

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



NGUYỄN THIÊN PHÚC

**XÂY DỰNG HỆ THỐNG QUẢN LÝ BÀI VIẾT MẠNG
XÃ HỘI DỰA TRÊN REACTJS**

ĐỒ ÁN

NGÀNH CÔNG NGHỆ THÔNG TIN

TP. HỒ CHÍ MINH, 2021

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



NGUYỄN THIÊN PHÚC

XÂY DỰNG HỆ THỐNG QUẢN LÝ BÀI VIẾT MẠNG
XÃ HỘI DỰA TRÊN REACTJS

Mã số sinh viên: 1851050115

ĐỒ ÁN
NGÀNH CÔNG NGHỆ THÔNG TIN

Giảng viên hướng dẫn: DƯƠNG HỮU THÀNH

TP. HỒ CHÍ MINH, 2021

TRƯỜNG ĐẠI HỌC MỞ

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

THÀNH PHỐ HỒ CHÍ MINH

Độc lập – Tự do – Hạnh phúc

KHOA CÔNG NGHỆ THÔNG TIN

GIẤY XÁC NHẬN

Tôi tên là: Nguyễn Thiên Phúc.....

Ngày sinh: 27-07-2000 Nơi sinh: Thành phố Hồ Chí Minh.....

Chuyên ngành: Công nghệ thông tin Mã sinh viên: 1851050115

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

Ký tên

(Ghi rõ họ và tên)

Phúc

Nguyễn Thiên Phúc

**Ý KIẾN CHO PHÉP BẢO VỆ ĐỒ ÁN TỐT NGHIỆP
CỦA GIẢNG VIÊN HƯỚNG DẪN**

Giảng viên hướng dẫn:

Sinh viên thực hiện: **Lớp:**

Ngày sinh: **Nơi sinh:**

Tên đề tài:

.....

.....

.....

.....

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

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

Người nhận xét

Dương Hữu Thành

LỜI CẢM ƠN

Trong khoảng thời gian được học tập tại trường Đại học Mở Thành phố Hồ Chí Minh. Dưới sự dẫn dắt của các giảng viên trong khoa Công nghệ thông tin, bên cạnh đó các bạn sinh viên cùng khóa cũng đã hỗ trợ em trong suốt quá trình học tại trường. Sau đây em xin gửi lời cảm ơn chân thành của mình đến Ban giám hiệu nhà trường Đại học Mở thành phố Hồ Chí Minh nói chung và Ban chủ nhiệm khoa công nghệ thông tin nói riêng lời cảm ơn chân thành nhất.

Quan trọng hơn hết, em xin gửi lời cảm ơn đến thầy ThS. Dương Hữu Thành, cảm ơn thầy vì đã tận tụy và luôn nhiệt tình hỗ trợ chúng em trong kì làm đồ án này cũng như những môn học từ trước đến nay để cho sinh viên tại em có một nền tảng kiến thức vững chắc nhất. Bên cạnh đó, em xin cảm ơn gia đình cũng như bạn bè đã tạo cho em một nguồn động lực và luôn giúp đỡ em để em có thể phát triển bản thân qua từng ngày.

Do trình độ chuyên môn còn ở mức cơ bản nên trong quá trình làm đồ án chắc chắn em sẽ có những sai sót nhất định. Vì lí do đó nên em hy vọng mình sẽ nhận được lời nhận xét của Quý thầy cô để em có thể làm tốt đồ án ngành của mình một cách tốt nhất có thể trong khả năng.

Lời cuối, em xin chân thành cảm ơn ngôi trường Đại học Mở thành phố Hồ Chí Minh đã giúp cho em và các bạn sinh viên khác có một kì thực hiện đồ án có lẽ là đáng nhớ trong thời kì covid này và một trải nghiệm học tập thật tốt trong khoảng thời gian được là sinh viên của trường.

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TÓM TẮT ĐỒ ÁN NGÀNH

Mạng xã hội từ thiện Charity PT là một website về mạng xã hội nhưng chủ yếu hoạt động riêng về lĩnh vực từ thiện. Người dùng cần phải đăng kí những thông tin cơ bản của mình để tham gia vào cộng đồng này. Đây là một trang mạng xã hội chủ yếu hoạt động về lĩnh vực từ thiện. Mọi người ở bất cứ khu vực hay ngành nghề nào cũng đều có thể tham gia hoạt động từ thiện của mình nhưng lại không làm mất quá nhiều thời gian cho hành động ý nghĩa này. Tuy vậy, mạng xã hội từ thiện vẫn có các chức năng cơ bản để phục cho người dùng như: đăng tải bài viết, thích và bình luận bài viết. Tuy nhiên, user có thể đăng vật phẩm đấu giá trên bài viết của mình. Sau khi bài viết được đấu giá thành công thì sẽ thanh toán qua Momo. Admin có thể xem thống kê số lượng bài viết cũng như lượt like, bình luận trên mạng xã hội.

MỤC LỤC

LỜI CẢM ƠN	3
NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN.....	4
TÓM TẮT ĐỒ ÁN NGÀNH.....	5
DANH MỤC TỪ VIẾT TẮT	9
DANH MỤC HÌNH VẼ	10
DANH MỤC BẢNG	11
MỞ ĐẦU.....	12
 Chương 1. GIỚI THIỆU ĐỀ TÀI.....	13
1.1. Giới thiệu và lí do chọn đề tài	13
1.2. Mục tiêu.....	14
1.3. Phạm vi và đối tượng nghiên cứu	14
1.3.1. Phạm vi.....	14
1.3.2. Đối tượng nghiên cứu.....	14
1.4. Bố cục đề tài.....	15
 Chương 2. CƠ SỞ LÝ THUYẾT	16
2.1. Giới thiệu ReactJS.....	16
2.1.1. ReactJS là gì	16
2.1.2. Tại sao nên sử dụng React để xây dựng website.....	16
2.1.3. Cài đặt môi trường.....	17
2.2. Kiến thức cơ bản	20
2.2.1. Khái niệm Component.....	20
2.2.2. Khái niệm JSX.....	21
2.2.3. Khái niệm Props	21
2.2.4. Khái niệm State	22
2.2.5. So sánh Props và State.....	23

2.3.	Vòng đời ReactJS	23
2.3.1.	Mounting	24
2.3.2.	Updating	24
2.3.3.	Unmounting	24
2.4.	Virtual DOM	24
2.5.	React Hook	25
2.5.1.	Hook State	26
2.5.2.	Hook Effect	26
2.6.	React Bootstrap	27
2.7.	Material UI (thư viện component UI)	28
2.8.	React Event	28
2.9.	React Router	29
2.10.	React Form	31
2.11.	React Context	33
2.12.	React Redux	34
2.13.	React Cookie	36
2.14.	React Axios	37
2.15.	Python Django	39
Chương 3.	HỆ THỐNG MẠNG XÃ HỘI TỪ THIỆN	40
3.1.	Xác định yêu cầu	40
3.2.	Phân tích yêu cầu	40
3.3.	Phương pháp triển khai	41
3.4.	Các bước triển khai đề tài	41
3.5.	Thiết kế CSDL	42
3.6.	Sơ đồ Use case	43
3.7.	Các API của server	44

Chương 4.	GIAO DIỆN VÀ CHỨC NĂNG MẠNG XÃ HỘI TỪ THIỆN	45
4.1.	Đăng nhập (Login)	45
4.2.	Đăng kí (Register)	46
4.3.	Bảng tin (Newsfeed)	46
4.4.	Bài viết (Post).....	48
4.5.	Trang cá nhân (Timeline).....	50
4.6.	Chi tiết bài viết (Post detail).....	49
4.7.	Thông tin User (About User)	50
4.8.	Giao diện hệ thống bên Python Django	51
Chương 5.	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	52
5.1.	Mục tiêu đạt được.....	52
5.2.	Hướng phát triển	52
TÀI LIỆU THAM KHẢO		53

DANH MỤC TỪ VIẾT TẮT

STT	Từ viết tắt	Tên đầy đủ	Ý nghĩa
1	HTML	Hypertext Markup Language.	Ngôn ngữ đánh dấu siêu văn bản.
2	CSS	Cascading Style Sheet.	Ngôn ngữ thiết kế giao diện cho HTML.
3	JS	Javascript.	Là một ngôn ngữ lập trình dành cho HTML và web.
4	JSX	Javascript – Extensible Markup Language.	Ngôn ngữ dùng để đánh dấu sự mở rộng.
5	UI	User Interface.	Giao diện người dùng.
6	DOM	Document Object Model.	Mô hình đối tượng tài liệu.
7	API	Application Programming Interface.	Cách thức trung gian dùng để kết nối ứng dụng và các thư viện khác nhau.
8	HTTP	Hypertext Transfer Protocol.	Giao thức truyền tải siêu văn bản.
9	CSDL	Cơ sở dữ liệu (Database).	Hệ thống dùng để lưu trữ thông tin.
10	CNTT	Công nghệ thông tin	

DANH MỤC HÌNH VẼ

Hình 2.1: Các phiên bản NodeJS.....	17
Hình 2.2: Tạo project ReactJS thành công.	18
Hình 2.3: Chạy project ReactJS thành công.....	19
Hình 2.4: Vòng đời ReactJS.	23
Hình 2.5: So sánh virtual DOM và Real DOM.	25
Hình 2.6: Event Handler.....	29
Hình 2.7: Cách hoạt động của Redux.....	35
Hình 2.8: Mô hình hoạt động của Django (MVT).	39
Hình 3.1: Sơ đồ UML của hệ thống.	42
Hình 3.2: Sơ đồ Use case.....	43
Hình 3.3: Các API của của python django.	44
Hình 4.1: Giao diện trang đăng nhập.....	45
Hình 4.2: Giao diện chức năng đăng kí.....	46
Hình 4.3: Giao diện trang Newsfeed.	47
Hình 4.4: Giao diện tạo bài viết mới.	47
Hình 4.5: Hiển thị bài viết của User.	48
Hình 4.6: Giao diện chi tiết bài viết.	49
Hình 4.7: Giao diện trang timeline.....	50
Hình 4.8: Giao diện trang thông tin cá nhân.	51
Hình 4.9: Giao diện admin phía backend.	51

DANH MỤC BẢNG

Bảng 2.1: So sánh props và state.	23
--	----

MỞ ĐẦU

Với sự phát triển hiện đại của lĩnh vực công nghệ trong thời đại 4.0 như hiện nay. Con người đã tiếp xúc với Internet khi còn rất trẻ. Vô hình chung con người gắn bó với môi trường ảo ngày càng nhiều. Đó không đơn giản như là cuộc cách mạng về công nghệ mà còn liên quan đến cách thức sử dụng dẫn đến mọi người tham gia vào mạng xã hội ảo ngày càng nhiều. Mạng xã hội đã và đang là một thế giới ảo không thể thiếu trong cuộc sống hiện đại của chúng ta. Nó giúp cho con người có thể tương tác với nhau một cách dễ dàng, giúp cho mọi người có thể chia sẻ sở thích và suy nghĩ của mình. Vì thế, mạng xã hội đã trở thành một ứng dụng có tác dụng dùng để lưu trữ những kỉ niệm khác nhau trong cuộc sống riêng của mỗi con người.

Sức hấp dẫn của mạng xã hội rất mạnh mẽ. Hầu như ảnh hưởng đến mọi lứa tuổi và mọi ngành nghề. Con người đang dần chuyển đời sống của mình Internet. Vì thế các doanh nghiệp, các dịch vụ, thương mại đã dần chuyển cách thức kinh doanh truyền thống sang phát triển doanh nghiệp riêng trên môi trường ảo.

Dưới sự bao phủ rộng của các trang mạng xã hội như thì hầu như không tránh khỏi được những ảnh hưởng những thứ không hay của mạng xã hội đến với giới trẻ như hiện nay. Vì thế, việc xây dựng một mạng xã hội từ thiện để hỗ trợ cho những người không được may mắn trong cuộc đời sẽ là một ảnh hưởng tích cực đến với giới trẻ hiện nay. Vì có những ý tưởng trên, đề tài “Nghiên cứu mạng xã hội từ thiện” được chọn là đề án của em.

Sau khi đã trình bày tổng quan một cách chi tiết về mạng xã hội, đề tài sẽ chủ yếu nhấn mạnh vào cách thức hoạt động của các trang mạng xã hội và tìm hiểu về cách thức hoạt động từ thiện cơ bản và thông thường khác. Từ đó đề tài sẽ có thêm những tính năng khác tạo thú vị hơn nhằm phát huy hiệu quả tốt hơn mạng xã hội từ thiện của mình..

Chương 1. GIỚI THIỆU ĐỀ TÀI

Chương đầu tiên trình bày những khái niệm tổng quát và cơ bản nhất về ngành công nghệ thông tin mà em đang được học và nêu những công nghệ được em áp dụng trong quá trình thực hiện đồ án này. Bên cạnh đó còn có những lí do em chọn đề tài mạng xã hội này và những mục tiêu mong muốn cho đề tài đó.

1.1. Giới thiệu và lí do chọn đề tài

Ngày nay, đất nước ta đang ở trên đường đua phát triển rất mạnh mẽ về công nghệ. Máy tính hay điện thoại đang dần trở thành trở thành một thiết bị không thể thiếu đối với mỗi cá nhân và hầu như đáp ứng được mọi nhu cầu của con người không chỉ ở khía cạnh công việc của mình mà còn trong lĩnh vực giải trí. Kể cả con người cũng có thể kết nối và giao lưu với nhau thông qua các trang mạng xã hội. Công nghệ đã giúp cho đời sống của chúng ta trở nên tiện ích và giản đơn hơn rất nhiều. Nhận thấy nhu cầu của con người ngày càng cao nên việc áp dụng khoa học và công nghệ vào lĩnh vực đời sống là vô cùng cần thiết. Do đó việc xây dựng cộng đồng, các trang mạng xã hội là một điều cần thiết cho sự kết nối với nhau.

Trong thời đại mà chúng ta đang được sống thì Internet đã trở nên quan trọng hơn bao giờ hết trong tất cả mọi lĩnh vực: chính trị, xã hội, kinh tế và thậm chí cả trong quân sự. Không những thế, Internet đang dần được lan rộng và được sử dụng ở mọi tầng lớp trong xã hội. Vì thế một trang mạng xã hội có tính nhân văn cao và có thể giúp được nhiều người kém may mắn và quan trọng hơn hết là lan tỏa niềm yêu thương với nhau.

Bên cạnh đó, do tính chất công việc và thời gian dành cho bản thân nên việc từ thiện trở nên quá phức tạp nên khiến những người muốn làm từ thiện mà cảm thấy phiền phức khi phải làm từ thiện. Vì thế, đổi mới cách thức làm từ thiện là việc làm cần thiết để giúp mọi người có thể lan tỏa và chia sẻ tình yêu thương lẫn nhau ngay cả khi mình không có nhiều thời gian.

Để lan tỏa niềm yêu thương đó của mọi người một cách đơn giản và tối ưu thời gian nên em đã chọn đề tài cho đồ án của mình là “Hệ thống quản lí bài viết mạng xã hội dựa trên ReactJS”.

1.2. Mục tiêu

Mục tiêu của em trong đồ án này là xây dựng một trang mạng xã hội dựa trên nền tảng công nghệ của ReactJS mà em đã học được từ môn “Các công nghệ lập trình hiện đại” vào đồ án của mình.

Xây dựng các website bằng các component để sau này tái sử dụng dễ dàng hơn. Bên cạnh đó em sẽ chủ động tìm hiểu thêm về cách hoạt động bên phía backend là Python Django để giúp tương tác bên backend và với cơ sở dữ liệu.

1.3. Phạm vi và đối tượng nghiên cứu

1.3.1. Phạm vi

Tìm hiểu kĩ về ReactJS cũng như cách thức hoạt động và tìm hiểu khái quát bên phía backend là Python Django. Về phía admin thì sẽ tiếp nhận các bài viết, lưu thông tin và quyết định người chiến thắng đấu giá của người dùng.

Mạng xã hội từ thiện sẽ có các chức năng như những trang mạng xã hội khác nhưng sẽ được phát triển thêm những tính năng có thể hỗ trợ giao dịch trực tuyến.

1.3.2. Đối tượng nghiên cứu

- Những người quan tâm đến hoạt động từ thiện.
- Công nghệ ReactJS.
- Công nghệ Python Django.
- Các website mạng xã hội.
- Các hình thức giao dịch online.
- Đối tượng người dùng tham gia.
- Lưu trữ dữ liệu bằng MySQL.
- Các thư viện liên quan của ReactJS.

1.4. Bố cục đề tài

- Chương 1: Giới thiệu đề tài, nêu lí do chọn đề tài này, mục tiêu và đối tượng nghiên cứu. Đồng thời kể về ứng dụng thực tiễn về ứng dụng mạng xã hội.
- Chương 2: Cơ sở lí thuyết, trình bày những khái niệm một cách cơ bản và dễ hiểu nhất về ReactJS cũng như cách tạo một project bằng ReactJS.
- Chương 3: Hệ thống mạng xã hội từ thiện Charity PT, các sơ đồ và trình bày về các chức năng cần có của hệ thống.
- Chương 4: Trình bày giao diện của hệ thống mạng xã hội Charity PT, các hình ảnh minh họa hỗ trợ cho việc sử dụng mạng xã hội này.
- Chương 5: Kết luận, trình bày về những chức năng website mà em đã làm được thành công, những vấn đề còn thiếu và phát triển đồ án này tốt hơn trong thời gian tới.

Chương 2. CƠ SỞ LÝ THUYẾT

Đây là chương em giới thiệu chủ yếu về công nghệ ReactJS, cách cài đặt, tạo một ứng dụng cũng như cách hoạt động của React. Ngoài ra, em sẽ nói một ít về công nghệ bên phía backend để mọi người có thể hiểu thêm về nó.

2.1. Giới thiệu ReactJS

2.1.1. ReactJS là gì

ReactJS được dùng để xây dựng giao diện mà phía người dùng có thể nhìn thấy được dựa trên thư viện Javascript phổ biến nhất hiện nay. Nó phản hồi trang web một cách nhanh chóng khi người dùng tác động lên trang web bằng cách chia nhỏ các thành phần như một layout để render trang web.

ReactJS được phát triển bởi Facebook cho phép phát triển các thành phần UI dựa trên component có khả năng tái sử dụng cao. ReactJS là mã nguồn mở xuất hiện vào năm 2003 của Javascript. Giờ đây ReactJS đã vượt qua Bootstrap và Angular (hai thư viện được sử dụng nhiều nhất lúc đó).

ReactJS được sử dụng cho cả phía client (front-end) và phía server (backend), chỉ chủ yếu giải quyết tập trung phần View trong mô hình MVC (Model-View-Controller).

2.1.2. Tại sao nên sử dụng React để xây dựng website

Ưu điểm:

React giúp cho các trang web load nhanh hơn các trang web thuần HTML thông thường vì các thành phần trong trang được chia nhỏ thành các component trong React giúp dễ quản lí, dễ sử dụng ở nhiều nơi.

Mỗi khi cần cập nhật dữ liệu, thay vì phải tải lại trang thì React chỉ cập nhật các component có liên quan. Điều này giúp cho trang web chạy mượt mà hơn và tăng khả năng mở rộng và tái sử dụng tốt trong tương lai. React không chỉ dùng phía client mà còn dùng được bên phía server.

Khuyết điểm:

Vì đây là thư viện của Facebook nên React chỉ hỗ trợ phần View nên sẽ không được hỗ trợ phần Model và Controller nên React cần phải kết hợp dùng với các framework khác. (Đó là lí do để em tìm hiểu và kết hợp với Python Django).

2.1.3. Cài đặt môi trường

- Bước 1: Cài đặt NodeJS: <https://nodejs.org/en/download/>

Downloads

Latest LTS Version: **16.13.0** (includes npm 8.1.0)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

The screenshot shows the Node.js download page. It has two main tabs: 'LTS' (Recommended For Most Users) and 'Current' (Latest Features). Below these are three main download options: 'Windows Installer' (node-v16.13.0-x64.msi), 'macOS Installer' (node-v16.13.0.pkg), and 'Source Code' (node-v16.13.0.tar.gz). Below these are links for 'Windows Binary (.zip)', 'macOS Binary (.tar.gz)', 'Linux Binaries (x64)', 'Linux Binaries (ARM)', and 'Source Code'. A table below lists the download links for various architectures:

Platform	Architecture	Download Link
Windows	32-bit	node-v16.13.0-x86.msi
	64-bit	node-v16.13.0-x64.msi
macOS	32-bit	node-v16.13.0.pkg
	64-bit	node-v16.13.0.pkg
Linux	32-bit	node-v16.13.0-linux-armv7l.tar.gz
	64-bit	node-v16.13.0-linux-x64.tar.gz
ARM	ARMv7	node-v16.13.0-linux-armv7l.tar.gz
	ARMv8	node-v16.13.0-linux-arm64.tar.gz

Hình 2.1: Các phiên bản NodeJS.

(Nguồn: <https://nodejs.org/en/download/>)

Để kiểm tra xem NodeJS và NPM đã được cài đặt trong máy hay chưa, chúng ta có thể mở terminal để gõ dòng lệnh:

```
node -v
```

```
npm -v
```

Khi cài đặt thành công, terminal sẽ hiển thị phiên bản NodeJS và NPM đã được cài đặt:

```
PS D:\DoAn\social-network-charity> npm -v
7.20.3
PS D:\DoAn\social-network-charity> node -v
v16.6.1
```

- Bước 2: Tạo ứng dụng ReactJS

Cài đặt bộ cài app React:

```
npm install -g create-react-app
```

Tạo project React với tên charity-social-network:

```
npm create react-app charity-social-network
```

Sau khi tạo thành công ta sẽ có một thư mục tên charity-social-network có chứa các thư mục như sau:



Hình 2.2: Tạo project ReactJS thành công.

(Nguồn: https://hocwebchuan.com/tutorial/reactjs/install_reactjs.php)

Trong đó có hai thư mục quan trọng là **public** và **src**:

- `index.html`: là trang html duy nhất để hiển thị giao diện.
 - `public`: dùng để chứa các file hình ảnh, css và javascript,...
 - `package.json` và `package-lock.json`: dùng để chứa các thông tin ứng dụng cần cài cho project.
 - `src`: dùng để chứa các file có các component ta viết trong quá trình làm.
- Bước 3: Chạy thư mục React

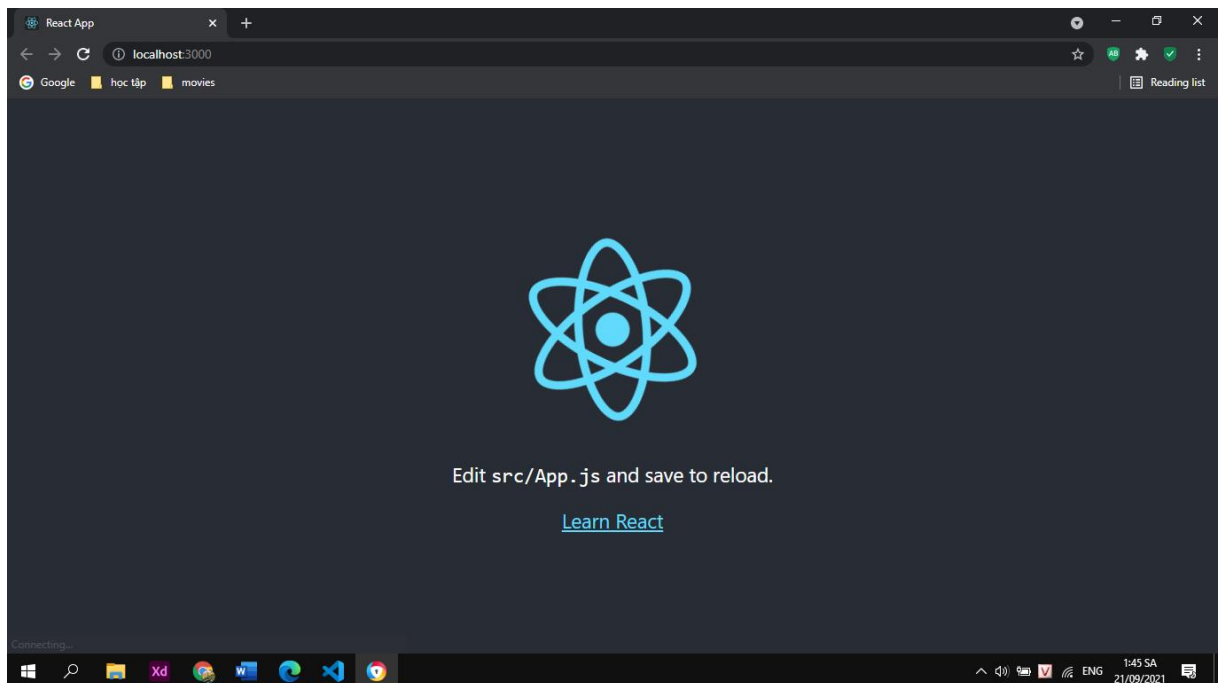
Đầu tiên trở tới thư mục vừa tạo bằng lệnh **cd**:

```
cd charity-social-network
```

Cuối cùng ta chạy project mới tạo bằng lệnh:

```
npm start
```

Lúc này trình duyệt sẽ tự động mở một trang web với đường link là: <http://localhost:3000/> và hiển thị như hình bên dưới là thành công:



Hình 2.3: Chạy project ReactJS thành công.

2.2. Kiến thức cơ bản

2.2.1. Khái niệm Component

React được xây dựng bằng các thành phần (component) chứ không dùng template như những trang web thông thường khác. Component khi được xây dựng xong thì chức năng của nó tương tự như các thẻ được viết trong HTML.

Component có hiệu quả trong việc tái sử dụng, quản lí và giúp cho trang web được bảo trì dễ dàng hơn.

Component trong React phải kế thừa (extend) thành phần `React.Component` và bắt buộc phải có phương thức `render` để trả về HTML.

Có hai cách để sử dụng Component:

- Class Component (dạng lớp)

```
function Component(props) {  
  return (  
    <div>  
      <h1> Nguyễn Thiên Phúc </h1>  
    </div>  
  )  
}
```

- Function Component (dạng hàm)

```
class Component extend React.Component {  
  render() {  
    return(  
      <div>  
        <h3>Xin chào Nguyễn Thiên Phúc</h3>  
      </div>  
    )  
  }  
}
```

2.2.2. Khái niệm JSX

JSX (là chữ viết tắt của Javascript XML) cho phép HTML được viết trong Javascript và chuyển các thẻ HTML thành các thành phần (component) của React.

Đây là một cú pháp mở rộng của Javascript. Biểu thức được viết trong cặp dấu hoặc nhon `{}`. Các thuộc tính trong JSX được viết theo quy tắc `lowerCamelCase`. Ví dụ: *background-color* thành *backgroundColor* và *class* thành *className*.

Ví dụ:

```
const name = "World"
const element = <div className="style">Hello {name}
</div>
```

Ta có thể trả ra nhiều thành phần nhưng chúng phải được bọc (wrap) trong một thành phần container.

React không bắt buộc sử dụng JSX, nhưng JSX giúp xây dựng component một cách dễ dàng và thực thi nhanh hơn javascript. Nhờ chuyển thành mã nguồn của Javascript nên nó đã được tối ưu hơn trước rất nhiều.

Việc tập trung component giúp chương trình dễ thiết kế template, dễ tái sử dụng, bảo trì và mở rộng. Các lỗi đều có thể phát hiện ra trong lúc biên dịch nên giúp chương trình chạy hiệu quả hơn.

2.2.3. Khái niệm Props

Props là thuộc tính được dùng để gửi dữ liệu đến component. Component được xem như là một hàm trong Javascript. Giá trị của props tại component con không được thay đổi.

```
function FullName(props) {
  return(
    <h1>
      {props.user.lastname} {props.user.firstname}
    </h1>
  )
}
```

Nếu component có phương thức constructor và truyền cho React.Component bằng phương thức super.

Ví dụ:

```
class ComponentConstructor extends React.Component {
  constructor(props) {
    super(props)
  }
}
```

2.2.4. Khái niệm State

State dùng để lưu giá trị của các thuộc tính trong component. State chỉ có hiệu lực trong phạm vi component.

Trong nhiều trường hợp chúng ta cần cập nhật state, giá trị state có thể thay đổi bằng phương thức `this.setState()`. Khi đối tượng state thay đổi (state có thể thay đổi bất cứ lúc nào) thì component sẽ nạp lại trang.

Muốn truy cập vào state ta sử dụng thuộc tính: `this.state.propsName`.

Ví dụ:

```
class ComponentState extends React.Component {
  constructor(props) {
    super(props);
    this.state = {position: "World"};
  }
  render() {
    return (
      <div>
        <h3>Hello: {this.state.position}</h3>
      </div>
    );
  }
}
```


2.2.5. So sánh Props và State

Giống:

- Đều là plain JS Object.
- Đều là trigger render update khi thay đổi.

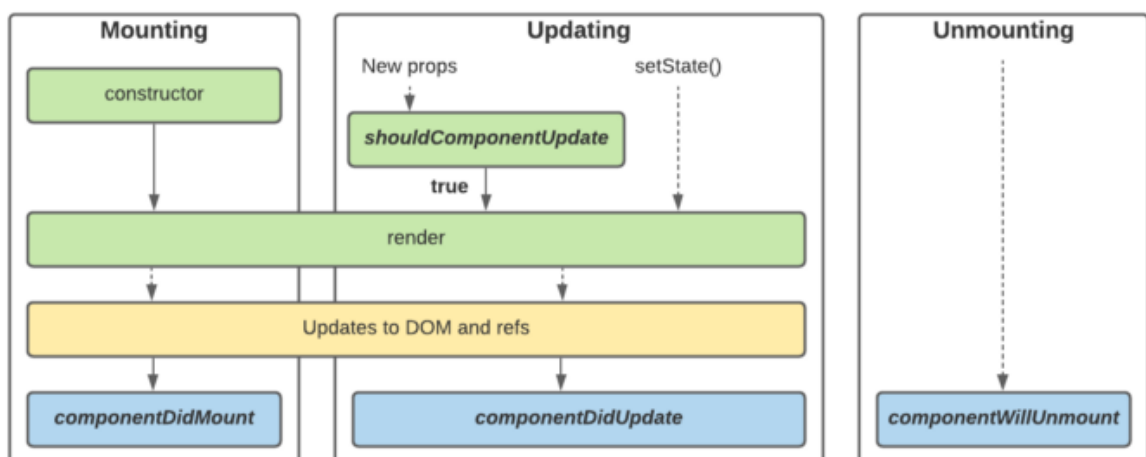
Khác:

	Props	State
Có thể nhận giá trị ban đầu từ thành phần gốc không?	✓	✓
Có thể được thay đổi bởi thành phần gốc không?	✓	X
Có thể đặt giá trị mặc định bên trong component không?	✓	✓
Có thể thay đổi bên trong thành phần component không?	X	✓
Có thể thay đổi trong các component con không?	✓	X

Bảng 2.1: So sánh props và state.

2.3. Vòng đời ReactJS

Component của React có 3 giai đoạn chính:



Hình 2.4: Vòng đời ReactJS.

(Nguồn: <https://projects.wojtekmaj.pl/react-lifecycle-methods-diagram/>)

2.3.1. Mounting

- Đưa các phần tử vào DOM và bắt buộc cần phải có phương thức render khi bạn định nghĩa chúng.
- Phương thức `componentDidMount()` được gọi sau khi thành phần được render.
- Đây là nơi chạy các câu lệnh yêu cầu thành phần đó đã được đặt trong DOM.

2.3.2. Updating

- Một component sẽ tự động thay đổi khi có bất cứ cập nhật nào về props hoặc state của component đó.
- Phương thức render là bắt buộc và luôn được gọi, các phương thức khác là tùy chọn và sẽ được gọi nếu bạn định nghĩa chúng.

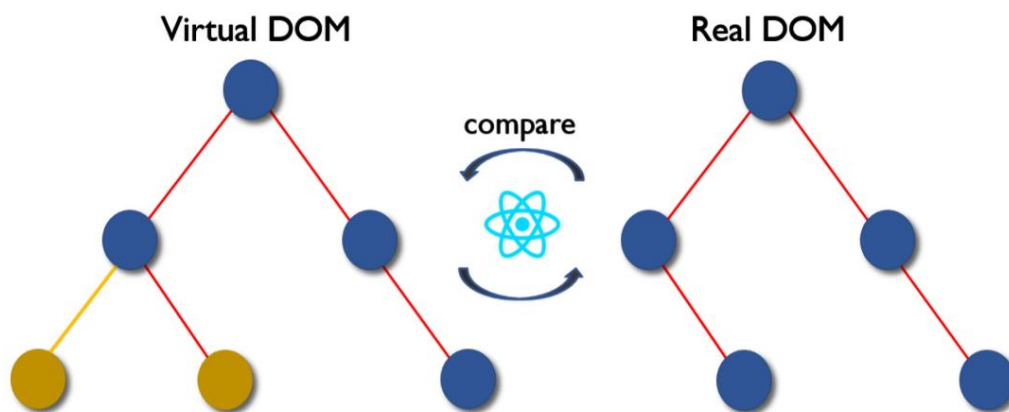
2.3.3. Unmounting

- Đây là giai đoạn cuối của component khi component sắp bị loại bỏ ra khỏi DOM.
- Phương thức `componentWillUnmount()` được gọi khi thành phần sắp bị xóa khỏi DOM.

2.4. Virtual DOM

ReactJS sử dụng DOM ảo (Virtual DOM) để cải thiện hiệu năng hoạt động của các Web ứng dụng thay vì sử dụng DOM thật truyền thống. Virtual DOM có đầy đủ đặc tính của DOM. Khi một Node trên Virtual DOM thay đổi, nó tìm node thay đổi bằng cách so sánh Virtual DOM và DOM, khi node tìm thấy sẽ tự động cập nhật mà không ảnh hưởng đến các node khác trên DOM.

Ví dụ: Nếu như một User có 3 bài viết trên trang cá nhân được hiển thị bằng ReactJS và nếu chúng ta chỉnh sửa bài viết thứ 2 thì chỉ bài viết thứ 2 đó được hiển thị lại. Còn những bài viết không có sự tác động thì sẽ giữ nguyên (không tải lại trang và cũng không hiển thị lại).



Hình 2.5: So sánh virtual DOM và Real DOM.

(Nguồn: <https://viblo.asia/p/ro-hon-mot-chut-ve-virtual-dom-trong-reactjs-maGK70ALZj2>)

2.5. React Hook

React Hook là phiên bản được hỗ trợ trong React 16.8 cho phép bạn dùng state và các tính năng khác trong React mà không cần phải viết lớp. Nhờ có hook mà chúng ta có thể tác động vào các tính năng của các component trong vòng đời.

Hook là các hàm Javascript nhưng cần tuân thủ hai quy tắc: chỉ gọi hook ở cấp cao nhất (không gọi hook bên trong vòng lặp và các hàm lồng nhau) và chỉ gọi hook từ các React function component.

Có hai dạng hook thông dụng là: `useState` và `useEffect`. `useState` dùng để thiết lập và tìm kiếm state. `useEffect` dùng để nắm bắt sự thay đổi của code trong chương trình.

Ví dụ import React Hook:

```
import React, {useState, useEffect} from 'react'
```

2.5.1. Hook State

Hook State là cách mới để khai báo trạng thái trong ứng dụng React. Hook sử dụng `useState()` để thiết lập và truy xuất trạng thái.

2.5.2. Hook Effect

Hook Effect cho phép thực hiện các hiệu ứng phụ (như một hành động) trong các hàm component. `UseEffect` không sử dụng các phương thức vòng đời có sẵn như trong các lớp component. Nói một cách dễ hiểu thì Hook Effect tương đương với các phương thức vòng đời `componentDidMount()`, `componentDidUpdate()` và `componentWillUnmount()`.

Ví dụ sử dụng `useState` và `useEffect` trong React Hook:

```
import React, {useState, useEffect} from 'react'

function ResponsitoryGrid ({ maSo }) {
  const [responsitory, setResponsitory] =
    useState([])
  const [load, setLoad] = useState(true)

  useEffect(() => {
    setLoad(true)
    fetchResponsitory(maSo).then((responsitory) => {
      setResponsitory(responsitory)
      setLoad(false)
    }, [maSo])
    if(load === true) {
      return <Loading />
    }
    return (<div />)
  })
}
```

2.6. React Bootstrap

Trong ReactJS, React Bootstrap được thay thế bằng Javascript Bootstrap, không phụ thuộc vào một số thư viện như jquery và các component của Bootstrap được xây dựng như các React component.

Cài đặt React Bootstrap:

```
npm install react-bootstrap-bootstrap@4.6.0
```

Ví dụ sử dụng lưới trong Bootstrap

```
import { Container, Row, Col } from "react-bootstrap"

{/* Hàm trong React*/}
...
<Container>
  <Row>
    <Col xs={12} md={8}> xs=12 md=8 </Col>
    <Col xs={6} md={4}> xs=6 md= 4</Col>
  </Row>

  <Row>
    <Col xs={6}> xs=6 </Col>
    <Col xs={6}> xs=6 </Col>
  </Row>
</Container>
...
```

2.7. Material UI (thư viện component UI)

Tương tự như Bootstrap thì Material UI cũng là một thư viện React Component được tích hợp thêm Material Design của Google. Nó giúp cho giao diện của trang web trở nên được hấp dẫn và sống động hơn.

Cài đặt Material UI

```
npm install material-ui
```

Cách sử dụng

```
import {Button} from '@material-ui/core';

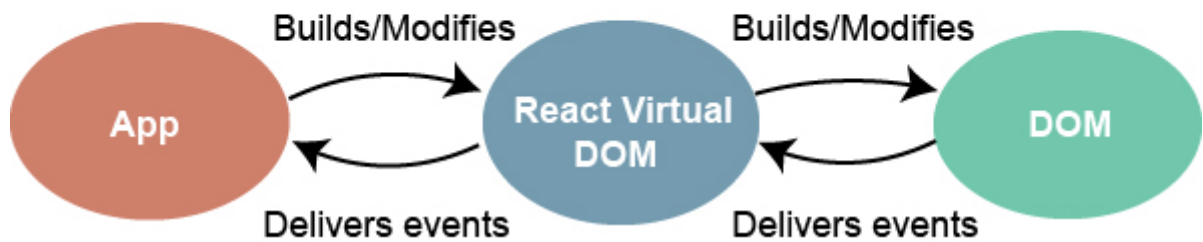
export default function MyButton() {
  return (
    <div>
      // Sử dụng Button như một thẻ trong React
      <Button> My Button </Button>
    </div>
  );
}
```

2.8. React Event

Event là tác động của người dùng lên trang web (click chuột, nhấn phím, tải trang,...) hoặc do hệ thống tạo ra. React có cơ chế xử lý sự kiện riêng, tương tự như trên các phần tử DOM.

Hệ thống React Event là một trình bao bọc trình duyệt chéo của sự kiện gốc của trình duyệt.

Events Handler



Hình 2.6: Event Handler.

(Nguồn: <https://www.javatpoint.com/react-events>)

2.9. React Router

React Router được dùng để chuyển các trang qua lại với nhau theo dạng Single page nhưng không cần phải tải lại trang web mà chỉ tải những các component cần thiết. React có 3 gói cho Routing:

- React-router: cung cấp các thành phần router chính cho ứng dụng.
- React-router-native: dùng cho các mobile-app (ứng dụng di động).
- React-router-dom: dùng cho các web-app (ứng dụng web).

Cách cài đặt:

```
npm install react-router-dom
```

- *BrowserRouter*: dùng để xử lý các URL động.
- *HashRouter*: dùng để xử lý các request tĩnh.
- *Router*: Chỉ định các component được nạp dựa trên đường dẫn chỉ định.
- *Switch*: dùng để chứa các Route để nạp các component.
- *Link*: dùng để thay đổi đường dẫn đến các URL mà không cần nạp lại trang.
- *NavLink*: tương tự như Link nhưng cho phép thêm style cho active link.
- *Redirect*: dùng để chuyển hướng đến các route khác trong ứng dụng nhưng vẫn duy trì route cũ.

Ví dụ: import các component trong App.js

```
import React from 'react';
import { Switch, BrowserRouter, Route } from 'react-router-dom';
import Newsfeed from '../webpage/Newsfeed';
import Timeline from '../webpage/Timeline';
import Login from '../webpage/Login';
import SignUp from '../webpage/SignUp';

export default function App() {
  return (
    <>
      <BrowserRouter>
        <div>
          <ul>
            <li>
              <Link to="/" component={Login} /> Login
            </li>
            <li>
              <Link to="/newsfeed" component={Newsfeed} /> Newsfeed
            </li>
            <li>
              <Link to="/timeline" component={Timeline} /> Timeline
            </li>
            <li>
              <Link to="/signup" component={SignUp} /> Sign up
            </li>
          </ul>
        </div>
      </BrowserRouter>
    </>
  );
}
```



```

        <Switch>
          <Route path="/" component={Login} />
          <Route path="/newsfeed/"
component={Newsfeed} />
          <Route path="/timeline/"
component={Timeline} />
          <Route path="/signup/" component={SignUp} />
        </Switch>
      </BrowserRouter>
    </>
  ) }

```

Chú ý: tất cả thuộc tính phải được bọc trong `BrowserRouter`, nếu để bên ngoài thì hệ thống sẽ báo lỗi.

2.10. React Form

Trong React, dữ liệu trong form thường được xử lý bởi các component khi dữ liệu được lưu trong state chứ không như HTML thường được xử lý trong DOM. Trước khi sử dụng, ta cần tạo state trong constructor.

Lệnh `event.target.value` dùng để truy cập các trường (field) trong trình xử lý sự kiện. Ta có thể kiểm soát các thay đổi bằng cách thêm xử lý sự kiện *onChange* trong thuộc tính. Để cập nhật trạng thái thì ta sẽ sử dụng [dấu ngoặc vuông] bên trong các thuộc tính.

Ta có thể sử dụng `useState` (Hook) để theo dõi giá trị đầu vào và cung cấp nguồn duy nhất cho toàn bộ ứng dụng.

Ví dụ React Form:

```
import { useState } from "react";

function MyForm() {
  const [inputs, setInputs] = useState({});
  const handleChange = (event) => {
    const name = event.target.name;
    setInputs(values => ({...values, [name]: value}))
  }
  const handleSubmit = (event) => {
    event.preventDefault();
    alert(inputs);
  }
  return (
    <form onSubmit={handleSubmit}>
      <div>Enter your name:
        <input
          type="text"
          name="username"
          value={inputs.username || ""}
          onChange={handleChange}
        />
      </div>
      <button type="submit" />
    </form>
  )
}

ReactDOM.render(<MyForm />,
  document.getElementById('root'));
```

2.11. React Context

React Context dùng để chuyển các component mẹ sang các component con bằng cách lưu trữ các props trong store (tương tự như Redux) nhưng cách làm này code viết ra sẽ dễ hiểu hơn so với đoạn mã của Redux.

Sử dụng context giúp ta dễ dàng truyền dữ liệu trong toàn ứng dụng.

Ví dụ sử dụng context:

```
const UseMyContext = React.createContext('light');
// Context cho ta chuyển giá trị vào trong component
// mà không phân luồng nó rõ ràng qua mọi component
class App extends React.Component {
  render() {
    return (
      <UseMyContext.Provider value="dark">
        <MyToolbar />
      </UseMyContext.Provider>
    );
  }
}
// Sử dụng Provider để chuyển component vào cây bên
// dưới

function MyToolbar() {
  return (
    <div>
      <MyThemeButton />
    </div>
  )
}
```

```
// Gán một contextType để đọc context hiện tại và
// React sẽ tìm Provider gần nhất để sử dụng giá trị
class MyThemeButton extends React.Component {
  static typeContext = MyContext;
  render() {
    return <Button theme={this.typeContext} />;
  }
}
```

2.12. React Redux

Redux là một vùng chứa những trạng thái có thể đoán trước được thiết kế để giúp bạn viết các ứng dụng Javascript hoạt động được hầu như ở mọi môi trường và dễ kiểm tra.

Với redux, web ứng dụng của em có trạng thái được lưu trong một bộ nhớ và mỗi thành phần có thể truy cập vào bất kì trạng thái nào mà nó cần từ bộ nhớ này.

Redux hoạt động dựa trên một bộ nhớ trung tâm lưu trữ toàn bộ trạng thái của ứng dụng. Mỗi component có thể truy cập trạng thái được lưu trữ mà không cần phải gửi các đạo cụ từ component này sang component khác.

Cài đặt Redux:

```
npm install --save redux
```

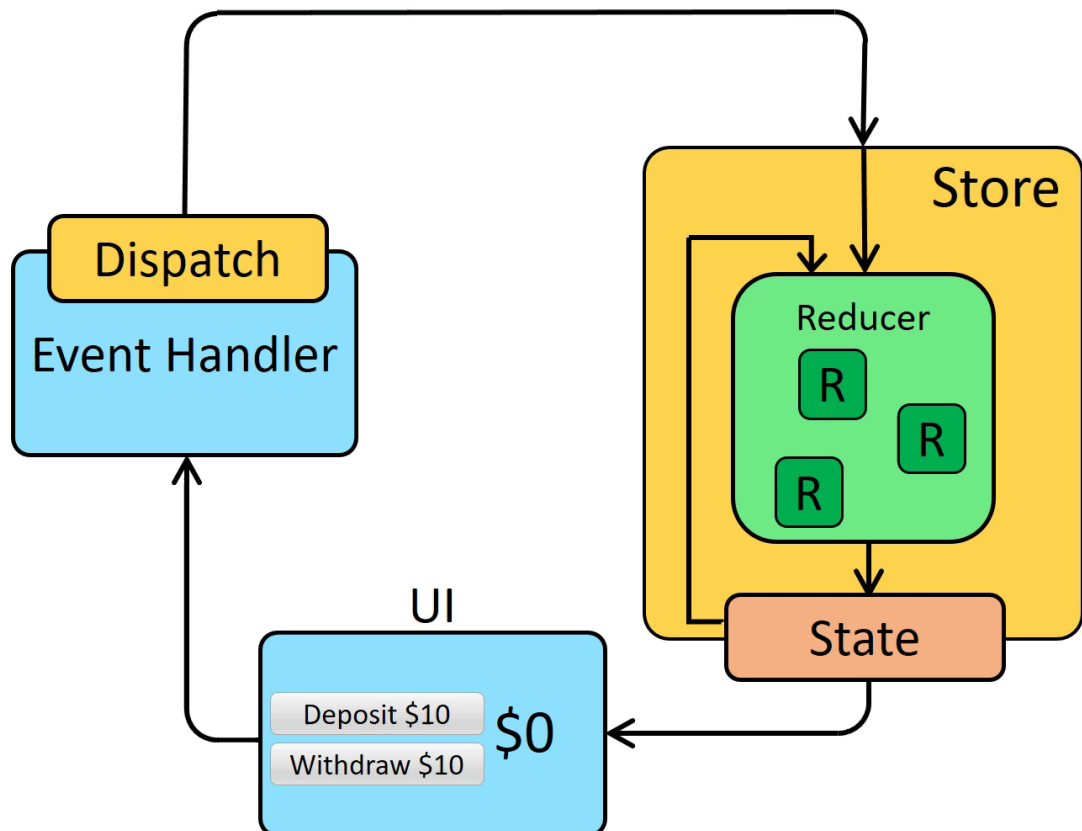
Cài Redux cùng với ứng dụng React:

```
npm install --save react-redux
```

Cơ chế hoạt động của Redux:

- Một action xảy ra khi user tương tác với ứng dụng. Hàm reducer được gọi với state hiện tại, action được truyền vào và trả về state mới.

- Store thông báo các view thực thi bằng cách gọi các hàm callback của chúng.
- React component đọc dữ liệu từ store và truyền action tới store để cập nhật dữ liệu.
- Redux theo dõi (subscribe) store để kiểm tra dữ liệu của component nào thay đổi sẽ nạp lại (rerender) component đó



Hình 2.7: Cách hoạt động của Redux.

(Nguồn: [Redux Fundamentals, Part 2: Concepts and Data Flow | Redux](#))

Trong đó:

- Action là sự kiện do tương tác của người dùng, gọi API hoặc từ form. Action là cách duy nhất để gửi dữ liệu đến store.
- Reducer là nơi duy nhất để thực hiện và cập nhật state trong Redux. Reducer nhận state hiện tại của đối tượng action để cập nhật state và trả về state mới.

Cú pháp tạo Reducer:

```
(state=initState, action) => newState
```

- Store là nơi lưu trữ duy nhất chứa state hiện hành của ứng dụng. Để tạo store cần truyền một reducer và sử dụng cú pháp createStore().

Cú pháp tạo store

```
const store = createStore(reducer)
```

Có ba phương thức khi làm việc với store:

- getState() : trả về state hiện tại.
- subscribe() : chỉ định hàm callback được gọi khi store được cập nhật.
- dispatch() : cho phép gửi action để thay đổi state, cập nhật UI tương ứng với state hiện tại.

2.13. React Cookie

Cookie là cơ chế dùng để lưu trữ dữ liệu cục bộ bên trong trình duyệt.

Cài đặt React Cookie:

```
npm install react cookie --save
```

Cách sử dụng:

```
import cookies from 'react-cookies'
```

Cú pháp:

```
const [cookies, setCookie, removeCookie] =  
useCookies(['cookie-name']);
```

Cookie cung cấp một component <CookiesProvider>, ta có thể dùng nó để bọc các thành phần gốc lại. Sau đó các component này được sử dụng bất kì đâu trên cookie.

Ví dụ: file *index.js*

```
import React from 'react'  
import {CookiesProvider} from 'React-cookie'  
import App from './App'
```

```
ReactDOM.render(  
  <CookiesProvider>  
    <App />  
  </CookiesProvider>  
  document.getElementById('root')  
) ;
```

2.14. React Axios

Axios là một thư viện phục vụ để tạo ra các yêu cầu HTTP hiện ra bên ngoài và lấy dữ liệu từ nguồn bên ngoài. Ta có thể dùng mã đọc được không đồng bộ có trong Javascript. Sau đó, ta có thể dễ dàng chặn hoặc hủy các yêu cầu khả năng tích hợp của các trang web giả mạo.

Axios là một ứng dụng promise-based HTTP, hoạt động cả trong môi trường client và server của NodeJS. Nó cung cấp một API duy nhất để xử lý XMLHttpRequests và giao diện http của nodeJS. Sau đây là cách sử dụng Axios:

- instance: thể hiện của axios.
- url (bắt buộc): endpoint API.
- method (bắt buộc): phương thức HTTP request.
- data: dữ liệu trong request body.
- params: dữ liệu trên query string.
- config: thông tin cấu hình cho axios.
- debounce: khoảng thời gian tối thiểu giữa các sự kiện request.
- onSuccess: sự kiện được gọi khi request thực hiện thành công.
- onError: sự kiện được gọi khi request thực hiện có lỗi.
- onLoading: thực hiện khi bắt đầu request.

Cài đặt React Axios

```
npm install axios
// và
npm install react-axios
```

Ví dụ get dữ liệu với axios

```
import React from 'react'
import axios from 'axios'
export default class ListProduct extends
React.Component {
  state = {
    people:[]
  }
  componentDidMount() {
    axios.get("<đường link>").then(res => {
      const products = res.data;
      this.setState({products});
    })
  }
}
```

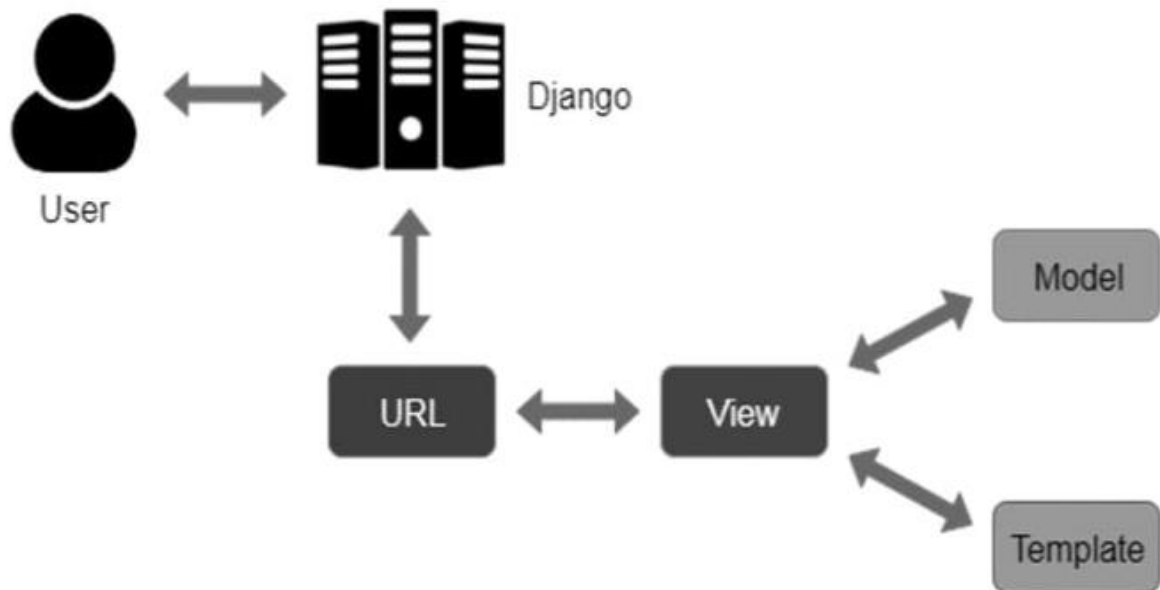
Ví dụ post dữ liệu với axios

```
axios ({
  method: 'post',
  url: '/products',
  data: {
    productName: "Noodle"
  }
});
```


2.15. Python Django

Django là một framework lập trình web bậc cao và là mã nguồn mở được viết bằng ngôn ngữ Python.

Django hoạt động theo mô hình MVT (Model-View-Template) chứ không phải mô hình MVC (Model-View-Controller). Ta có thể thay thế phần Template trong mô hình MVT bằng framework ReactJS.



Hình 2.8: Mô hình hoạt động của Django (MVT).

(Nguồn: <https://niithanoi.edu.vn/django-la-gi.html>)

Khi người dùng gửi yêu cầu, Django sẽ xử lý yêu cầu của người đó bằng Model, View và Template. Mã trong Controller được viết dùng để kiểm tra các tác động qua lại với nhau giữa Model và View.

Để kiểm tra hoạt động của Controller có tính ứng dụng tốt hay không bằng cách ánh xạ đường dẫn URL. Nếu URL ánh xạ thành công thì View sẽ tương tác qua lại với Model và trả lại request cho Template.

Chương 3. **HỆ THỐNG MẠNG XÃ HỘI TỪ THIỆN**

Chương này em sẽ trình bày về hệ thống của trang mạng xã hội từ thiện Charity PT. Trình bày cách thức hoạt động cũng như các chức năng cần phải có của một trang mạng xã hội. Bên cạnh đó em có trình bày sơ đồ để mọi người có thể hiểu thêm hệ thống.

3.1. Xác định yêu cầu

Mạng xã hội từ thiện Charity PT có các chức năng sau:

- Đăng nhập và đăng kí vào mạng xã hội.
- Thêm/xóa/sửa bài viết.
- User có thể thêm hoặc không thêm hashtag vào bài viết mà User sắp tạo.
- Hiển thị chi tiết của bài viết (hiển thị thông tin và comment).
- Hiển thị profile của User (số bài viết, hình ảnh,...)
- User có thể xem lại thông tin cá nhân của mình đã đăng kí tài khoản.
- Thích, chia sẻ, bình luận bài viết.
- User đăng tải các vật phẩm muốn đấu giá lên bài viết của mình.
- User có thể đấu giá các vật phẩm đang được đăng đấu giá.
- User được xây dựng profile riêng cho trang của mình.
- Admin được xem thống kê các bài viết, số lượng người thích, bình luận hoặc chia sẻ bài viết khác.
- Admin có thể xem thống kê số lượng bài viết, số bình luận và số like trên bài viết.
- User thắng đấu giá có thể chuyển tiền cho người bán đấu giá.
-

3.2. Phân tích yêu cầu

- Giao diện chính sẽ hiển thị các bài viết đã được duyệt nên user nào cũng có thể xem nó thông qua tài khoản riêng của mình.
- Đăng nhập có hai vai trò (admin và user). Em sẽ phát triển cho User đăng nhập bằng facebook và google trong thời gian tới.
- Form đăng kí dành riêng cho các user chưa có tài khoản và muốn đăng kí để tham gia vào mạng xã hội.

3.3. Phương pháp triển khai

Để có được một trang mạng xã hội hoàn hảo, hoạt động mượt mà trong thực tế thì ta cần có những kế hoạch và quy trình thiết kế web đảm bảo được tính trọn vẹn, thông tin người dùng được bảo mật tốt và phải thỏa mãn được nhu cầu sử dụng của người dùng.

Mạng xã hội từ thiện Charity PT không chỉ xây dựng bằng React mà còn sử dụng thêm những thư viện bên ngoài như Material UI (thư viện UI component) nhằm mang đến một giao diện bắt mắt nhưng cũng rất là tiện lợi. Các component của thư viện cũng được chia nhỏ thành nhiều các icon thành phần để dễ dàng cho việc sử dụng lại nhiều lần.

Trong quá trình phát triển đồ án sẽ có thể phát sinh thêm những chức năng mà cần sử dụng đến dependency nhằm giải quyết các vấn đề phức tạp mà các lập trình viên thường gặp, giúp cho chúng ta giảm được thời gian phát triển website và giúp cho các lập trình viên xử lý dễ dàng hơn.

3.4. Các bước triển khai đề tài

Bước 1: Xác định phân khúc người dùng mà website hướng tới, thiết lập các chức năng dựa trên các yêu cầu của một mạng xã hội.

Bước 2: Tính toán dữ liệu được xử lý khi gọi API.

Bước 3: Lên ý tưởng cho việc thiết kế giao diện bằng phương pháp liệt kê các chức năng trong hệ thống.

Bước 4: Tiến hành xây dựng giao diện. Trước tiên, ta phải phân chia các thư mục rõ ràng để dễ dàng thao tác và quản lý chúng. Tiếp theo, ta sẽ thiết kế giao diện bằng HTML, CSS và Javascript,... vào các thư mục đã chia.

Bước 5: Sau khi đã hoàn thành xong bước 4, em sẽ tiến hành cắt nhỏ file HTML thành từng component nhỏ để tiện cho việc sử dụng lại. Mỗi component có thể là một phần hiển thị ra website hay chỉ là một chức năng đơn thuần của web.

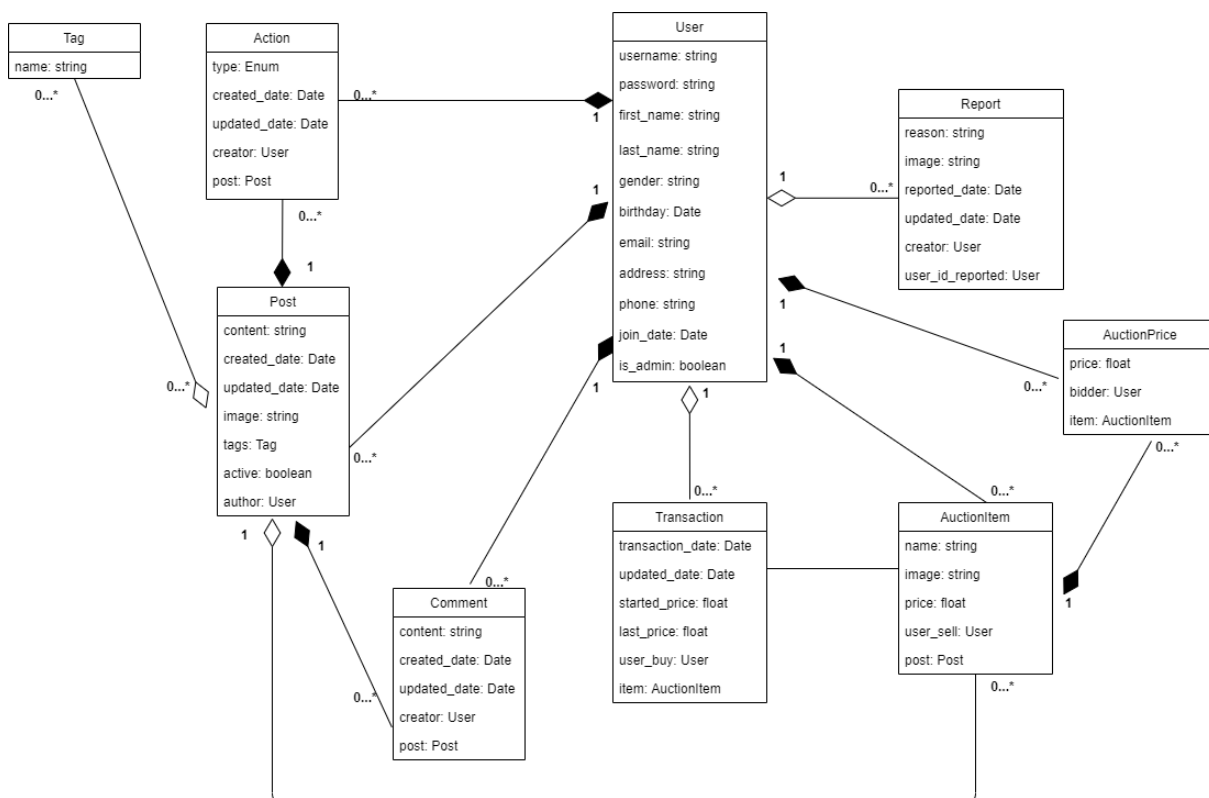
Bước 6: Tạo các phiên quản lí chúng bằng Cookie, LocalStorage để có thể xác thực người dùng cũng như để bảo mật thông tin.

Bước 7: Tạo các chức năng cho giao diện đồng thời kết nối với API, định tuyến thông qua props và state và context để truy vấn dữ liệu.

Bước 8: Kiểm thử giao diện người dùng, bước này cần thực hiện khi hoàn thành xong một chức năng nào đó thì chúng ta tiến hành kiểm thử luôn để tránh sau này có lỗi chúng ta có thể dễ dàng khoanh vùng và sửa lại chức năng đó.

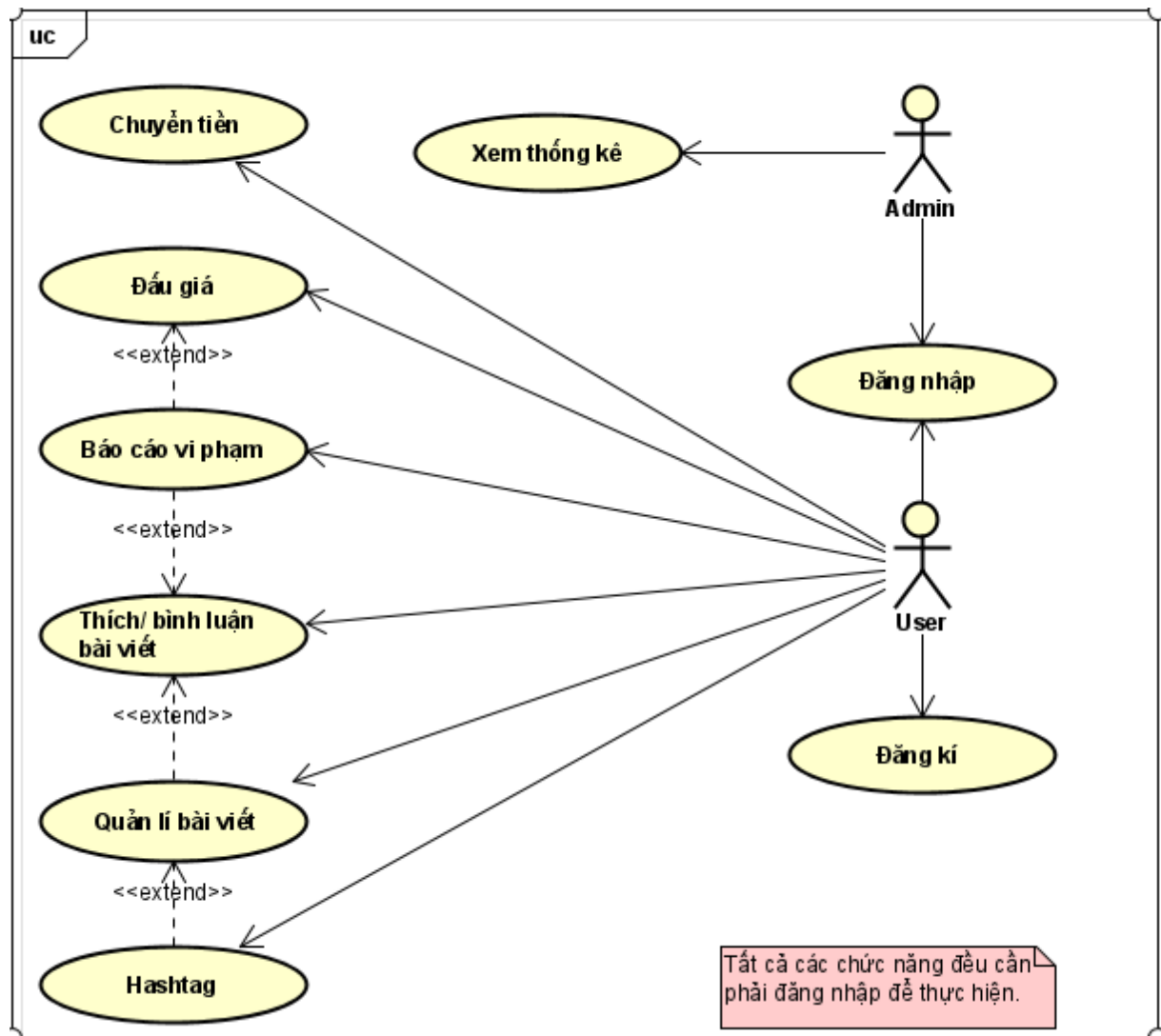
Bước 9: Đem website vào vận hành, bảo trì và nâng cấp.

3.5. Thiết kế CSDL



Hình 3.1: Sơ đồ UML của hệ thống.

3.6. Sơ đồ Use case



Hình 3.2: Sơ đồ Use case.

Nhìn vào sơ đồ thì ta đã có thể thấy được hệ thống này có hai quyền đó là quyền admin và User (người dùng hệ thống). Cả hai quyền này trước khi sử dụng bất cứ chức năng nào thì đều phải đăng nhập vào hệ thống. Riêng User nào nếu như chưa có tài khoản riêng thì cần phải đăng kí tài khoản riêng cho cá nhân thì mới đăng nhập được. Chức năng của User sẽ quyết định độ hấp dẫn và giúp cho mạng xã hội này được lan tỏa tốt hơn đến với nhiều người. Còn Admin thì chỉ có thể thống kê số lượng User tham gia và số lượng bài post, đồng thời cũng có thể đăng bài viết nếu muốn nhưng khả năng lan tỏa thì hệ thống cần sự tham gia của nhiều người dùng để lan tỏa đến với tất cả mọi người giúp cho hệ thống phát triển tốt hơn trong tương lai.

3.7. Các API của server

actions			▼
GET	/actions/	actions_list	🔒
auction_items			▼
GET	/auction_items/	auction_items_list	🔒
POST	/auction_items/{id}/add-price/	auction_items_add_price	🔒
comments			▼
PUT	/comments/{id}/	comments_update	🔒
PATCH	/comments/{id}/	comments_partial_update	🔒
DELETE	/comments/{id}/	comments_delete	🔒
oauth2-info			▼
GET	/oauth2-info/	oauth2-info_list	🔒
posts			▼
GET	/posts/	posts_list	🔒
POST	/posts/	posts_create	🔒
GET	/posts/{id}/	posts_read	🔒
PUT	/posts/{id}/	posts_update	🔒
PATCH	/posts/{id}/	posts_partial_update	🔒
DELETE	/posts/{id}/	posts_delete	🔒
POST	/posts/{id}/add-comment/	posts_add_comment	🔒
POST	/posts/{id}/add-item/	posts_add_item	🔒
POST	/posts/{id}/add-tags/	posts_add_tag	🔒
GET	/posts/{id}/get-comments/	posts_get_comments	🔒
POST	/posts/{id}/take-action/	posts_take_action	🔒
reports			▼
POST	/reports/	reports_create	🔒
GET	/reports/{id}/	reports_read	🔒
DELETE	/reports/{id}/	reports_delete	🔒
tags			▼
GET	/tags/	tags_list	🔒
transactions			▼
GET	/transactions/	transactions_list	🔒
POST	/transactions/	transactions_create	🔒
users			▼
GET	/users/	users_list	🔒
POST	/users/	users_create	🔒
GET	/users/current-user/	users_get_current_user	🔒

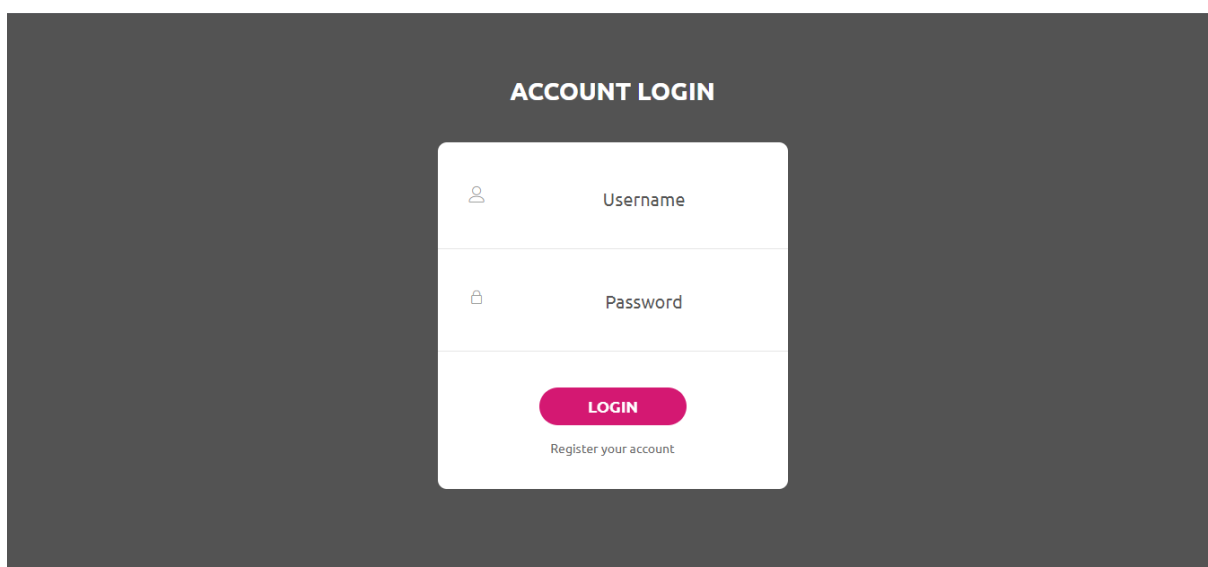
Hình 3.3: Các API của của python django.

Chương 4. GIAO DIỆN VÀ CHỨC NĂNG MẠNG XÃ HỘI TỪ THIỆN

Ở chương này trình bày sâu về giao diện của mạng xã hội từ thiện và các chức năng khi chúng ta đăng nhập vào hệ thống cũng như cách tạo bài viết và bình luận trên bài viết đó. Mọi thứ đều sẽ được trình bày một cách đơn giản nhưng mà chi tiết để mọi người có thể hiểu được cách sử dụng của mạng xã hội này.

4.1. Đăng nhập (Login)

Khi người dùng truy cập vào trang web mạng xã hội Charity PT thì sẽ thấy giao diện của trang đăng nhập và người dùng phải nhập đúng thông tin đã đăng kí thì mới có thể truy cập được trang web. Khi người dùng nhập sai thông tin đăng kí thì hệ thống sẽ hiển thị thông báo nhập sai thông tin. Trong trường hợp người dùng chưa có tài khoản riêng thì cần nhấp vào “Register your account” để thực hiện đăng kí tài khoản.

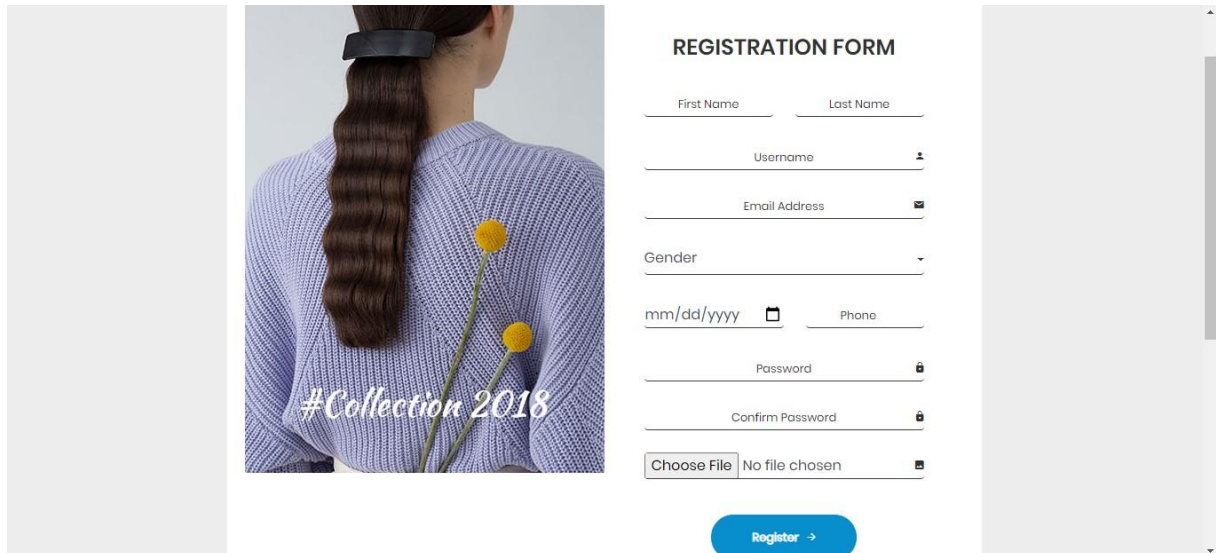


Hình 4.1: Giao diện trang đăng nhập.

Chức năng của giao diện đăng nhập dùng để cho user đã có tài khoản riêng của mình đăng nhập vào hệ thống mạng xã hội từ thiện. User cần phải nhập đúng thông tin của Username và Password như đã đăng kí tài khoản thì mới thực hiện chuyển trang. Đây là là trang người dùng sẽ thấy khi thoát khỏi tài khoản của mình.

4.2. Đăng kí (Register)

Trang đăng kí dùng để cho người dùng đăng kí tài khoản riêng để tham gia vào cộng đồng. Người dùng cần phải điền đầy đủ thông tin mà form đăng kí yêu cầu như trong hình. Sau đó nhấn Register thì hệ thống kiểm tra và sẽ trả về kết quả.

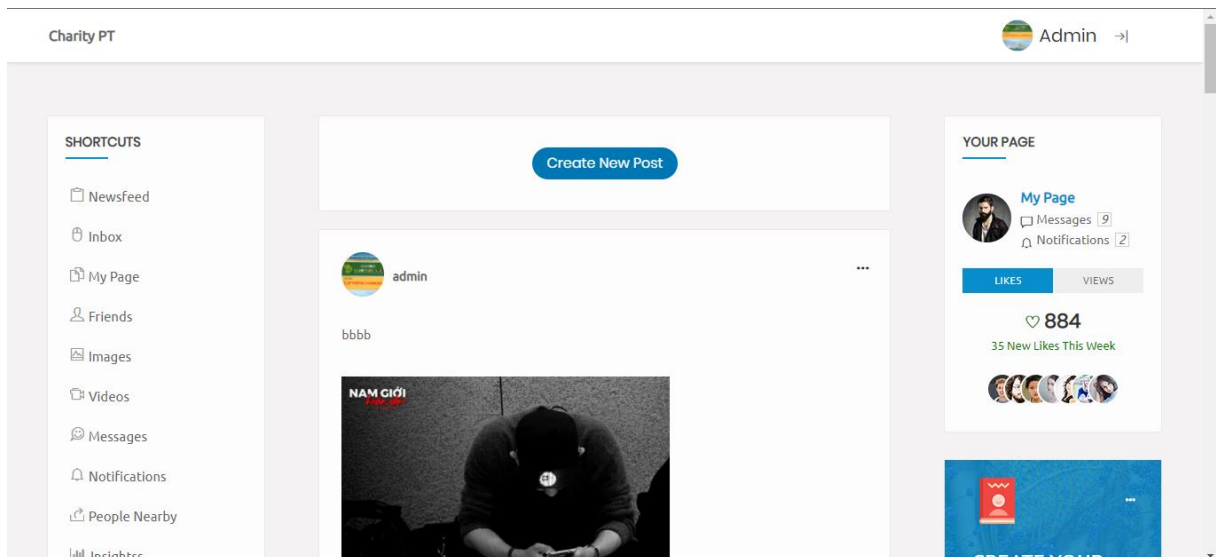


Hình 4.2: Giao diện chức năng đăng kí.

Chức năng của trang đăng kí dùng để cho người dùng đăng kí tài khoản riêng cho cá nhân khi chưa có tài khoản tham gia vào mạng xã hội Charity PT. User đăng kí phải điền đầy đủ thông tin trong form đăng kí. Khi đăng kí được thực hiện thành công, hệ thống sẽ tự động chuyển đến trang đăng nhập.

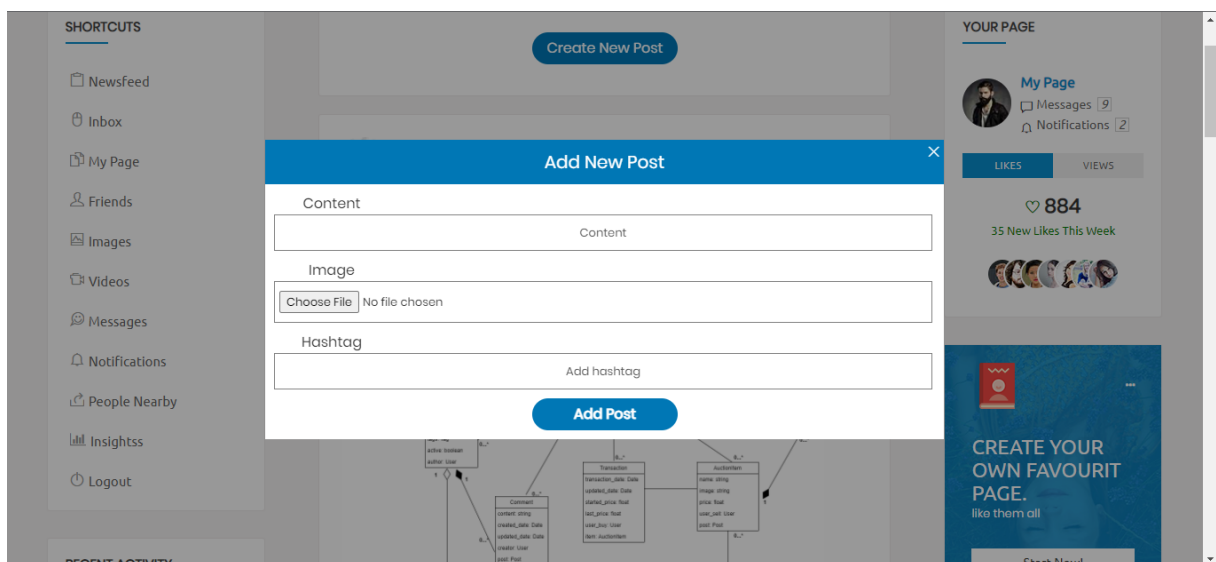
4.3. Bảng tin (Newsfeed)

Trang này giống như trang chủ của trang mạng xã hội khi bạn đã có tài khoản riêng cho mình và đã đăng nhập thành công. Trang này hiển thị các bài viết của những User khác và bạn có thể kéo xuống để xem được nhiều thông tin hơn từ các bài viết đó. Bên cạnh đó khi đăng nhập vào thì hệ thống sẽ hiển thị ra Username và ảnh của người đăng kí và khi User muốn đăng xuất khỏi hệ thống thì họ chỉ cần nhấn dấu mũi tên phía bên phải Username thì sẽ đăng xuất ra khỏi hệ thống và quay lại trang login.



Hình 4.3: Giao diện trang Newsfeed.

Chức năng của trang Newsfeed cập nhật các bài viết mới nhất trên hệ thống mạng xã hội từ thiện. Các bài viết sẽ hiển thị ảnh và Username của người đã tạo bài viết đó. Và nếu như User muốn tạo bài viết riêng cho mình thì chỉ cần bấm nút “Create New Post”, giao diện sẽ hiển thị form như sau:



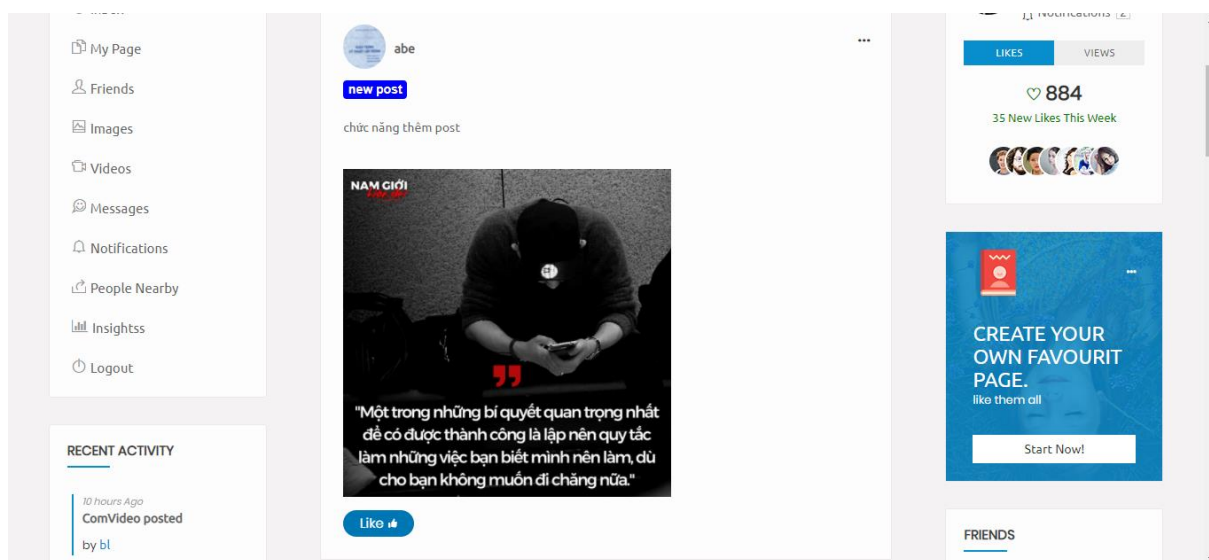
Hình 4.4: Giao diện tạo bài viết mới.

Lúc này, User chỉ cần điền thông tin mà mình muốn thêm vào cho bài viết của mình. Ảnh và hashtag không bắt buộc nhưng nội dung (content) của bài viết phải bắt buộc nhập thì mới thực hiện tạo bài viết. Sau khi điền xong thông tin, User chỉ cần bấm nút “Add Post” thì bài viết của User đã được tạo và hiện lên trang Newsfeed.

4.4. Bài viết (Post)

Sau khi thêm bài viết thành công, trang newsfeed sẽ tự động tải lại trang web và cập nhật bài viết của User mới vừa thêm vào.

Trên bài viết này, các người dùng khác sẽ thấy được các thông tin mà người tạo bài viết đã thêm vào bài viết của mình như là nội dung bài viết, hình ảnh và hashtag. Và khi User hover lên icon ba chấm thì sẽ hiển thị ra một danh sách. Khi User vào trang chi tiết bài viết (Post detail) thì User sẽ được bình luận ở trên bài viết này đồng thời cũng có thể xem được bình luận của những User khác và thích bài viết này.



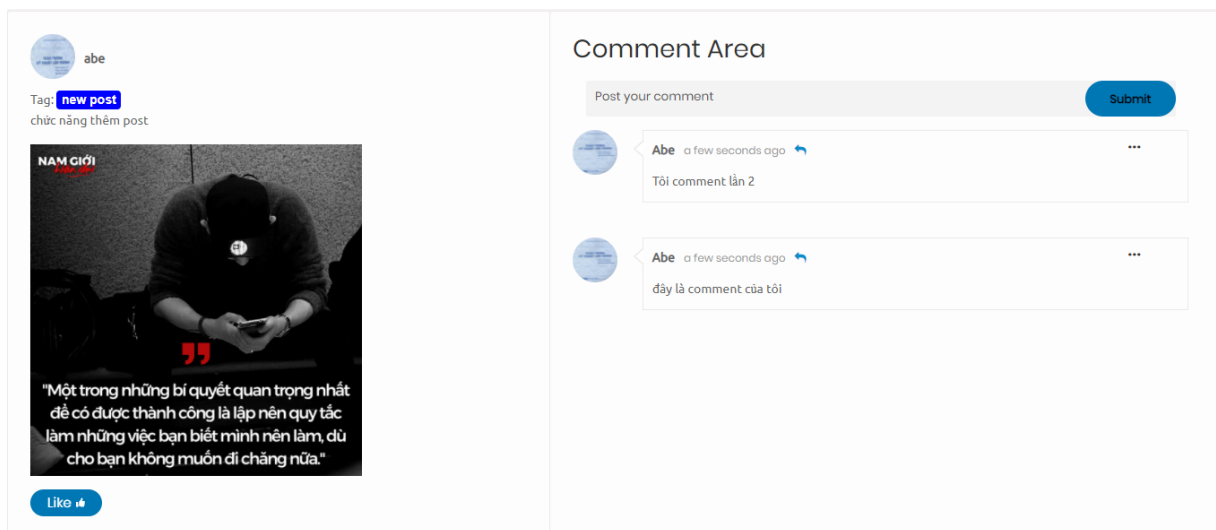
Hình 4.5: Hiển thị bài viết của User.

Chỉnh sửa bài viết (update post) và xóa bài viết (delete post) chỉ có thể được hoạt động khi User là người tạo bài viết này. Những người dùng khác không phải chủ của bài viết sẽ không dùng được chức năng này. Khi chủ bài viết bấm update post thì bài

viết sẽ hiển thị lên một form dùng để cho chủ bài viết chỉnh sửa thông tin trên bài viết của mình. Còn nếu chọn delete post thì bài viết sẽ xóa ra khỏi hệ thống và sẽ không có User nào thấy được bài post đã bị xóa.

4.5. Chi tiết bài viết (Post detail)

Trang này dùng để cho người dùng xem tổng quát hơn chi tiết của một bài viết và xem những bình luận của các User khác đã bình luận về bài viết này.

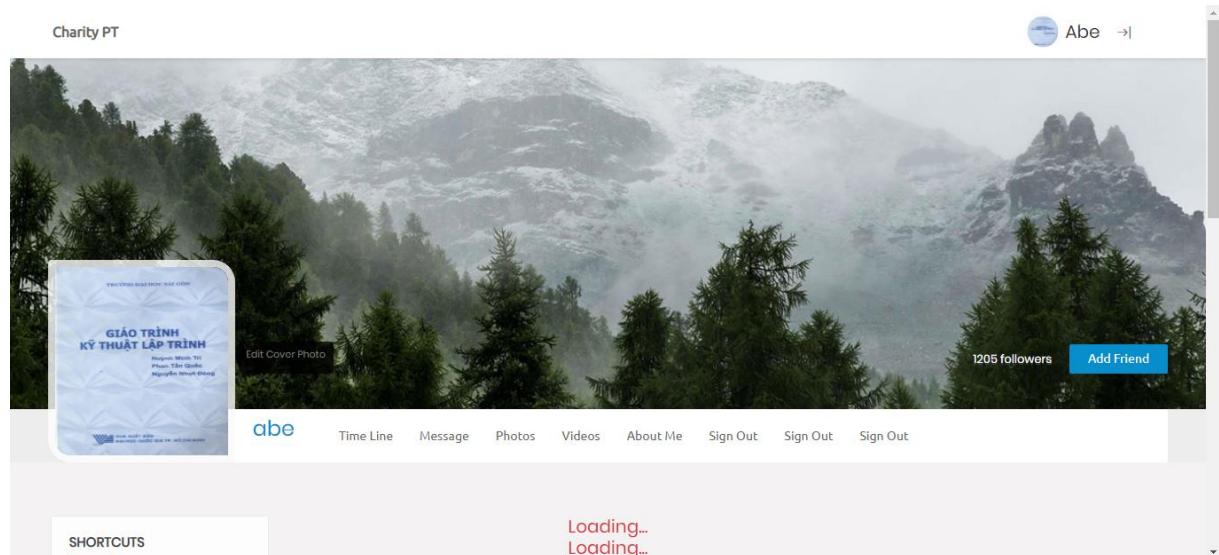


Hình 4.6: Giao diện chi tiết bài viết.

Chức năng của trang này dùng để bình luận bài viết và giúp cho người dùng xem bài viết một cách tổng quan nhất. Khi vào trang này, mọi User đều có thể bình luận về bài viết này. Hệ thống sẽ lấy thông tin của người đã bình luận và hiển thị ảnh cũng như Username của người bình luận, các User khác sẽ thấy bình luận của User đã bình luận trên bài viết này. Hệ thống cũng có hỗ trợ sửa, xóa bình luận khi mà người dùng lỡ nhập sai thông tin bình luận hay không muốn bình luận bài viết này nữa. Chỉ cần trỏ chuột vào hình ba chấm, người dùng có thể chọn sửa hoặc xóa bình luận.

4.6. Trang cá nhân (Timeline)

Trang này dùng để hiển thị thông tin cá nhân User đã đăng kí tài khoản và lưu lịch sử hoạt động cũng như các bài viết mà User đó đã đăng hoặc chia sẻ lên trang cá nhân của mình.

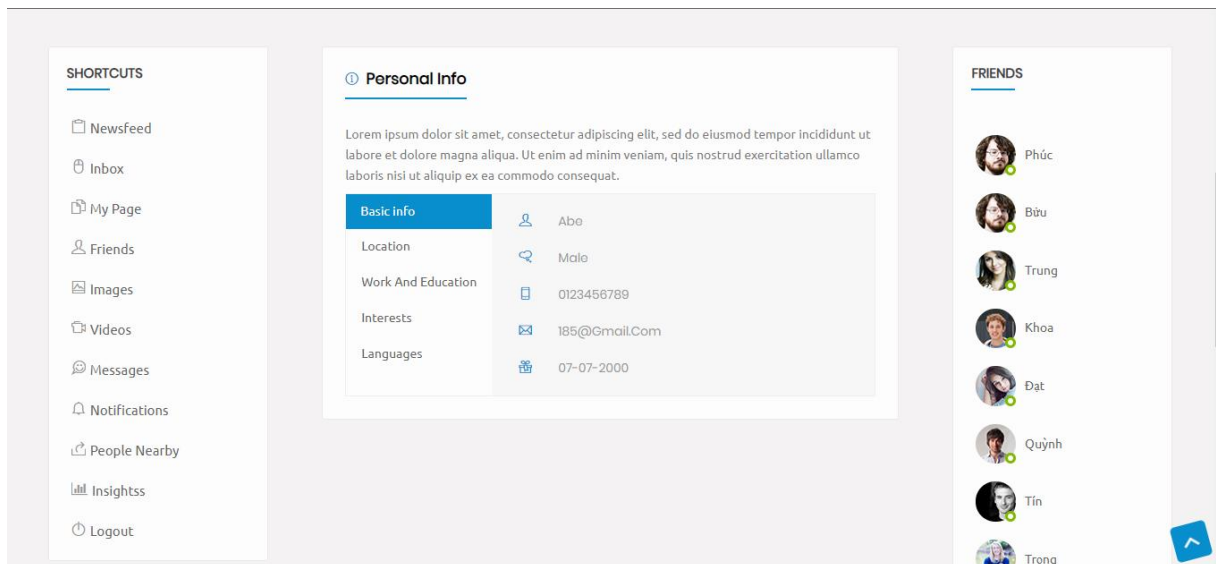


Hình 4.7: Giao diện trang timeline.

Chức năng của trang Timeline dùng để lưu các hoạt động của User tương tự như là một trang cá nhân để User có thể xem lại mọi hoạt động của mình cũng như nhớ lại những kỉ niệm của mình khi đã tham gia vào mạng xã hội Charity PT.

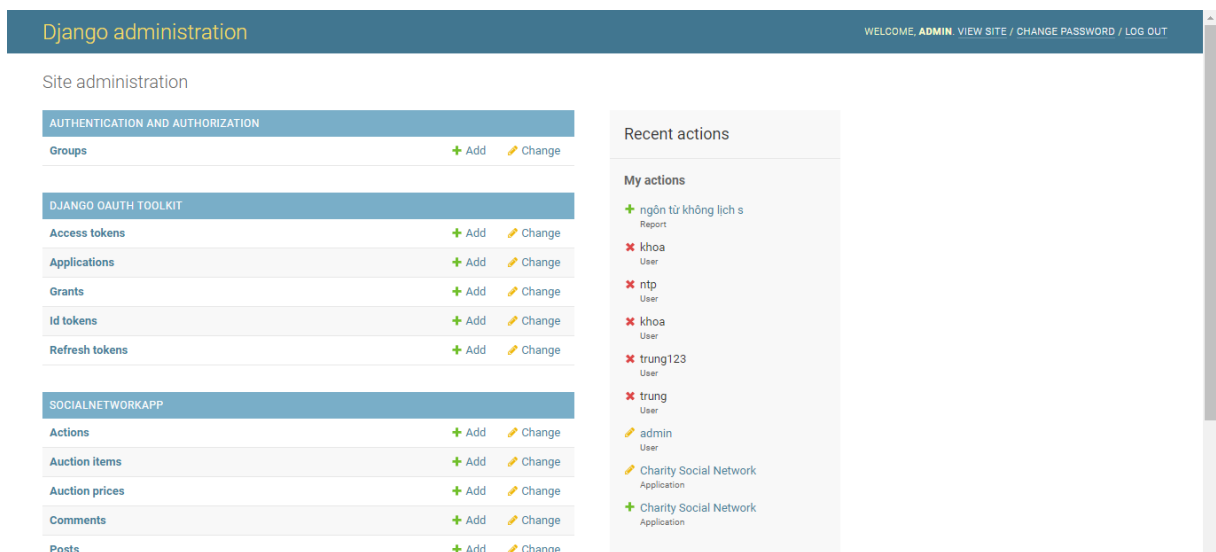
4.7. Thông tin User (About User)

Trang này có thể được xem như là một trang con của Timeline vì trang này được dùng để hiển thị thông tin cá nhân của User đã đăng kí tài khoản như: Username, họ tên, địa chỉ hay là số điện thoại của User đó. Chỉ có chủ tài khoản mới được xem thông tin của mình. Chức năng của Trang này dùng để cho User thay đổi thông tin cá nhân của mình khi không còn ở nơi cũ cũng như là sử dụng số điện thoại cũ.



Hình 4.8: Giao diện trang thông tin cá nhân.

4.8. Giao diện hệ thống bên Python Django



Hình 4.9: Giao diện admin phía backend.

Đây chính là giao diện admin bên phía hệ thống backend sử dụng Python Django. Đúng như tên gọi thì đây là trang dùng để quản lý mạng xã hội từ thiện. Hệ thống này giúp cho admin quản lý hệ thống của mình cũng như là thống kê được số lượng User đã tham gia vào hệ thống hay số lượng bài viết đã có trên hệ thống này.

Chương 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Chương cuối em sẽ trình bày về kết quả của mình và những gì mình đã học được trong khoản thời gian hiện đồ án ngành cũng như nêu ra những thiếu sót trong hệ thống và có những định hướng sẽ khắc phục và phát triển hệ thống trở nên mạnh mẽ hơn trong thời gian sắp tới.

5.1. Mục tiêu đạt được

Sau một khoản thời gian thực hiện đồ án, em đã học được cách mà ReactJS hoạt động cũng như cách cắt các component thành từng phần nhỏ và cách thức truyền dữ liệu qua lại giữa các component thông qua props và state. Đồng thời lấy được API do phía backend cung cấp để đổ dữ liệu ra giao diện.

Website đã thực hiện được tốt các chức năng cơ bản của một mạng xã hội thông thường. Tuy nhiên vẫn còn một số điểm hạn chế như: website còn nhỏ, hệ thống không được thông minh như những trang mạng xã hội khác, người dùng chưa thể chat trực tuyến được với nhau, không có chức năng tìm kiếm nội dung. Trải nghiệm người dùng không được tốt. Đầu giá vẫn chưa thể thực hiện thành công.

5.2. Hướng phát triển

Trong thời gian tới em sẽ cố gắng hoàn thành các chức năng còn thiếu như đã nêu trên và sẽ thêm những chức năng mới đa dạng hơn để người dùng có được trải nghiệm tốt nhất như:

- Tạo giao diện trên điện thoại.
- Tạo chức năng chat trực tuyến.
- Cải thiện khả năng bảo mật cũng như hiệu năng của trang web.
- Tạo những game đơn giản có sức hấp dẫn để thu hút mọi người tham gia vào hoạt động từ thiện.
- Cải thiện giao diện thống kê cho admin.
- Đăng nhập đăng kí bằng google và facebook.
- Cải thiện chức năng đầu giá sản phẩm.
- Áp dụng thêm những công nghệ mới như Cloud, Docker.

TÀI LIỆU THAM KHẢO

- [1] “Cài đặt React.js | Hướng dẫn học react | Học web chuẩn.”
https://hocwebchuan.com/tutorial/reactjs/install_reactjs.php (accessed Nov. 12, 2021).
- [2] “How to Implement Material-UI in React | Engineering Education (EngEd) Program | Section.” <https://www.section.io/engineering-education/how-to-implement-material-ui-in-react/> (accessed Nov. 12, 2021).
- [3] “Hướng dẫn học React.js | Học web chuẩn.”
<https://hocwebchuan.com/tutorial/reactjs/> (accessed Oct. 05, 2021).
- [4] “React – A JavaScript library for building user interfaces.” <https://reactjs.org/> (accessed Oct. 05, 2021).
- [5] “ReactJS Tutorial - javatpoint,” www.javatpoint.com.
<https://www.javatpoint.com/reactjs-tutorial> (accessed Oct. 05, 2021).
- [6] “Redux - A predictable state container for JavaScript apps. | Redux.”
<https://redux.js.org/> (accessed Oct. 05, 2021).
- [7] “Redux Fundamentals, Part 2: Concepts and Data Flow | Redux.”
<https://redux.js.org/tutorials/fundamentals/part-2-concepts-data-flow> (accessed Nov. 12, 2021).
- [8] “Rõ hơn một chút về Virtual DOM trong ReactJS.” <https://viblo.asia/p/ro-hon-mot-chut-ve-virtual-dom-trong-reactjs-maGK70ALZj2> (accessed Nov. 12, 2021).
- [9] “Sử dụng Material-UI trong dự án dùng reactjs,” *Viblo*, Apr. 27, 2017.
<https://viblo.asia/p/su-dung-material-ui-trong-du-an-dung-reactjs-XL6lAgamKek> (accessed Nov. 12, 2021).
- [10] “Viblo | Free service for technical knowledge sharing,” *Viblo*.
<https://viblo.asia/newest> (accessed Oct. 05, 2021).
- [11] “Lập trình Web với Django? Django là gì?” <https://niithanoi.edu.vn/django-la-gi.html> (accessed Nov. 12, 2021).