

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for writing. There are no margins, text, or other markings on the page.

(Ký tên và ghi rõ họ tên)

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting or typing. There are no margins, text, or other markings on the page.

Thành viên hội đồng
(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Để hoàn thành đồ án chuyên ngành này, em xin tỏ lòng biết ơn đến giáo viên hướng dẫn người đã tận tình hướng dẫn trong suốt quá trình thực hiện đồ án.

Em xin chân thành cảm ơn những quý thầy, cô trong khoa Kỹ thuật và Công nghệ Trường Đại học Trà Vinh đã tận tình truyền đạt kiến thức trong những năm em học tập. Với những kiến thức được tiếp thu trong quá trình học không chỉ là nền tảng cho quá trình thực hiện đồ án mà còn là hành trang quý báu để em bước vào đời một cách vững chắc và tự tin.

Trong quá trình thực hiện đồ án và viết báo cáo, do còn hạn chế về kiến thức, báo cáo của em khó tránh khỏi những thiếu sót. Em kính mong nhận được những ý kiến đóng góp ý kiến từ thầy cô để có thể học hỏi thêm nhiều kinh nghiệm. Cuối cùng, em xin kính chúc quý thầy cô luôn mạnh khỏe và đạt nhiều thành công trong sự nghiệp giảng dạy.

MỤC LỤC

| | |
|--|------|
| LỜI CẢM ƠN | ii |
| MỤC LỤC | iv |
| DANH MỤC HÌNH ẢNH | vi |
| DANH MỤC BẢNG BIỂU | vii |
| TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH | viii |
| MỞ ĐẦU | 1 |
| CHƯƠNG 1 : TỔNG QUAN | 2 |
| 1.1 Đặt vấn đề | 2 |
| 1.2 Mục đích | 2 |
| CHƯƠNG 2 : NGHIÊN CỨU LÝ THUYẾT | 3 |
| 2.1 Ngôn ngữ JavaScript | 3 |
| 2.1.1 Khái niệm | 3 |
| 2.1.2 Ứng dụng của JavaScript | 3 |
| 2.2 Tailwind CSS | 4 |
| 2.2.1 Khái niệm | 4 |
| 2.2.2 Ưu điểm của Tailwind CSS | 4 |
| 2.3 React | 5 |
| 2.3.1 Khái niệm | 5 |
| 2.3.2 Cấu trúc cơ bản của Component | 5 |
| 2.3.3 JSX | 6 |
| 2.3.4 Virtual DOM | 6 |
| 2.3.5 Luồng dữ liệu một chiều | 6 |
| 2.3.6 State và Props | 6 |
| 2.3.7 Ưu điểm của React | 7 |
| 2.4 Express.js | 8 |
| 2.4.1 Khái niệm | 8 |
| 2.4.2 Đặc điểm của Express.js | 8 |
| 2.4.3 Ưu điểm của Express.js | 9 |
| 2.5 RESTful API | 9 |
| 2.5.1 Khái niệm | 9 |
| 2.5.2 Cách hoạt động của RESTful API | 10 |
| 2.6 Node.js | 11 |
| 2.6.1 Khái niệm | 11 |
| 2.6.2 Đặc điểm của Node.js | 11 |
| 2.6.3 Ưu điểm của Node.js | 12 |
| 2.7 Hệ quản trị cơ sở dữ liệu MongoDB | 13 |
| 2.7.1 Khái niệm | 13 |
| 2.7.2 Những tính năng nổi bật của MongoDB | 14 |
| 2.7.3 Ưu điểm của MongoDB | 15 |
| 2.7.4 Nhược điểm của MongoDB | 15 |
| CHƯƠNG 3 : HIỆN THỰC HÓA NGHIÊN CỨU | 16 |
| 3.1 Mô tả bài toán | 16 |
| 3.2 Yêu cầu chức năng | 16 |
| 3.2.1 Người dùng | 16 |
| 3.2.2 Admin | 16 |
| 3.3 Yêu cầu phi chức năng | 17 |
| 3.4 Cơ sở dữ liệu | 17 |
| 3.4.1 Các collection trong cơ sở dữ liệu | 18 |
| 3.4.2 Mối quan hệ giữa các collection trong hệ thống | 23 |
| 3.4.3 Sơ đồ use case | 23 |
| 3.5 Cài đặt Node.js và React | 24 |

| | | |
|---|--|----|
| 3.5.1 | Cài đặt Node.js..... | 24 |
| 3.5.2 | Cài đặt React..... | 25 |
| CHƯƠNG 4 : KẾT QUẢ NGHIÊN CỨU | | 26 |
| 4.1 | Trang chủ..... | 26 |
| 4.2 | Tìm kiếm công thức | 27 |
| 4.3 | Trang xem thông chi tiết công thức | 27 |
| 4.4 | Trang đăng nhập và đăng ký..... | 29 |
| 4.4.1 | Đăng nhập..... | 29 |
| 4.4.2 | Đăng ký | 29 |
| 4.4.3 | Cập nhật thông tin người dùng | 30 |
| 4.5 | Quản trị..... | 30 |
| 4.5.1 | Thêm công thức | 31 |
| 4.5.2 | Quản lý công thức | 31 |
| 4.5.3 | Chỉnh sửa công thức..... | 32 |
| 4.5.4 | Quản lý tài khoản người dùng | 34 |
| 4.5.5 | Quản lý bình luận..... | 34 |
| CHƯƠNG 5 : KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN | | 35 |
| 5.1 | Kết luận | 35 |
| 5.2 | Hướng phát triển | 35 |
| DANH MỤC TÀI LIỆU THAM KHẢO | | 36 |
| PHỤ LỤC | | 37 |

DANH MỤC HÌNH ẢNH

| | |
|--|-------------------------------------|
| Hình 2.1. Logo React | 5 |
| Hình 2.2. Express.js..... | 8 |
| Hình 2.3. Cách hoạt động của RESTful API | 10 |
| Hình 2.4. Node.js | 11 |
| Hình 2.5. MongoDB..... | 13 |
| Hình 3.1. Sơ đồ use case..... | 24 |
| Hình 3.2 Node.js LTS..... | 25 |
| Hình 4.1 Trang chủ..... | 26 |
| Hình 4.2 Trang tìm kiếm | 27 |
| Hình 4.3 Trang chi tiết công thức 1..... | 27 |
| Hình 4.4 Trang chi tiết công thức 2..... | 28 |
| Hình 4.5 Trang chi tiết công thức 3..... | 28 |
| Hình 4.6 Trang đăng nhập | 29 |
| Hình 4.7 Trang đăng ký | 29 |
| Hình 4.8 Cập nhật thông tin người dùng | 30 |
| Hình 4.9 Giao diện trang quản trị..... | 30 |
| Hình 4.10 Trang thêm công thức 1 | 31 |
| Hình 4.11 Trang thêm công thức 2 | 31 |
| Hình 4.12 Trang quản lý công thức..... | 32 |
| Hình 4.13 Trang chỉnh sửa công thức 1 | 32 |
| Hình 4.14 Trang chỉnh sửa công thức 2 | 33 |
| Hình 4.15 Trang quản lý tài khoản..... | 34 |
| Hình 4.16 Trang quản lý bình luận | 33 |
| Hình 4.17 Trang thống kê 1 | Error! Bookmark not defined. |
| Hình 4.18 Trang thống kê 2 | 34 |

DANH MỤC BẢNG BIỂU

| | |
|---|----|
| Bảng 3.1 Danh sách các collection..... | 17 |
| Bảng 3.2 Chi tiết các thuộc tính của collection ratings | 20 |
| Bảng 3.3 Chi tiết các thuộc tính của collection users..... | 20 |
| Bảng 3.4 Chi tiết các thuộc tính của collection comments..... | 21 |
| Bảng 3.5 Chi tiết các thuộc tính của collection visits | 21 |
| Bảng 3.6 Chi tiết các thuộc tính của collection recipes..... | 22 |

TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

1. Tóm tắt vấn đề nghiên cứu:

Website chia sẻ công thức nấu ăn là một website được xây dựng nhằm tạo ra một nền tảng trực tuyến thân thiện và tiện ích cho những người có chung sở thích cũng như đam mê nấu ăn. Website này cho phép người dùng đăng ký tài khoản và đăng nhập vào hệ thống. Sau khi đăng nhập người dùng có thể tìm kiếm công thức nấu ăn mình cần và để lại đánh giá, bình luận về những công thức nấu ăn của người khác.

2. Các hướng tiếp cận:

Phân tích yêu cầu của hệ thống: Xác định các chức năng cần có của website như đăng ký, đăng nhập, tìm kiếm công thức nấu ăn, đăng tải công thức nấu ăn và quản lý công thức nấu ăn.

Xây dựng cơ sở dữ liệu: Thiết kế các bảng như users, recipes, comments, ratings để lưu trữ thông tin người dùng, công thức nấu ăn, bình luận và đánh giá.

Phát triển giao diện: Sử dụng Framework React để xây dựng giao diện người dùng với nhiều chức năng và khả năng tương tác đa dạng với người dùng.

Sử dụng Node.js và Framework Express.js để xây dựng backend và phát triển các API cho phép thực hiện các thao tác với cơ sở dữ liệu như thêm, sửa, xóa dữ liệu.

3. Cách giải quyết vấn đề:

Xây dựng chức năng đăng ký và đăng nhập để xác thực người dùng.

Xây dựng chức năng đăng công thức cho phép đăng công thức nấu ăn kèm theo hình ảnh, video và các bước thực hiện.

Xây dựng chức năng tìm kiếm công thức nấu ăn dựa theo các thông tin như tên công thức, nguyên liệu, cách chế biến giúp người dùng dễ dàng tìm được công thức nấu ăn vừa ý.

Phát triển chức năng đánh giá và bình luận giúp người dùng có thể đánh giá công thức và bình luận ngay dưới mỗi công thức, giúp tăng tính tương tác và đóng góp ý kiến từ cộng đồng.

Xây dựng một trang quản lý công thức để chỉnh sửa, xóa hoặc cập nhật các công thức đã đăng.

4. Một số kết quả đạt được:

Có được kiến thức cơ bản và sử dụng thành thạo các công cụ, ngôn ngữ: Javascript, Tailwind CSS, React, Express.js, MongoDB.

Xây dựng được website chia sẻ công thức với giao diện thân thiện, trực quan với những tính năng như đăng ký, đăng nhập tài khoản. Tính năng quản lý, tìm kiếm, đánh giá, bình luận công thức nấu ăn.

MỞ ĐẦU

1. Lý do chọn đề tài:

Với sự phát triển và phổ biến của công nghệ, ngày càng có nhiều người có nhu cầu tìm kiếm, chia sẻ công thức nấu ăn và tìm hiểu các món ăn mới. Tuy nhiên không phải nền tảng nào cũng đáp ứng được các nhu cầu ở trên. Nên em quyết định thực hiện đề tài "Xây dựng website chia sẻ công thức nấu ăn" nhằm tạo ra một nền tảng có thể đáp ứng các nhu cầu này, giúp người dùng không chỉ tìm kiếm và tham khảo công thức nấu ăn mà còn tương tác, đóng góp và quản lý công thức của riêng mình. Đây sẽ là một ứng dụng hữu ích cho người yêu thích nấu ăn.

2. Mục tiêu:

Mục tiêu của đề tài là thiết kế và xây dựng một website chuyên về chia sẻ công thức nấu ăn với các tính năng như đăng ký, đăng nhập, đăng tải công thức, tìm kiếm công thức nấu ăn, và cho phép đánh giá, bình luận ở mỗi bài đăng công thức nấu ăn.

3. Đối tượng:

Đề tài tập trung vào việc nghiên cứu và phát triển các tính năng liên quan đến công thức nấu ăn.

4. Phạm vi nghiên cứu:

Chức năng chính của website: Các tính năng đăng ký/dăng nhập, đăng tải và quản lý công thức, tìm kiếm công thức theo từ khóa, nguyên liệu và kiểu chế biến, đánh giá và bình luận.

Thiết kế và phát triển website: Sử dụng công nghệ HTML, CSS, JavaScript cho giao diện; Node.js, Express.js và MongoDB cho back-end và cơ sở dữ liệu.

Phạm vi đối tượng sử dụng: Người dùng có tài khoản đăng nhập vào website, đăng tải công thức và tương tác với các bài viết của người dùng khác.

CHƯƠNG 1: TỔNG QUAN

1.1 Đặt vấn đề

Với sự phát triển của công nghệ và sự phổ biến của Internet, việc truy cập thông tin trở nên dễ dàng và tiện lợi hơn bao giờ hết. Điều này đã mở ra cơ hội cho nhiều lĩnh vực phát triển, trong đó có ngành ẩm thực. Hiện nay, nhu cầu tìm kiếm và chia sẻ các công thức nấu ăn qua các nền tảng trực tuyến đang gia tăng. Người dùng có xu hướng tra cứu các công thức nấu ăn mới để bổ sung vào thực đơn hàng ngày, học hỏi cách chế biến các món ăn và cải thiện kỹ năng nấu ăn của bản thân.

Tuy nhiên, khi tìm kiếm công thức trên các nền tảng khác nhau người dùng thường gặp một số khó khăn do một số công thức được chia sẻ không đầy đủ hoặc thiếu hướng dẫn chi tiết, gây khó khăn trong quá trình thực hiện. Có nhiều nền tảng không có tính năng quản lý công thức cũng như đánh giá hoặc bình luận về công thức.

Trước những hạn chế này, việc xây dựng một website chia sẻ công thức nấu ăn không chỉ đáp ứng nhu cầu của người dùng mà còn tạo cơ hội kết nối cộng đồng những người yêu thích nấu ăn. Website sẽ cung cấp một không gian cho phép người dùng tìm kiếm thông tin về các công thức nấu ăn, sau đó để lại đánh giá và bình luận của bản thân về công thức nấu ăn đó. Đây là nền tảng để thúc đẩy sự sáng tạo và chia sẻ kiến thức trong cộng đồng.

1.2 Mục đích

Cung cấp kho tàng công thức nấu ăn phong phú và đáng tin cậy cho phép người dùng đăng tải công thức nấu ăn kèm theo các hướng dẫn chi tiết giúp người xem dễ dàng hiểu và thực hiện.

Xây dựng các tính năng bình luận, đánh giá công thức giúp người dùng trao đổi ý kiến, đóng góp và học hỏi lẫn nhau về các món ăn và kỹ năng nấu nướng. Xây dựng chức năng tìm kiếm món ăn với các tiêu nguyên liệu, cách chế biến, tên món ăn, giúp người dùng nhanh chóng tìm thấy công thức phù hợp với nhu cầu. Cung cấp chức năng quản lý các công thức đã đăng tải, cho phép chỉnh sửa hoặc lưu trữ công thức một cách dễ dàng và thuận tiện.

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1 Ngôn ngữ JavaScript

2.1.1 Khái niệm

JavaScript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Từ làm mới bảng tin trên trang mạng xã hội đến hiển thị hình ảnh động và bản đồ tương tác, các chức năng của JavaScript có thể cải thiện trải nghiệm người dùng của trang web. Là ngôn ngữ kịch bản phía máy khách, JavaScript là một trong những công nghệ cốt lõi của World Wide Web [3].

JavaScript giúp tạo ra các trang web động bằng cách thay đổi nội dung của trang dựa trên hành động của người dùng, như nhấp chuột, gõ phím, hoặc các sự kiện khác. Nó cũng có thể được sử dụng để giao tiếp với các API (Interface lập trình ứng dụng) để lấy dữ liệu từ máy chủ và cập nhật nội dung trang mà không cần tải lại trang web.

2.1.2 Ứng dụng của JavaScript

JavaScript có nhiều ứng dụng và được sử dụng rộng rãi trong phát triển web và nhiều lĩnh vực khác. Sau đây là một số ứng dụng của JavaScript.

Phát triển web: JavaScript là ngôn ngữ chủ yếu cho phát triển phía client trong web. Nó giúp thêm các tương tác người dùng, hiệu ứng trực quan, và làm cho trang web trở nên động đầy mà không cần phải tải lại trang.

JavaScript không đồng bộ và XML: JavaScript được sử dụng để thực hiện các yêu cầu mà không làm tải lại trang, giúp cải thiện trải nghiệm người dùng bằng cách tải dữ liệu một cách linh hoạt.

Thiết kế giao diện người dùng: JavaScript cung cấp các công cụ và thư viện để tạo ra giao diện người dùng đẹp mắt và tương tác, làm tăng trải nghiệm người dùng.

Phát triển back-end: JavaScript cũng có thể được sử dụng để phát triển phía server thông qua các nền tảng như Node.js, cho phép viết mã JavaScript cho cả phía client và phía server.

Phát triển trò chơi: JavaScript có thể được sử dụng để phát triển trò chơi trực tuyến và trò chơi đa người chơi.

Phát triển ứng dụng di động: Sử dụng các framework như React Native hoặc Flutter, JavaScript có thể được sử dụng để xây dựng ứng dụng di động cho cả iOS và Android.

Browser extensions: JavaScript được sử dụng để phát triển tiện ích mở rộng cho các trình duyệt web, cung cấp các chức năng mở rộng cho trình duyệt.

2.2 Tailwind CSS

2.2.1 Khái niệm

Tailwind CSS là một framework CSS tiện ích tương tự như Bootstrap, nó cho phép xây dựng giao diện người dùng nhanh chóng và dễ dàng thông qua các lớp CSS sẵn có. Điểm đặc biệt của Tailwind CSS là nó cung cấp một số lượng lớn các lớp CSS với các thuộc tính và quy tắc CSS khác nhau. Thay vì sử dụng các lớp CSS được định nghĩa sẵn cho các thành phần cụ thể, Tailwind CSS tập trung vào việc cung cấp các lớp CSS cơ bản mà chúng ta có thể kết hợp để xây dựng giao diện.

2.2.2 Ưu điểm của Tailwind CSS

Tính linh hoạt cao: Tailwind CSS cho phép tùy chỉnh thiết kế đến từng chi tiết nhờ các lớp tiện ích nhỏ gọn (utility classes). Điều này giúp bạn tạo ra các giao diện tùy biến mà không bị giới hạn bởi các mẫu giao diện có sẵn như trong Bootstrap.

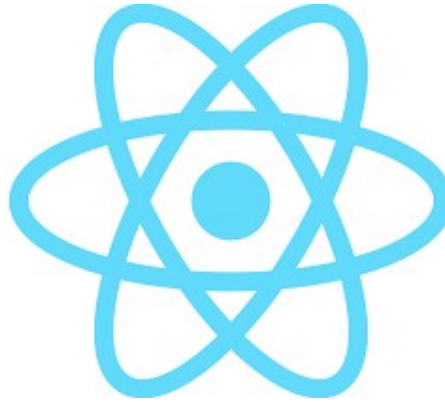
Không bị ràng buộc về thiết kế: Tailwind CSS không cung cấp các thành phần giao diện UI mặc định, giúp bạn dễ dàng tự do sáng tạo mà không phải tuân theo phong cách nhất định như Bootstrap. Bạn có thể tạo nên một giao diện độc đáo và phù hợp hoàn toàn với thiết kế yêu cầu.

Tối ưu hóa hiệu suất: Với PurgeCSS tích hợp sẵn, Tailwind có thể tự động xóa các lớp CSS không sử dụng khi build dự án, giúp giảm kích thước file CSS đáng kể và tăng tốc độ tải trang.

Khả năng tùy chỉnh mạnh mẽ: Tailwind cung cấp file cấu hình `tailwind.config.js`, cho phép tùy chỉnh mọi thứ từ màu sắc, font chữ, kích thước, khoảng cách đến các breakpoint cho responsive, giúp bạn tạo ra một theme hoàn toàn riêng.

2.3 React

2.3.1 Khái niệm



Hình 2.1. Logo React

React là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, ra mắt vào năm 2013, bản thân nó là một thư viện Javascript được dùng để xây dựng những tương tác với các thành phần trên website. Một trong những điểm nổi bật nhất của React đó là việc render dữ liệu không chỉ thực hiện được trên tầng Server mà còn ở dưới Client nữa.

React là một thư viện JavaScript chuyên giúp các nhà phát triển xây dựng giao diện người dùng. Trong lập trình ứng dụng front-end, lập trình viên thường sẽ phải làm việc chính trên 2 thành phần sau: Giao diện người dùng và xử lý tương tác của người dùng. Giao diện người dùng là tập hợp những thành phần có thể nhìn thấy được trên bất kỳ một ứng dụng nào, ví dụ như: menu, thanh tìm kiếm, những nút nhấn,...

2.3.2 Cấu trúc cơ bản của Component

React sử dụng kiến trúc dựa trên component, trong đó giao diện người dùng được chia thành các thành phần nhỏ, độc lập và có thể tái sử dụng. Mỗi component trong React là một lớp hoặc một hàm JavaScript, chứa logic và giao diện riêng biệt. Một component có thể nhỏ như một nút, hoặc lớn như toàn bộ trang. Tên component của React phải luôn bắt đầu bằng chữ in hoa. Các component có thể lồng vào nhau để tạo ra cấu trúc giao diện người dùng phức tạp [1].

2.3.3 JSX

JSX là một cú pháp mở rộng của ngôn ngữ lập trình JavaScript nó cho phép các lập trình viên viết mã HTML trực tiếp trong mã JavaScript. JSX giúp cho lập trình viên dễ dàng tạo ra các component hơn vì cú pháp của JSX gần giống với HTML.

Lợi ích khi sử dụng JSX để lập trình web là dễ sử dụng hơn JavaScript và cho phép trích dẫn HTML cũng như việc sử dụng các cú pháp thẻ HTML để render các subcomponent. JSX tối ưu hóa mã khi biên soạn, vì vậy nó chạy nhanh hơn so với mã JavaScript tương đương.

2.3.4 Virtual DOM

DOM (Document Object Model) là cấu trúc mô phỏng của một trang web, dùng để truy cập và quản lý các thành phần HTML. Virtual DOM là bản sao của DOM trên trang web và React sử dụng bản copy đó để tìm kiếm đúng phần mà DOM thật cần cập nhật khi trạng thái ứng dụng thay đổi, React sẽ so sánh Virtual DOM với DOM thật và chỉ cập nhật đúng các thành phần cần thiết, giúp ứng dụng chạy nhanh và mượt mà hơn.

2.3.5 Luồng dữ liệu một chiều

React không có những module chuyên dụng để xử lý data, vì vậy React chia nhỏ view thành các component nhỏ có mối quan hệ chặt chẽ với nhau. React hỗ trợ ràng buộc dữ liệu một chiều, giúp dữ liệu chỉ đi theo một hướng từ cha sang con, giúp dễ dàng kiểm soát luồng dữ liệu trong ứng dụng. Điều này làm cho việc quản lý và gỡ lỗi ứng dụng trở nên dễ dàng và rõ ràng hơn.

2.3.6 State và Props

State là một đối tượng quản lý dữ liệu động trong component. Khi state thay đổi, React sẽ tự động cập nhật giao diện để phản ánh các thay đổi này. State thường được sử dụng trong các component stateful, nghĩa là các component có trạng thái thay đổi theo thời gian [1] .

Props (viết tắt của properties) là các giá trị được truyền từ component cha xuống component con. Props giúp truyền dữ liệu và các hàm giữa các component và không thể thay đổi trong component con [1].

2.3.7 Ưu điểm của React

React có cú pháp đơn giản và dễ hiểu, đặc biệt là đối với những người đã quen thuộc với JavaScript. JSX, một phần mở rộng của JavaScript, cho phép lập trình viên viết mã HTML trong JavaScript, làm cho quá trình phát triển trở nên trực quan và dễ dàng hơn. Điều này giúp các lập trình viên mới dễ dàng tiếp cận và bắt đầu sử dụng React nhanh chóng.

React cho phép xây dựng các thành phần giao diện người dùng nhỏ gọn và độc lập, gọi là component. Các component này có thể được tái sử dụng nhiều lần trong cùng một ứng dụng hoặc trong các dự án khác nhau, giúp tiết kiệm thời gian và công sức phát triển, đồng thời đảm bảo tính nhất quán của giao diện người dùng.

Cấu trúc cơ bản của component trong React giúp việc viết và quản lý các thành phần giao diện người dùng trở nên dễ dàng và hiệu quả hơn. Lập trình viên có thể tách biệt các phần của giao diện người dùng thành các component nhỏ hơn, dễ dàng kiểm thử và bảo trì. Việc viết các component này trở nên đơn giản nhờ JSX, cho phép kết hợp JavaScript và HTML một cách tự nhiên và dễ hiểu.

React sử dụng Virtual DOM để tối ưu hóa hiệu suất ứng dụng. Thay vì cập nhật toàn bộ DOM thật mỗi khi có thay đổi, React sẽ cập nhật Virtual DOM trước, sau đó so sánh với DOM thật để chỉ thay đổi những phần cần thiết. Điều này giúp giảm thiểu số lượng thao tác trên DOM thật, từ đó cải thiện tốc độ và hiệu suất của ứng dụng.

Cấu trúc dựa trên component của React giúp mã nguồn dễ dàng quản lý, bảo trì và mở rộng. Mỗi component có thể được phát triển, kiểm thử và bảo trì độc lập, giúp việc sửa lỗi và thêm tính năng mới trở nên đơn giản hơn. Điều này cũng giúp các nhóm phát triển có thể làm việc song song trên các phần khác nhau của ứng dụng mà không gây ra xung đột.

Với React Native, một framework dựa trên React, lập trình viên có thể phát triển các ứng dụng di động cho cả iOS và Android sử dụng cùng một codebase. Điều này giúp tiết kiệm thời gian và nguồn lực, đồng thời đảm bảo tính nhất quán của ứng dụng trên các nền tảng khác nhau.

2.4 Express.js

2.4.1 Khái niệm



Hình 2.2. Express.js

Express.js là một framework web phổ biến được sử dụng để phát triển các ứng dụng web và API qua Node.js. Nền tảng được xem là một phương thức xử lý các yêu cầu HTTP, quản lý các tuyến đường, xử lý phần mềm trung gian và nhiều tính năng khác để phát triển hiệu quả ứng dụng web [2].

Express.js tập trung vào công việc tối ưu hóa việc xây dựng web ứng dụng bằng cách cung cấp một cấu trúc hoạt động và chỉ định rõ ràng việc xử lý yêu cầu và phản hồi. Nền tảng cũng hỗ trợ tích hợp các phần mềm trung gian bên ngoài để mở rộng chức năng của ứng dụng [2].

2.4.2 Đặc điểm của Express.js

Định tuyến (Routing): Express.js cho phép định tuyến các yêu cầu HTTP tới các endpoint cụ thể dựa trên đường dẫn và phương thức yêu cầu. Điều này giúp tách biệt logic xử lý cho từng phần của ứng dụng và quản lý các route một cách linh hoạt.

Trích xuất dữ liệu từ yêu cầu (Request parsing): Express.js cung cấp các công cụ để trích xuất dữ liệu từ yêu cầu HTTP như tham số đường dẫn, tham số truy vấn, thân yêu cầu, header và các loại dữ liệu khác.

Phản hồi yêu cầu (Response handling): Express.js cho phép phản hồi yêu cầu với headers và status code tương ứng, cũng như trả về dữ liệu dưới nhiều định dạng như JSON, HTML hoặc phản hồi khác.

Middleware: Trung gian (middleware) trong Express.js có thể được sử dụng để xử lý các yêu cầu trước khi chúng đến các endpoint cụ thể, giúp thực

hiện các thao tác như xác thực người dùng, xử lý lỗi, nén dữ liệu và nhiều tác vụ khác

RESTful API Development: Express.js thường được sử dụng để xây dựng các RESTful API với sự hỗ trợ tốt cho việc tạo, lấy, cập nhật và xóa dữ liệu thông qua các phương thức HTTP.

2.4.3 Ưu điểm của Express.js

Dễ sử dụng: Với cú pháp đơn giản và dễ hiểu, Express.js giúp việc phát triển ứng dụng trở nên nhanh chóng và hiệu quả, giảm thiểu độ phức tạp trong việc quản lý yêu cầu và phản hồi.

Hiệu suất cao: Vì được xây dựng trên Node.js, Express.js tận dụng các đặc tính của Node.js như xử lý bất đồng bộ và hiệu suất cao, giúp ứng dụng web hoặc API chạy nhanh.

Phát triển nhanh chóng: Express.js cung cấp nhiều công cụ và tính năng hỗ trợ việc phát triển web nhanh chóng, như hỗ trợ các template engine, middleware và khả năng xử lý lỗi dễ dàng.

Cộng đồng lớn: Express.js có một cộng đồng phát triển rộng lớn và rất nhiều tài liệu hỗ trợ, giúp người dùng dễ dàng tìm kiếm giải pháp và nhận sự trợ giúp khi gặp vấn đề.

Mở rộng linh hoạt: Express.js cho phép dễ dàng mở rộng ứng dụng bằng cách sử dụng các module hoặc middleware từ bên thứ ba, tăng tính linh hoạt của ứng dụng.

2.5 RESTful API

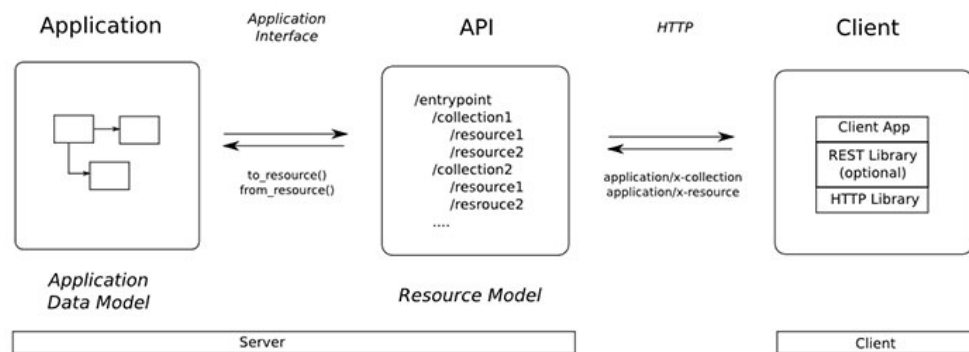
2.5.1 Khái niệm

RESTful API là một loại API tuân theo các nguyên tắc của kiến trúc REST (Representational State Transfer), được sử dụng như một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web để tiện cho việc quản lý các resource. RESTful API chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động...), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP [7].

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

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

2.5.2 Cách hoạt động của RESTful API



Hình 2.3. Cách hoạt động của RESTful API

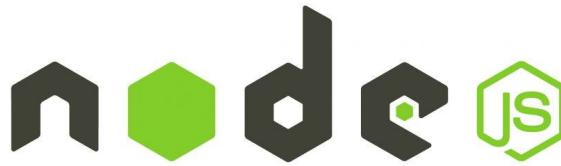
REST hoạt động dựa trên nguyên lý của kiến trúc REST chủ yếu dựa vào giao thức HTTP để giao tiếp với tài nguyên của hệ thống thông qua phương thức HTTP. Các phương thức HTTP thường được sử dụng:

- GET (SELECT): Trả về một sản phẩm hoặc một danh sách sản phẩm.
- POST (CREATE): Tạo mới một sản phẩm.
- PUT (UPDATE): Cập nhật thông tin cho sản phẩm.
- DELETE (DELETE): Xóa một sản phẩm.

Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.

2.6 Node.js

2.6.1 Khái niệm



Hình 2.4. Node.js

Được phát hành vào năm 2009, NodeJS, hay còn được biết với tên gọi chính thức là Node.js, là một nền tảng JavaScript phía máy chủ (server-side) cho phép chạy mã JavaScript ngoài trình duyệt, chủ yếu dùng để phát triển ứng dụng web, ứng dụng mạng, và các dịch vụ web. Node.js được xây dựng trên V8 JavaScript engine của Google Chrome, giúp mã JavaScript chạy nhanh và hiệu quả. Node.js cho phép các lập trình viên tạo cả ứng dụng front-end và back-end bằng JavaScript [6].

2.6.2 Đặc điểm của Node.js

Nguồn mở (Open-source): Mã nguồn của Node.js được công bố công khai, điều này có nghĩa là bất kỳ ai cũng có thể truy cập, sử dụng, và đóng góp vào mã nguồn. Node.js được duy trì bởi cộng đồng lập trình viên trên toàn thế giới, và hướng dẫn đóng góp của Node.js hướng dẫn bạn cách để bạn có thể góp phần phát triển nó [6].

Đa nền tảng (Cross-platform): Node.js không phụ thuộc vào bất kỳ hệ điều hành nào cụ thể nào, nghĩa là nó có thể chạy trên Linux, macOS hoặc Windows. Điều này làm cho Node.js trở thành một lựa chọn linh hoạt cho các nhà phát triển muốn xây dựng các ứng dụng có thể hoạt động trên nhiều nền tảng khác nhau mà không cần thay đổi mã nguồn [6].

Môi trường thực thi JavaScript (JavaScript runtime environment): Để mã JavaScript có thể được thực thi, nó cần một môi trường chạy phù hợp. Trong khi trình duyệt như Chrome và Firefox cung cấp một môi trường thực thi cho JavaScript, Node.js mở rộng khả năng này ra ngoài trình duyệt. Node.js cho phép chạy JavaScript trên máy chủ, hoặc trong bất kỳ môi trường máy tính nào khác, không chỉ trong trình duyệt.

Dựa trên V8 JavaScript Engine: Node.js được xây dựng dựa trên V8, động cơ JavaScript được phát triển bởi Google cho trình duyệt Chrome. Điều này giúp Node.js có khả năng thực thi JavaScript nhanh và hiệu quả, đồng thời hỗ trợ các tính năng mới nhất của ngôn ngữ JavaScript.

2.6.3 Ưu điểm của Node.js

Hiệu suất cao: Node.js được xây dựng trên động cơ JavaScript V8 của Google Chrome, cho phép biên dịch mã JavaScript thành mã máy nhanh chóng. Nhờ đó, thời gian thực thi của Node.js rất nhanh, làm tăng hiệu suất của các ứng dụng.

Hệ sinh thái phong phú: Với hơn 50,000 gói có sẵn trong Node Package Manager (NPM), các nhà phát triển có thể dễ dàng tìm và sử dụng các thư viện theo nhu cầu của họ mà không cần phải viết lại từ đầu, tiết kiệm đáng kể thời gian và công sức.

Xử lý bất đồng bộ và không chặn (Asynchronous and Non-blocking): Node.js hoạt động một cách bất đồng bộ và không chặn các hoạt động I/O, nghĩa là nó không cần chờ đợi API trả về dữ liệu trước khi tiếp tục xử lý yêu cầu tiếp theo. Điều này làm cho Node.js trở nên lý tưởng cho việc xây dựng các ứng dụng web thời gian thực và xử lý dữ liệu lớn.

Tính nhất quán trong mã nguồn: Node.js cho phép sử dụng cùng một ngôn ngữ lập trình (JavaScript) cho cả phía máy chủ và máy khách. Điều này không chỉ giúp giảm thiểu sự không đồng bộ giữa client và server mà còn làm cho việc bảo trì và quản lý mã nguồn trở nên dễ dàng hơn.

Khả năng mở rộng: Node.js hỗ trợ xây dựng các ứng dụng có khả năng mở rộng cao thông qua mô hình sự kiện và bất đồng bộ của mình. Điều này cho phép xử lý hàng ngàn kết nối đồng thời mà không làm giảm hiệu suất.

Ngôn ngữ quen thuộc: Vì Node.js là một khung làm việc JavaScript, nó trở thành lựa chọn lý tưởng cho những nhà phát triển đã quen thuộc với JavaScript. Điều này làm cho quá trình học tập và phát triển dự án với Node.js trở nên dễ dàng hơn nhiều.

2.7 Hệ quản trị cơ sở dữ liệu MongoDB

2.7.1 Khái niệm



Hình 2.5. MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở, rất phổ biến hiện nay. Thay vì lưu trữ dữ liệu trong các bảng và dòng như các hệ quản trị SQL truyền thống, MongoDB lưu dữ liệu dưới dạng các tài liệu (documents) trong các bộ sưu tập (collections). Các dữ liệu được lưu trữ trong tài liệu kiểu JSON nên tốc độ truy vấn sẽ rất nhanh.

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 [5].

MongoDB là một nền tảng cơ sở dữ liệu tài liệu NoSQL linh hoạt, có thể mở rộng được thiết kế để khắc phục cách tiếp cận cơ sở dữ liệu quan hệ và những hạn chế của các giải pháp NoSQL khác. Các sản phẩm MongoDB nổi tiếng với khả năng mở rộng theo chiều ngang và cân bằng tải, điều này đã mang lại cho các nhà phát triển ứng dụng một mức độ linh hoạt và khả năng mở rộng chưa từng có [4].

Tính đến năm 2020, MongoDB đã được tải xuống hơn 30 triệu lần với hơn 730.000 đăng ký MongoDB University. Có trình điều khiển cho hơn 10 ngôn ngữ, với hàng chục ngôn ngữ khác được cộng đồng thêm vào. Hơn hết, MongoDB hoàn toàn miễn phí để sử dụng. Ứng dụng này cung cấp cho các nhà phát triển một số khả năng hữu ích, cho dù bạn cần chạy riêng tư trên trang web hay trên đám mây công cộng [4].

MongoDB có thể chạy trong các môi trường sau:

MongoDB Atlas: Đây là dịch vụ được quản lý để triển khai MongoDB trên đám mây.

MongoDB Enterprise: Các Phiên bản MongoDB tự quản lý, dựa trên đăng ký.

MongoDB Community Edition: Các phiên bản MongoDB có sẵn, sử dụng miễn phí và tự quản lý.

2.7.2 Những tính năng nổi bật của MongoDB

Truy vấn ad-hoc cho phân tích thời gian thực: MongoDB hỗ trợ các truy vấn ad-hoc linh hoạt giúp tăng hiệu quả khi xử lý khối lượng dữ liệu lớn với các biến động không dự đoán trước, giúp tối ưu hóa hiệu suất.

Lập chỉ mục (Indexing) để cải thiện tốc độ truy vấn: Lập chỉ mục phù hợp tăng tốc độ truy vấn và cải thiện hiệu suất. MongoDB cho phép tạo chỉ mục theo yêu cầu trên các trường khác nhau, bao gồm cả các trường lồng trong mảng.

Sao chép để tăng tính ổn định và khả dụng dữ liệu: Sao chép dữ liệu trên nhiều máy chủ giúp tăng tính ổn định và khả năng khôi phục khi gặp sự cố.

Phân mảnh: Phân mảnh dữ liệu trên các máy chủ khác nhau giúp tăng khả năng mở rộng, giảm tải và hỗ trợ ứng dụng có số lượng người dùng lớn.

Cân bằng tải: Với tính năng sao chép và phân mảnh, MongoDB có thể xử lý hàng triệu yêu cầu đồng thời mà vẫn đảm bảo độ chính xác và tính nhất quán của dữ liệu.

2.7.3 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 ghi đệm lên RAM, hạn chế truy cập vào ổ cứng nên tốc độ đọc và ghi cao.

2.7.4 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 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.

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

3.1 Mô tả bài toán

Website chia sẻ công thức nấu ăn không chỉ là một nền tảng trực tuyến đơn thuần mà còn là nơi để mọi người khám phá về những món ăn mới và trải nghiệm sự đa dạng của ẩm thực. Với mong muốn làm cho việc vào bếp vui hơn mỗi ngày, website sẽ cung cấp một nền tảng trực tuyến thân thiện, dễ sử dụng, nơi người dùng có thể tìm kiếm công thức nấu ăn mong muốn và có thể để lại những đánh giá, bình luận của bản thân về công thức nấu ăn. Nội dung trên nền tảng sẽ được quản lý bởi admin nhằm đảm bảo tính minh bạch và chất lượng. Nền tảng sẽ hỗ trợ hai nhóm người dùng chính: người dùng thông thường và quản trị viên, với các chức năng được thiết kế tối ưu hóa trải nghiệm sử dụng và đảm bảo hiệu quả trong quản lý.

3.2 Yêu cầu chức năng

3.2.1 Người dùng

Đăng nhập và đăng ký: Người dùng có thể đăng ký tài khoản bằng email, mật khẩu sau đó đăng nhập vào hệ thống để sử dụng các tính năng.

Quản lý tài khoản cá nhân: Người dùng có thể chỉnh sửa, cập nhật thông tin cá nhân của mình.

Tìm kiếm công thức nấu ăn: Người dùng tìm kiếm công thức nấu ăn theo tên công thức, nguyên liệu, kiểu chế biến.

Bình luận, đánh giá công thức nấu ăn: Người dùng có thể đánh giá và bình luận dưới mỗi bài đăng công thức nấu ăn.

3.2.2 Admin

Quản lý tài khoản người dùng: Xem danh sách tài khoản người dùng, xem thông tin chi tiết của từng tài khoản, xóa tài khoản người dùng vi phạm.

Quản lý công thức nấu ăn: Xem danh sách công thức nấu ăn đã đăng. Thêm mới, chỉnh sửa hoặc xóa công thức nấu ăn.

Quản lý đánh giá và bình luận: Xem danh sách các bình luận, đánh giá dưới mỗi bài đăng công thức. Admin cũng có thể xóa các bình luận không phù hợp.

3.3 Yêu cầu phi chức năng

Hiệu năng của hệ thống: Tốc độ tải trang phải nhanh và có khả năng chịu tải ở mức khá.

Khả năng mở rộng: Cấu trúc hệ thống linh hoạt để có thể tích hợp thêm tính năng trong cho hệ thống trong tương lai.

Khả năng tương thích: Có khả năng hỗ trợ các trình duyệt phổ biến như Chrome, Edge, Fire Fox, Cốc Cốc,...

Tính bảo mật: Mật khẩu của người dùng sẽ được mã hóa bằng thuật toán mạnh như bcrypt.

Quản lý mã nguồn rõ ràng: Cấu trúc mã nguồn của website cần rõ ràng, dễ hiểu và dễ dàng cho các lập trình viên thực hiện các thay đổi khi cần thiết.

3.4 Cơ sở dữ liệu

3.4.1 Danh sách các collection

Bảng 3.1 Danh sách các collection

| STT | Tên | Mô tả |
|-----|----------|--|
| 1 | users | Lưu trữ thông tin người dùng của hệ thống, bao gồm tên, email, mật khẩu, vai trò, và thông tin liên quan đến tài khoản. |
| 2 | recipes | Chứa các công thức nấu ăn với thông tin về tên món ăn, mô tả, nguyên liệu, hướng dẫn chế biến, thời gian chuẩn bị và chế biến. |
| 3 | ratings | Lưu trữ các đánh giá của người dùng về các công thức nấu ăn, bao gồm điểm số (1-5 sao), người đánh giá, công thức được đánh giá và thời gian đánh giá. |
| 4 | comments | Chứa các bình luận của người dùng về công thức nấu ăn, với thông tin người bình luận, nội dung bình luận, thời gian đăng. |
| 5 | visits | Ghi lại thông tin về lượt truy cập, thời gian truy cập vào website. |

3.4.2 Cấu trúc các collection trong hệ thống

Collection người dùng:

```
{
  "_id": ObjectId("..."),
  "username": "string",
  "email": "string",
  "password": "string",
  "createdAt": ISODate("..."),
  "role": "string",
  "recipes": [ObjectId("...")]
}
```

Collection bình luận:

```
{
  "_id": ObjectId("..."),
  "userId": ObjectId("..."),
  "recipeId": ObjectId("..."),
  "content": "string",
  "createdAt": ISODate("...")
}
```

Collection visits:

```
{
  "_id": ObjectId("..."),
  "date": "YYYY-MM-DD",
  "visits": 0,
  "createdAt": ISODate("...")
}
```

Collection đánh giá:

```
{
  "_id": ObjectId("..."),
  "recipeId": ObjectId("..."),
  "userId": ObjectId("..."),
  "rating": "number",
  "createdAt": ISODate("...")
}
```

Collection công thức:

```
{
  "_id": ObjectId("..."),
  "title": "string",
  "images": "string",
  "cookingStyle": "string",
  "cookingTime": "string",
  "ingredients": ["string"],
  "stepsDescriptions": ["string"],
  "stepsImages": ["string"],
  "video": "string",
  "createdAt": ISODate("..."),
  "updatedAt": ISODate("..."),
  "ratings": [ObjectId("...")],
  "comments": [ObjectId("...")]
}
```

3.4.3 Chi tiết các collection

Collection: ratings

Bảng 3.2 Chi tiết các thuộc tính của collection ratings

| STT | Thuộc tính | Diễn giải | Kiểu dữ liệu |
|-----|------------|--|--------------|
| 1 | _id | ID duy nhất của đánh giá. | ObjectId |
| 2 | recipeId | ID của công thức được đánh giá. | ObjectId |
| 3 | userId | ID của người dùng đã thực hiện đánh giá. | ObjectId |
| 4 | rating | Điểm đánh giá của công thức (thường là số từ 1-5 sao). | Number |
| 5 | createdAt | Thời gian tạo đánh giá. | ISODate |

Collection: users

Bảng 3.3 Chi tiết các thuộc tính của collection users

| STT | Thuộc tính | Mô tả | Kiểu dữ liệu |
|-----|------------|---|--------------|
| 1 | _id | ID duy nhất của document, tự động tạo bởi MongoDB. | ObjectId |
| 2 | username | Tên người dùng, dùng để đăng nhập vào hệ thống. | String |
| 3 | email | Địa chỉ email của người dùng, thường dùng để xác thực và giao tiếp. | String |
| 4 | password | Mật khẩu của người dùng (nên mã hóa khi lưu trữ). | String |
| 5 | createdAt | Thời gian tạo tài khoản người dùng. | ISODate |
| 6 | role | Vai trò của người dùng, có thể là user hoặc admin. | String |
| 7 | recipes | Mảng chứa các ID công thức mà người dùng đã tạo hoặc lưu. | [ObjectId] |

Collection: comments

Bảng 3.4 Chi tiết các thuộc tính của collection comments

| STT | Thuộc tính | Diễn giải | Kiểu dữ liệu |
|-----|------------|--|--------------|
| 1 | _id | ID duy nhất của bình luận. | ObjectId |
| 2 | userId | ID của người dùng đã viết bình luận. | ObjectId |
| 3 | recipeId | ID của công thức mà bình luận liên quan đến. | ObjectId |
| 4 | content | Nội dung của bình luận. | String |
| 5 | createdAt | Thời gian tạo bình luận. | ISODate |

Collection: visits

Bảng 3.5 Chi tiết các thuộc tính của collection visits

| STT | Thuộc tính | Diễn giải | Kiểu dữ liệu |
|-----|------------|--|--------------|
| 1 | _id | ID duy nhất của bản ghi lượt truy cập. | ObjectId |
| 2 | date | Ngày ghi nhận lượt truy cập (theo định dạng YYYY-MM-DD). | String |
| 3 | visits | Số lượt truy cập vào công thức trong ngày đó. | Number |
| 4 | createdAt | Thời gian tạo bản ghi lượt truy cập. | ISODate |

Collection: recipes

Bảng 3.6 Chi tiết các thuộc tính của collection recipes

| STT | Thuộc tính | Diễn giải | Kiểu dữ liệu |
|-----|-------------------|--|--------------|
| 1 | _id | ID duy nhất của công thức, tự động tạo bởi MongoDB. | ObjectId |
| 2 | title | Tiêu đề của công thức, thường là tên món ăn. | String |
| 3 | images | URL hoặc đường dẫn tới ảnh minh họa của công thức. | String |
| 4 | cookingStyle | Cách chế biến món ăn | String |
| 5 | cookingTime | Thời gian chế biến công thức (có thể là "10 phút", "1 giờ", ...). | String |
| 6 | ingredients | Danh sách các nguyên liệu cần thiết cho công thức, lưu dưới dạng mảng. | [String] |
| 7 | stepsDescriptions | Mô tả các bước thực hiện trong quá trình chế biến công thức. | [String] |
| 8 | stepsImages | Danh sách các ảnh minh họa cho từng bước chế biến. | [String] |
| 9 | video | Đường dẫn video hướng dẫn nấu ăn (nếu có). | String |
| 10 | createdAt | Thời gian tạo công thức. | ISODate |
| 11 | updatedAt | Thời gian cập nhật công thức. | ISODate |
| 12 | ratings | Mảng chứa các ID của các đánh giá liên quan đến công thức. | [ObjectId] |
| 13 | comments | Mảng chứa các ID của các bình luận liên quan đến công thức. | [ObjectId] |

3.4.4 Mỗi quan hệ giữa các collection trong hệ thống

Collection users(người dùng) và recipe (công thức) có mối quan hệ **1-n** do mỗi người dùng có thể đăng nhiều công thức nấu ăn.

Collection users(người dùng) và comment (bình luận) có mối quan hệ **1-n** do một người dùng có thể viết nhiều bình luận.

Collection users(người dùng) và ratings (đánh giá) có mối quan hệ **1-n** do một người dùng có thể đánh giá nhiều công thức nấu ăn.

Collection recipe (công thức) và comments (bình luận) có mối quan hệ **1-n** do một công thức nấu ăn có thể có nhiều bình luận.

Collection recipe (công thức) và ratings (đánh giá) có mối quan hệ **1-n** do một công thức nấu ăn có thể nhận được nhiều đánh giá.

3.5 Sơ đồ use case

Actors:

Người dùng: Thực hiện các chức năng như đăng nhập, đăng ký, tìm kiếm, đánh giá, bình luận và quản lý tài khoản.

Admin: Quản lý hệ thống tài khoản, quản lý công thức, bình luận. **Use case:**

Admin:

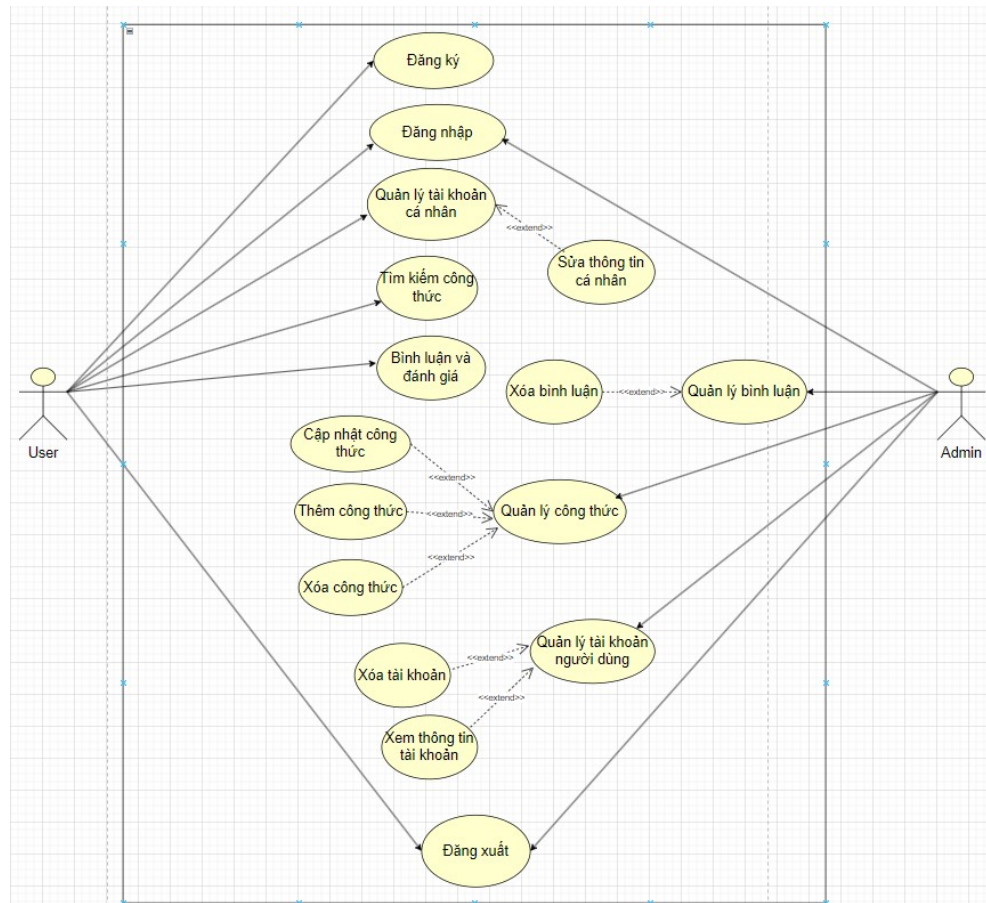
- Đăng nhập
- Quản lý tài khoản người dùng.
- Quản lý công thức.
- Quản lý bình luận.
- Đăng xuất

Người dùng:

- Đăng nhập
- Đăng ký.
- Quản lý tài khoản cá nhân.

- Tìm kiếm công thức.
- Bình luận và đánh giá công thức.
- Đăng xuất

Sơ đồ use case:



Hình 3.1. Sơ đồ use case

3.6 Cài đặt Node.js và React

3.6.1 Cài đặt Node.js

Bước 1: Truy cập trang chính thức của Node.js tại địa chỉ <https://nodejs.org/en>

Bước 2: Tải phiên bản Node.js LTS (Long-Term Support)



Hình 3.2 Node.js LTS

Bước 3: Bước 3: Chạy file cài đặt và làm theo các bước hướng dẫn trên màn hình.

Bước 4: Sau khi cài đặt, kiểm tra xem Node.js đã được cài đặt vào máy tính hay chưa bằng cách mở terminal và gõ lệnh `node -v` và `npm -v`. Nếu xuất hiện phiên bản của Node.js và npm thì đã cài đặt thành công.

3.6.2 Cài đặt React

Bước 1: Mở terminal và di chuyển đến thư mục muốn tạo dự án bằng lệnh `cd đường-dẫn-thư-mục`.

Bước 2: Sau khi di chuyển vào đường dẫn thư mục mong muốn thì gõ lệnh `npm create vite@latest tên-dự-án`. Sau đó chọn Framework React và ngôn ngữ JavaScript.

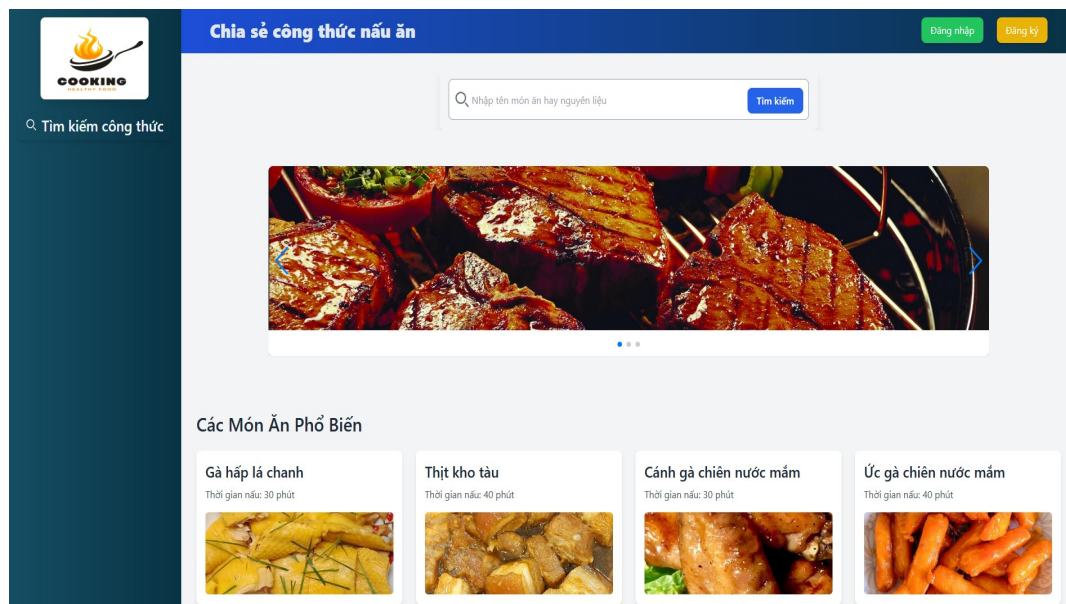
Bước 3: Di chuyển vào thư mục dự án bằng lệnh `cd tên-dự-án` và cài đặt các gói phụ thuộc bằng cách gõ lệnh `npm install`.

Bước 4: Chạy server phát triển bằng cách gõ lệnh `npm-run-dev`.

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

4.1 Trang chủ

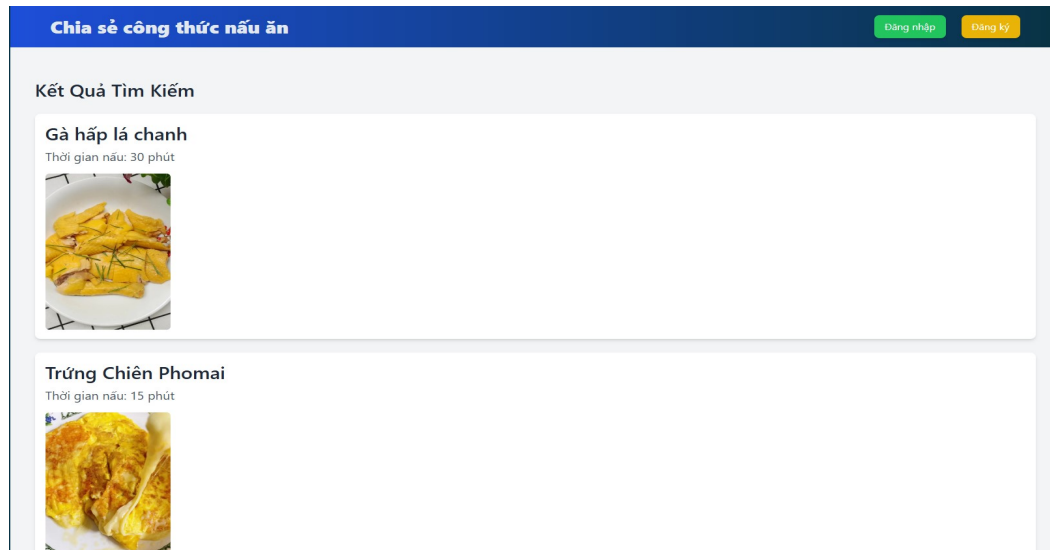
Giao diện trang chủ: Ở trên cùng của giao diện trang chủ là Navbar với hai nút là đăng nhập và đăng ký. Nút "Đăng nhập" sẽ dẫn người dùng đến một trang hoặc một trang cho phép họ nhập thông tin đăng nhập của mình và Nút "Đăng ký" sẽ dẫn người dùng đến một trang hoặc một trang nơi họ có thể tạo tài khoản mới. Ở phía dưới là thanh tìm kiếm và một banner hiển thị hình ảnh của một số món ăn. Thanh tìm kiếm có chức năng cơ bản là cho phép người dùng nhập từ khóa và tìm kiếm công thức nấu ăn phù hợp. Tiếp theo là danh sách các công thức phổ biến, có thể xem thông tin chi tiết của món ăn bằng cách nhấn vào giao diện của các công thức trên màn hình.



Hình 4.1 Trang chủ

4.2 Tìm kiếm công thức

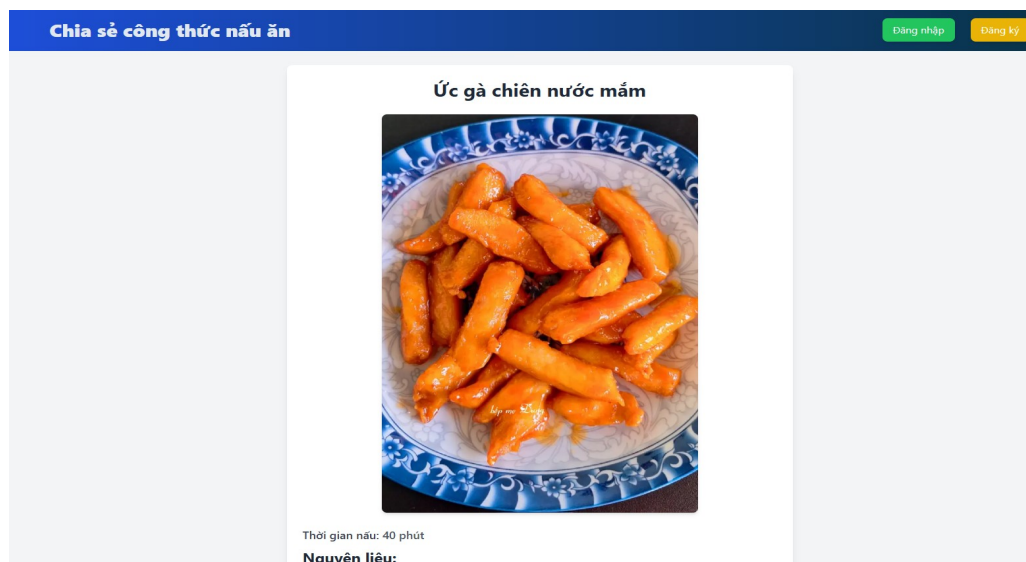
Giao diện trang tìm kiếm công thức: Hiển thị danh sách các công thức nấu ăn được tìm kiếm với những thông tin như tên công thức, hình ảnh của công thức. Người dùng có thể xem thông tin chi tiết của công thức bằng cách nhấn vào giao diện của mỗi công thức.



Hình 4.2 Trang tìm kiếm

4.3 Trang xem thông chi tiết công thức

Trang xem chi tiết công thức: Là nơi người dùng có thể tham khảo thông tin đầy đủ về một công thức nấu ăn cụ thể. Trang này giúp người dùng hiểu rõ về cách chế biến, nguyên liệu cần thiết, và các bước thực hiện để làm món ăn



Hình 4.3 Trang chi tiết công thức 1

Nguyên liệu:

- 500 gram ức gà
- 2 muỗng canh bột chiên giòn
- 2 muỗng canh đường
- 2 muỗng canh nước mắm
- 2 muỗng canh tương ớt
- Dầu ăn, muối

Các bước thực hiện:

1. Ức gà rửa sạch với nước muối loãng, rửa lại với nước sạch, để ráo. Cắt miếng dài con chì. Cho Ức gà vào tô, cho bột chiên giòn vào rồi trộn đều.



2. Cho dầu ăn vào chảo, nhiều dầu 1 chút, nấu cho dầu nóng rồi cho gà vào chiên vàng giòn. Vớt gà ra rây cho ráo dầu



3. Nước sốt: Cho nước mắm + đường + tương ớt + nước lọc vào chén khuấy đều cho tan đường.



Hình 4.4 Trang chi tiết công thức 2



Video hướng dẫn:

https://www.youtube.com/shorts/0bGL2d_cBd0

Đánh giá:

Điểm trung bình: 5.0 ★

👤 Nguyễn Văn A 5 ★

0

Đánh giá

Bình luận:

👤 Nguyễn Văn A

món ăn ngon!!!

Vào lúc: 12/27/2024, 12:28:34 PM

Nhập bình luận...

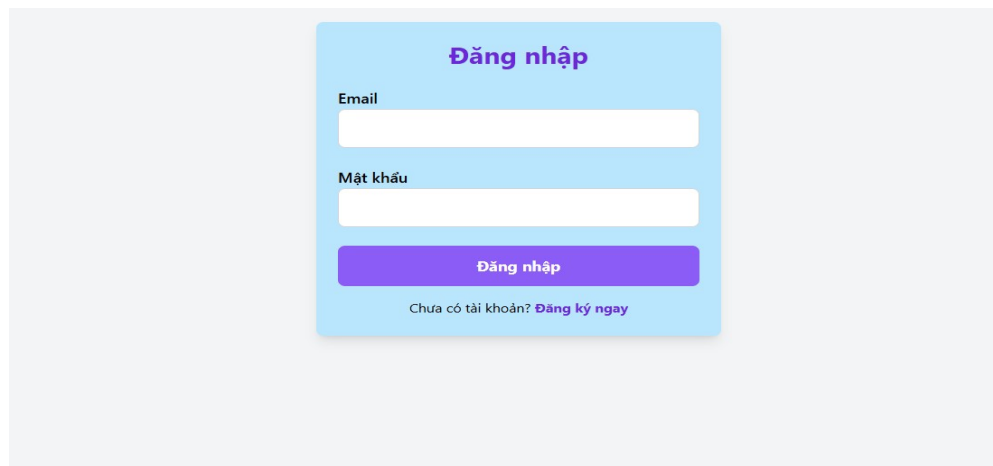
Bình luận

Hình 4.5 Trang chi tiết công thức 3

4.4 Trang đăng nhập và đăng ký

4.4.1 Đăng nhập

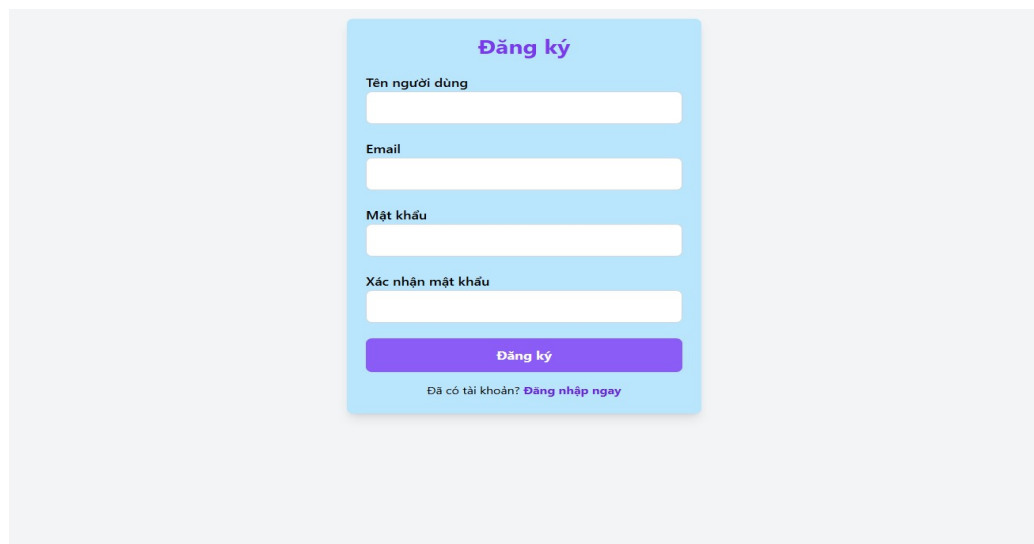
Trang đăng nhập cho phép người dùng truy cập vào tài khoản của họ. Sau khi điền đủ thông tin, người dùng nhấn nút "Đăng nhập" để vào hệ thống. Nếu người dùng chưa có tài khoản, người dùng có thể nhấn vào liên kết để chuyển hướng tới trang đăng ký.



Hình 4.6 Trang đăng nhập

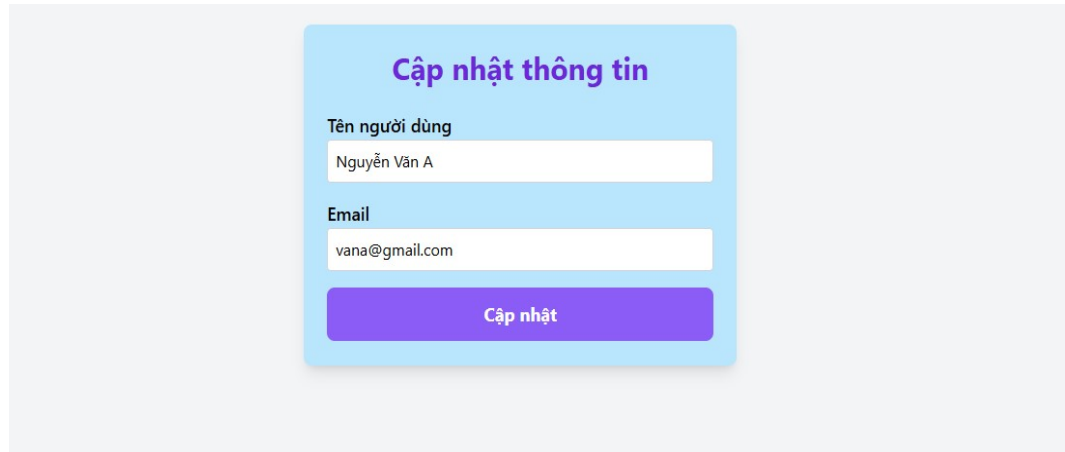
4.4.2 Đăng ký

Trang đăng ký cho phép người dùng tạo tài khoản mới trên website. Form đăng ký yêu cầu người dùng nhập mật khẩu hai lần để xác nhận rằng họ không mắc lỗi khi nhập. Sau khi điền đầy đủ thông tin, người dùng nhấn nút "Đăng ký" để tạo tài khoản mới.



Hình 4.7 Trang đăng ký

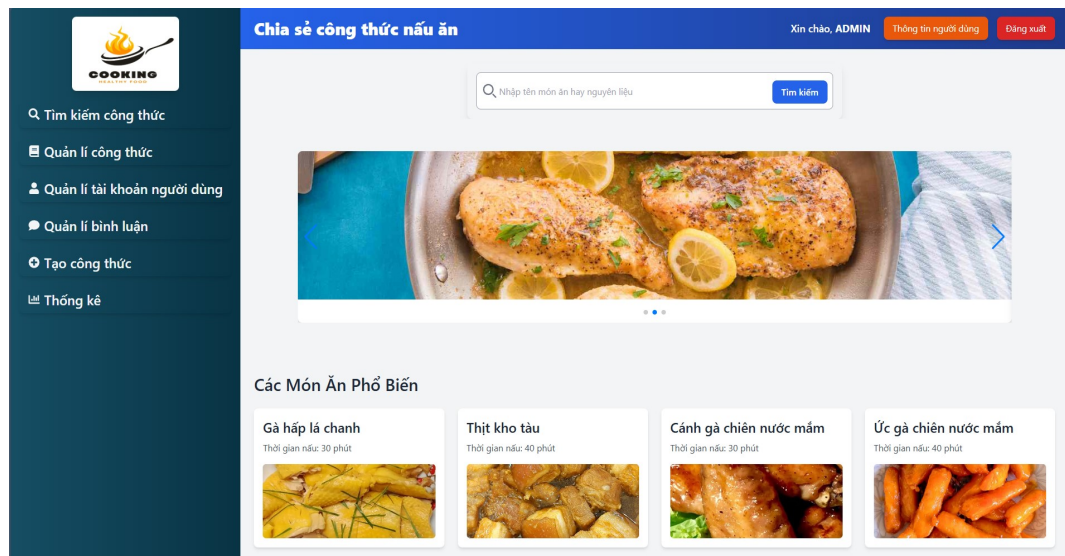
4.4.3 Cập nhật thông tin người dùng



The image shows a web form titled "Cập nhật thông tin" (Update information) in a light blue box. It contains two input fields: "Tên người dùng" (Username) with the value "Nguyễn Văn A" and "Email" with the value "vana@gmail.com". Below the fields is a purple button labeled "Cập nhật" (Update).

Hình 4.8 Cập nhật thông tin người dùng

4.5 Quản trị



Hình 4.9 Giao diện trang quản trị

4.5.1 Thêm công thức

Trang thêm công thức là nơi các admin đã đăng nhập có quyền thêm mới một công thức nấu ăn vào hệ thống để chia sẻ với cộng đồng.

The screenshot shows a web interface for adding a recipe. At the top, there's a blue header with the text 'Chia sẻ công thức nấu ăn' and a user greeting 'Xin chào, ADMIN'. Below the header, the main form is titled 'Thêm Công Thức Nấu Ăn'. It contains several input fields: 'Tên món ăn' (Recipe Name), 'Ảnh món ăn' (Recipe Image) with a 'Choose File' button, 'Nguyên liệu' (Ingredients) with a '+ Thêm nguyên liệu' link, 'Cách chế biến' (Cooking Method) with a dropdown menu, and 'Thời gian nấu' (Cooking Time). The user is logged in as 'ADMIN'.

Hình 4.10 Trang thêm công thức 1

This screenshot shows the 'Các bước thực hiện' (Steps) section of the recipe form. It includes a 'Mô tả bước' (Step Description) field, a 'Choose File' button, a '+ Thêm bước' link, and a 'URL video' field. A blue button labeled 'Thêm công thức' (Add Recipe) is at the bottom.

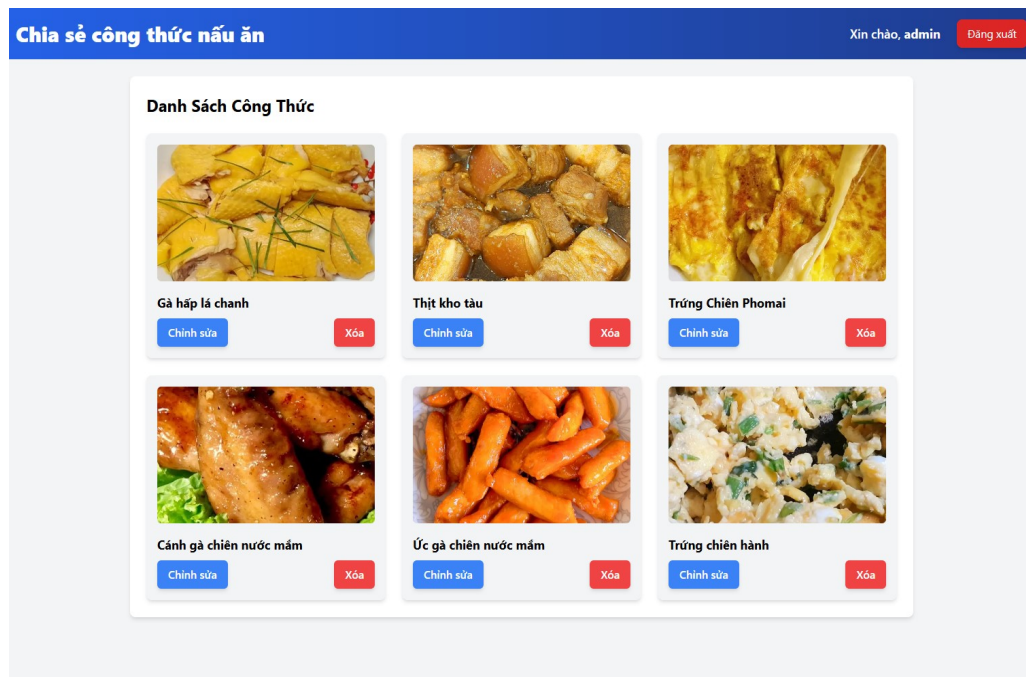
Hình 4.11 Trang thêm công thức 2

4.5.2 Quản lý công thức

Trang quản lý công thức là nơi admin có quyền quản lý các công thức đã thêm vào hệ thống. Nó hiển thị danh sách các công thức và các công cụ để chỉnh sửa hoặc xóa công thức.

Nút chỉnh sửa: Được cung cấp bên cạnh mỗi công thức, cho phép admin chỉnh sửa công thức đó.

Nút xóa: Được cung cấp bên cạnh mỗi công thức, cho phép admin xóa công thức khỏi hệ thống. Cần có xác nhận trước khi xóa để tránh việc xóa nhầm công thức.

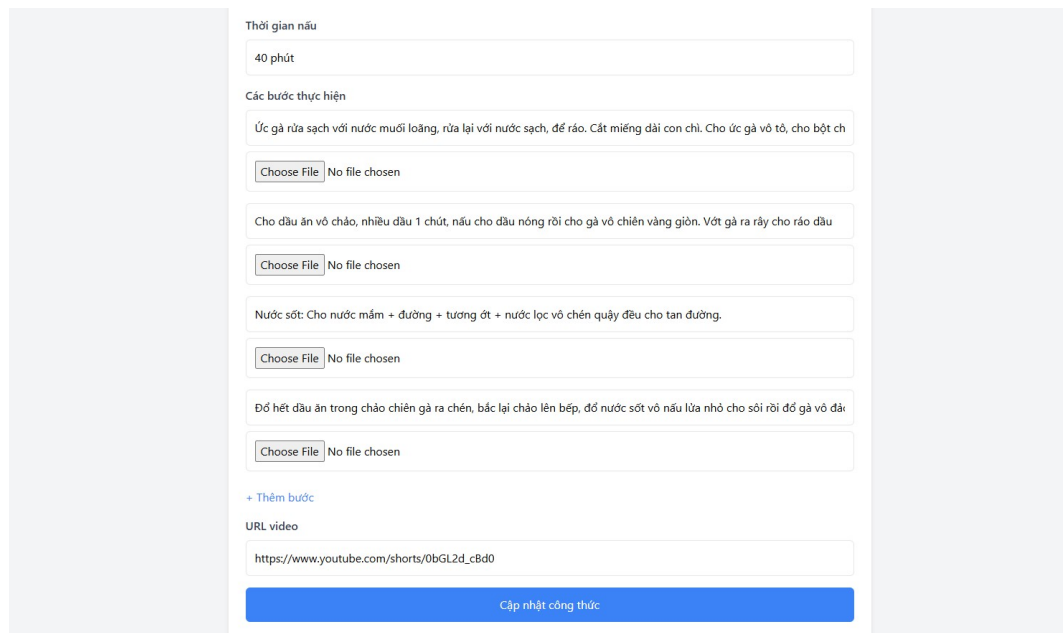


Hình 4.12 Trang quản lý công thức

4.5.3 Chỉnh sửa công thức

Trang chỉnh sửa công thức cho phép admin chỉnh sửa công thức đã được thêm vào hệ thống. Chức năng này giúp cập nhật lại công thức khi có thay đổi.

Hình 4.13 Trang chỉnh sửa công thức 1



Thời gian nấu

40 phút

Các bước thực hiện

Ức gà rửa sạch với nước muối loãng, rửa lại với nước sạch, để ráo. Cắt miếng dài con chì. Cho ức gà vào tô, cho bột ch

Choose File No file chosen

Cho dầu ăn vào chảo, nhiều dầu 1 chút, nấu cho dầu nóng rồi cho gà vào chiên vàng giòn. Vớt gà ra rây cho ráo dầu

Choose File No file chosen

Nước sốt: Cho nước mắm + đường + tương ớt + nước lọc vào chén khuấy đều cho tan đường.

Choose File No file chosen

Đổ hết dầu ăn trong chảo chiên gà ra chén, bắc lại chảo lên bếp, đổ nước sốt vào nấu lửa nhỏ cho sôi rồi đổ gà vào đảo.

Choose File No file chosen

+ Thêm bước

URL video

https://www.youtube.com/shorts/0bGL2d_cBd0

Cập nhật công thức

Hình 4.14 Trang chỉnh sửa công thức 2

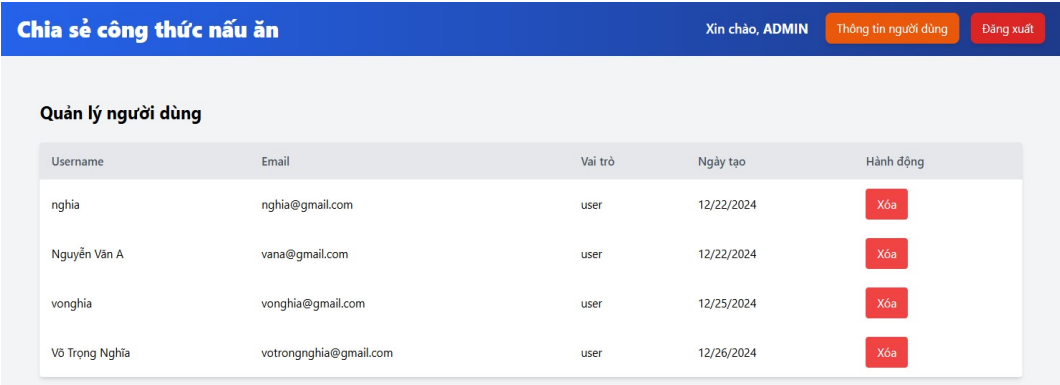
4.5.4 Quản lý bình luận

Trang này sẽ hiển thị danh sách các bình luận trong hệ thống, admin có thể xem và xóa các bình luận không phù hợp.

| Chia sẻ công thức nấu ăn | | | | |
|--------------------------|----------------------|----------------|-----------------|--|
| | | | Xin chào, ADMIN | Thông tin người dùng Đăng xuất |
| Quản lý bình luận | | | | |
| Người dùng | Recipe | Nội dung | Ngày tạo | Hành động |
| ngghia | Thịt kho tàu | abc | 12/25/2024 | Xóa |
| ADMIN | Thịt kho tàu | i am admin | 12/25/2024 | Xóa |
| vongghia | Gà hấp lá chanh | vongghia | 12/25/2024 | Xóa |
| Nguyễn Văn A | Ức gà chiên nước mắm | món ăn ngon!!! | 12/27/2024 | Xóa |
| Nguyễn Văn A | Gà hấp lá chanh | abc | 12/27/2024 | Xóa |
| ngghia | Gà hấp lá chanh | gà quá ngon!! | 12/27/2024 | Xóa |

Hình 4.15 Trang quản lý bình luận

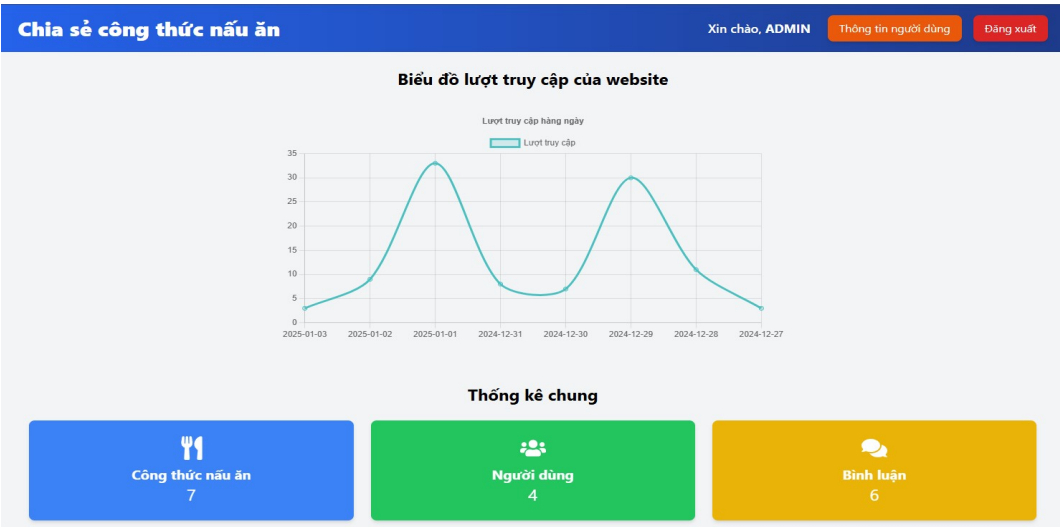
4.5.5 Quản lý tài khoản người dùng



Hình 4.16 Trang quản lý tài khoản

4.5.6 Trang thống kê

Trang thống kê hiển thị biểu đồ lượt truy cập theo thời gian, giúp quản trị viên theo dõi xu hướng truy cập. Đồng thời, trang cũng hiển thị các thống kê như số lượng công thức nấu ăn, số lượng người dùng, và số lượng bình luận.



Hình 4.15 Trang thống kê 1

| Thống kê lượt truy cập | |
|------------------------|---------------|
| Thời gian | Lượt truy cập |
| 2025-01-03 | 3 |
| 2025-01-02 | 9 |
| 2025-01-01 | 33 |
| 2024-12-31 | 8 |
| 2024-12-30 | 7 |
| 2024-12-29 | 30 |
| 2024-12-28 | 11 |
| 2024-12-27 | 3 |

Hình 4.16 Trang thống kê 2

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Xây dựng được website chia sẻ công thức nấu ăn với các chức năng cơ bản.

Sử dụng thành thạo các công cụ đã dùng để phát triển website.

Đọc hiểu các tài liệu chuyên ngành.

Cải thiện được tư duy lập trình của bản thân.

5.2 Hướng phát triển

Từ nền tảng đề tài này, trong tương lai hướng đến việc mở rộng và tối ưu hóa quy mô của website. Xây dựng diễn đàn hoặc nhóm thảo luận để người dùng trao đổi kinh nghiệm, mẹo vặt khi nấu ăn. Tối ưu hóa giao diện dành cho các thiết bị di động hỗ trợ người dùng truy cập và sử dụng nền tảng mọi lúc, mọi nơi. Thêm chức năng lưu trữ công thức yêu thích vào danh sách cá nhân và cho phép người dùng tạo và chia sẻ công thức của riêng mình. Tăng cường bảo mật bằng những phương pháp như xác thực hai yếu tố (2FA), mã hóa dữ liệu.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] P. M. Khoa, "React là gì? Lộ trình trở thành lập trình viên ReactJS," Top Dev, 10 1 2015. [Online]. Available: <https://topdev.vn/blog/react-la-gi-lo-trinh/>. [Accessed 10 11 2024].
- [2] N. Liên, "Expressjs là gì? Cách sử dụng Expressjs trong lập trình đạt kết quả tốt nhất," FPT Shop, 8 3 2012. [Online]. Available: <https://fptshop.com.vn/tin-tuc/danh-gia/express-js-la-gi-174976>. [Accessed 12 11 2024].
- [3] Amazon Web Service, "Javascript (JS) là gì?," Amazon Web Service, 3 2006. [Online]. Available: <https://aws.amazon.com/vi/what-is/javascript/>. [Accessed 12 11 2024].
- [4] MongoDB, "MongoDB Features," MongoDB, 11 2 2009. [Online]. Available: <https://www.mongodb.com/resources/products/fundamentals/features>. [Accessed 12 11 2024].
- [5] Top Dev, "MongoDB là gì? Định nghĩa đầy đủ và chi tiết nhất về MongoDB," Top Dev, 2015. [Online]. Available: <https://topdev.vn/blog/mongodb-la-gi/>. [Accessed 11 11 2024].
- [6] Top Dev, "Node.js là gì? Tổng hợp kiến thức NodeJS," Top Dev, 2015. [Online]. Available: <https://topdev.vn/blog/node-js-la-gi>. [Accessed 11 11 2024].
- [7] Top Dev, "RESTful API là gì? Cách thiết kế RESTful API," Top Dev, 2015. [Online]. Available: <https://topdev.vn/blog/restful-api-la-gi/>. [Accessed 12 11 2024].

PHỤ LỤC