

**TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN**

Tel. (84-236) 3736949, Fax. (84-236) 3842771
Website: <http://dut.udn.vn/khoacntt>, E-mail: cntt@dut.udn.vn



BÁO CÁO THỰC TẬP CÔNG NHÂN

PHẦN THIẾT KẾ WEBSITE

ĐỀ TÀI:

MẠNG XÃ HỘI CHIA SẺ CÔNG THỨC MÓN ĂN

TÊN NHÓM: Nhóm 11_12

NHÓM SINH VIÊN THỰC HIỆN

Nguyễn Văn Đại (nhóm trưởng)	17T2
Châu Việt Trung	17T2
Phan Văn Vũ	17T2
Trần Thị Thu Công	17T2
Phạm Thị Thùy Dương	17T2

Đà Nẵng, 12/2020

MỤC LỤC

MỞ ĐẦU 1

CHƯƠNG 1: TỔNG QUAN VỀ CÔNG NGHỆ3

1.1. NGÔN NGỮ LẬP TRÌNH VÀ THƯ VIỆN.....3	3
1.1.1. ReactJS.....3	3
1.1.2. NodeJs:.....3	3
1.1.3. ExpressJs:.....4	4
1.2. CƠ SỞ DỮ LIỆU.....4	4
1.3. MÔI TRƯỜNG DEPLOY.....5	5
1.3.1. Heroku.....5	5

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ VÀ TRIỂN KHAI HỆ THỐNG6

2.1. KHẢO SÁT VÀ ĐẶC TẢ YÊU CẦU.....6	6
2.1.1. Phân quản lý trong admin.....6	6
2.1.2. Giao diện người dùng.....6	6
2.2. YÊU CẦU PHI CHỨC NĂNG.....6	6
2.3. Phân tích thiết kế hệ thống.....6	6
2.3.1. Các chức năng của hệ thống.....6	6
2.3.2. Các tác nhân của hệ thống.....7	7
2.4. USE CASE.....9	9
2.4.1. Use case tổng quát.....9	9
2.4.2. Use case đăng nhập.....9	9
2.4.3. Use case đăng ký.....10	10
2.4.4. Use case hiển thị chức năng của user:.....10	10
2.4.5. Use case hiển thị chức năng của admin:.....11	11
2.4.6. Use case xem bài viết.....11	11
2.4.7. Use case quản lý thông tin cá nhân.....12	12
2.4.8. Use case đăng bài viết:.....12	12
2.4.9. Use case quản lý bài viết.....13	13
2.5. THIẾT KẾ CƠ SỞ DỮ LIỆU.....13	13
2.5.1. Danh sách các bảng.....13	13
2.5.2. Bảng User.....13	13
2.5.3. Bảng Post.....14	14
2.5.4. Bảng comment.....15	15
2.5.5. Bảng hashtag.....15	15
2.5.6. Bảng friend.....15	15
2.5.7. Bảng emotion.....16	16
2.5.8. Bảng notification.....16	16

CHƯƠNG 3: CÀI ĐẶT VÀ KẾT QUẢ ĐẠT ĐƯỢC.....17

3.1.	CÀI ĐẶT	17
3.1.1.	Backend	17
3.1.2.	FrontEnd	20
3.1.3.	Deploy chương trình.....	20
3.2.	GIAO DIỆN TRANG WEB.....	22
3.2.1.	Giao diện người dùng	22
3.2.2.	Giao diện trang Admin	28

DANH SÁCH HÌNH ẢNH

Hình 1 Use case tổng quát	9
Hình 2 Use case đăng nhập	9
Hình 3 Use case đăng ký	10
Hình 4 Use case hiển thị chức năng của user	10
Hình 5 Use case hiển thị chức năng của admin	11
Hình 6 Use case xem bài viết	11
Hình 7 Use case quản lý trang cá nhân	12
Hình 8 Use case đăng bài viết	12
Hình 9 Use case quản lý bài viết	13
Hình 10 Cấu trúc cây controller	17
Hình 11 Cấu trúc cây route	18
Hình 12 Logo Postman	18
Hình 13 Logo Chai	18
Hình 14 Logo CircleCI	19
Hình 15 Màn hình MongoDB Compass Communication	19
Hình 16 Giao diện của MongoDB Cloud	20
Hình 17 Tạo app mới ở Heroku	21
Hình 18 Chọn Repository để deploy trên Heroku	21
Hình 19 Chọn nhánh để deploy trên Heroku	21
Hình 20 Màn hình Application Log	22
Hình 21 Giao diện khi mở ứng dụng lần đầu	22
Hình 22 Giao diện đăng ký	23
Hình 23 Giao diện đăng nhập	24
Hình 24 Giao diện khi đăng nhập thành công	24
Hình 25 Giao diện thêm bài viết	25
Hình 26 Giao diện xem chi tiết bài viết và comment	25
Hình 27 Giao diện xem chi tiết trang cá nhân	26
Hình 28 Giao diện chỉnh sửa thông tin cá nhân	26
Hình 29 Giao diện kết quả tìm kiếm	27
Hình 30 Giao diện trang Admin	28

DANH SÁCH BẢNG

Bảng 1 Các yêu cầu phi chức năng của ứng dụng	6
Bảng 2: Chức năng của Admin	7
Bảng 2: Chức năng của người dùng.....	8
Bảng 3: Danh sách các bảng cơ sở dữ liệu	13
Bảng 4: Bảng User	13
Bảng 5: Bảng Post.....	14
Bảng 6: Bảng comment.....	15
Bảng 7: Bảng hashtag	15
Bảng 8: Bảng friend	15
Bảng 9: Bảng emotion	16
Bảng 10: Bảng notification	16

DANH SÁCH TỪ VIẾT TẮT

Từ viết tắt	Diễn giải
URI	Uniform Resource Identifier
JSON	JavaScript Object Notation
JWT	JSON Web Token
API	Application Programming Interface
URL	Uniform Resource Locator
CSS	Cascading Style Sheets
ES6	ECMAScript 2015

MỞ ĐẦU

1. Tổng quan về đề tài

Mạng xã hội về ẩm thực, chia sẻ công thức món ăn “Cookii” là một nền tảng giúp giới thiệu, quảng bá những món ăn từ đơn giản đến phức tạp, từ dân dã đến độc đáo. Trang web “Social Cookii” cho phép người dùng chia sẻ những công thức về những món ăn hàng ngày và có thể học hỏi được những món ăn mới lạ. Và từ đó họ có thể giúp đỡ nhau tạo nên những món ăn hấp dẫn, đẹp mắt qua những lời bình luận chia sẻ của nhau.

Chúng em viết ra trang web này vì nhận thấy hiện nay, nhu cầu tìm hiểu về ẩm thực ngày càng tăng nhưng vẫn chưa có nhiều trang web để cho người dùng có thể chia sẻ được món ăn mà họ muốn được chia sẻ hay tìm hiểu thêm các món ăn độc đáo hơn.

2. Mục đích và ý nghĩa của đề tài

2.1. Mục đích

Social Cookii có mục đích xây dựng nên một cộng đồng chia sẻ và thảo luận văn minh. Tạo ra một môi trường giao lưu, học hỏi, chia sẻ những món ăn mà người dùng muốn chia sẻ với nhau để tạo ra những món ăn hấp dẫn và độc đáo.

2.2. Ý nghĩa

Social Cookii giúp cho người dùng có thể bày tỏ được những tâm huyết mà họ đã làm ra những món ăn đó và chia sẻ chúng để cho nhiều người được biết. Mặt khác, người dùng cũng học hỏi thêm những món ăn mới, những kinh nghiệm mới từ những người xung quanh trên trang web.

3. Bố cục của đề tài

Báo cáo đề tài bao gồm các nội dung sau:

Mở đầu

Chương 1: TỔNG QUÁT VỀ CÔNG NGHỆ

Chương 2: PHÂN TÍCH THIẾT KẾ VÀ PHÁT TRIỂN HỆ THỐNG

Chương 3: CÀI ĐẶT VÀ KẾT QUẢ ĐẠT ĐƯỢC

Kết luận và hướng phát triển.

CHƯƠNG 1: TỔNG QUAN VỀ CÔNG NGHỆ

1.1. NGÔN NGỮ LẬP TRÌNH VÀ THƯ VIỆN

1.1.1. ReactJS

React (ReactJS hay React.js) là một thư viện JavaScript mã nguồn mở dùng để xây dựng giao diện người dùng hoặc các thành phần UI. Được phát triển bởi Facebook và cộng đồng các nhà phát triển và các công ty. React có thể sử dụng như một cơ sở để phát triển các ứng dụng đơn hoặc ứng dụng di động.

React giúp việc tạo giao diện người dùng trở nên dễ dàng. Thiết kế các khung nhìn đơn giản cho từng trạng thái trong ứng dụng và React sẽ cập nhật và hiển thị hiệu quả các thành phần phù hợp khi dữ liệu của bạn thay đổi.

React giúp đóng gói các thành phần cùng với trạng thái của chúng, sau đó sử dụng chúng vào các giao diện người dùng phức tạp. Vì logic các thành phần được viết bằng JavaScript nên có thể dễ dàng chuyển dữ liệu trong ứng dụng.

React Hooks là những hàm cho phép bạn "kết nối" React State và Lifecycle vào các Functional Components. Với Hooks bạn có thể sử dụng State và Lifecycles mà không cần dùng ES6 class.

Lợi ích của React Hook:

- Khiến các component trở nên gọn nhẹ hơn
- Giảm đáng kể số lượng code, dễ tiếp cận.
- Cho phép chúng ta sử dụng state ngay trong function component.

1.1.2. NodeJs:

NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp. NodeJS là một mã nguồn mở được sử dụng rộng rãi bởi hàng ngàn lập trình viên trên toàn thế giới.

NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ Window cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.

Các đặc tính của NodeJS:

- Không đồng bộ
- Chạy rất nhanh
- Đơn luồng nhưng khả năng mở rộng cao
- Không đệm
- Có giấy phép

1.1.3. ExpressJs:

ExpressJS là một trong những framework phổ biến dùng để xây dựng API và Website phổ biến nhất của NodeJS. Nó được sử dụng rộng rãi đến mức hầu như mọi dự án Web nào bắt đầu bằng việc tích hợp Express.

Lý do để chọn ExpressJS:

- Có nhiều tính năng hỗ trợ tất cả những gì bạn cần trong việc xây dựng Web và API
- Quản lý các route dễ dàng
- Cung cấp một nền tảng phát triển cho các API
- Hỗ trợ nhiều thư viện và plugin
- Bảo mật và an toàn hơn so với việc code thuần
- Hỗ trợ cộng đồng tuyệt vời.

1.2. CƠ SỞ DỮ LIỆU

MongoDB là database mã nguồn mở NoSQL hỗ trợ đa nền tảng được thiết kế theo hướng đối tượng. Các ‘bảng’ (trong MongoDB gọi là collection) có cấu trúc linh hoạt cho phép dữ liệu không cần tuân theo dạng cấu trúc nào. Nó có thể dùng để lưu trữ dữ liệu có cấu trúc phức tạp và đa dạng. Đặc biệt, chương trình này lưu trữ dữ liệu vào collection theo hướng tài liệu kiểu JSON thay vì bảng nên có hiệu suất cao và tính khả dụng cao.

MongoDB lần đầu ra đời bởi MongoDB Inc., tại thời điểm đó là thế hệ 10, vào tháng Mười năm 2007, nó là một phần của sản phẩm PaaS (Platform as a Service) tương tự như Windows Azure và Google App Engine. Sau đó nó đã được chuyển thành nguồn mở từ năm 2009.

Ưu điểm của MongoDB

- Dữ liệu lưu trữ phi cấu trúc, không có tính ràng buộc, toàn vẹn nên tính sẵn sàng cao, hiệu suất lớn và dễ dàng mở rộng lưu trữ.
- Dữ liệu được caching (ghi đệm) lên RAM, hạn chế truy cập vào ổ cứng nên tốc độ đọc và ghi cao.

Nhược điểm của MongoDB

- Không ứng dụng được cho các mô hình giao dịch nào có yêu cầu độ chính xác cao do không có ràng buộc.
- Không có cơ chế transaction (giao dịch) để phục vụ các ứng dụng ngân hàng.
- Dữ liệu lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn.
- Mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

Mongoose là một **Object Document Mapper (ODM)**. Do đó, Mongoose cho phép định nghĩa các object (đối tượng) với một schema được định nghĩa rõ ràng, được ánh xạ tới một MongoDB document.

Mongoose là một framework JavaScript thường được sử dụng trong ứng dụng Node.js với database là MongoDB.

1.3. MÔI TRƯỜNG DEPLOY

1.3.1. Heroku

Heroku là nền tảng đám mây cho phép các lập trình viên xây dựng, triển khai, quản lý và mở rộng ứng dụng (PaaS – Platform as a service).

Nó rất linh hoạt và dễ sử dụng, cung cấp cho một con đường đơn giản nhất để đưa sản phẩm tiếp cận người dùng. Nó giúp các nhà phát triển tập trung vào phát triển sản phẩm mà không cần quan tâm đến việc vận hành máy chủ hay phần cứng...

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ VÀ TRIỂN KHAI HỆ THỐNG

2.1. KHẢO SÁT VÀ ĐẶC TẢ YÊU CẦU

2.1.1. Phần quản lý trong admin

- Quản lý người dùng
- Quản lý bài viết
- Quản lý hashtag
- Quản lý comment
- Bao gồm tất cả chức năng của người dùng thông thường.

2.1.2. Giao diện người dùng

- Website có giao diện đẹp, dễ nhìn, không quá phức tạp.
- Font chữ đơn giản, dễ nhìn, màu sắc hài hòa.
- Thanh menu thật đơn giản.
- An toàn và bảo mật dữ liệu.
- Kiểm tra website có tương thích với các trình duyệt phổ biến hay không.

2.2. YÊU CẦU PHI CHỨC NĂNG

Bảng 1 Các yêu cầu phi chức năng của ứng dụng

Mục	Tên yêu cầu	Mô tả yêu cầu
1	Giao diện	Giao diện hệ thống phải dễ sử dụng, trực quan, thân thiện với người dùng.
2	Tốc độ xử lý	Hệ thống phải xử lý nhanh chóng và chính xác.
3	Bảo mật	Tính bảo mật và độ an toàn cao.
4	Tương thích	Tương thích với đa phần các trình duyệt web hiện tại.

2.3. Phân tích thiết kế hệ thống

2.3.1. Các chức năng của hệ thống

Các chức năng của Admin:

- Quản lý user
- Quản lý bài viết
- Quản lý hastag

Các chức năng của người dùng:

- Đăng ký tài khoản
- Đăng nhập tài khoản
- Xem bài viết
- Tìm kiếm bài viết
- Đăng bài viết
- Chỉnh sửa bài viết
- Xóa bài viết
- Chỉnh sửa thông tin cá nhân
- Bình luận
- Bày tỏ cảm xúc bài viết
- Theo dõi người dùng

2.3.2. Các tác nhân của hệ thống

Dựa vào mô tả trên, ta có thể nhận thấy được các tác nhân chính của hệ thống như sau:

Tác nhân Admin: người thực hiện kiểm tra các báo xấu, từ đó xóa những bài viết vi phạm tiêu chuẩn trang web. Nếu người dùng đó vi phạm nhiều lần, Admin có quyền khóa tài khoản người dùng đó.

Tác nhân người dùng: người dùng là người có tài khoản hoặc là khách vãng lai. Nếu là khách vãng lai, người đó chỉ có thể xem bài viết, tìm kiếm bài viết. Còn đối với người dùng có tài khoản, người đó sẽ có thêm quyền đăng bài, bình luận, bày tỏ cảm xúc, theo dõi người khác và chỉnh sửa thông tin cá nhân.

- Chức năng Admin

Bảng 2: Chức năng của Admin

Mục	Tên chức năng	Mô tả
1	Xóa bài viết	Thực hiện xóa bài viết của người dùng. Bài viết này thường sẽ là những bài viết bị báo cáo và vi phạm tiêu chuẩn trang web

2	Khóa tài khoản người dùng	Thực hiện khóa tài khoản của người dùng. Người dùng sẽ không có quyền đăng nhập. Từ đó, sẽ không thể thực hiện đăng bài, bình luận, Upvote, Downvote...
3	Xác minh tài khoản	Thực hiện xác minh lại tài khoản cho người dùng nếu người dùng chỉnh sửa hoặc quên tài khoản
4	Phê duyệt bài viết	Cho phép đăng bài nếu người dùng đăng nội dung phù hợp mục đích của trang web
5	Thống kê bài viết theo hashtag	Liệt kê số lượng bài viết liên quan đến hashtag
6	Thống kê số lượng người dùng	Liệt kê số lượng người dùng đăng ký tài khoản trang web

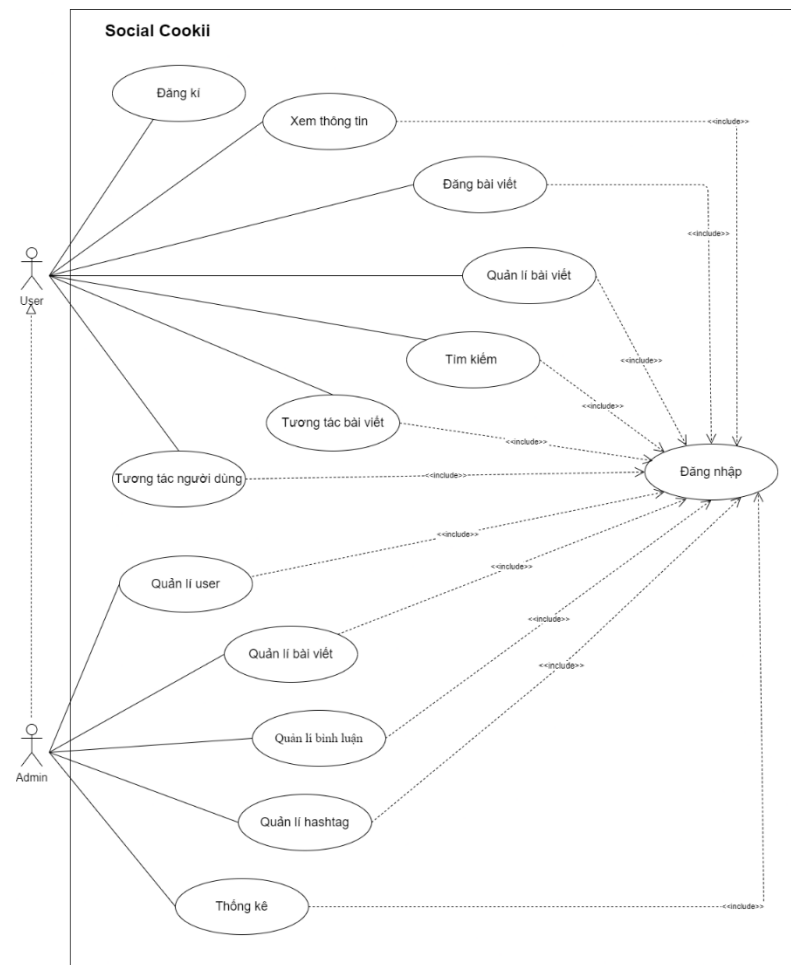
- Chức năng người dùng

Bảng 2: Chức năng của người dùng

Mục	Tên chức năng	Mô tả
1	Đăng ký tài khoản	Người dùng có thể đăng ký tài khoản dựa trên các thông tin: Username, Fullname, Email, Phone, Gender, Date of birth , Password, Confirm Password.
2	Đăng nhập tài khoản	Người dùng có thể đăng nhập tài khoản dựa trên username và password
3	Xem bài viết	Người dùng có thể xem các bài viết tại trang chủ
4	Tìm kiếm bài viết	Người dùng có thể tìm kiếm bài viết dựa trên tiêu đề của bài viết.
5	Chỉnh sửa thông tin cá nhân	Người dùng có thể cập nhật username, email, password, avatar.
6	Bình luận	Người dùng có thể bình luận vào dưới các bài viết.
7	Bày tỏ cảm xúc	Người dùng có thể bày tỏ cảm xúc vào bài viết mà mình yêu thích để tăng lượt tương tác, ủng hộ cho tác giả.
8	Theo dõi người khác	Người dùng có thể chọn theo dõi những người mình thích để có thể xem được nhiều bài viết và không bỏ lỡ bài viết của họ.
9	Chỉnh sửa bài viết	Người dùng có thể chỉnh sửa bài viết của mình
10	Xóa bài viết	Người dùng có thể xóa bài viết của mình

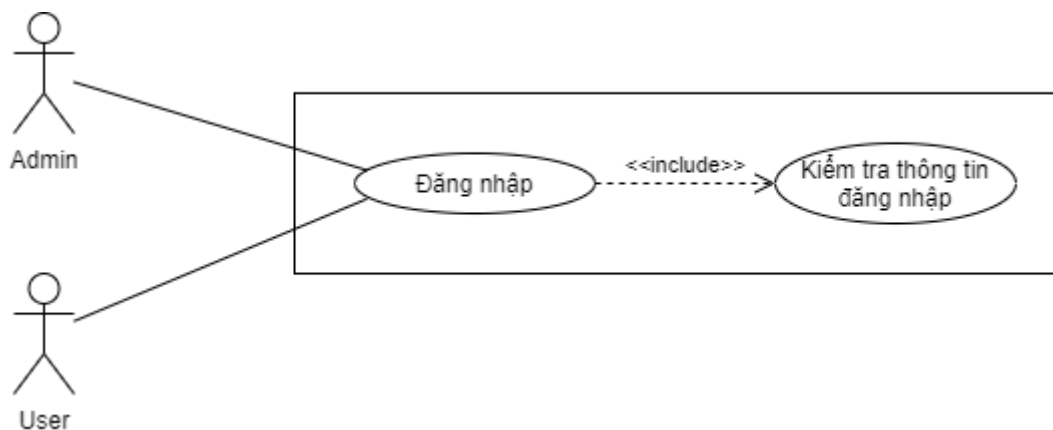
2.4. USE CASE

2.4.1. Use case tổng quát



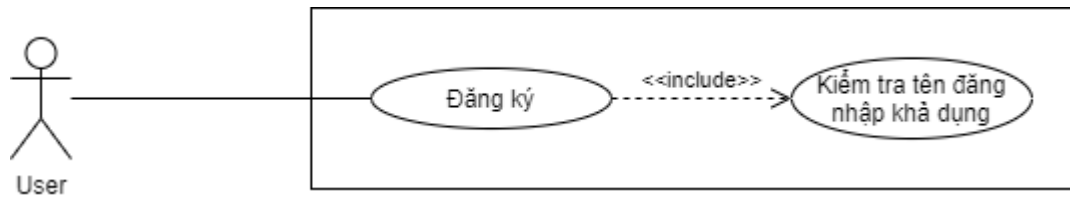
Hình 1 Use case tổng quát

2.4.2. Use case đăng nhập



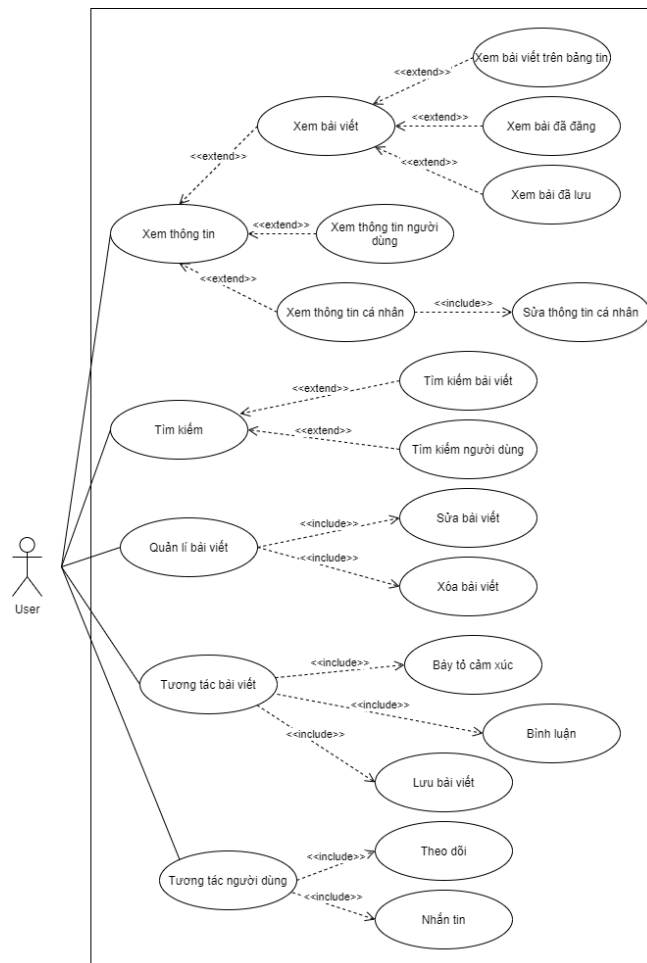
Hình 2 Use case đăng nhập

2.4.3. Use case đăng ký



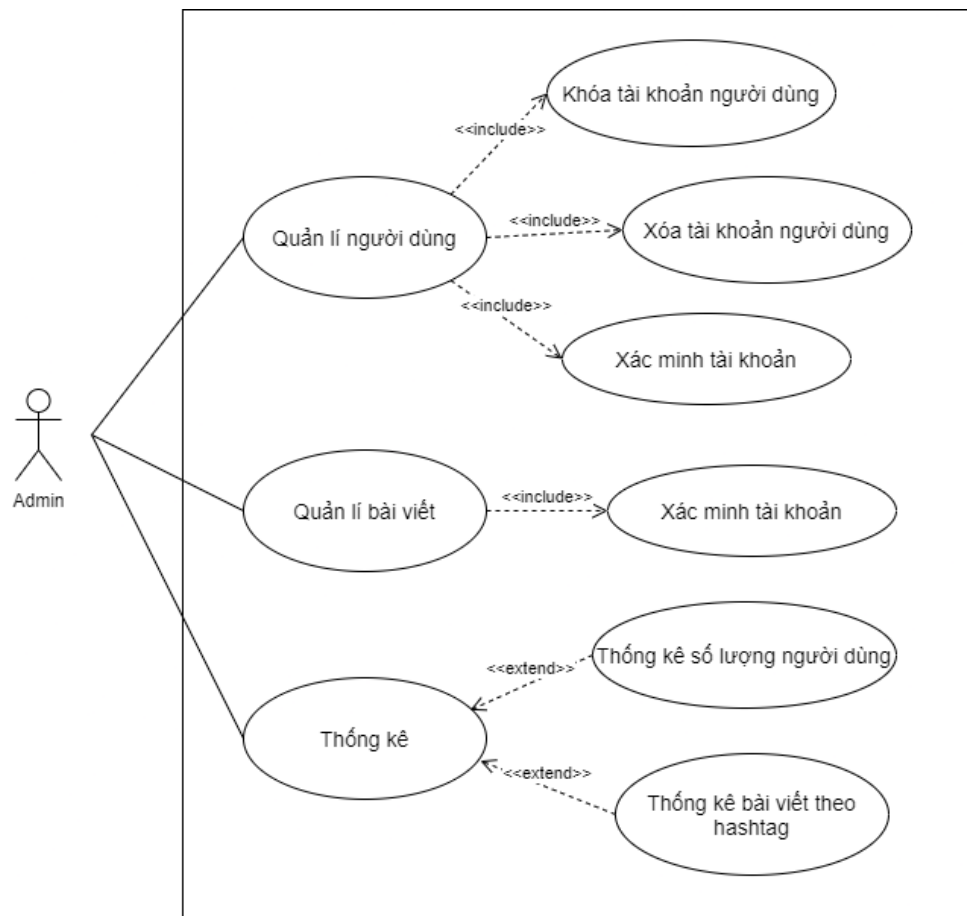
Hình 3 Use case đăng ký

2.4.4. Use case hiển thị chức năng của user:



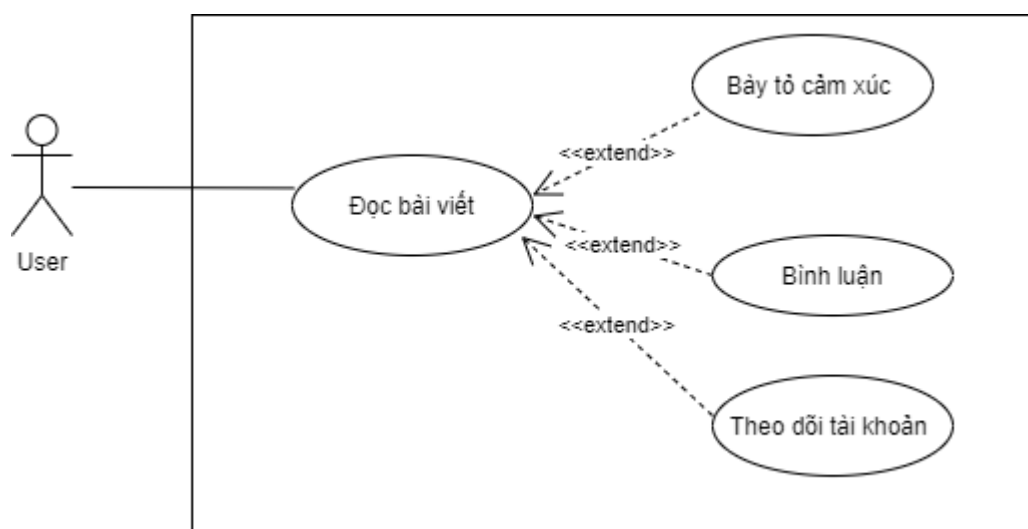
Hình 4 Use case hiển thị chức năng của user

2.4.5. Use case hiển thị chức năng của admin:



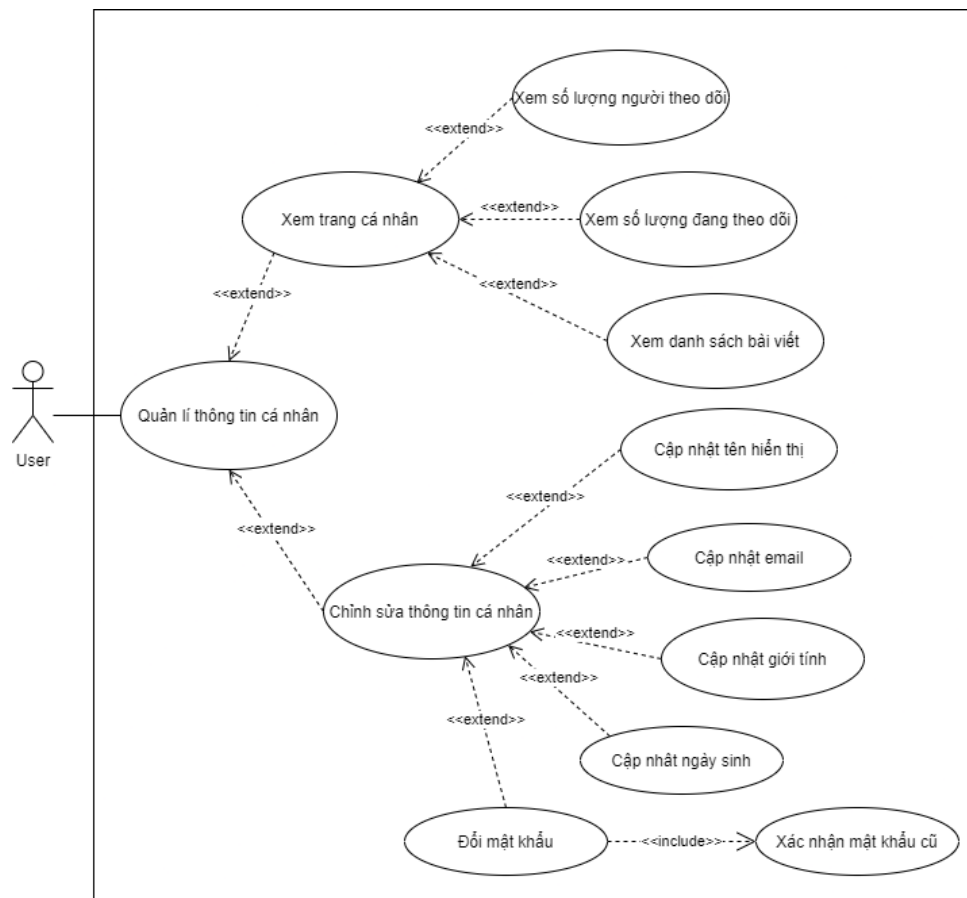
Hình 5 Use case hiển thị chức năng của admin

2.4.6. Use case xem bài viết



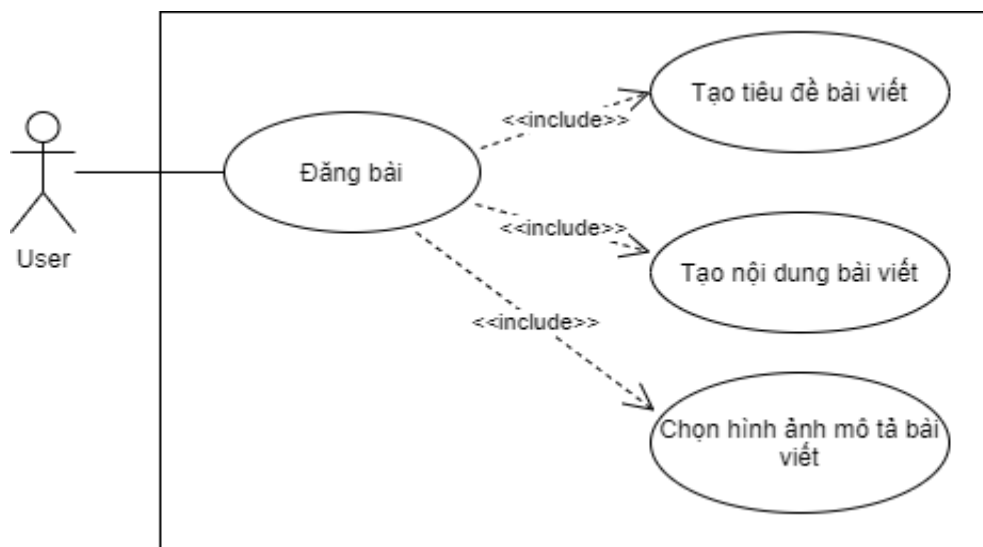
Hình 6 Use case xem bài viết

2.4.7. Use case quản lý thông tin cá nhân



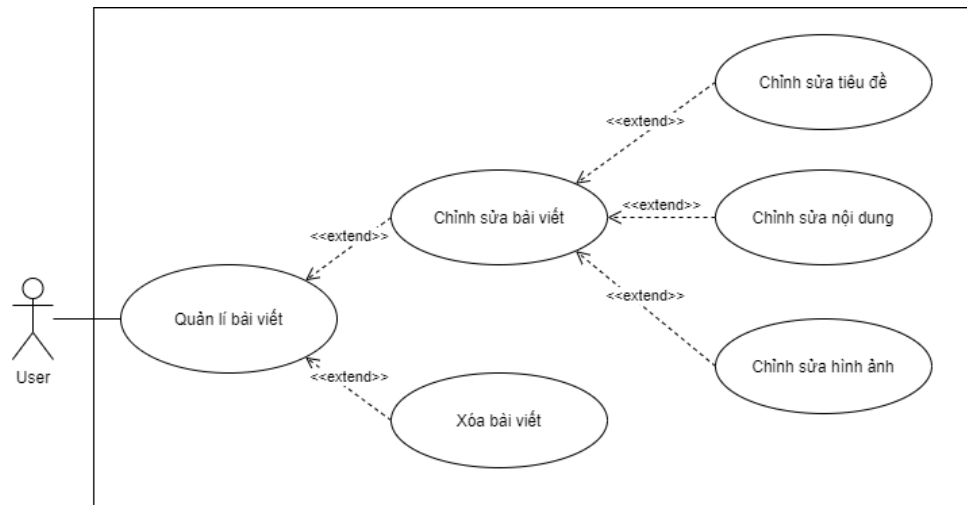
Hình 7 Use case quản lý trang cá nhân

2.4.8. Use case đăng bài viết:



Hình 8 Use case đăng bài viết

2.4.9. Use case quản lý bài viết



Hình 9 Use case quản lý bài viết

2.5. THIẾT KẾ CƠ SỞ DỮ LIỆU

2.5.1. Danh sách các bảng

Bảng 3: Danh sách các bảng cơ sở dữ liệu

STT	Tên bảng	Mô tả
1	User	Lưu thông tin danh sách các User
2	Post	Lưu thông tin các bài đăng
3	Comment	Lưu thông tin các bình luận
4	Hashtag	Lưu thông tin các hashtag
5	Friend	Lưu thông tin các bạn bè của User
6	Emotion	Lưu thông tin các cảm xúc của hệ thống
7	Notification	Lưu thông tin các thông báo

2.5.2. Bảng User

Bảng 4: Bảng User

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	_id	ObjectId	ID của User
2	username	String	Tên đăng nhập của user
3	fullname	String	Tên người dùng đầy đủ
4	email	String	Email của user

5	password	String	Mật khẩu đã được mã hoá
6	gender	String	Giới tính của User
7	birthday	String	Ngày sinh của User
8	phone	String	Số điện thoại của User
9	createAt	String	Ngày tạo tài khoản
10	updateAt	String	Ngày cập nhật tài khoản
11	isActive	Boolean	Trạng thái của User
12	verificationCode	String	Xác minh tài khoản
13	role	String	Vai trò của người dùng
14	avatar	String	Hình đại diện của người dùng
15	followed	Array	Danh sách tài khoản mà người dùng theo dõi
16	following	Array	Danh sách tài khoản theo dõi người dùng

2.5.3. Bảng Post

Bảng 5: Bảng Post

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	_id	ObjectId	ID của bài viết
2	title	String	Tiêu đề bài viết
3	description	String	Mô tả bài viết
4	createdBy	ObjectId	Id của người viết
5	createdAt	Date	Ngày tạo bài viết
6	deletedAt	Date	Ngày xóa bài viết
7	editedAt	Date	Ngày sửa bài viết
8	thumbnails	Array	Danh sách hình ảnh của bài viết
9	recipe	Array	Công thức của món ăn
10	videos	Array	Danh sách video của bài viết
11	comments	ObjectId	Danh sách chứa bình luận của bài viết
12	likeUserIds	ObjectId	Danh sách chứa Id của người like bài viết
13	state	String	Trạng thái bài đăng

14	hashtagIds	ObjectId	Danh sách chứa Id của các hashtag trong bài viết
----	------------	----------	--------------------------------------------------

2.5.4. Bảng comment

Bảng 6: Bảng comment

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	_id	ObjectId	ID của bình luận
2	postId	ObjectId	Id của bài viết
3	userId	ObjectId	Id của người bình luận
4	content	String	Nội dung của bình luận
5	createdAt	Date	Ngày bình luận
6	editedAt	Date	Ngày sửa bình luận
7	deletedAt	Date	Ngày xóa bình luận
8	repliedCommentId	ObjectId	Id của những bình luận trả lời
9	likeUserIds	ObjectId	Danh sách Id của người like bình luận

2.5.5. Bảng hashtag

Bảng 7: Bảng hashtag

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	_id	ObjectId	ID của hashtag
2	hashtagName	String	Tên hashtag

2.5.6. Bảng friend

Bảng 8: Bảng friend

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	_id	ObjectId	ID của friend
2	isFriend	Boolean	Trạng thái bạn bè
3	senderUserId	ObjectId	ID của người gửi lời mời kết bạn
4	receiverUserId	ObjectId	ID của người nhận lời mời kết bạn
5	isPending	Boolean	Trạng thái chờ xử lý
6	deletedAt	Date	Ngày xóa friend
7	createdAt	Date	Ngày tạo friend

2.5.7. Bảng emotion

Bảng 9: Bảng emotion

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	_id	ObjectId	ID của Emotion
2	emotionName	String	Tên của Emotion
3	imageFilename	String	Hình ảnh của Emotion

2.5.8. Bảng notification

Bảng 10: Bảng notification

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	_id	ObjectId	ID thông báo
2	url	String	Đường dẫn của thông báo
3	content	String	Nội dung thông báo
4	senderUserId	ObjectId	ID người gửi thông báo
5	receiveUserId	ObjectId	ID người nhận thông báo
6	isSeen	Boolean	Trạng thái xem thông báo hay chưa
7	createdAt	Date	Thời gian thông báo
8	deletedAt	Date	Thời gian xóa thông báo

CHƯƠNG 3: CÀI ĐẶT VÀ KẾT QUẢ ĐẠT ĐƯỢC

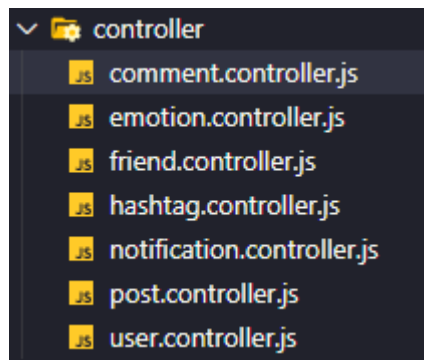
3.1. CÀI ĐẶT

3.1.1. Backend

Sau khi thiết kế database, chúng tôi cài đặt phần Backend bằng NodeJS, ExpressJS và MongoDB.

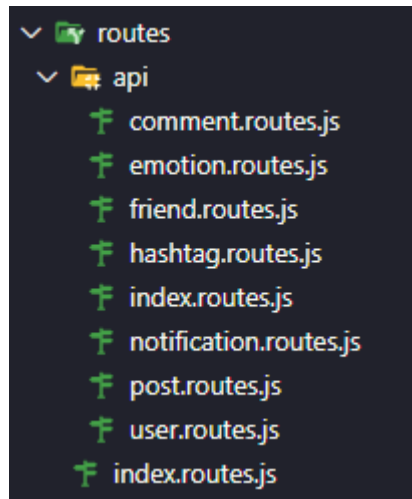
Cách cài đặt code:

1. Tiến hành xây dựng các đối tượng Model Mongoose đã được thiết kế ở trên.
2. Xây dựng các service giao tiếp với database
3. Xây dựng các controller gửi về các trạng thái và nội dung hồi đáp khi nhận được các yêu cầu (request) từ Client. Trong chương trình này, đa số các yêu cầu đều cần phải truy cập vào cơ sở dữ liệu. Vì vậy lớp này sẽ giao tiếp với lớp service.



Hình 10 Cấu trúc cây controller

4. Xây dựng các đường route cho Server.
 - Mỗi một collection có một đường route riêng để các client có thể thực hiện gửi yêu cầu đến.
 - Ngoài những đường route cho các collection, xây dựng thêm các đường route truy cập đến các file cho phép truy cập.



Hình 11 Cấu trúc cây route

5. Cài đặt server như truy cập cơ sở dữ liệu, cho phép truy cập dữ liệu, cho phép đọc gói tin...
6. Cấu hình các lệnh khởi chạy server trong file **Package.json** và thực hiện khởi chạy server.
7. Test các API của Server
 - Để đạt được sự hiệu quả, chính xác của API và tiết kiệm thời gian, công sức trong thời gian dò lỗi, chúng ta cần kiểm thử (test) các API, trước khi thực hiện gửi yêu cầu đến từ Client. Chúng tôi sử dụng đến phần mềm Postman, đồng thời sử dụng đến thư viện Chai. Bạn cũng có thể thực hiện kiểm thử trên Remote Repository với CircleCI.



Hình 12 Logo Postman

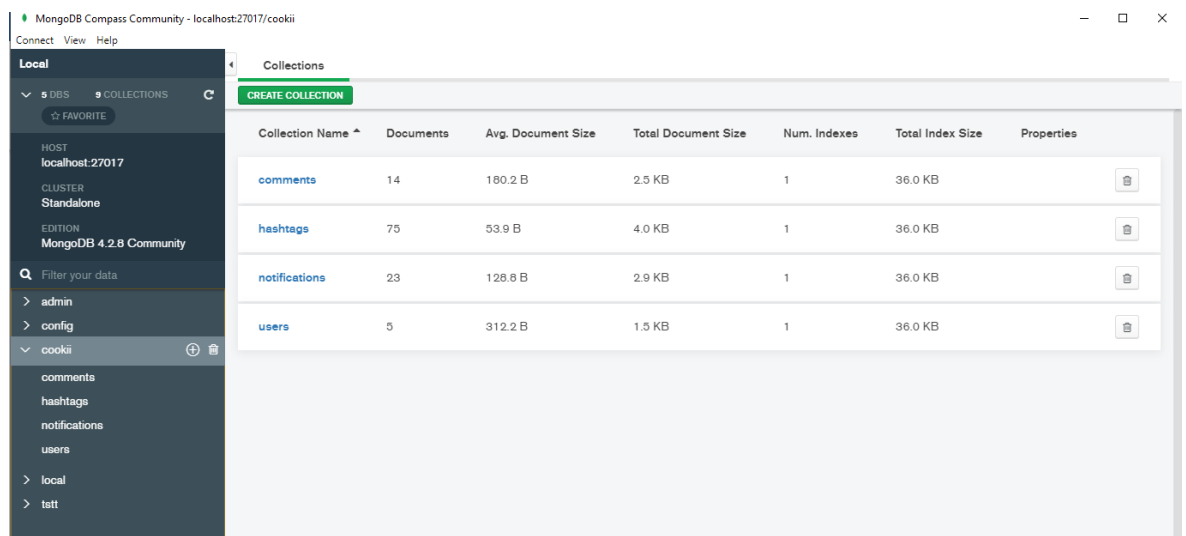


Hình 13 Logo Chai



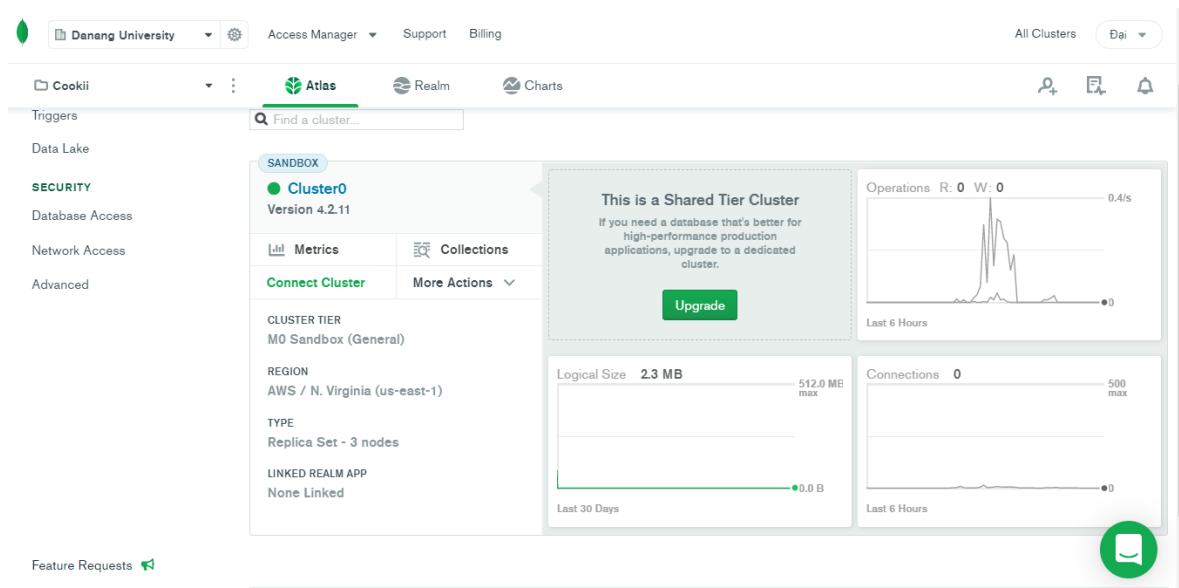
Hình 14 Logo CircleCI

Về cơ sở dữ liệu, trong quá trình phát triển ứng dụng chúng tôi sử dụng ứng dụng MongoDB Compass Community để lưu trữ và truy vấn dữ liệu. Việc sử dụng ứng dụng local, sẽ đảm bảo tốc độ truy cập nhanh và linh hoạt, tùy ý cập nhật.



Hình 15 Màn hình MongoDB Compass Communication

Trong quá trình deploy ứng dụng, chúng tôi sử dụng MongoDB Cloud để tiện cho việc truy cập và đồng bộ dữ liệu. Tuy nhiên, tồn tại một số nhược điểm riêng như tốc độ truy cập không cao, kém linh hoạt.



Hình 16 Giao diện của MongoDB Cloud

3.1.2. FrontEnd

8. Cài đặt các template của chương trình.
 - Dựa vào những kiến thức HTML, CSS và Javascript đã được học, chúng tôi tiến hành cài đặt các template của chương trình theo những usecase thiết kế.
9. Cài đặt các Component React
 - Bước này là bước khá phức tạp và mất nhiều thời gian.
 - Thực hiện các bước gửi yêu cầu đến API Server để render ra kết quả trên giao diện của người dùng.
10. Cài đặt chuyển trang, liên kết trang dựa vào thư viện 'react-router-dom'

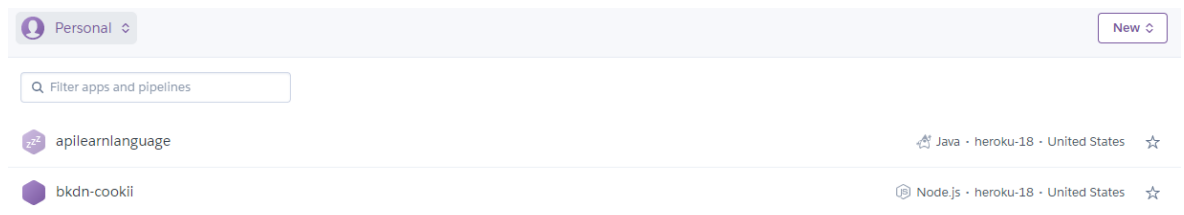
3.1.3. Deploy chương trình

Chúng tôi chọn Heroku là môi trường để deploy sản phẩm

1. Cầu hình lệnh chạy và các biến môi trường sẵn

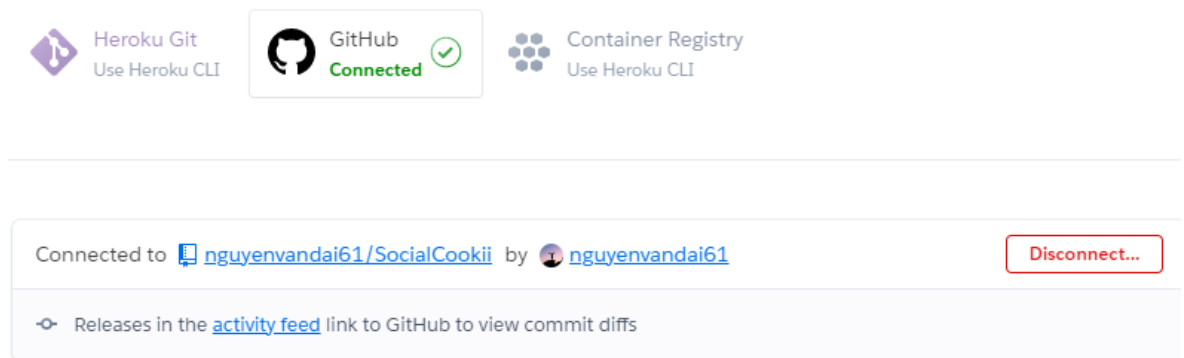
➔ `Process.MONGO_URI = mongodb+srv://teamanmy:<password>@cluster0.jojjg.mongodb.net/test`

2. Sử dụng GitHub để lưu giữ Source code.
3. Đăng nhập vào Heroku, rồi tạo app mới ở Dashboard



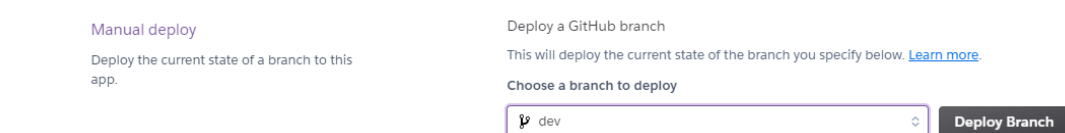
Hình 17 Tạo app mới ở Heroku

4. Chọn GitHub là Deployment Method, sau đó chọn Repository muốn deploy



Hình 18 Chọn Repository để deploy trên Heroku

5. Chọn nhánh cần deploy, ở đây chúng tôi chọn nhánh “dev”



Hình 19 Chọn nhánh để deploy trên Heroku

6. Chương trình chạy các lệnh cấu hình và hiển thị các truy cập ở màn hình Application Logs

```
Application Logs ALL PROCESSES

2020-12-30T04:29:30.360716+00:00 app[web.1]: {}
2020-12-30T04:29:30.361642+00:00 heroku[router]: at=info method=GET path="/api/user/userInfo/5febce933f486b0017597447" host=bkdn-cookii.herokuapp.com request_id=1c8fe123-8ca7-4477-a493-c398264f912c fwd="180.93.33.58" dyno=web.1 connect=0ms service=5ms status=304 bytes=296 protocol=https
2020-12-30T04:29:30.472283+00:00 heroku[router]: at=info method=GET path="/api/post" host=bkdn-cookii.herokuapp.com request_id=e29e25bb-c882-4021-8858-7e6541a0f732 fwd="180.93.33.58" dyno=web.1 connect=1ms service=112ms status=304 bytes=300 protocol=https
2020-12-30T04:29:30.741399+00:00 app[web.1]: lala
2020-12-30T04:29:30.742607+00:00 heroku[router]: at=info method=GET path="/image/default/avatar/avatardefault.png" host=bkdn-cookii.herokuapp.com request_id=fbf6118b-5890-4184-830d-34f07cb95ba2 fwd="180.93.33.58" dyno=web.1 connect=0ms service=3ms status=304 bytes=384 protocol=https
2020-12-30T04:29:30.749260+00:00 app[web.1]: aa
2020-12-30T04:29:30.749335+00:00 app[web.1]: {}
2020-12-30T04:29:30.856599+00:00 heroku[router]: at=info method=GET path="/api/post" host=bkdn-cookii.herokuapp.com request_id=fe91a62f-e935-4e07-be05-ac28b1396337 fwd="180.93.33.58" dyno=web.1 connect=2ms service=108ms status=304 bytes=300 protocol=https
2020-12-30T04:29:31.021954+00:00 heroku[router]: at=info method=GET path="/manifest.json" host=bkdn-cookii.herokuapp.com request_id=a59a2d77-7ee7-419a-9ea4-b2acce34864d fwd="180.93.33.58" dyno=web.1 connect=2ms service=2ms status=304 bytes=383 protocol=https
2020-12-30T04:29:31.299900+00:00 heroku[router]: at=info method=GET path="/logo192.png" host=bkdn-cookii.herokuapp.com request_id=e7b2e68d-4c0b-4e6c-9bbf-c7093ff1fc23 fwd="180.93.33.58" dyno=web.1 connect=1ms service=3ms status=304 bytes=384 protocol=https

☒ Autoscroll with output Save
```

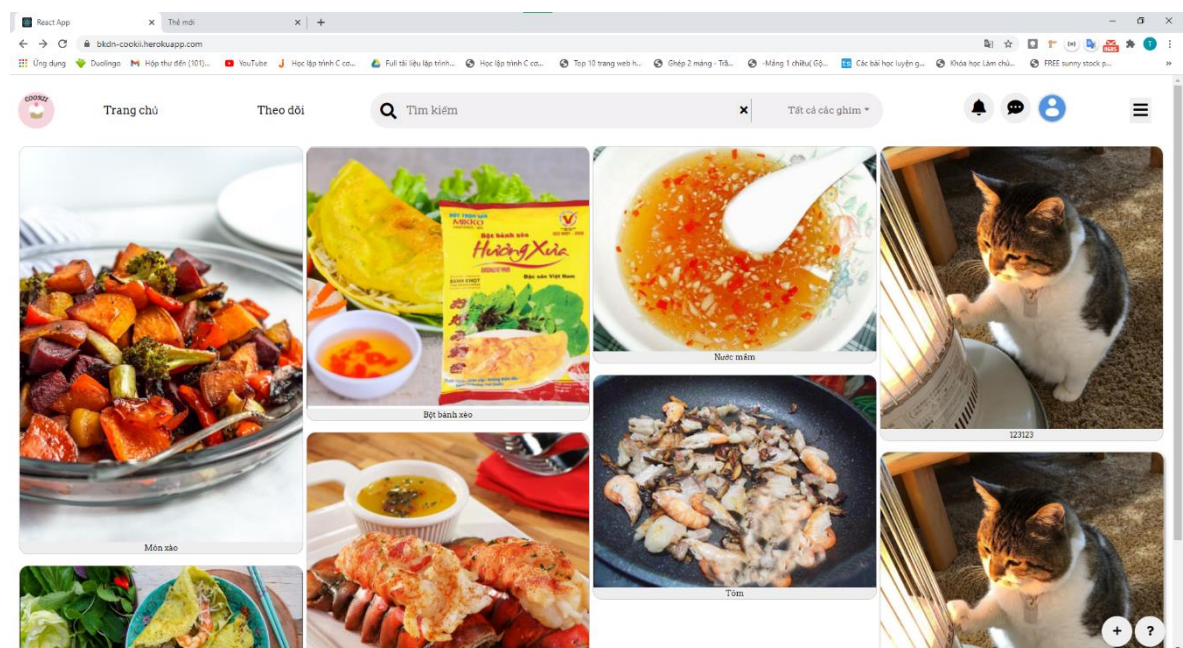
Hình 20 Màn hình Application Log

Sau khi deploy, địa chỉ của Server API sẽ là <https://bkdn-cookii.herokuapp.com>.

3.2. GIAO DIỆN TRANG WEB

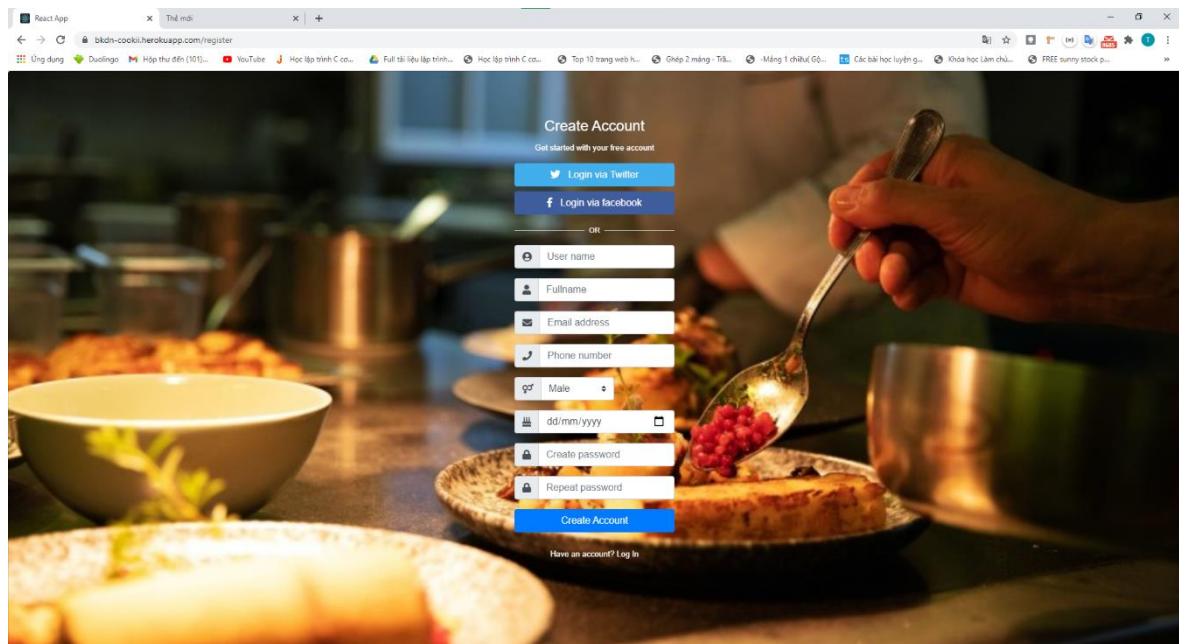
3.2.1. Giao diện người dùng

- Giao diện khi mở ứng dụng lần đầu



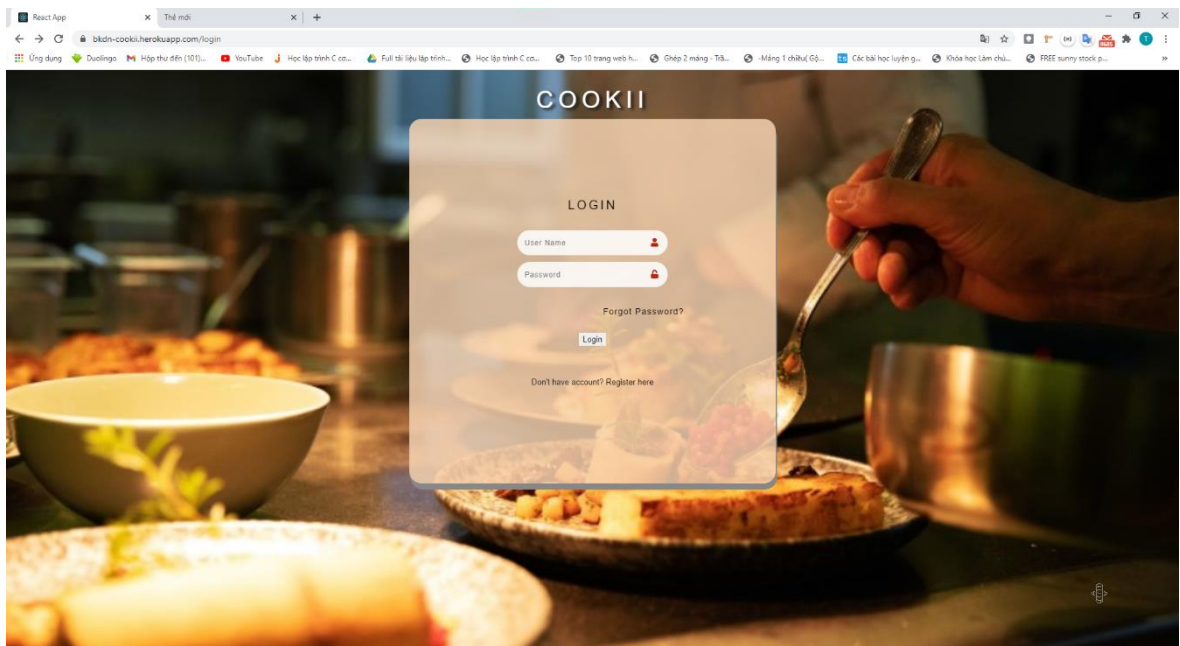
Hình 21 Giao diện khi mở ứng dụng lần đầu

- Giao diện đăng ký



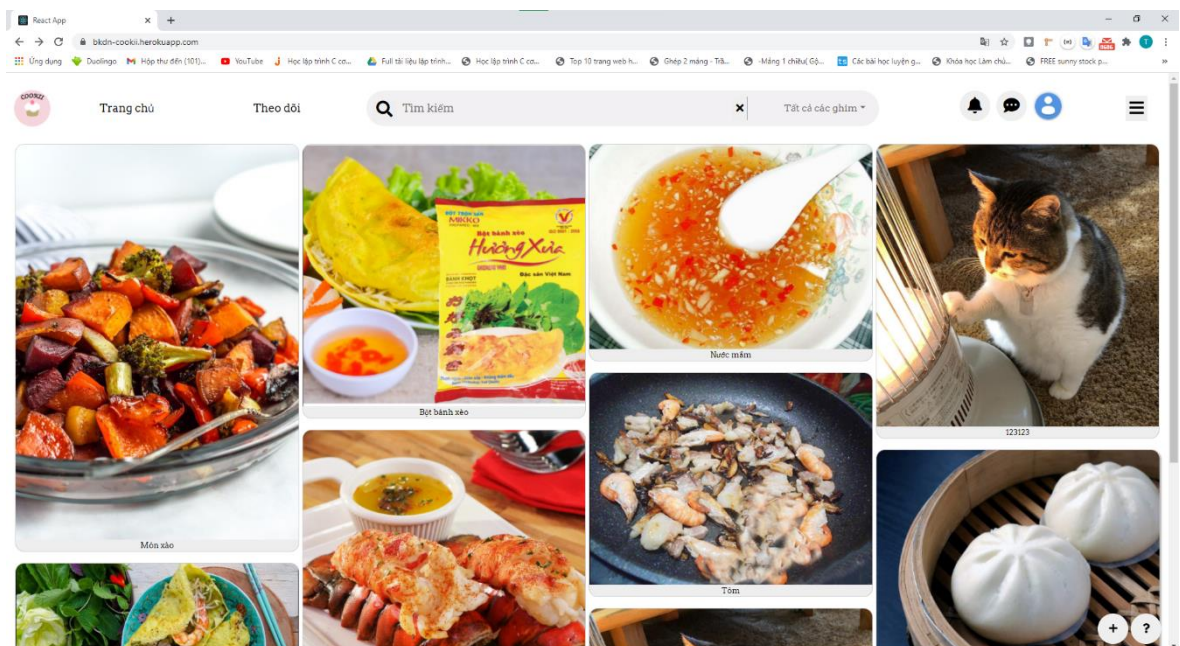
Hình 22 Giao diện đăng ký

- Giao diện đăng nhập



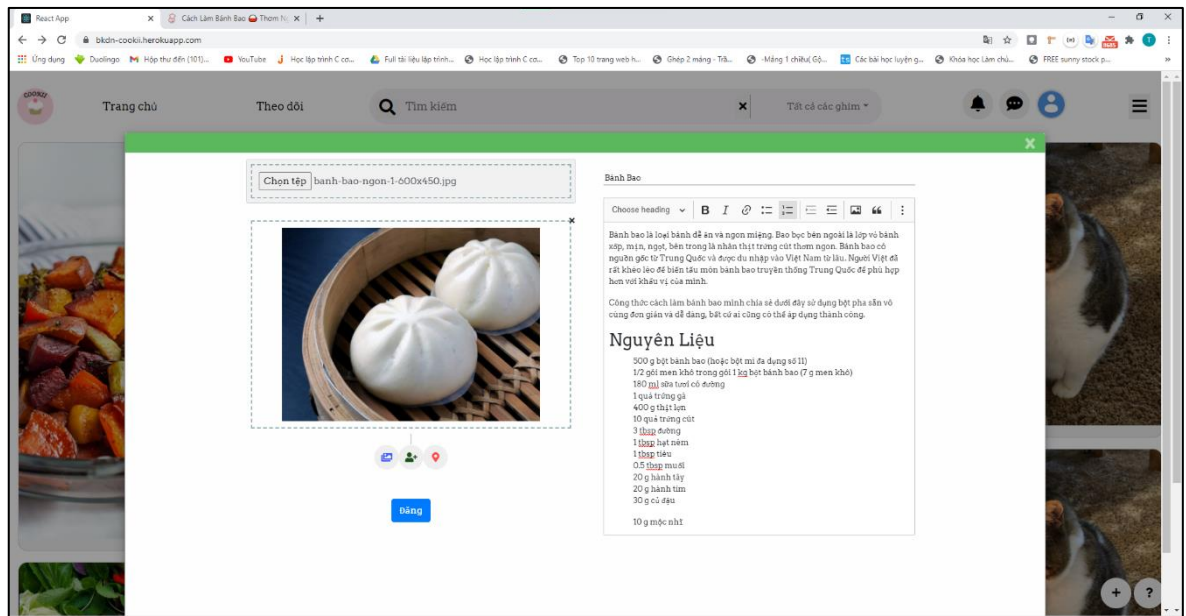
Hình 23 Giao diện đăng nhập

- Giao diện khi đăng nhập thành công



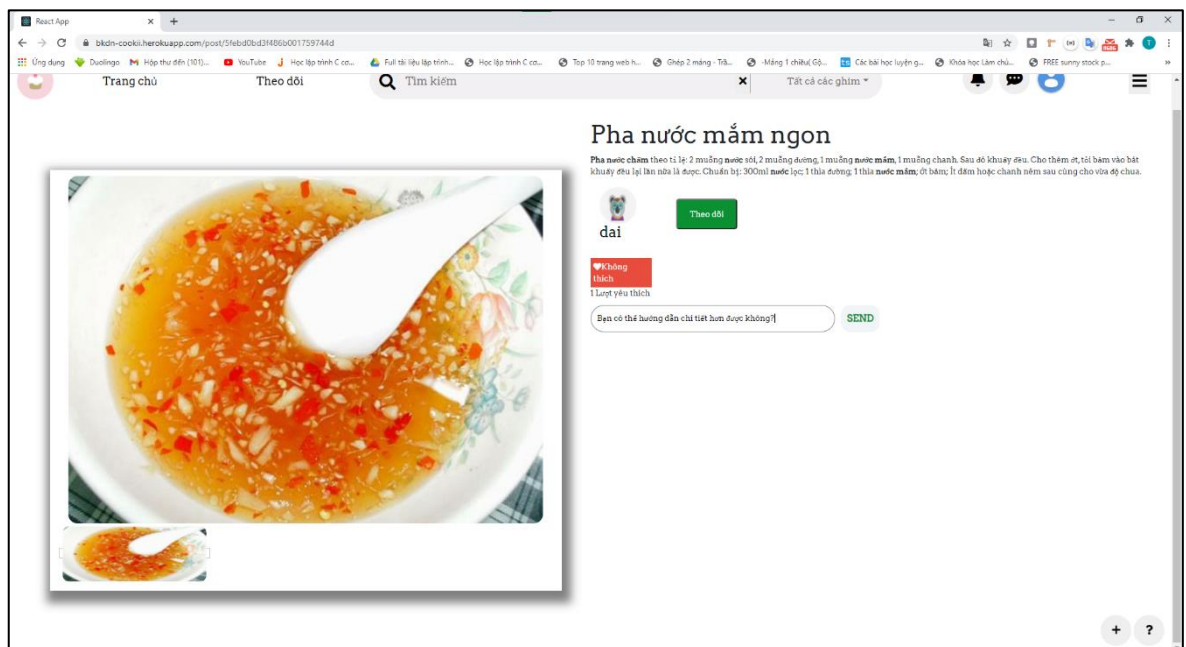
Hình 24 Giao diện khi đăng nhập thành công

- Giao diện thêm bài viết



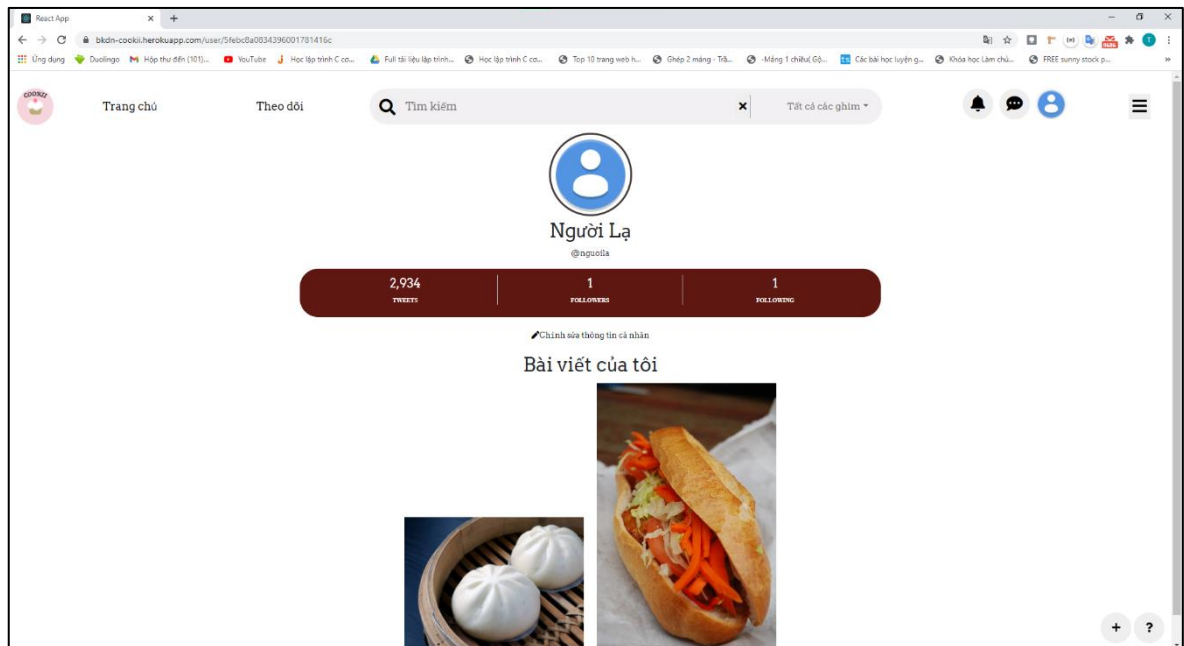
Hình 25 Giao diện thêm bài viết

- Giao diện xem chi tiết bài viết và comment



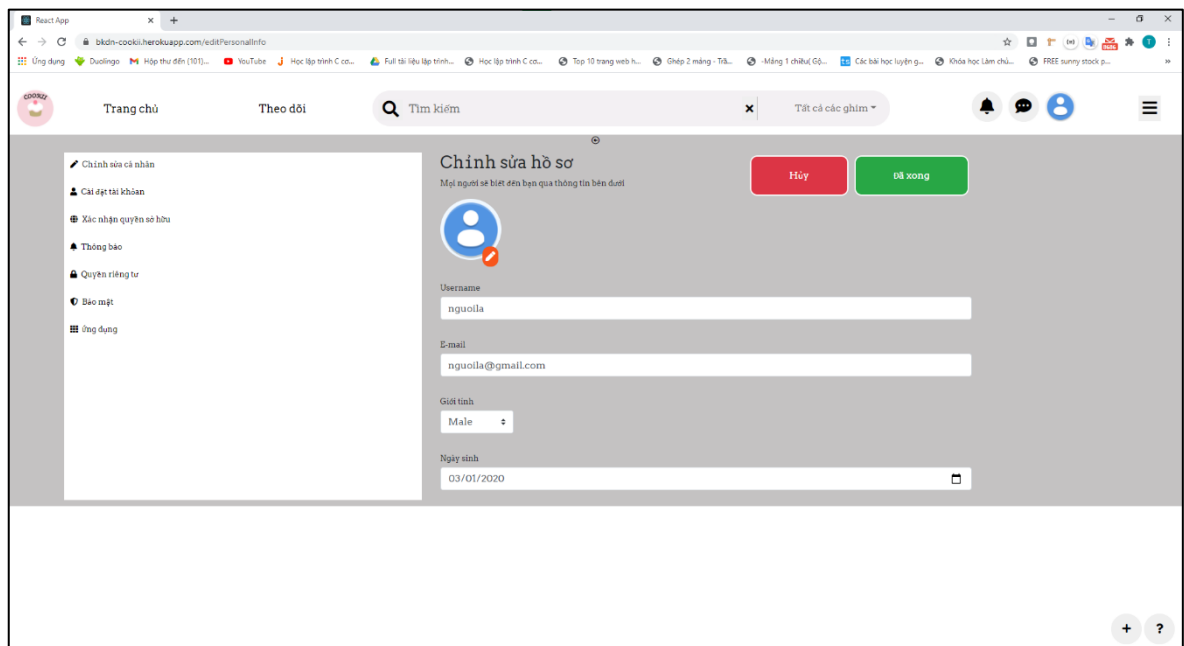
Hình 26 Giao diện xem chi tiết bài viết và comment

- Giao diện xem chi tiết trang cá nhân



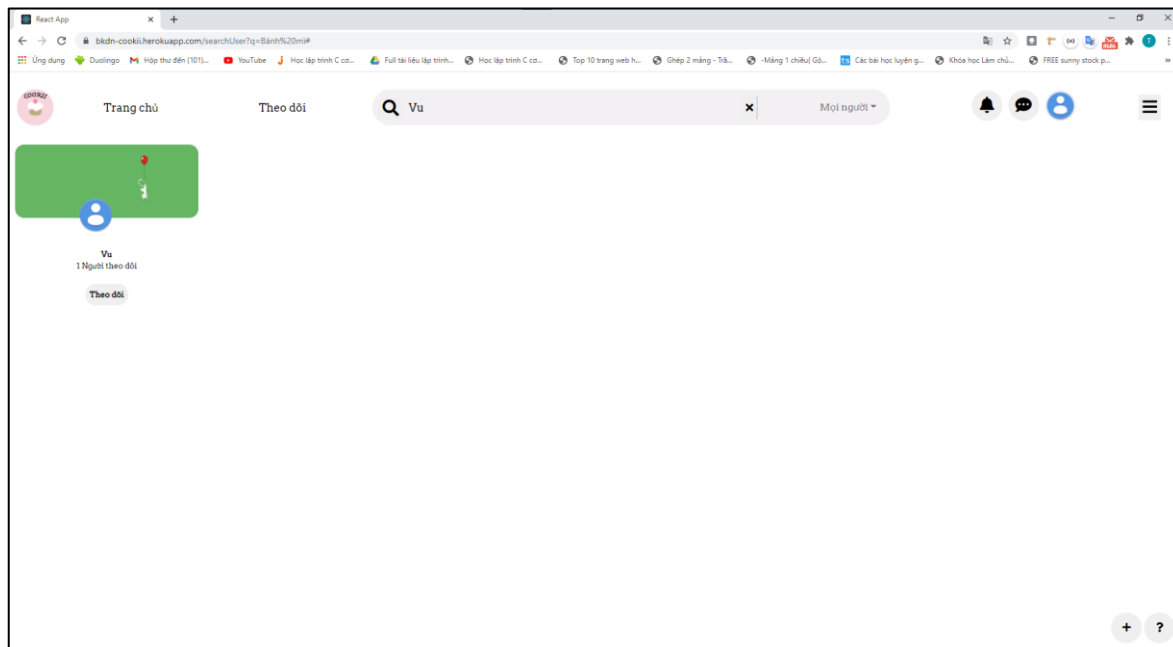
Hình 27 Giao diện xem chi tiết trang cá nhân

- Giao diện chỉnh sửa thông tin cá nhân



Hình 28 Giao diện chỉnh sửa thông tin cá nhân

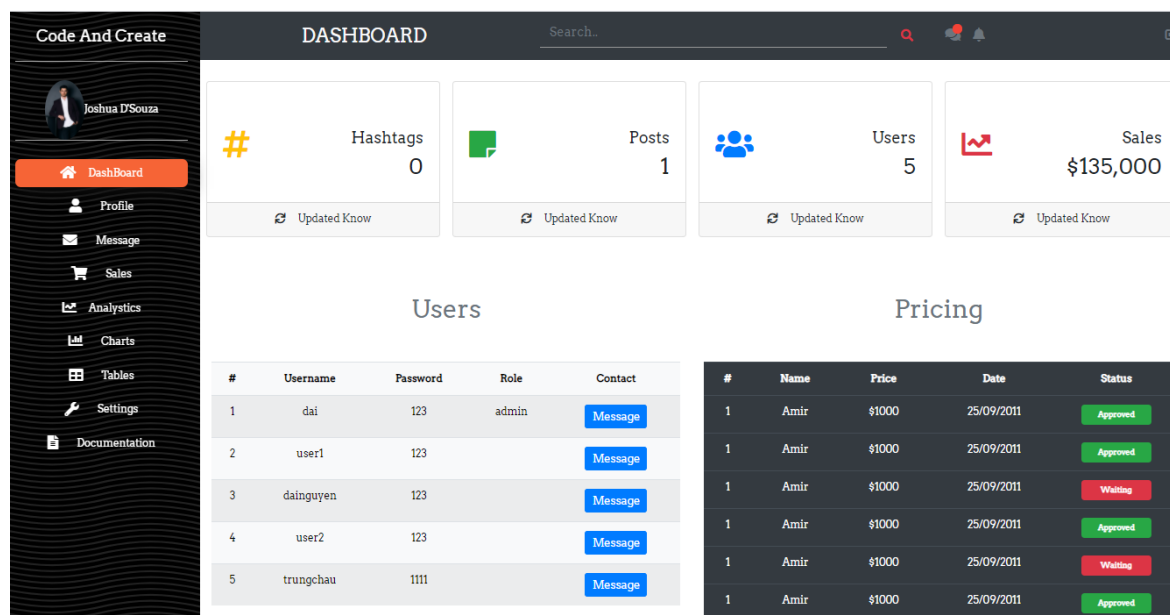
- Giao diện kết quả tìm kiếm



Hình 29 Giao diện kết quả tìm kiếm

3.2.2. Giao diện trang Admin

- Giao diện trang Admin



Hình 30 Giao diện trang Admin

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. KẾT QUẢ ĐẠT ĐƯỢC

Trong thời gian tìm hiểu, nghiên cứu cơ sở lý thuyết và triển khai ứng dụng công nghệ, đề tài đã đạt được những kết quả sau:

Về mặt lý thuyết: trang Web đã có được hầu hết các chức năng ở phần phân tích thiết kế hệ thống. Các chức năng chạy ổn định. Các yêu cầu phi chức năng đã được đáp ứng. Giao diện Web đẹp mắt, tốc độ xử lý khá nhanh, tính bảo mật ở mức khá. Trang Web chạy tốt trên nền tảng trình duyệt Web của cả máy tính và điện thoại.

Về mặt thực tiễn ứng dụng: sau khi hoàn thành đề tài, chúng em đã học được thêm rất nhiều kiến thức mới. Ngoài các kiến thức mới được học, các bạn trong nhóm đã học được cách làm việc nhóm, quản lý thời gian và sử dụng git để quản lý code một cách hiệu quả. Những kinh nghiệm này sẽ là nền tảng để giúp các chúng em nâng cao trình độ bản thân, kinh nghiệm trong thực tế để có thể làm việc ngoài các doanh nghiệp sau này.

Tuy nhiên, đề tài còn tồn tại các vấn đề như sau:

- Chưa có thông báo đến người dùng khi nhận được comment, like,...
- Chưa phát triển được hệ thống nhắn tin
- Chưa phát triển được hashtag cho bài viết
- Phần admin vẫn còn thiếu nhiều chức năng.

2. HƯỚNG PHÁT TRIỂN

Một số hướng nghiên cứu và phát triển của đề tài như sau:

- Phát triển tính năng thông báo đến người dùng khi nhận được comment, like,...
- Phát triển được hệ thống nhắn tin
- Gắn hashtag cho bài viết
- Phát triển thêm chức năng cho admin

TÀI LIỆU THAM KHẢO

Internet

- [1] <https://expressjs.com/>
- [2] <https://reactjs.org/>
- [3] <https://dashboard.heroku.com/apps/>
- [4] <https://topdev.vn/blog/heroku-la-gi/>
- [5] <https://cloud.mongodb.com>
- [6] Giao diện admin của Joshua D'Souza