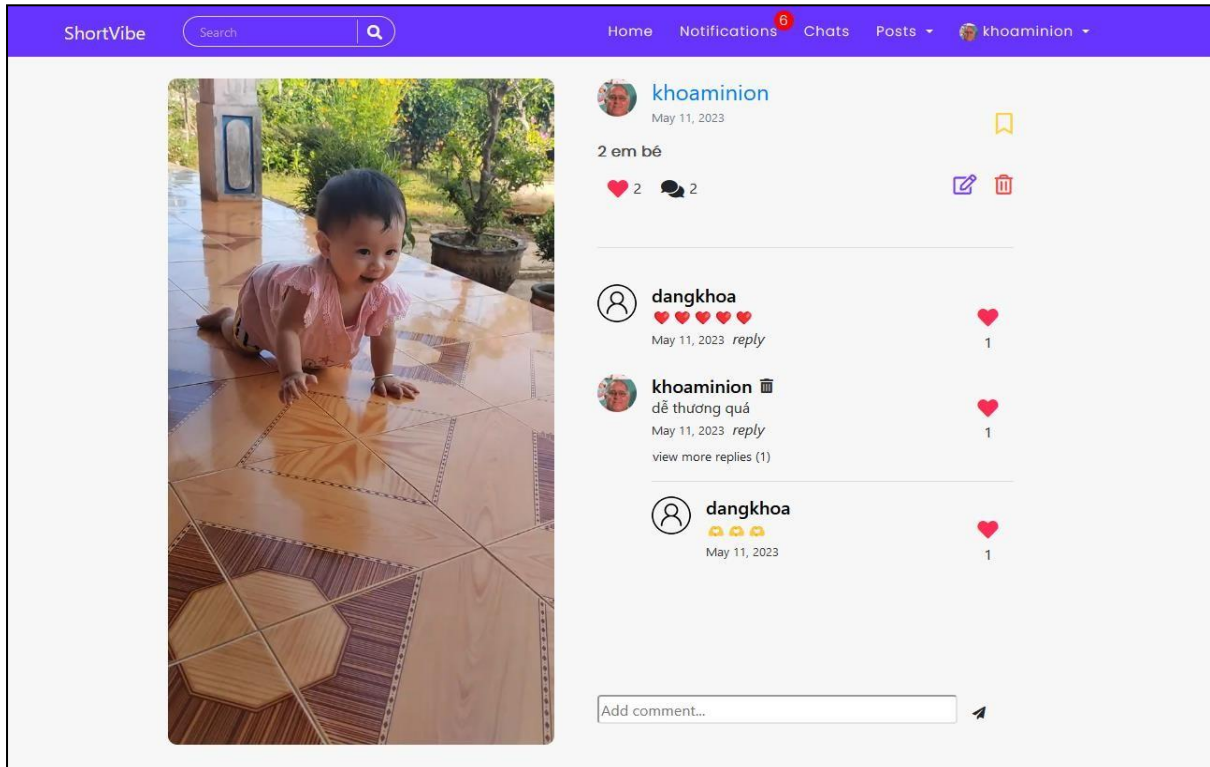
Hình 4.15 Giao diện trang chủ

4.8 Giao diện chi tiết bài đăng

Ở màn hình này, sẽ hiển thị chi tiết nội dung của bài đăng với video sẽ có thêm thanh điều khiển, danh sách bình luận của bài đăng được sắp xếp theo thứ tự thời gian tạo. Có thêm biểu tượng lưu video để xem sau, 2 biểu tượng chỉnh sửa bài đăng và xóa bài đăng sẽ chỉ xuất hiện nếu chủ của bài đăng chính là người dùng.

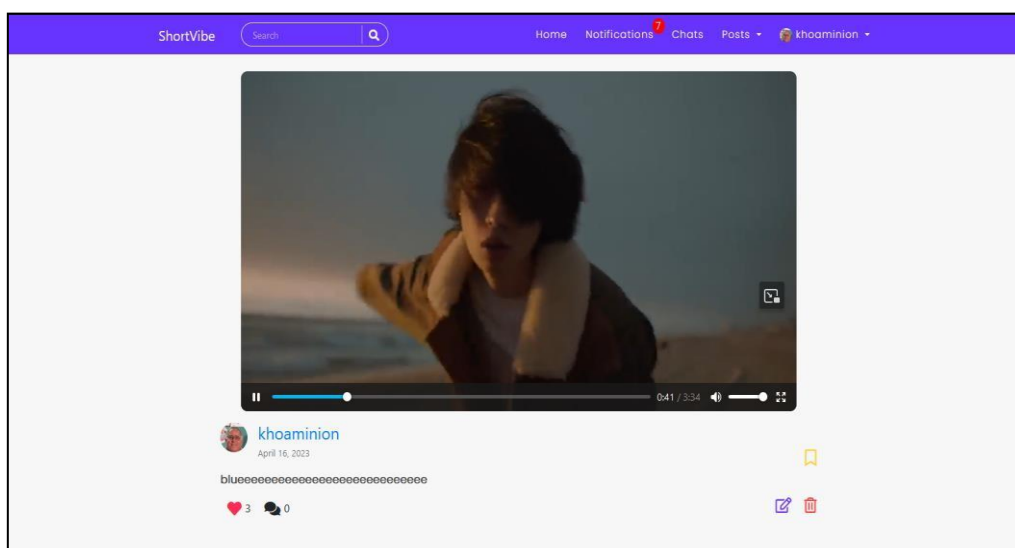
Ở phần bình luận cũng có thể thêm yêu thích cho bình luận và trả lời bình luận, nếu bình luận hiển thị là của người dùng thì sẽ có thêm biểu tượng để xóa bình luận đó.

Tùy vào tỉ lệ là ngang hay học của video mà trang chi tiết bài đăng sẽ có cách sắp xếp giao diện khác nhau để có được sự tối ưu về trải nghiệm nhất có thể cho người dùng.



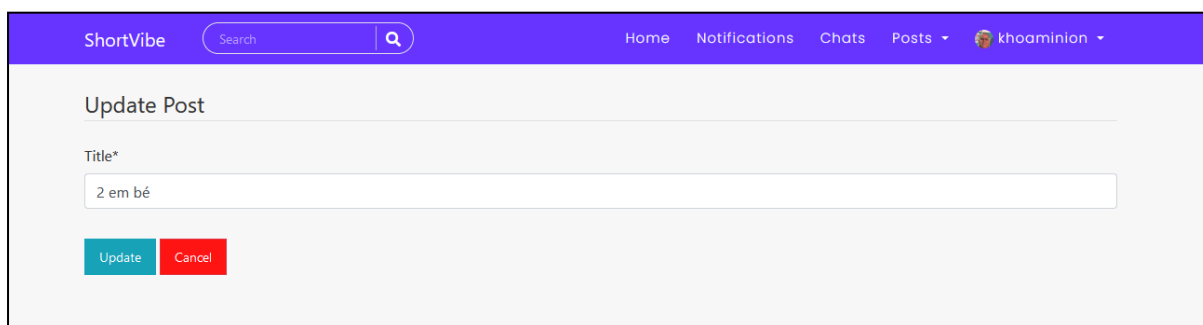
Hình 4.16 Giao diện trang chi tiết bài đăng với video tỉ lệ dọc

Giao diện trang chi tiết bài đăng với video tỉ lệ ngang.



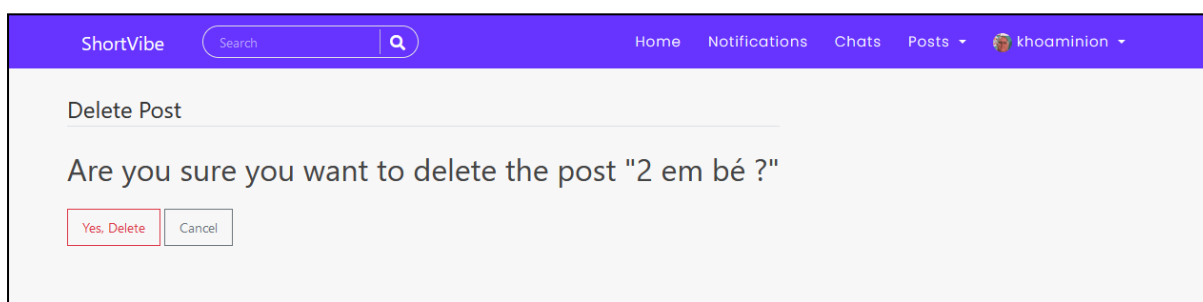
Hình 4.17 Giao diện trang chi tiết bài đăng với video tỉ lệ ngang

Giao diện cập nhật tiêu đề bài đăng.



Hình 4.18 Giao diện cập nhật tiêu đề bài đăng

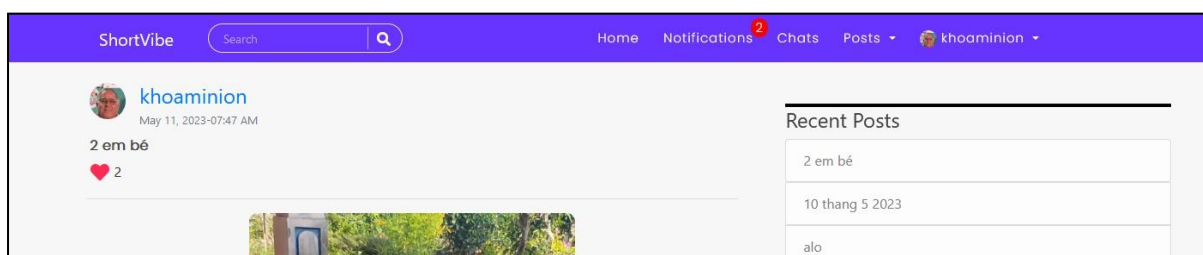
Giao diện xóa bài đăng.



Hình 4.19 Giao diện xóa bài đăng

4.9 Giao diện trang thông báo

Khi người dùng có thông báo thì ở sẽ có icon thông báo hiển thị kèm theo số lượng thông báo chưa xem của người dùng.



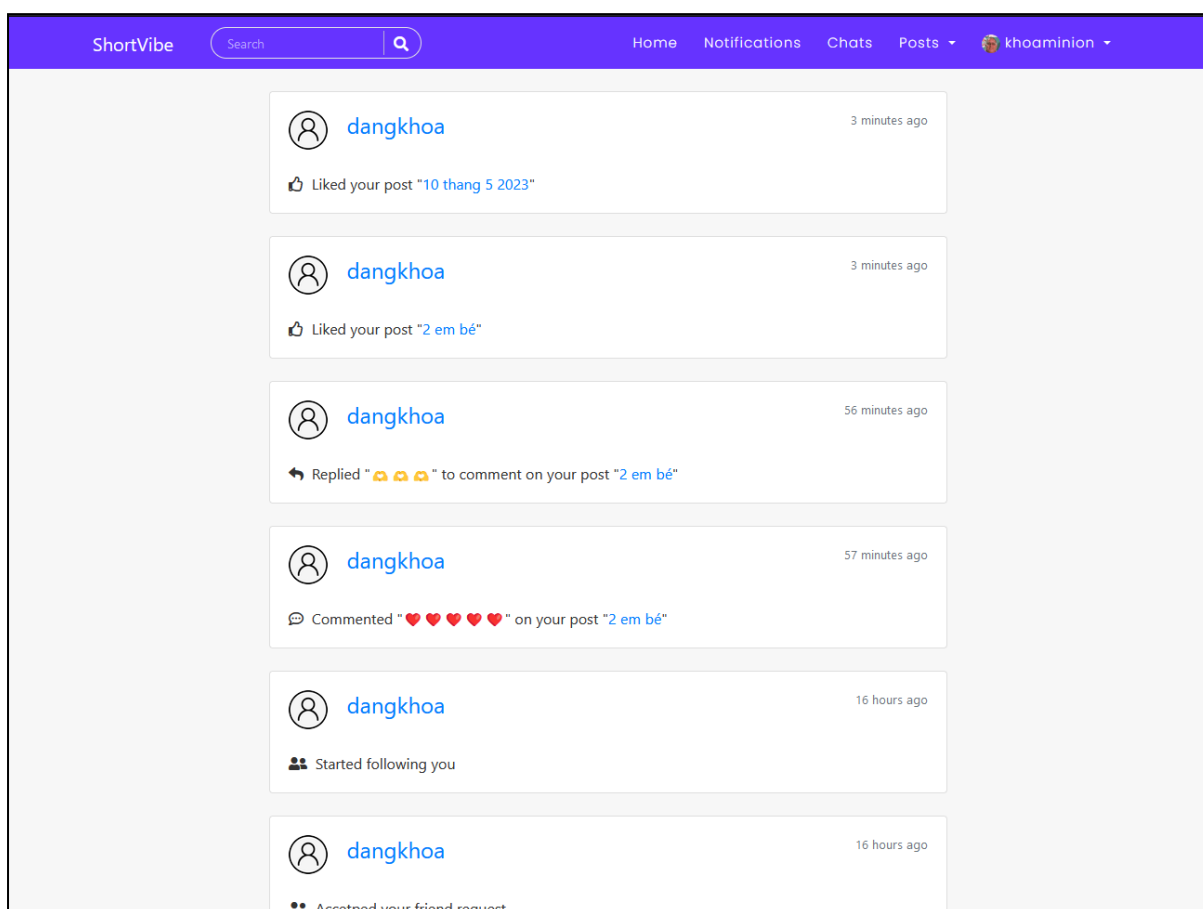
Hình 4.20 Giao diện icon thông báo

Khi người dùng chuyển đến trang danh sách thông báo thì người dùng có thể xem được toàn bộ thông báo được sắp xếp theo mới nhất đến cũ nhất, và mặc định khi đang có thông báo, nếu người dùng chuyển đến trang thông báo thì toàn bộ thông báo

đều được đánh dấu là đã đọc, tương ứng icon thông báo sẽ biến mất.

Thông báo sẽ được tạo tự động và gửi đến người dùng khi người dùng khác có những hành động trên hệ thống có liên quan đến người dùng, chẳng hạn gửi lời mời kết bạn, theo dõi người dùng, trả lời bình luận của người dùng,...

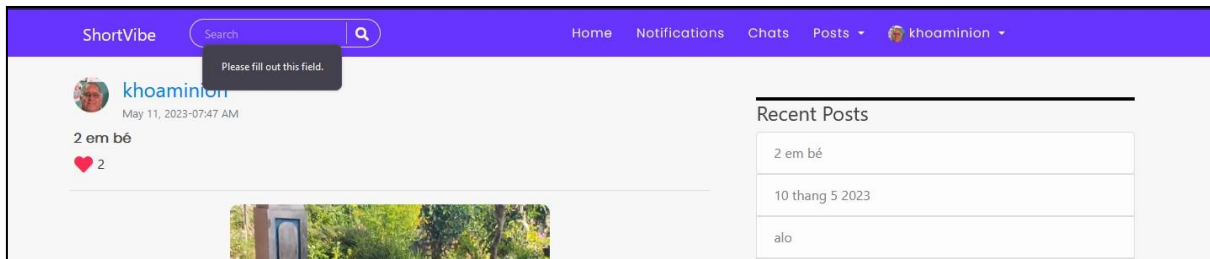
Từ danh sách thông báo này, có thể chuyển đến đối tượng được đề cập đến trong thông báo bằng cách chọn vào nó.



Hình 4.21 Giao diện danh sách thông báo

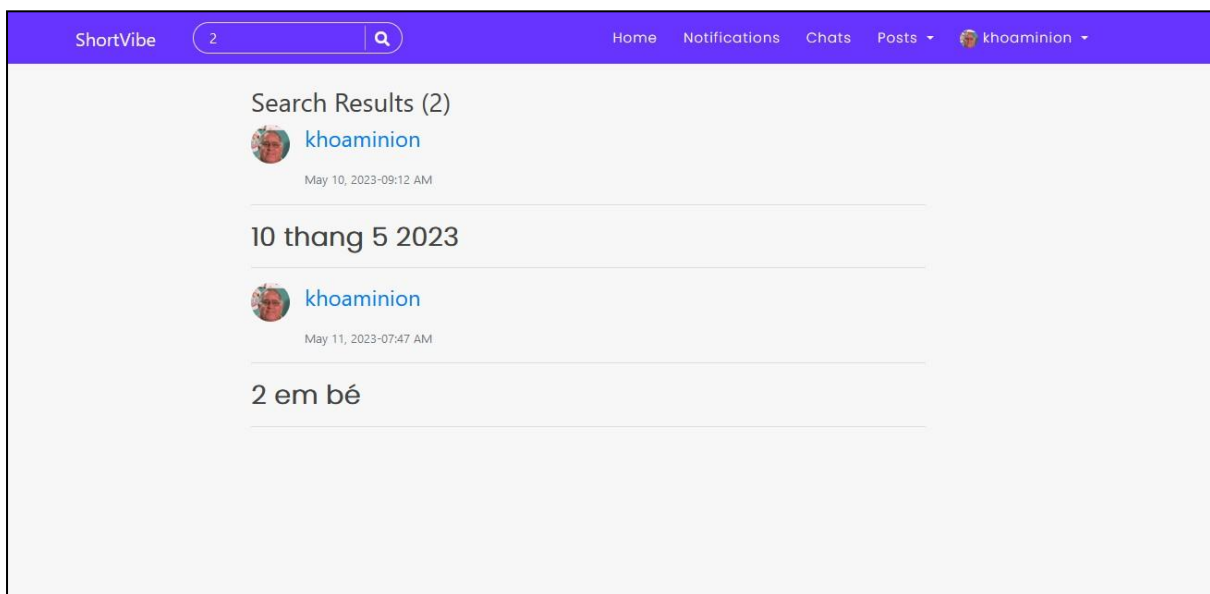
4.10 Giao diện trang tìm kiếm bài đăng

Từ thanh tìm kiếm phải nhập giá trị vào và nhấn tìm kiếm thì sẽ chuyển đến trang giao diện tìm kiếm bài đăng.



Hình 4.22 Giao diện thanh tìm kiếm yêu cầu phải có giá trị để có thể thao tác

Ở trang này sẽ hiện thị các bài đăng có tiêu đề giống với nội dung tìm kiếm nhất có thể. Từ đây có thể di chuyển nhanh đến trang chi tiết bài đăng thông qua bài đăng.



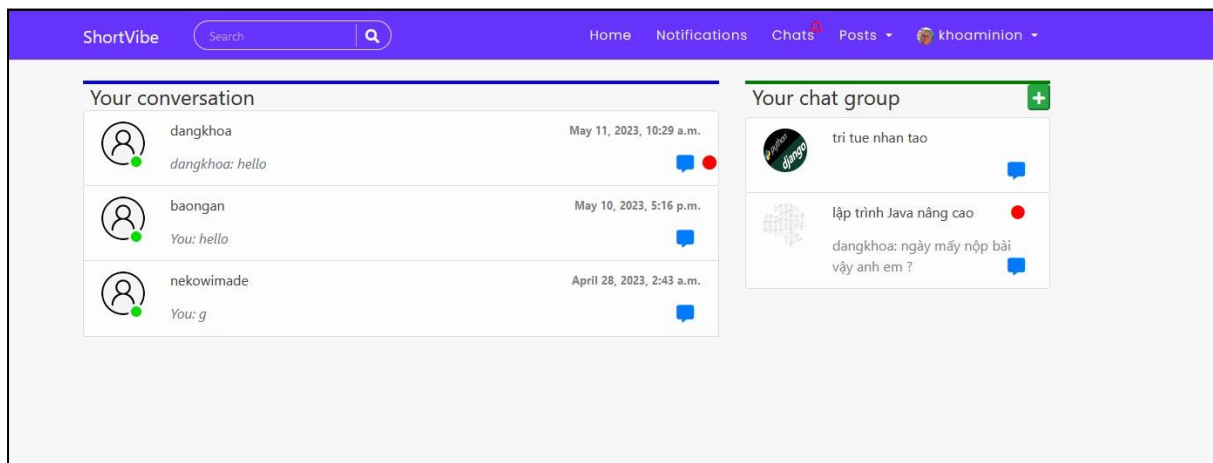
Hình 4.23 Giao diện trang tìm kiếm hiển thị bài đăng theo nội dung tìm kiếm

4.11 Giao diện danh sách hội thoại riêng tư và hội thoại nhóm

Ở giao diện này, phần bên trái là danh sách các cuộc hội thoại riêng tư, hội thoại riêng tư yêu cầu 2 người phải là bạn bè thì mới có thể sử dụng được. Phía bên phải là danh sách các cuộc hội thoại nhóm mà người dùng đang tham gia, cả 2 danh sách đều hiển thị 1 số thông tin cơ bản như tin nhắn cuối cùng, tên nhóm hội thoại hoặc là tên của người hội thoại riêng tư kèm theo trạng thái hoạt động.

Icon màu đỏ xuất hiện ở các cuộc hội thoại đồng nghĩa với việc có tin nhắn mới chưa xem, và với việc dấu chấm màu đỏ còn xuất hiện thì đồng nghĩa menu chat trên thanh điều hướng cũng sẽ xuất hiện biểu tượng chiếc chuông, và khi người dùng truy

cập vào màn hình của cuộc hội thoại thì đồng nghĩa mọi tin nhắn mới đã được xem, icon thông báo sẽ biến mất.

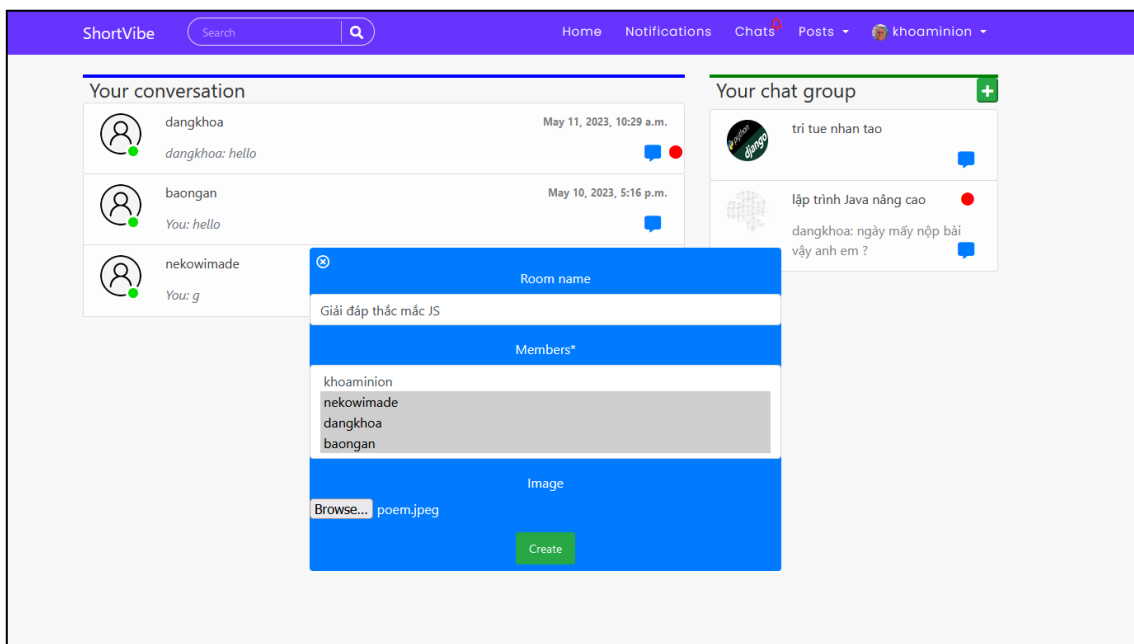


Hình 4.24 Giao diện danh sách hội thoại riêng tư và hội thoại nhóm

Bất kỳ người dùng nào cũng có thể tạo nhóm chat với việc nhấn vào biểu tượng dấu “+” phía bên phải của danh sách nhóm hội thoại.

Cửa sổ thêm tạo nhóm hiện lên, người dùng sẽ nhập tên nhóm, chọn các thành viên sẽ được thêm vào nhóm, danh sách các thành viên được hiển thị ở đây chính là danh sách bạn bè của người dùng. Người dùng cũng có thể chọn ảnh đại diện cho nhóm hội thoại và nhấn “Create” để tạo. Mặc định người tạo nhóm sẽ là trưởng nhóm, và trong 1 nhóm hội thoại chỉ có 1 trưởng nhóm.

Sau khi tạo thành công thì người dùng sẽ được chuyển thẳng đến màn hình nhóm hội thoại vừa mới tạo.

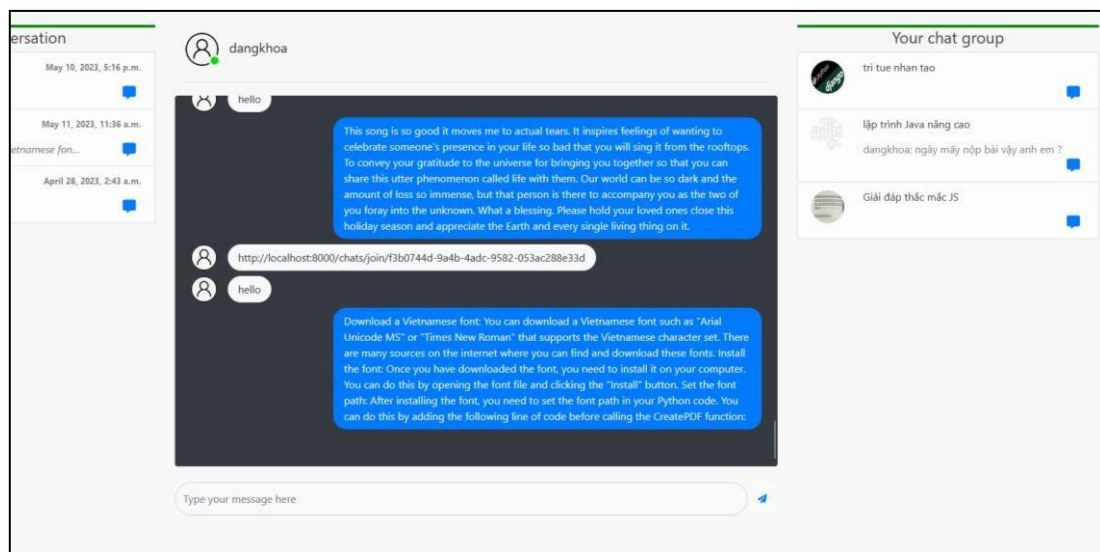


Hình 4.25 Giao diện cửa sổ tạo nhóm hội thoại.

4.12 Giao diện phòng hội thoại

Giao diện phòng hội thoại riêng tư, khu vực chính giữa chính là những tin nhắn của cuộc hội thoại giữa 2 người, khu vực bên trái sẽ hiển thị danh sách hội thoại riêng tư của người dùng, khu vực bên phải là danh sách các nhóm hội thoại mà người dùng tham gia.

Khu vực tin nhắn có vùng để nhập tin nhắn và nút gửi để gửi tin nhắn cho đối phương.



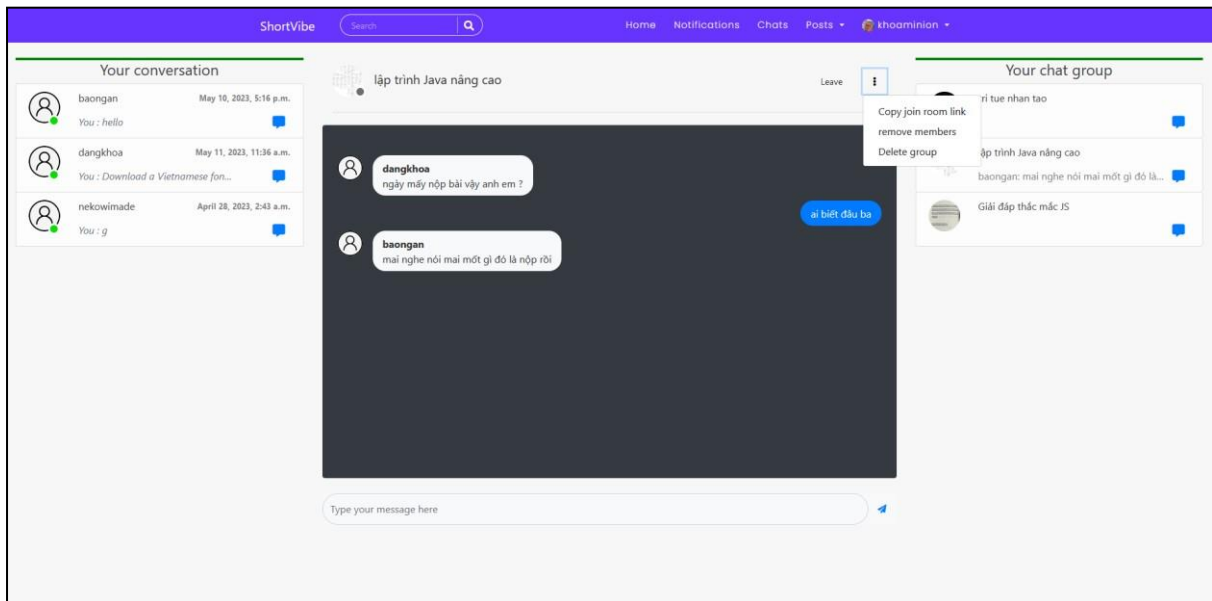
Hình 4.26 Giao diện phòng hội thoại riêng tư

Giao diện phòng hội thoại nhóm, tương tự chức năng và giao diện như phòng hội thoại riêng tư, tuy nhiên, tin nhắn sẽ kèm theo tên người gửi, và có thêm chức năng rời nhóm, lấy link tham gia nhóm giành cho thành viên, riêng đối với thành viên là trưởng nhóm thì có thêm chức năng mời thành viên khác ra khỏi nhóm và xóa nhóm hội thoại.

Với chức năng lấy link tham gia nhóm thì khi người dùng chọn vào, link tham gia nhóm sẽ được tự động lưu vào bộ nhớ tạm, bây giờ, chỉ cần chia sẻ link cho người dùng khác, khi người dùng khác truy cập vào link, nếu điều kiện thỏa mãn thì sẽ tự động tham gia nhóm.

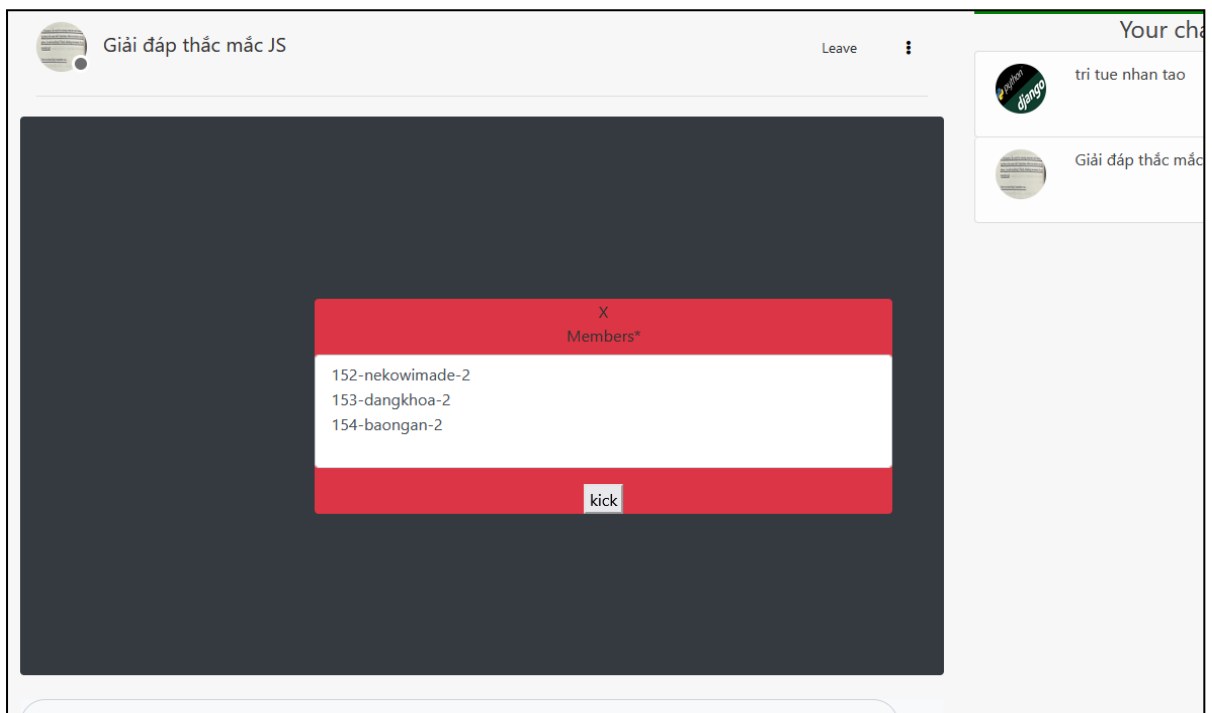
Với chức năng mời người dùng khác ra khỏi nhóm thì sẽ có 1 cửa sổ xuất hiện với danh sách thành viên trong nhóm, người trưởng nhóm chỉ cần chọn những ai cần rời khỏi nhóm và thực thi.

Giao diện phòng hội thoại nhóm của người dùng trưởng nhóm.



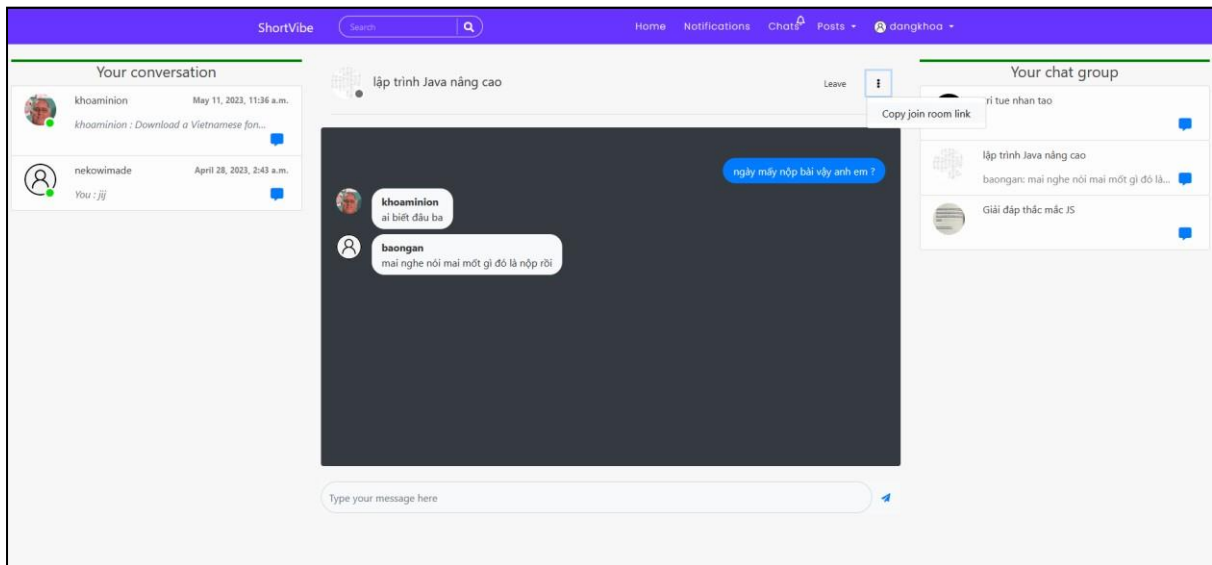
Hình 4.27 Giao diện phòng hội thoại nhóm với thành viên là trưởng nhóm

Giao diện cửa sổ mời thành viên ra khỏi nhóm.



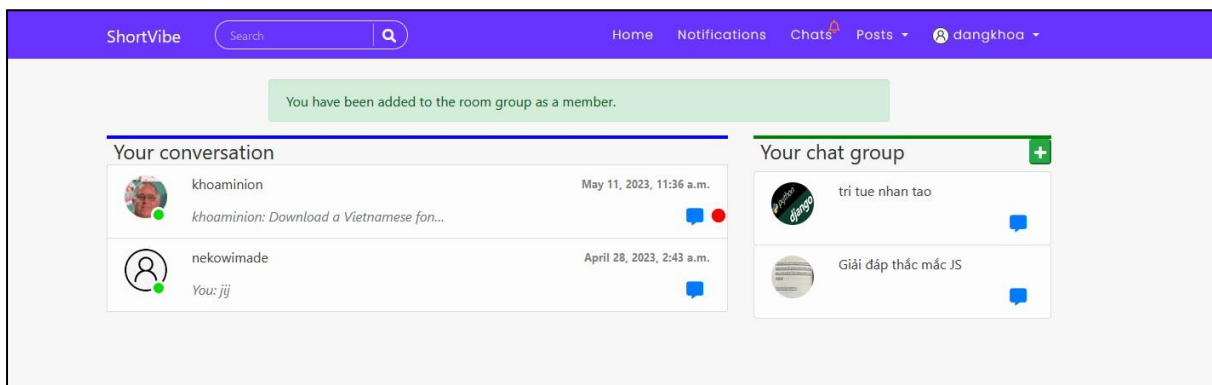
Hình 4.28 Giao diện của sổ mời thành viên ra khỏi nhóm của thành viên trưởng nhóm

Giao diện phòng hội thoại với người dùng là thành viên.



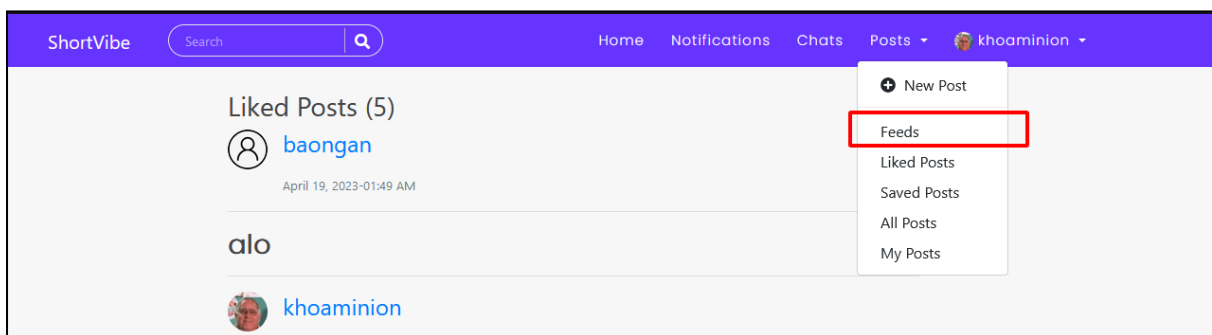
Hình 4.29 Giao diện phòng hội thoại với người dùng là thành viên

Thông báo người dùng đã tham gia nhóm hội thoại thành công.



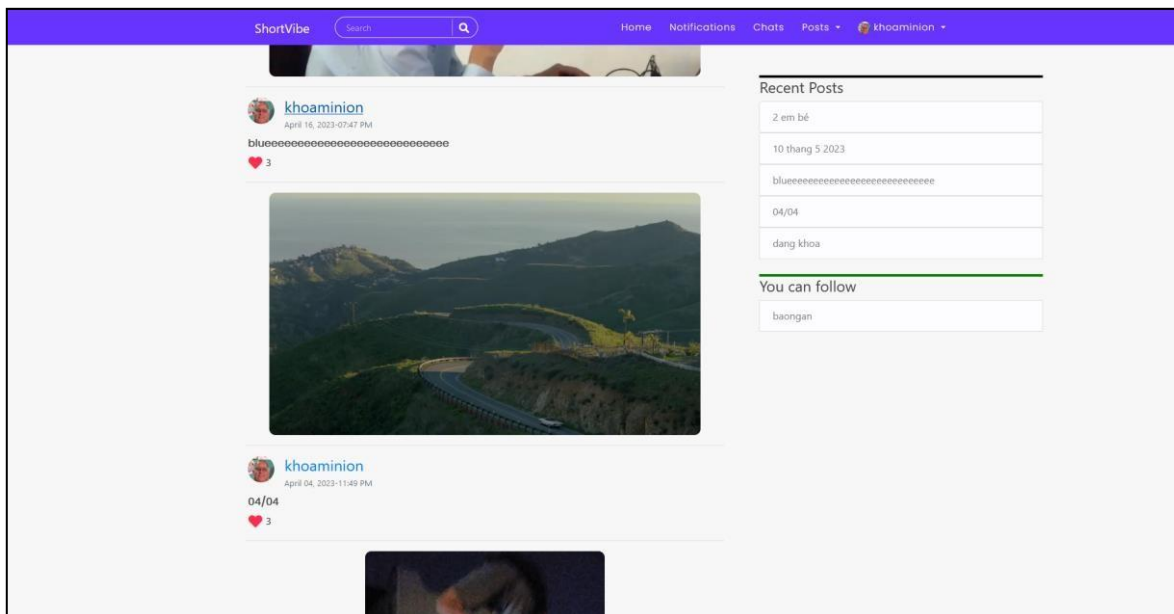
Hình 4.30 Giao diện thông báo người dùng đã tham gia nhóm hội thoại thành công

4.13 Giao diện danh sách bài đăng của những người dùng đang theo dõi



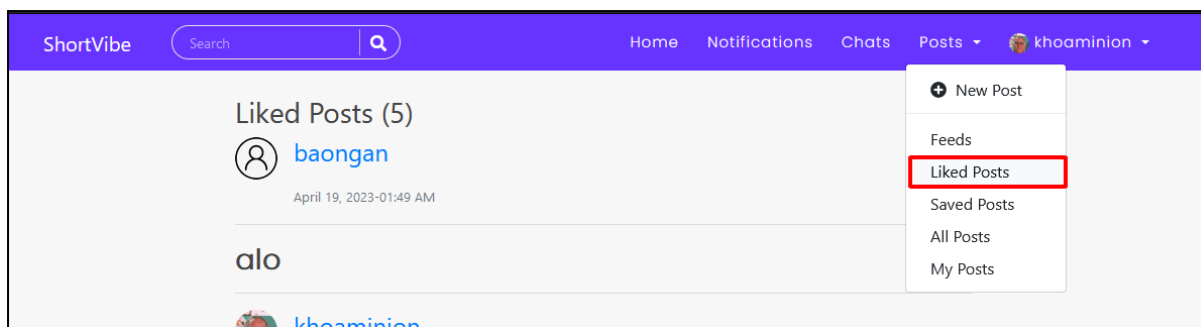
Hình 4.31 Lối tắt dẫn đến danh sách bài đăng của người user đang theo dõi

Tương tự như giao diện trang chủ, chỉ khách những bài đăng được hiển thị chỉ thuộc những người dùng mà user đang theo dõi cũng như những bài đăng của bản thân.



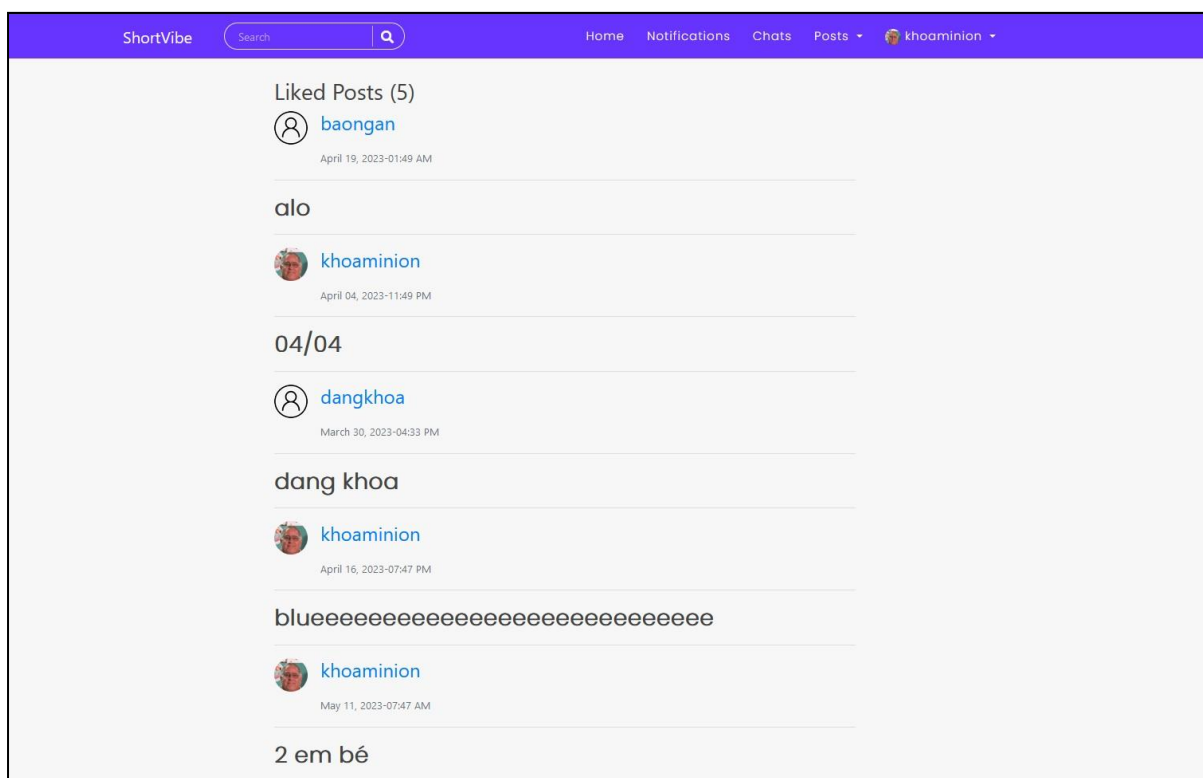
Hình 4.32 Giao diện danh sách bài đăng của những người dùng mà user theo dõi

4.14 Giao diện danh sách những bài đăng mà người dùng đã yêu thích



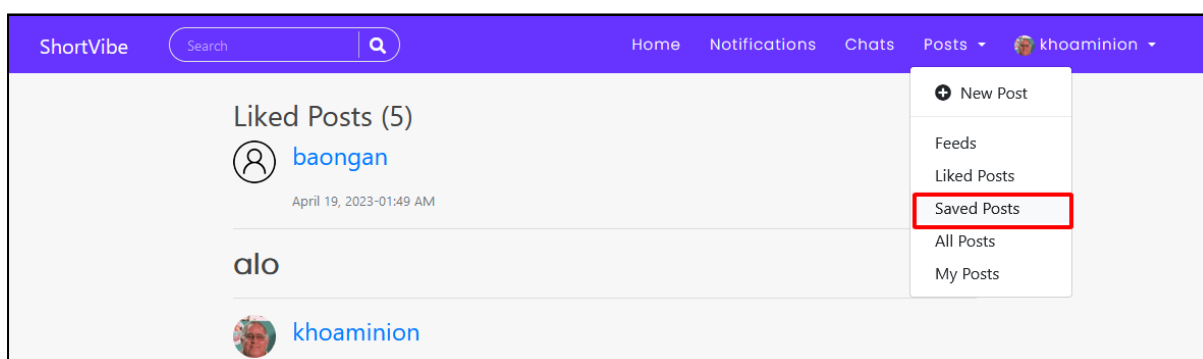
Hình 4.33 Lối tắt dẫn đến danh sách các bài đăng người dùng đã yêu thích

Hiển thị tất cả bài đăng mà người dùng đã yêu thích.



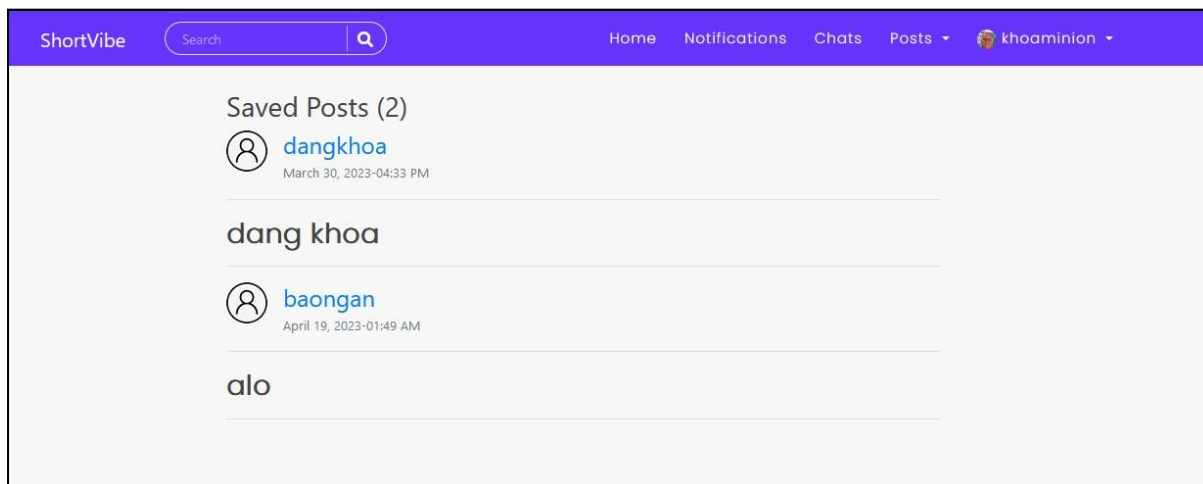
Hình 4.34 Giao diện danh sách những bài đăng mà người dùng đã yêu thích.

4.15 Giao diện danh sách bài đăng người dùng đã thêm vào xem sau



Hình 4.35 Lối tắt đến danh sách các bài đăng đã thêm vào xem sau.

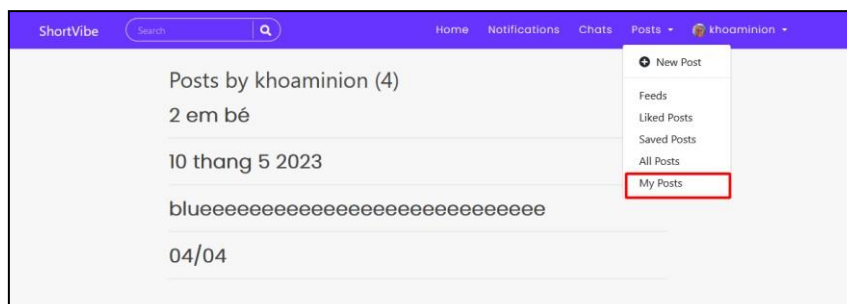
Hiển thị danh sách các bài đăng mà người dùng đã thêm vào mục xem sau.



Hình 4.36 Danh sách bài đăng người dùng đã thêm vào xem sau.

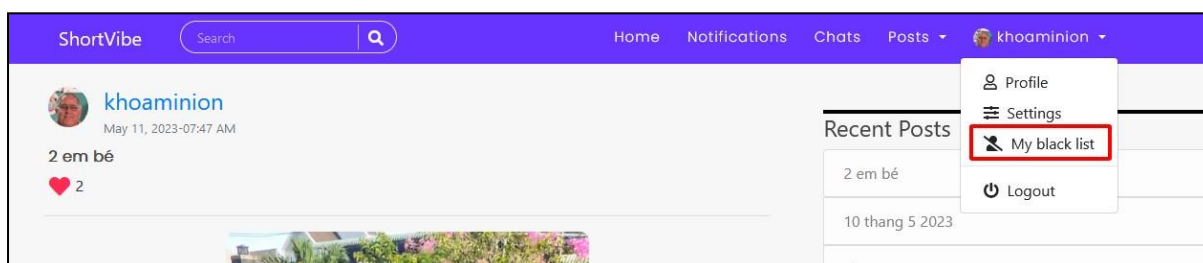
4.16 Giao diện danh sách bài đăng của bản thân người dùng.

Hiển thị toàn bộ bài đăng của bản thân người dùng.



Hình 4.37 Giao diện danh sách bài đăng của bản thân người dùng.

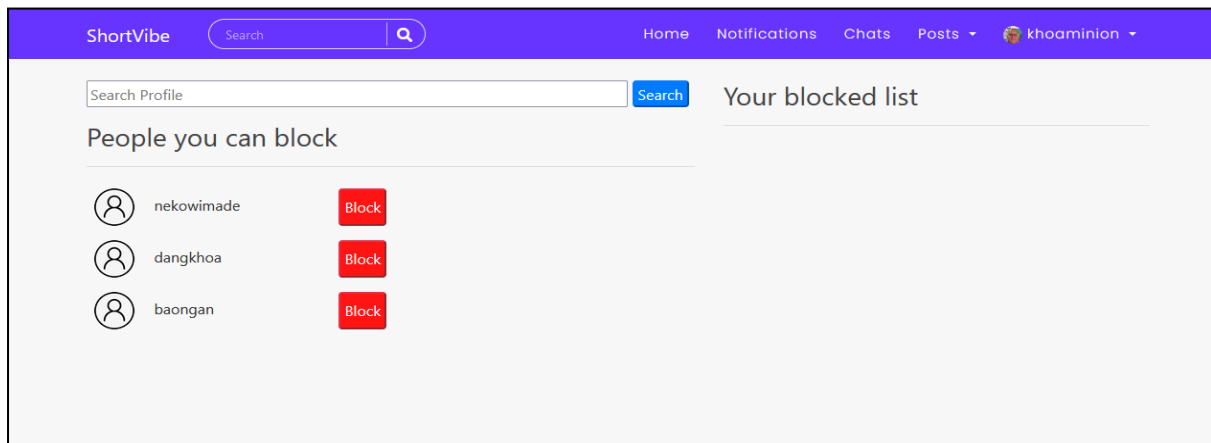
4.17 Giao diện quản lý danh sách người dùng hạn chế



Hình 4.38 Lối tắt đến trang quản lý người dùng hạn chế

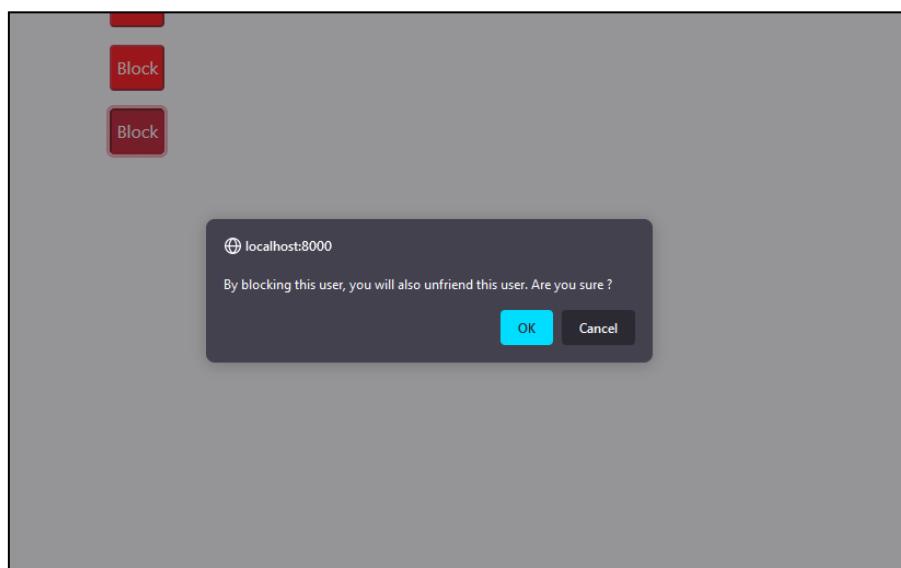
Hiện thị giao diện quản lý danh sách bị hạn chế đối với người dùng. Màn hình sẽ hiển thị toàn bộ người dùng có trong hệ thống, liệt kê thành danh sách người dùng có thể chặn được và danh sách còn lại là những người đã bị hạn chế.

Khi danh sách dài quá thì có thể tìm kiếm username của người dùng từ thanh tìm kiếm. Sau khi nhấn nút block, sẽ có hộp thoại xác nhận và người bị chặn sẽ được chuyển sang danh sách đã chặn.



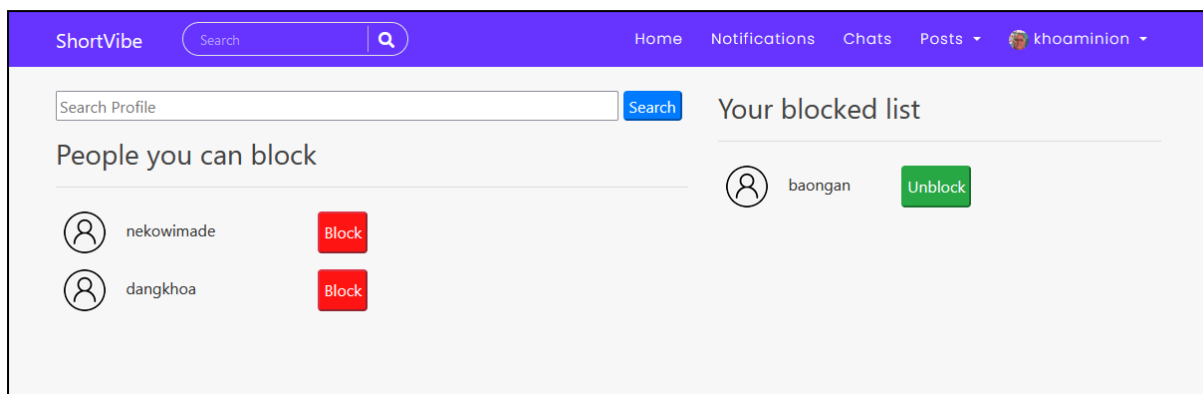
Hình 4.39 Giao diện trang quản lý người dùng hạn chế

Cửa sổ xác nhận thêm người dùng vào danh sách hạn chế. Khi thêm vào danh sách hạn chế thì quan hệ bạn bè giữa 2 người dùng cũng sẽ bị xóa bỏ.



Hình 4.40 Xác nhận đưa người dùng vào danh sách hạn chế

Người dùng đã bị đưa vào danh sách hạn chế. Để gỡ người dùng ra khỏi danh sách hạn chế chỉ việc click vào nút unblock.

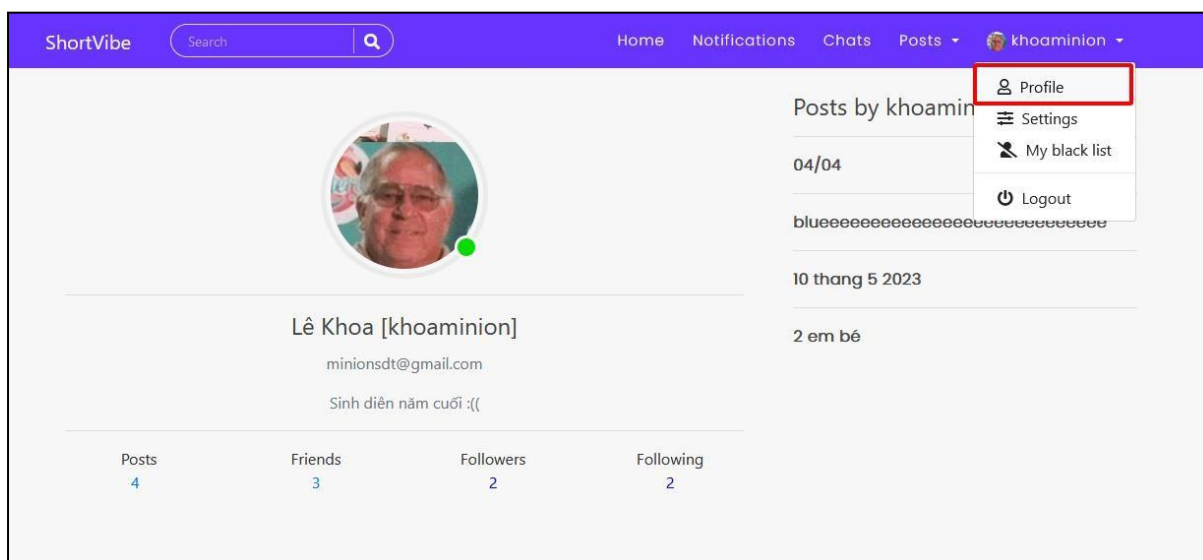


Hình 4.41 Người dùng đã được đưa vào danh sách hạn chế

4.18 Giao diện trang cá nhân

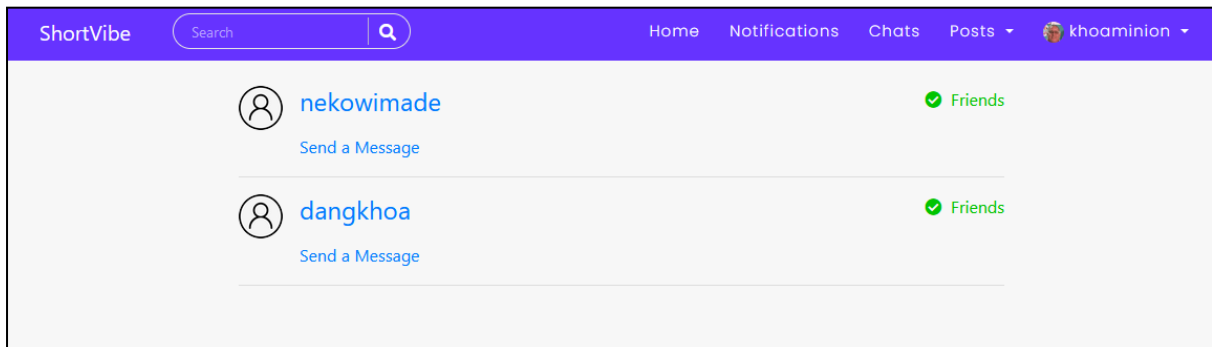
Giao diện trang cá nhân của bản thân người dùng. Bao gồm các thông tin cơ bản như ảnh đại diện, họ và tên kèm theo tên đăng nhập, email, ngày sinh, tiểu sử,...

Phần bên phải liệt kê các bài đăng của bản thân. Có thống kê số bạn bè, những người mà người dùng cùng theo dõi và những người theo dõi người dùng, có thể xem được tất cả những thông tin này.



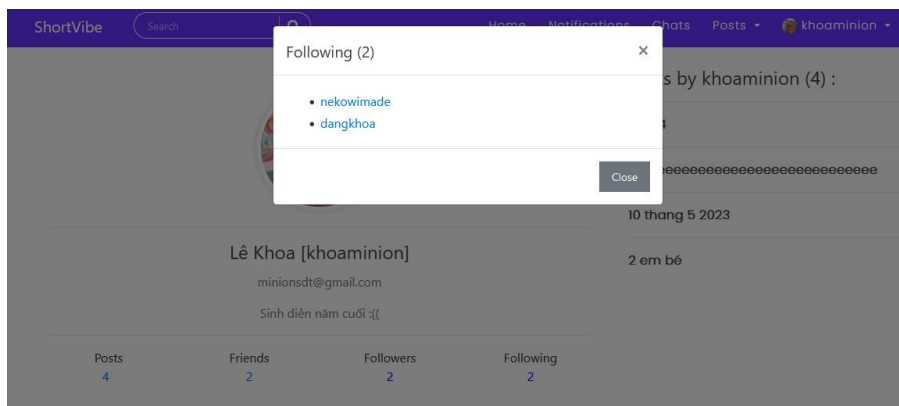
Hình 4.42 Lối tắt đến trang cá nhân của người dùng

Giao diện danh sách bạn bè.



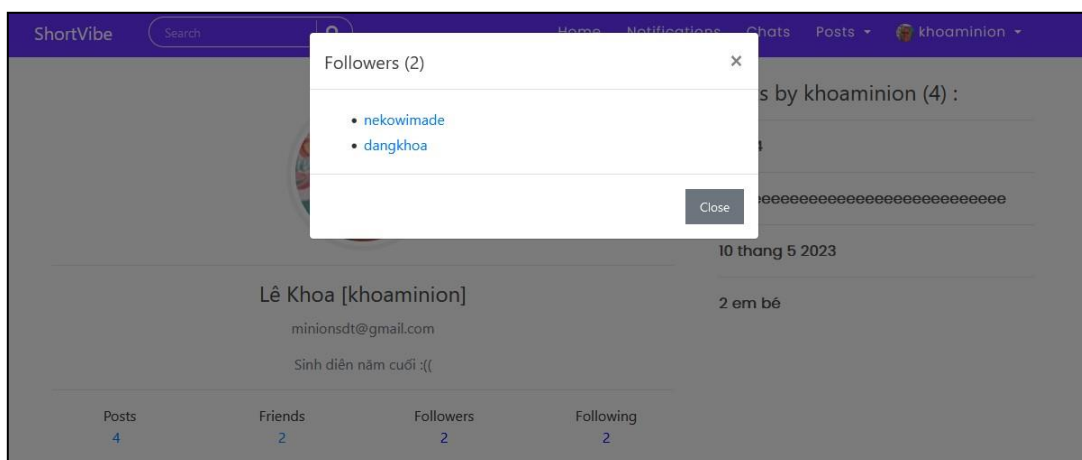
Hình 4.43 Giao diện danh sách bạn bè của bản thân người dùng.

Giao diện những người đang theo dõi.



Hình 4.44 Giao diện danh sách những người đang theo dõi người dùng.

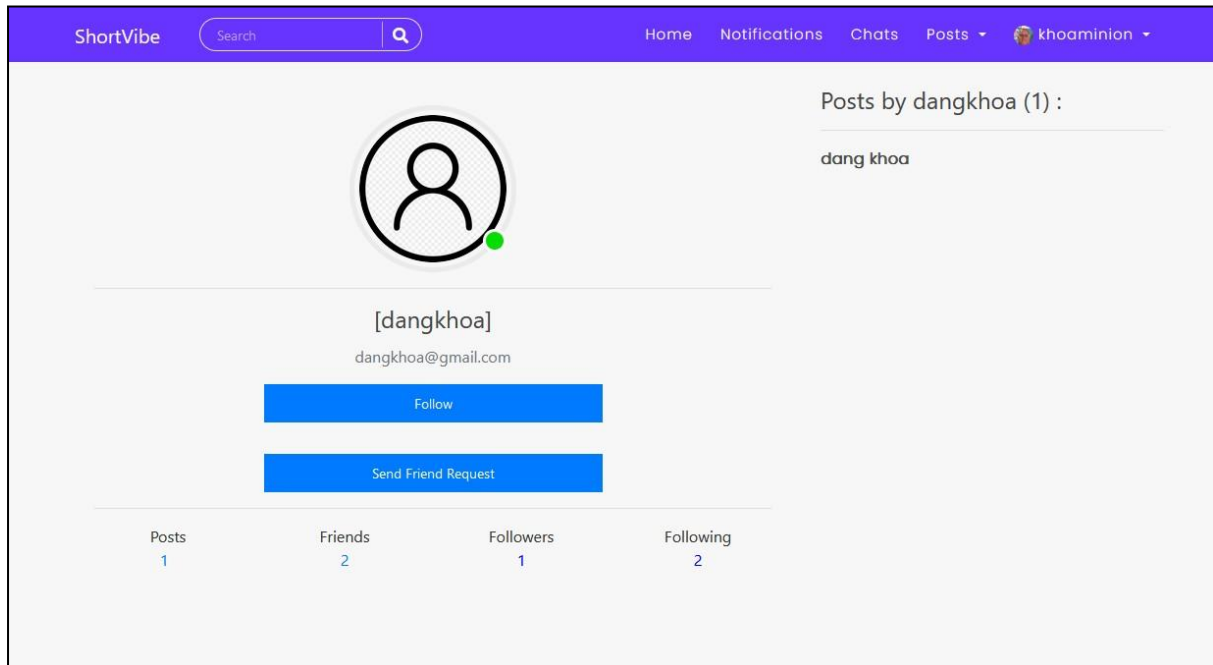
Giao diện những người khác đang theo dõi người dùng.



Hình 4.45 Giao diện danh sách những người mà người dùng đang theo dõi.

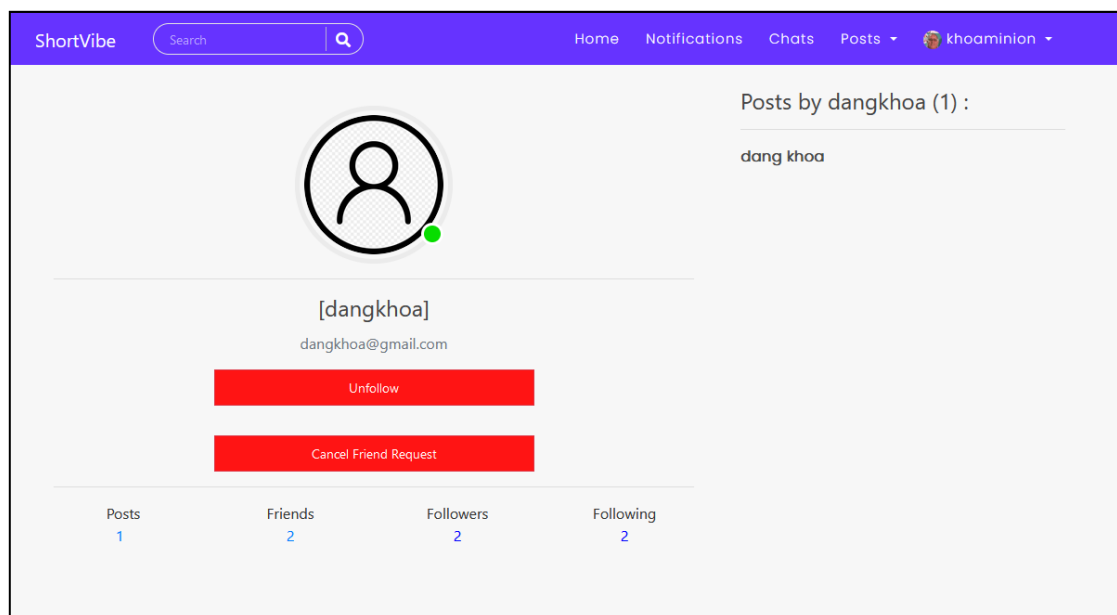
Giao diện trang cá nhân của người dùng khác.

Khi truy cập vào trang cá nhân của người dùng khác, có thể xem thông tin của người dùng đó, theo dõi và bỏ theo dõi người dùng, gửi lời mời kết bạn.



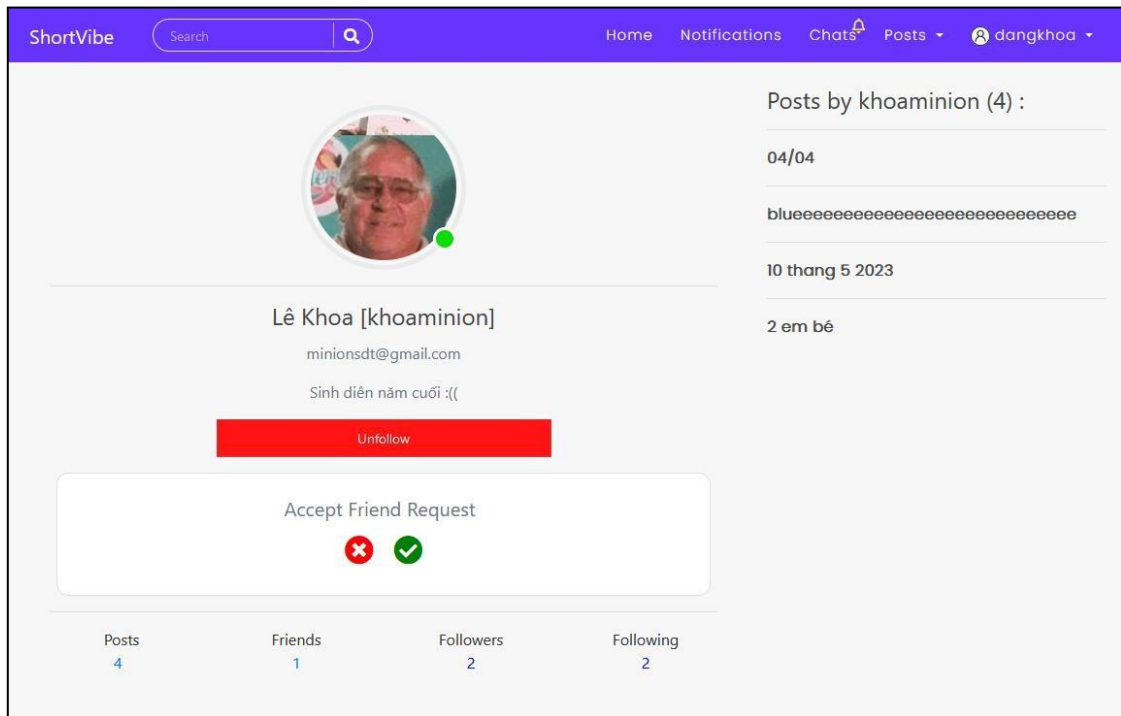
Hình 4.46 Giao diện trang cá nhân của người dùng khác.

Giao diện sau khi đã theo dõi và gửi lời mời kết bạn.



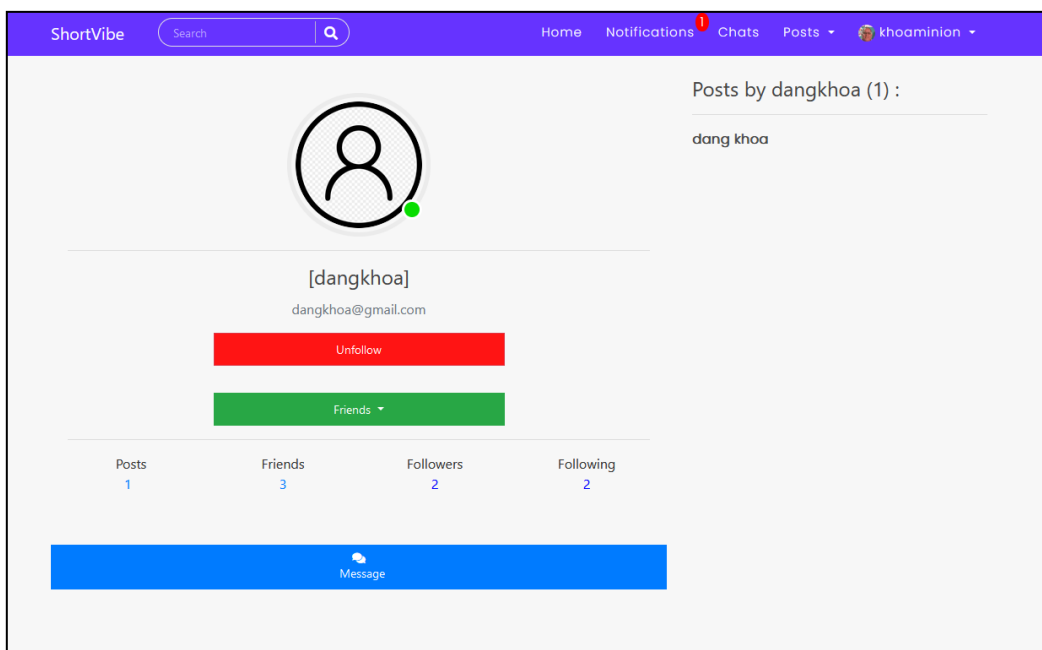
Hình 4.47 Giao diện trang cá nhân của người dùng khác khi đã gửi lời mời và theo dõi.

Giao diện bên phía người dùng nhận được lời mời kết bạn., người nhận được lời mời kết bạn có quyền chấp nhận hoặc từ chối lời mời.



Hình 4.48 giao diện bên phía người dùng nhận được lời mời kết bạn.

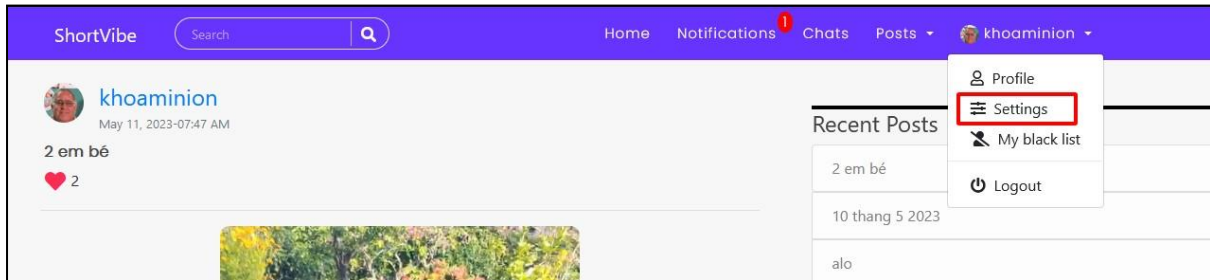
Giao diện trang cá nhân của người dùng khác sau khi đã trở thành bạn bè.



Hình 4.49 Giao diện trang cá nhân của người dùng khác khi cả 2 là bạn bè

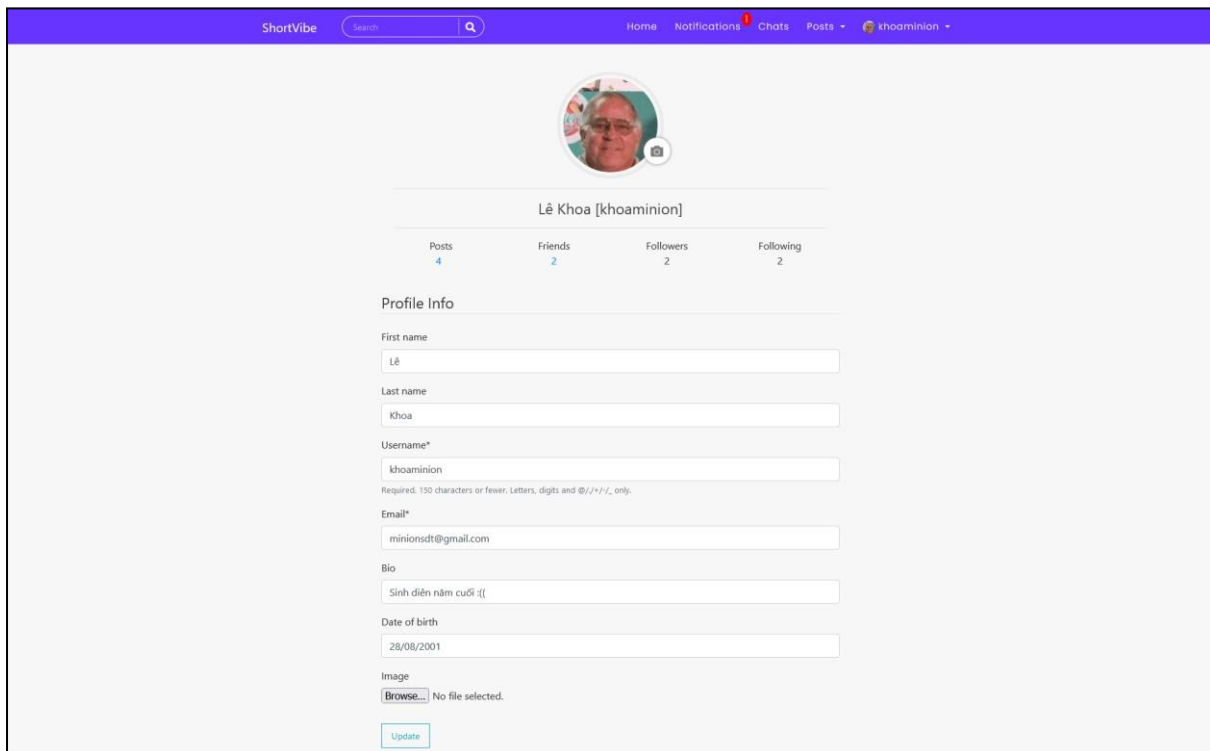
Lúc này sẽ có thêm lựa chọn để có thể tạo hội thoại và trao đổi thông tin với nhau, nếu nhấn vào nút message thì sẽ chuyển đến phòng hội thoại riêng tư của 2 người dùng.

4.19 Giao diện cập nhật thông tin người dùng



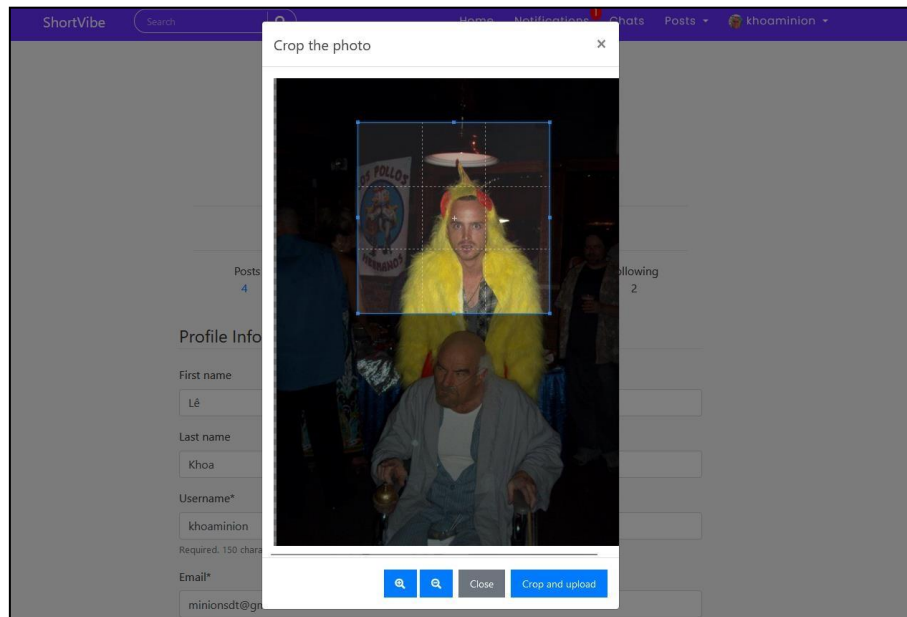
Hình 4.50 lối tắt đến trang quản lý thông tin bản thân người dùng.

Giao diện hiển thị toàn bộ thông tin của người dùng, muốn cập nhật thông tin nào chỉ việc thay đổi thông tin ở ô tương ứng và nhấn update.



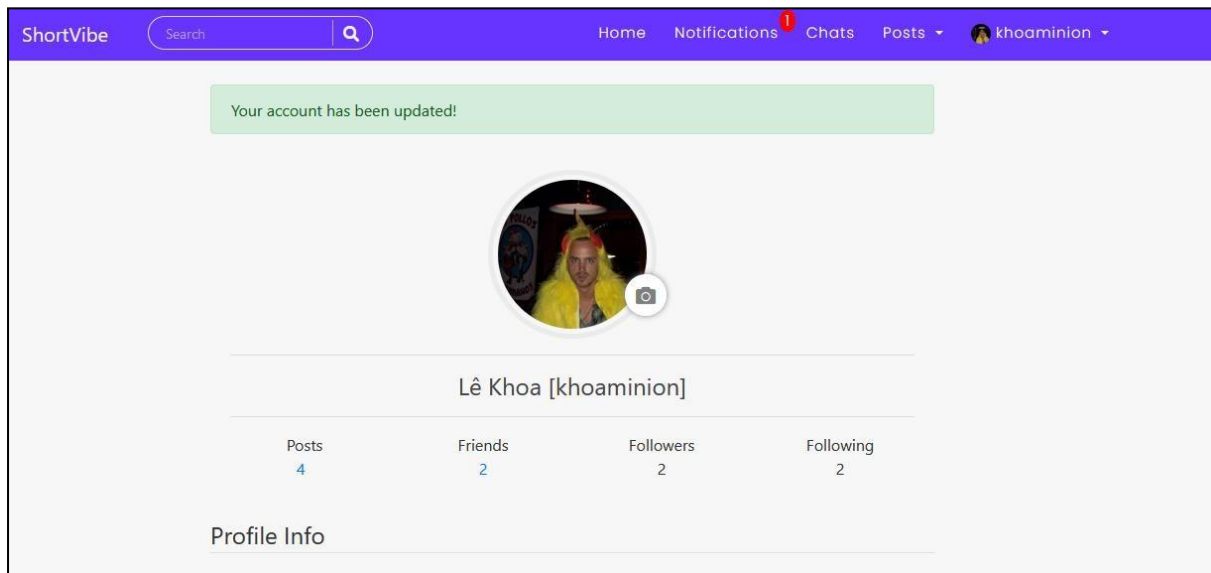
Hình 4.51 Giao diện quản lý thông tin của người dùng.

Cửa sổ để điều chỉnh kích thước ảnh đại diện khi cập nhật.



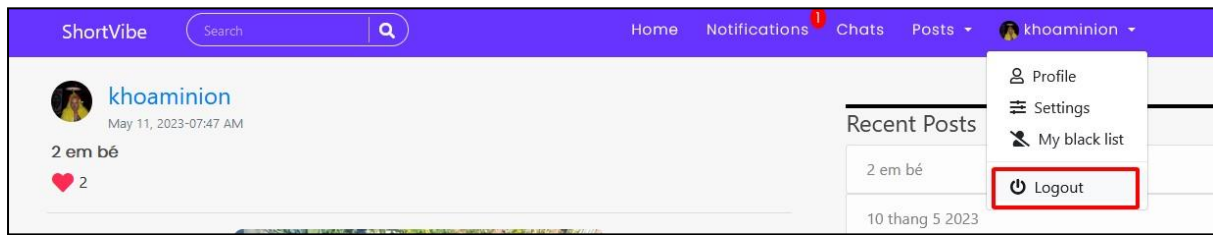
Hình 4.52 Giao diện cửa sổ điều chỉnh kích thước ảnh đại diện.

Thông báo sau khi cập nhật thông tin thành công.



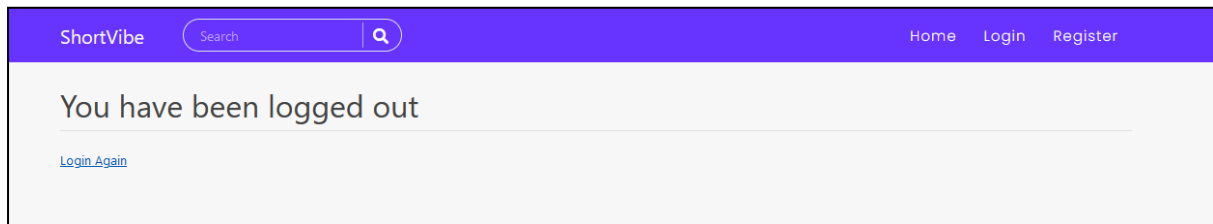
Hình 4.53 Thông báo khi thông tin người dùng được cập nhật thành công.

4.20 Giao diện khi đăng xuất



Hình 4.54 Lối tắt đến chức năng đăng xuất.

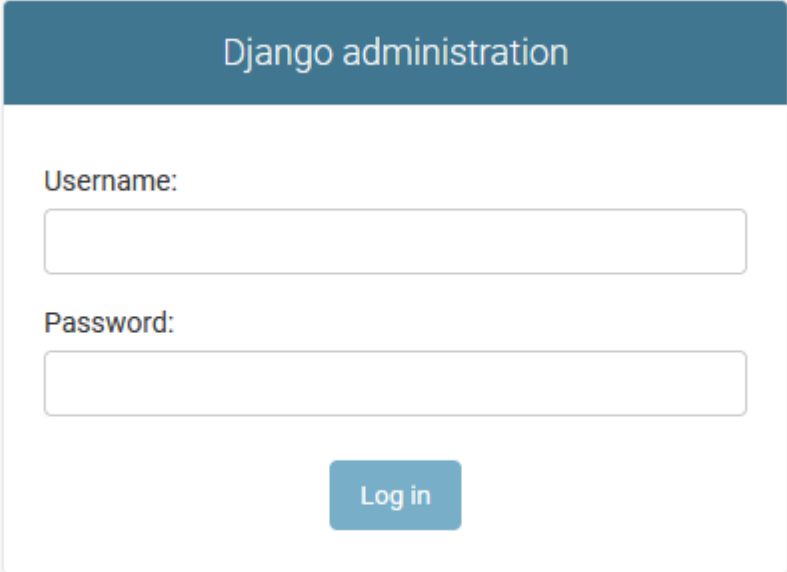
Đăng xuất và chuyển đến màn hình đã đăng xuất.



Hình 4.55 Giao diện xác nhận đã đăng xuất.

4.21 Giao diện người dùng quản trị

Giao diện đăng nhập.

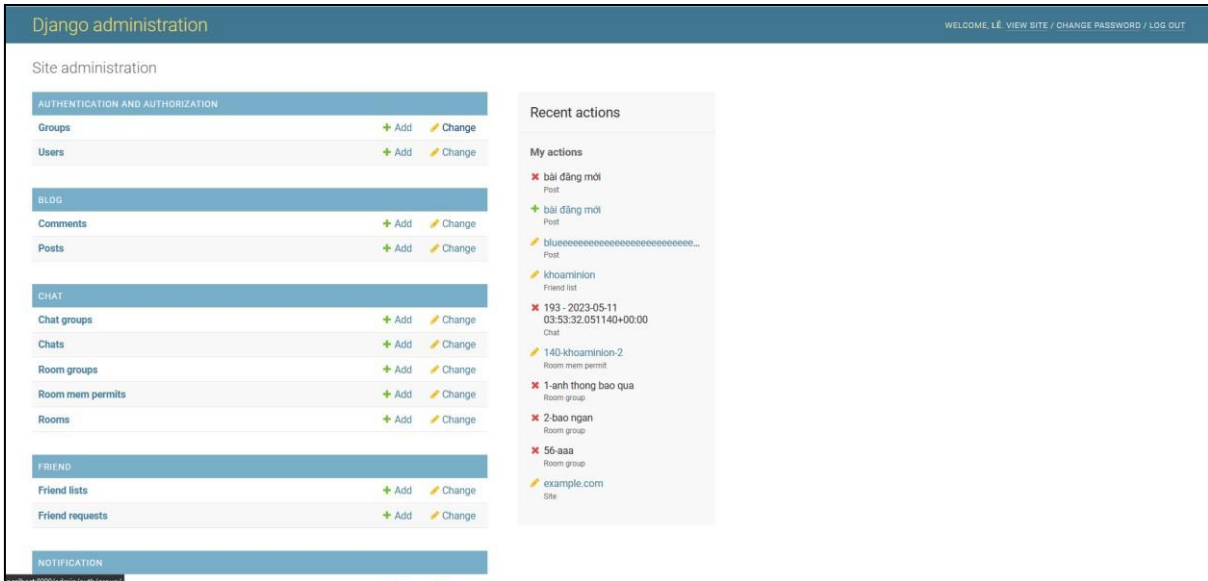


The image shows a screenshot of the Django administration login interface. It features a dark blue header with the text "Django administration". Below the header, there are two input fields: "Username:" and "Password:". A blue "Log in" button is centered below the password field.

Hình 4.56 Giao diện đăng nhập trang quản trị

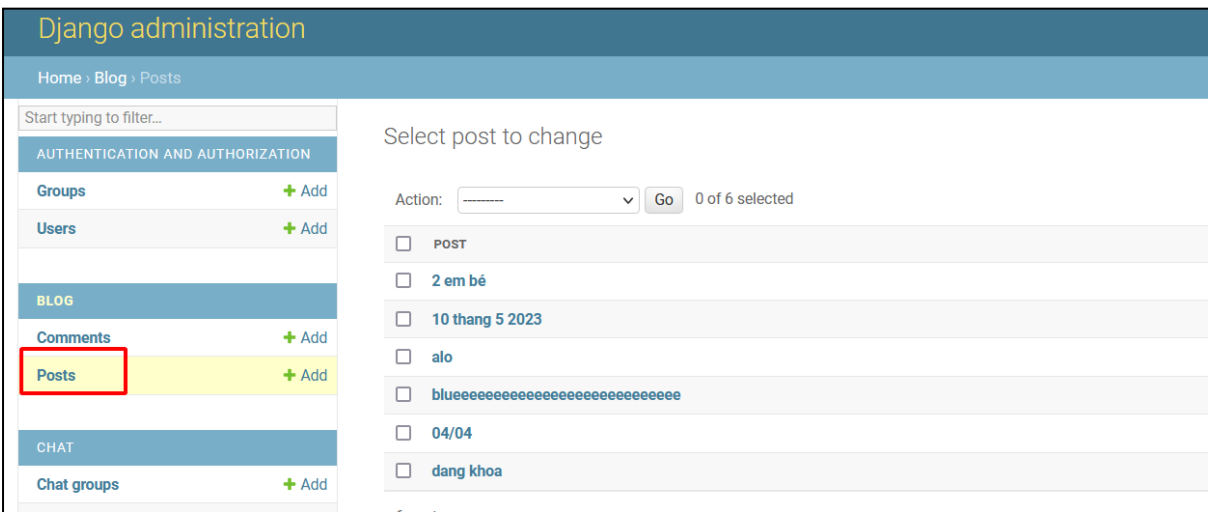
Giao diện quản lý tất cả các đối tượng trong hệ thống. Các đối tượng có trong danh sách này là tất cả những đối tượng có thể tương tác trong hệ thống mà người dùng có thể tiếp cận được. Riêng đối với người quản trị thì sẽ có giao diện đặc biệt để quản lý. Mọi dữ liệu bên trong hệ thống đều dưới quyền kiểm soát của người dùng quản trị.

Lưu ý: Người quản trị phải là người hiểu rõ về hệ thống và có 1 số chuyên môn nhất định để có thể quản trị hệ thống tron tru, tránh được sai sót.



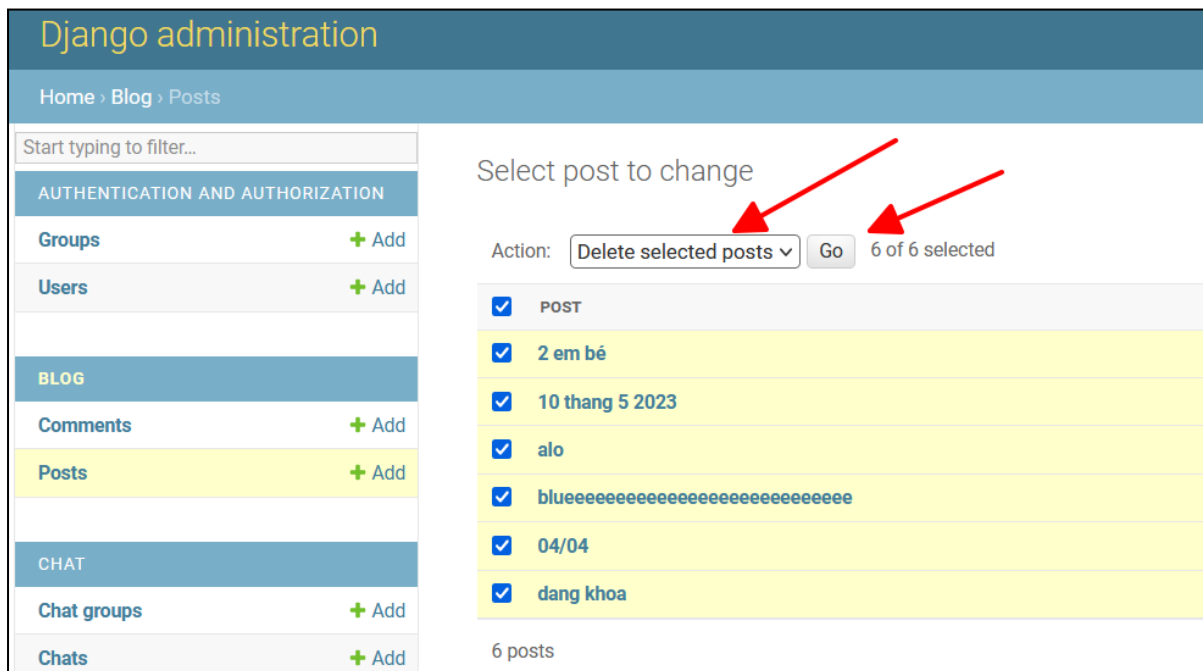
Hình 4.57 Giao diện danh sách các đối tượng có thể quản lý từ trang quản trị

Đại diện giao diện danh sách posts để làm ví dụ, posts sẽ tương ứng với bài đăng trên giao diện người dùng.



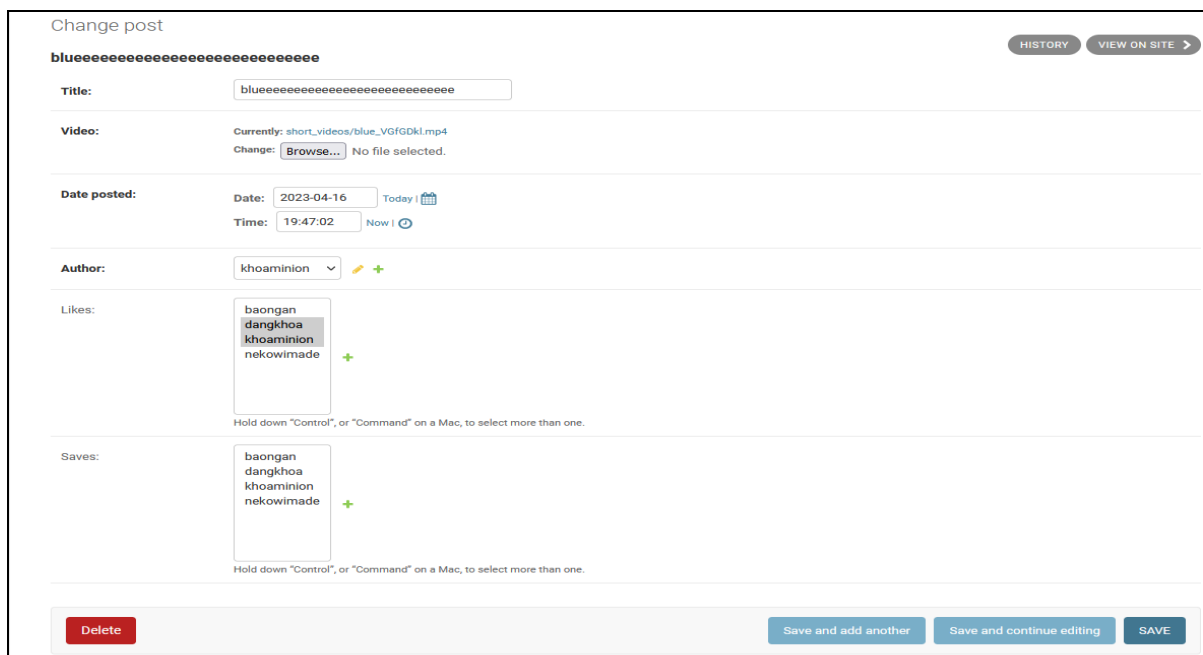
Hình 4.58 Giao diện danh sách các bài đăng được liệt kê trong trang quản trị

Ở màn hình này sẽ có check box và action có thể chọn được là delete, để có thể thực hiện xóa được đồng loạt 6 bài đăng cùng 1 lúc.



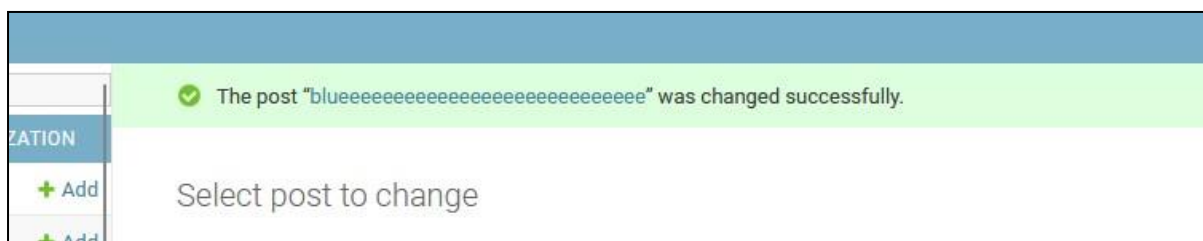
Hình 4.59 Giao diện các bài đăng đã được chọn để thực thi trong trang quản trị.

Khi chọn vào 1 bài đăng bất kỳ thì sẽ chuyển sang màn hình thông tin của bài đăng đó và có thể cập nhật được thông tin cho bài đăng hoặc xóa bài đăng. Ở màn hình này có thể thấy được tất cả thuộc tính có liên quan đến bài đăng như tiêu đề, những người yêu thích, ngày tạo, ... và tất nhiên mọi thông tin đều có thể thay đổi được.



Hình 4.60 giao diện thông tin của bài đăng từ trang quản trị

Sau khi đã cập nhật bài đăng thì sẽ quay lại màn hình danh sách bài đăng kèm theo thông báo thành công.



Hình 4.61 Thông báo cập nhật bài đăng thành công từ trang quản trị.

Muốn tạo bài đăng mới chỉ cần nhấn nút add post từ màn hình danh sách, giao diện sẽ chuyển sang màn hình tạo bài đăng mới.

Ví dụ tạo 1 bài đăng mới với tác giả là “baongan”, là tiêu đề là “bài đăng mới nè”.

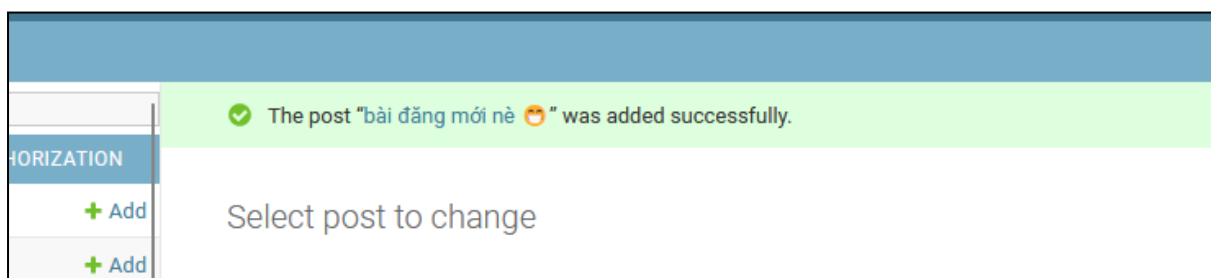
The screenshot shows a form titled "Add post" with the following fields and options:

- Title:** A text input field containing "bài đăng mới nè 🥰".
- Video:** A "Browse..." button followed by the file path "Snaptik.app_72189...182438808837.mp4".
- Date posted:** Two sub-fields: "Date:" with a date picker set to "2023-05-15" and a "Today" button; "Time:" with a time picker set to "23:39:03" and a "Now" button.
- Author:** A dropdown menu set to "baongan" with a plus icon to add more authors.
- Likes:** A list of users: "baongan", "dangkhoa", "khoaminion", "nekowimade" with a plus icon to add more.
- Saves:** A list of users: "baongan", "dangkhoa", "khoaminion", "nekowimade" with a plus icon to add more.

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

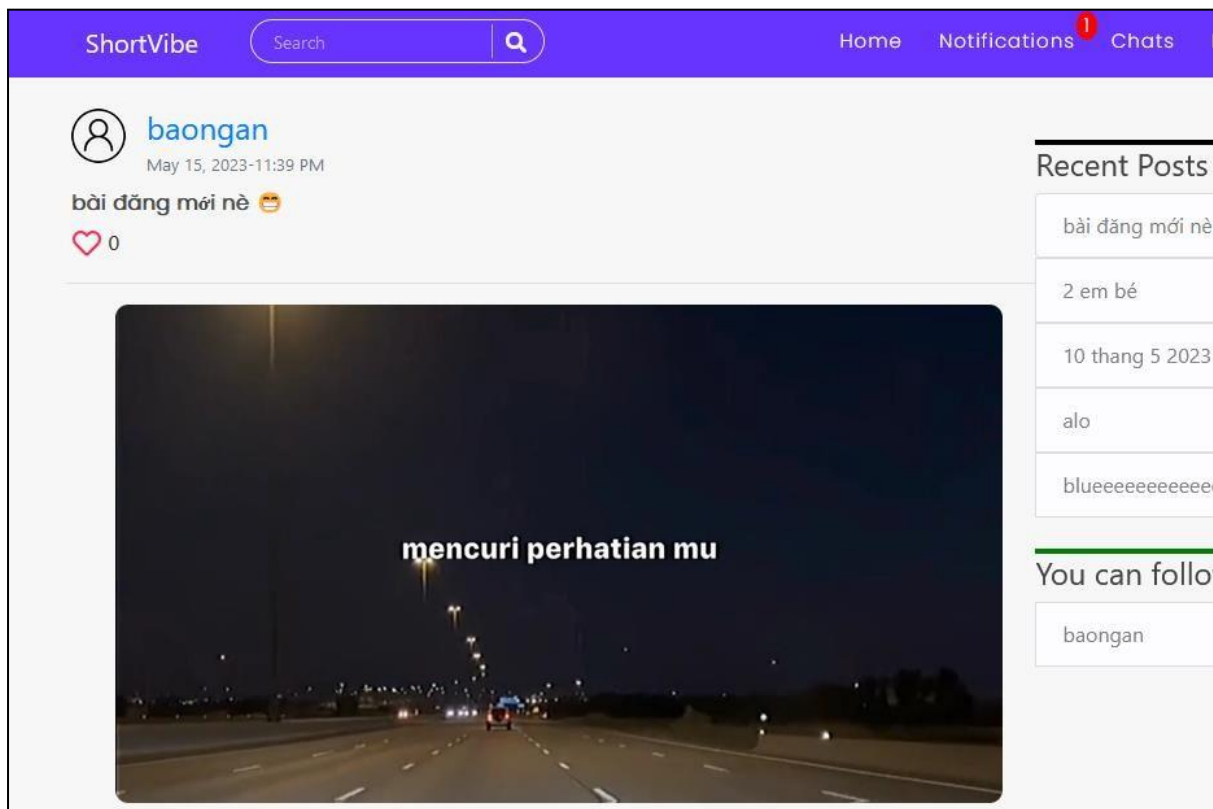
Hình 4.62 Giao diện tạo mới bài đăng từ trang quản trị

Thông báo thành công.



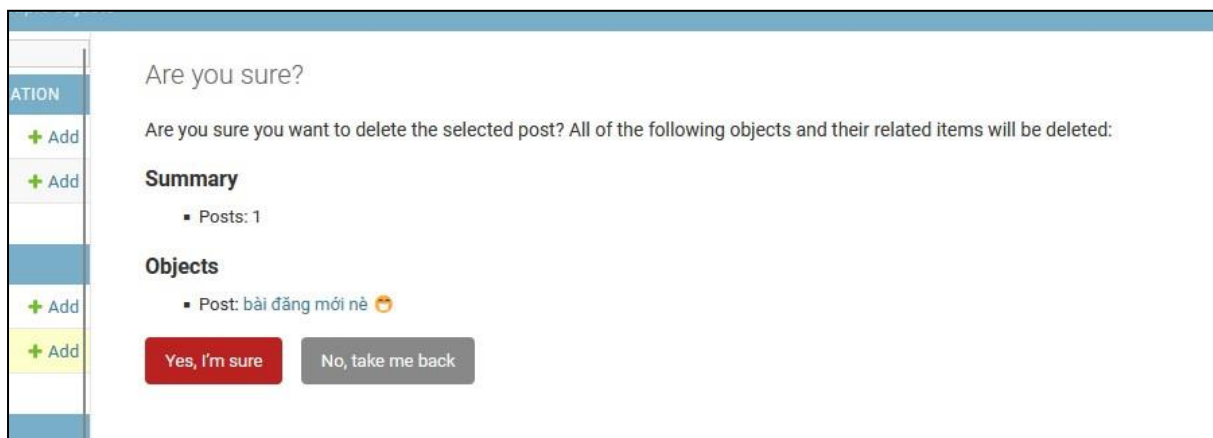
Hình 4.63 Thông báo thêm bài đăng thành công từ trang quản trị

Kiểm tra bên giao diện người dùng.



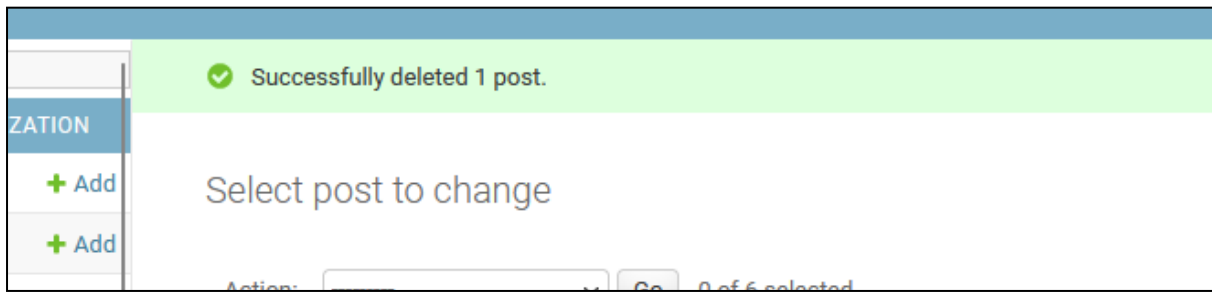
Hình 4.64 Kiểm tra bài đăng vừa thêm từ trang quản trị trên giao diện người dùng

Xóa bài đăng vừa tạo.



Hình 4.65 Tiến hành xóa bài đăng từ trang quản trị.

Thông báo thành công.



Hình 4.66 Thông báo xóa bài đăng thành công từ trang quản trị.

Tương ứng với ví dụ sử dụng bài đăng, mọi đối tượng được liệt kê trong giao diện quản trị đều có thể kiểm soát được với những thao tác tương tự.

CHƯƠNG 5 KẾT LUẬN

5.1 Kết quả đạt được

- Vận dụng, áp dụng được những kiến thức đã học để xây dựng được đồ án.
- Khai thác được các công nghệ đã tìm hiểu và đưa được vào đồ án.
- Nâng cao được khả năng tự tìm tòi và nghiên cứu thông tin, quá trình thực hiện đồ án luôn yêu cầu bản thân phải tìm kiếm thông tin dữ liệu thật đầy đủ và chính xác.
- Nâng cao khả năng viết và trình bày, như cấu trúc câu, định dạng văn bản,...
- Có thêm kiến thức về chuyên môn.
- Xây dựng được mối quan hệ khi làm đồ án, quá trình thực hiện đồ án yêu cầu trao đổi và làm việc giữa sinh viên với giảng viên và giữa sinh viên với sinh viên.
- Xây dựng được ứng dụng với giao diện trực quan, bắt mắt, dễ dàng thao tác và sử dụng.
- Xây dựng được cơ sở dữ liệu có tính toàn vẹn và chính xác.
- Nắm bắt được tâm lý người dùng và tạo ra ứng dụng mang lại trải nghiệm tốt nhất có thể.
- Hoàn hiện được phần lớn chức năng đã phân tích trong đồ án.

5.2 Hướng phát triển

- Nâng cấp cho giao diện người dùng thêm sinh động hơn, đẹp hơn và sắp xếp giao diện hợp lý để nâng tầm trải nghiệm.
- Hoàn thiện các chức năng còn dang dở.
- Phát triển thêm nhóm bài đăng để có thể chia sẻ bài đăng trong 1 nhóm kín.
- Phát triển chức năng phân loại cho bài đăng để có thể sắp xếp và phân chia theo loại, từ đó phát triển thêm tính năng gợi ý cho người dùng xem những video theo sở thích của người dùng.
- Phát triển chức năng báo cáo bài đăng, giúp người quản trị có thể kiểm soát được và loại bỏ những nội dung độc hại, vi phạm các tiêu chuẩn cộng đồng,...
- Cập nhật lại hội thoại để có thể gửi được thêm các định dạng mới như hình ảnh, tệp âm thanh, ...
- Rà soát kiểm tra các lỗ hổng, áp dụng các công nghệ mới để tăng tính bảo mật cho hệ thống.

Do lượng kiến thức còn hạn chế, cũng như hạn chế về mặt thời gian nên việc mắc phải những sai sót trong bài báo cáo này là việc không thể tránh khỏi. Em xin ghi nhận mọi phê bình cũng như góp ý của quý thầy cô và Hội đồng bảo vệ đồ án trường Đại học Bà Rịa – Vũng Tàu.

Em xin chân thành cảm ơn quý thầy cô trong Hội đồng bảo vệ, thầy cô trong khoa Kỹ thuật – Công nghệ, ban giám hiệu nhà trường và đặc biệt là lời cảm ơn chân thành nhất dành cho thầy Phan Ngọc Hoàng, người trực tiếp hướng dẫn em, đã không tiếc thời gian quý báu để chỉ dạy, đưa ra gợi ý, những hướng đi tốt nhất để em hoàn thiện được đồ án này.

TÀI LIỆU THAM KHẢO

[1] cách sử dụng Ajax

https://www.w3schools.com/jquery/jquery_ajax_get_post.asp

[2] cách sử dụng Django-Channels

<https://channels.readthedocs.io/en/stable/>

[3]Documentation Django

<https://docs.djangoproject.com/en/4.2/>

[4] Tham khảo cách tạo thông báo theo thời gian thực bằng Django Channels

<https://auberginesolutions.com/blog/django-channels-real-time-notifications/>

[5]cách cài đặt và chạy redis trên WSL(Windows subsystem for linux)

<https://redis.io/docs/getting-started/st>

[6] Tham khảo cách fix lỗi trên stackoverflow

<https://stackoverflow.com/>

[7] tham khảo các video về ứng dụng bffbook trên kênh Pyplane

https://www.youtube.com/watch?v=ozr6NEomLQw&list=PLgJw1dR712joFJvX_WKlUglbR1SNCeno1&index=1

PHỤ LỤC

Cấu trúc thư mục

Thư mục Root

	<p>blog: quản lý app blog</p> <p>chat: quản lý app chat</p> <p>friend: quản lý app friend</p> <p>media: chứa file upload của project</p> <p>myproject: quản lý project</p> <p>notification: quản lý app notification</p> <p>users: quản lý app users</p> <p>.env: chứa các biến người dùng định nghĩa</p> <p>Manage.py: file quản lý project</p> <p>Requirements.txt: chứa danh sách thư viện</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Bảng thư mục 1 Thư mục root

Thư mục myproject

	<p>__pycache__: hệ thống tạo để phân biệt module.</p> <p>__init__.py: hệ thống tạo để phân biệt module.</p> <p>asgi.py: file cấu hình asgi cho project</p> <p>Wsgi.py: file cấu hình wsgi cho project</p> <p>Routing.py: khai báo URL của channels.</p> <p>settings.py: khai báo các hàm kiểm tra.</p> <p>Urls.py: khai báo các URL của project.</p> <p>vali.py: định nghĩa module mới.</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

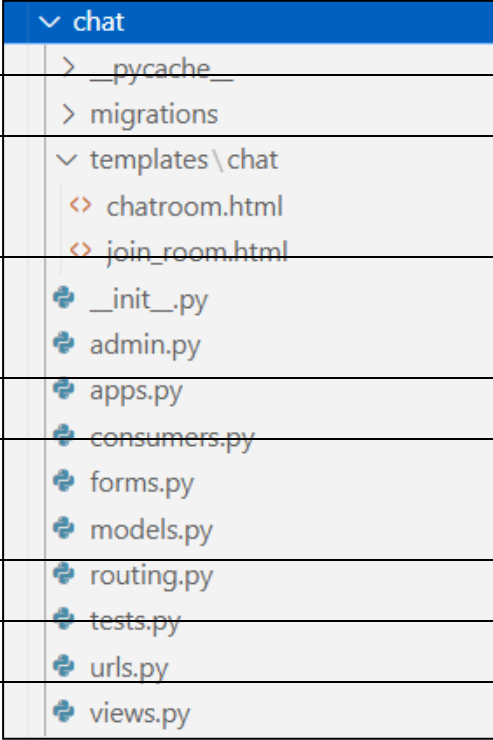
Bảng thư mục 2 Thư mục myprojetc

Thư mục blog

	__pycache__ : hệ thống tạo để phân biệt module.
	Migrations: quản lý các phiên bản cấu trúc database.
<ul style="list-style-type: none"> ▼ blog 	
<ul style="list-style-type: none"> > __pycache__ 	Static: chứa các file tĩnh.
<ul style="list-style-type: none"> > migrations 	Images: chứa các file ảnh.
<ul style="list-style-type: none"> ▼ static\blog 	
<ul style="list-style-type: none"> > images 	Js: chứa các file javascript.
<ul style="list-style-type: none"> > js 	Templates: chứa file HTML.
<ul style="list-style-type: none"> # bootstrap.min.css # main.css # vc.css 	__init__.py: hệ thống tạo để phân biệt module.
<ul style="list-style-type: none"> > templates 	admin.py: file chứa các hàm định dạng trang quản lý.
<ul style="list-style-type: none"> 🔗 __init__.py 🔗 admin.py 🔗 apps.py 🔗 consumers.py 🔗 forms.py 	apps.py: khai báo app.
<ul style="list-style-type: none"> 🔗 models.py 	consumers.py: khai báo các hàm xử lý sự kiện channels (hỗ trợ realtime).
<ul style="list-style-type: none"> 🔗 routing.py 🔗 tests.py 🔗 urls.py 	Forms.py: định nghĩa các form.
<ul style="list-style-type: none"> 🔗 routing.py 🔗 tests.py 🔗 urls.py 	Models.py: khai báo các đối tượng database.
<ul style="list-style-type: none"> 🔗 routing.py 🔗 tests.py 🔗 urls.py 	Routing.py: khai báo URL của channels.
<ul style="list-style-type: none"> 🔗 routing.py 🔗 tests.py 🔗 urls.py 	Test.py: khai báo các hàm kiểm tra.
	Urls.py: khai báo các URL của app.
	Utils.py: hỗ trợ xử lý các thư viện khác.
	Views.py: khai báo các hàm xử lý giao diện

Bảng thư mục 3 Thư mục blog

Thư mục chat

	__pycache__: hệ thống tạo để phân biệt module.
	Migrations: quản lý các phiên bản cấu trúc database.
	Templates: chứa file HTML.
	__init__.py: hệ thống tạo để phân biệt module.
	admin.py: file chứa các hàm định dạng trang quản lý.
	apps.py: khai báo app.
	consumers.py: khai báo các hàm xử lý sự kiện channels (hỗ trợ realtime).
	Forms.py: định nghĩa các form.
	Models.py: khai báo các đối tượng database.
	Routing.py: khai báo URL của channels.
	Test.py: khai báo các hàm kiểm tra.
	Urls.py: khai báo các URL của app.
	Views.py: khai báo các hàm xử lý giao diện

Bảng thư mục 4 Thư mục chat

Thư mục friend

	__pycache__: hệ thống tạo để phân biệt module.
<ul style="list-style-type: none"> ▼ friend > __pycache__ > migrations ▼ templates\friend <ul style="list-style-type: none"> > snippets <> friend_list.html <> friend_requests.html 🔗 __init__.py 🔗 admin.py 🔗 apps.py 🔗 friend_request_status.py 🔗 models.py 🔗 tests.py 🔗 urls.py 🔗 utils.py 🔗 views.py 	<p>Migrations: quản lý các phiên bản cấu trúc database.</p> <p>Templates: chứa file HTML.</p> <p>__init__.py: hệ thống tạo để phân biệt module.</p> <p>admin.py: file chứa các hàm định dạng trang quản lý.</p> <p>apps.py: khai báo app.</p> <p>Friend_request_status.py: chứa các biến của lời mời kết bạn.</p> <p>Models.py: khai báo các đối tượng database.</p> <p>Test.py: khai báo các hàm kiểm tra.</p> <p>Urls.py: khai báo các URL của app.</p> <p>Views.py: khai báo các hàm xử lý giao diện.</p> <p>Utils.py: định nghĩa module mới.</p>

Bảng thư mục 5 Thư mục friend

Thư mục notification

<ul style="list-style-type: none"> ▼ notification > __pycache__ > migrations 🔗 __init__.py 🔗 admin.py 🔗 apps.py 🔗 models.py 🔗 tests.py 🔗 urls.py 🔗 views.py 	<p>__pycache__: hệ thống tạo để phân biệt module.</p> <p>Migrations: quản lý các phiên bản cấu trúc database.</p> <p>__init__.py: hệ thống tạo để phân biệt module.</p> <p>admin.py: file chứa các hàm định dạng trang quản lý.</p> <p>apps.py: khai báo app.</p> <p>Models.py: khai báo các đối tượng database.</p> <p>Test.py: khai báo các hàm kiểm tra.</p> <p>Urls.py: khai báo các URL của app.</p> <p>Views.py: khai báo các hàm xử lý giao diện.</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Bảng thư mục 6 Thư mục notification

Thư mục users

<ul style="list-style-type: none"> ▼ users > __pycache__ > migrations > templates 🔗 __init__.py 🔗 admin.py 🔗 apps.py 🔗 forms.py 🔗 models.py 🔗 tests.py 🔗 urls.py 🔗 views.py 	<p>__pycache__: hệ thống tạo để phân biệt module.</p> <p>Migrations: quản lý các phiên bản cấu trúc database.</p> <p>Templates: chứa file HTML.</p> <p>__init__.py: hệ thống tạo để phân biệt module.</p> <p>admin.py: file chứa các hàm định dạng trang quản lý.</p> <p>apps.py: khai báo app.</p> <p>Forms.py: định nghĩa các form.</p> <p>Models.py: khai báo các đối tượng database.</p> <p>Test.py: khai báo các hàm kiểm tra.</p> <p>Urls.py: khai báo các URL của app.</p> <p>Views.py: khai báo các hàm xử lý giao diện.</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Bảng thư mục 7 Thư mục users