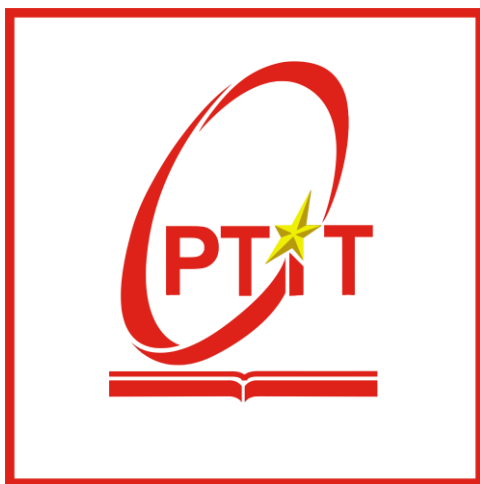


HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN 1



Website SmartBytes
Hệ thống gợi ý công thức nấu ăn
theo nhu cầu dinh dưỡng

Môn học: Thực tập cơ sở
Giảng viên: Kim Ngọc Bách
Họ tên: Trần Quang Anh
Mã SV: B22DCCN044
Lớp: E22CQCN04-B

Mục Lục

Phần Mở Đầu	4
CHƯƠNG I: TỔNG QUAN VỀ ỨNG DỤNG	6
1.1 Mục tiêu	6
1.2 Tính năng chính	6
1.3 Phạm vi dự án	7
1.4 Các bài toán cần giải quyết	8
1.4.1 Các bài toán.....	8
1.4.2 Giải pháp	8
CHƯƠNG II: PHÂN TÍCH THIẾT KẾ HỆ THỐNG ỨNG DỤNG.....	11
2.1 Mô tả bằng ngôn ngữ tự nhiên.....	11
2.2 Yêu cầu người dùng	11
2.3 Mô tả yêu cầu phần mềm	12
2.3.1 Usecase cho khách vãng lai	12
2.3.2 Yêu cầu chức năng	13
2.3.3 Yêu cầu phi chức năng.....	13
2.3.4 Vẽ biểu đồ UC tổng quát	14
2.3.5 Kịch bản của các chức năng	15
CHƯƠNG III: CÔNG NGHỆ SỬ DỤNG.....	19
3.1 Các công nghệ sử dụng cho Frontend	19
3.1.1 Giới thiệu về React	19
3.1.2 Một số thư viện khác được sử dụng	22
3.2 Các công nghệ sử dụng cho Backend	24
3.2.1 Giới thiệu hệ quản trị cơ sở dữ liệu MongoDB.....	24
3.2.2 Giới thiệu Nodejs	26
3.2.3 Giới thiệu Express.....	28
3.3 Các công nghệ sử dụng cho xử lý dữ liệu và training model gợi ý món ăn	29
3.3.1 Mô hình K-Nearest Neighbors (KNN).....	29
3.3.2 Công nghệ xử lý dữ liệu.....	32
CHƯƠNG IV: XÂY DỰNG ỨNG DỤNG SMARTBYTES.....	33
4.1 Phân tích dữ liệu mong muốn.....	33
4.1.1 Mục tiêu và đặc điểm của dữ liệu	33
4.1.2 Nguồn dữ liệu.....	34

4.1.3 Tiền xử lí dữ liệu.....	35
4.1.3.1 Dữ liệu cho mô hình	35
4.1.3.2 Dữ liệu cho hệ thống	37
4.2 Phân tích mô hình KNN trong hệ thống gợi ý món ăn	38
4.2.1 Biểu Diễn Văn Bản(TfidfVectorizer)	38
4.2.2 Chuẩn hóa dữ liệu (StandardScaler).....	41
4.2.3 Huấn luyện mô hình	43
4.2.4 Đưa dữ liệu đầu vào và cho ra kết quả	43
CHƯƠNG V: CÀI ĐẶT VÀ THỰC NGHIỆM	45
5.1 Cài đặt	45
5.1.1 Cài đặt NodeJS.....	45
5.1.2 cài đặt MongoDB (Window).....	47
5.1.3 Môi trường để tiền xử lí và huấn luyện mô hình.	50
5.1.4 Cấu hình BackEnd	50
5.1.5 Cấu hình Frontend.....	52
5.2 Kết quả thực nghiệm	53
5.2.1Trang chủ.....	53
5.2.2 Trang công thức.....	53
5.2.3 Trang Thông tin công thức	54
5.2.4 Trang gợi ý món ăn.....	55
5.2.5 Trang lưu món ăn.....	55
CHƯƠNG VI: ĐÁNH GIÁ KẾT QUẢ.....	57
6.1 Đánh giá kết quả	57
6.2 Hạn chế.....	57
6.3 Hướng phát triển.....	58
TÀI LIỆU THAM KHẢO	59

Phần Mở Đầu

Trong bối cảnh cuộc sống hiện đại ngày càng bận rộn, việc duy trì một chế độ ăn uống lành mạnh và cân bằng dinh dưỡng trở nên vô cùng quan trọng. Tuy nhiên, không phải ai cũng đủ thời gian và kiến thức để lên thực đơn, lựa chọn món ăn phù hợp với nhu cầu cơ thể. Mọi người thường gặp khó khăn trong việc tìm kiếm các công thức nấu ăn phù hợp với nhu cầu dinh dưỡng cá nhân, đặc biệt là khi cần kiểm soát lượng calo, chất béo, chất xơ và các thành phần dinh dưỡng khác.

Nhận thức được nhu cầu thực tế này, đề tài thực tập "Xây dựng website công thức món ăn với hệ thống gợi ý theo nhu cầu dinh dưỡng" được thực hiện nhằm tạo ra một giải pháp công nghệ hữu ích, giúp người dùng dễ dàng tìm kiếm và lưu trữ các công thức nấu ăn phù hợp với mục tiêu dinh dưỡng của mình.

Website được thiết kế nhằm cung cấp một kho tàng công thức nấu ăn đa dạng, cho phép người dùng dễ dàng khám phá và lưu trữ các món ăn yêu thích mà không cần phải đăng nhập, tạo sự thuận tiện tối đa. Điểm nổi bật của hệ thống là tính năng gợi ý món ăn thông minh, dựa trên các tiêu chí dinh dưỡng cụ thể như lượng calo, chất béo (fat) và chất xơ (fiber),... Cụ thể, hệ thống sẽ phân tích nhu cầu của người dùng và đề xuất ba món ăn phù hợp nhất, giúp họ xây dựng thực đơn không chỉ ngon miệng mà còn đáp ứng các mục tiêu sức khỏe cá nhân. Tính năng này không chỉ hỗ trợ những người quan tâm đến lối sống lành mạnh mà còn phù hợp với các đối tượng có nhu cầu đặc biệt, như giảm cân, tăng cường chất xơ hoặc kiểm soát chất béo.

Dự án không chỉ dừng lại ở việc cung cấp một nền tảng kỹ thuật, mà còn hướng đến việc tạo ra giá trị thực tiễn, góp phần nâng cao nhận thức cộng đồng về tầm quan trọng của dinh dưỡng. Thông qua việc ứng dụng công nghệ hiện đại và giao diện thân thiện, website hứa hẹn mang đến trải nghiệm mượt mà, dễ tiếp cận và phù hợp với mọi đối tượng người dùng. Trong quá trình phát triển, báo cáo này sẽ tập trung vào việc phân tích các công nghệ sử dụng, thiết kế kiến trúc hệ thống, triển khai các tính năng cốt lõi như tìm kiếm công thức, gợi ý món ăn đồng thời đánh giá hiệu suất và đề xuất hướng cải tiến trong tương lai.

Báo cáo không chỉ là báo cáo kỹ thuật mà còn là cơ hội để rút ra những bài học kinh nghiệm trong quá trình xây dựng một sản phẩm công nghệ từ ý tưởng đến hiện thực. Với tâm huyết và nỗ lực, em hy vọng rằng dự án này không chỉ hoàn

thành tốt các mục tiêu học thuật mà còn đặt nền móng cho một giải pháp thực tiễn, hỗ trợ người dùng trong việc xây dựng lối sống lành mạnh và bền vững. Website công thức nấu ăn này là bước đầu tiên trong hành trình ứng dụng công nghệ để cải thiện chất lượng cuộc sống, và em tin rằng nó sẽ mang lại những giá trị thiết thực cho cộng đồng.

Bố cục báo cáo gồm 6 chương:

- Chương I: tổng quan về ứng dụng.
- Chương II: Phân tích thiết kế hệ thống ứng dụng.
- Chương III: Công nghệ sử dụng.
- Chương IV: Xây dựng ứng dụng SmartBytes.
- Chương V: Cài đặt và thực nghiệm.
- Chương VI: Đánh giá kết quả.

CHƯƠNG I: TỔNG QUAN VỀ ỨNG DỤNG

Chào mừng bạn đến với SmartBites - website công thức nấu ăn thông minh, nơi bạn có thể khám phá các món ăn ngon và nhận gợi ý phù hợp với nhu cầu dinh dưỡng của mình! SmartBites không chỉ là một thư viện công thức đa dạng mà đề xuất món ăn dựa trên các yếu tố mà bạn quan tâm. Với giao diện trực quan và thân thiện, chúng tôi mong muốn mang đến trải nghiệm ẩm thực cá nhân hóa, giúp bạn dễ dàng nấu những bữa ăn vừa ngon miệng vừa tốt cho sức khỏe.

1.1 Mục tiêu

- Cá nhân hóa gợi ý: Cho phép người dùng chọn yếu tố dinh dưỡng họ muốn tập trung (ví dụ: giảm chất béo, tăng chất xơ) và đề xuất công thức phù hợp.
- Giới thiệu món ăn trực quan: Xây dựng giao diện đẹp mắt để hiển thị công thức và thông tin dinh dưỡng chi tiết.
- Hỗ trợ sức khỏe cá nhân: Giúp người dùng đạt được mục tiêu dinh dưỡng riêng thông qua các gợi ý thông minh.
- Khám phá giá trị thực tiễn: Phát triển một nền tảng hữu ích, áp dụng kiến thức vào việc giải quyết nhu cầu ẩm thực thực tế của người dùng.
- Thúc đẩy lối sống lành mạnh: Bằng cách đơn giản hóa quá trình lựa chọn và chuẩn bị bữa ăn dinh dưỡng, SmartBytes khuyến khích người dùng hình thành thói quen ăn uống khoa học, góp phần nâng cao sức khỏe và chất lượng cuộc sống.

1.2 Tính năng chính

- Kho công thức nấu ăn phong phú:
 - SmartBytes cung cấp một danh mục công thức đa dạng, từ các món ăn truyền thống đến hiện đại, phù hợp với nhiều khẩu vị và văn hóa ẩm thực khác nhau.
 - Mỗi công thức đi kèm thông tin chi tiết về thành phần, hướng dẫn chế biến, và giá trị dinh dưỡng (calo, chất béo, chất xơ, v.v.), giúp người dùng dễ dàng nắm bắt thông tin.
- Lưu công thức không cần đăng nhập:
 - Người dùng có thể lưu các món ăn yêu thích trực tiếp trên website mà không cần tạo tài khoản hoặc đăng nhập, mang lại sự tiện lợi tối đa.

- Tính năng này được thiết kế để phù hợp với những người dùng muốn sử dụng nhanh chóng, không bị ràng buộc bởi các bước đăng ký phức tạp.
- Hệ thống gợi ý món ăn thông minh:
 - Điểm nổi bật của SmartBytes là hệ thống gợi ý món ăn dựa trên nhu cầu dinh dưỡng cá nhân. Người dùng có thể nhập các tiêu chí như lượng calo mong muốn, mức chất béo (fat), hoặc chất xơ (fiber), và hệ thống sẽ tự động đề xuất ba món ăn phù hợp nhất.
 - Hệ thống sử dụng thuật toán phân tích dữ liệu dinh dưỡng để đảm bảo các gợi ý không chỉ đáp ứng yêu cầu mà còn cân bằng về hương vị và tính thực tiễn trong chế biến.
- Giao diện thân thiện và dễ sử dụng:
 - Website được thiết kế với giao diện trực quan, dễ điều hướng, phù hợp với mọi đối tượng người dùng, từ người mới bắt đầu nấu ăn đến những đầu bếp gia đình dày dạn kinh nghiệm.

1.3 Phạm vi dự án

- Phạm vi chức năng
 - Hệ thống hiển thị và quản lý cơ sở dữ liệu công thức nấu ăn với thông tin chi tiết về nguyên liệu, cách chế biến và giá trị dinh dưỡng.
 - Tính năng tìm kiếm và lọc công thức theo tên món ăn, nguyên liệu chính và loại món.
 - Hệ thống gợi ý thông minh dựa trên nhu cầu dinh dưỡng, cho phép người dùng nhập các tiêu chí về calo, chất béo, chất xơ, protein,... và nhận được 3 gợi ý món ăn phù hợp nhất.
 - Chức năng lưu trữ món ăn yêu thích sử dụng bộ nhớ trình duyệt, không yêu cầu đăng nhập.
 - Giao diện responsive thân thiện với người dùng, tối ưu cho cả desktop và mobile.
- Phạm vi kỹ thuật
 - Dự án được giới hạn trong việc phát triển ứng dụng web front-end với các công nghệ web và framework chuẩn (HTML, CSS, JavaScript, React). Dữ liệu công thức được lưu trữ dưới dạng cơ sở dữ liệu đơn giản. Hệ thống không bao gồm các tính năng back-end phức tạp như

xử lý thanh toán, quản lý người dùng nâng cao hay tích hợp API bên thứ ba.

1.4 Các bài toán cần giải quyết

1.4.1 Các bài toán

Để xây dựng một website SmartBytes hiệu quả và hoạt động ổn định, chúng tôi cần giải quyết một số bài toán kỹ thuật quan trọng, đặc biệt là khi tối ưu việc truy xuất dữ liệu và tích hợp các tính năng thông minh. Các bài toán chính bao gồm:

- Làm sạch và chuẩn hóa dữ liệu
 - Dữ liệu công thức nấu ăn(nguyên liệu, cách chế biến, giá trị dinh dưỡng) thường được thu thập từ nguồn trên Kaggle:
 - Dữ liệu không đồng nhất: có nguyên liệu sẽ “kcal”, còn có nguyên liệu chỉ là “cal”
 - Thiếu thông tin: Một số công thức có thể thiếu một số giá trị dinh dưỡng(calor, chất béo, chất xơ)
 - Dữ liệu không đồng nhất: dữ liệu như hướng dẫn nấu ăn hoặc nguyên liệu sẽ được định dạng khác nhau
- Hiện thị danh sách công thức với số lượng công thức lớn
 - Website cần hiển thị danh sách 50,000 công thức nấu ăn, nhưng nếu tải toàn bộ danh sách cùng lúc, hệ thống dữ liệu sẽ không chịu được và xảy ra lỗi, nếu hệ thống chịu được thì thời gian tải trang sẽ lâu, gây ảnh hưởng đến trải nghiệm người dùng.
- Cách backend đọc được model training
 - Hệ thống gợi ý món ăn dựa trên nhu cầu dinh dưỡng(calor, chất béo, chất xơ) yêu cầu một mô hình(model) để phân tích và đề xuất công thức phù hợp. Yêu cầu backend phải đọc được mô hình để xử lý yêu cầu gợi ý từ người dùng.

1.4.2 Giải pháp

- Làm sạch và chuẩn hóa dữ liệu
 - Chuẩn hóa
 - Chuẩn hóa lại định dạng dữ liệu bằng các hàm xử lý văn bản hoặc regex trong python.
 - Chuẩn hóa các đơn vị dinh dưỡng về một chuẩn chung để thuận tiện cho việc training.

- Xử lý dữ liệu thiếu
 - Nếu một công thức bị thiếu thông tin dinh dưỡng, sử dụng giá trị bằng 0 hoặc trung bình tham khảo từ các công thức tương tự.
- Công cụ và công nghệ:
 - Sử dụng python và thư viện pandas để làm sạch dữ liệu.
 - Lưu dữ liệu đã làm sạch vào cơ sở dữ liệu (MongoDB) và file csv.
- Hiển thị danh sách công thức
 - Triển khai lazy loading trên frontend.
 - Sử dụng JavaScript (hoặc framework như React/Vue.js) để chỉ tải một số lượng nhỏ công thức (ví dụ: 20 công thức) khi người dùng mở trang.
 - Khi người dùng cuộn xuống cuối danh sách, gửi yêu cầu API để lấy thêm 20 công thức tiếp theo.
 - Sử dụng Intersection Observer API trong JavaScript để phát hiện khi người dùng cuộn đến cuối trang, kích hoạt tải thêm dữ liệu.
 - Tối ưu backend để hỗ trợ lazy loading
 - Backend (Node.js) cung cấp API phân trang (pagination), ví dụ: `/api/recipes?page=1&limit=20`.
 - Sử dụng truy vấn cơ sở dữ liệu hiệu quả để chỉ lấy một phần dữ liệu mỗi lần.
 - Tạo chỉ mục (index) cho các cột thường xuyên truy vấn (như tên món, danh mục) để tăng tốc độ tìm kiếm.
- Cách back end đọc được model training
 - Lưu trữ và triển khai model
 - Lưu mô hình dưới dạng tệp (ví dụ: `model.pkl` nếu dùng Python và thư viện `scikit-learn`).
 - Cách Back end đọc model

- Backend (Node.js) tải tệp mô hình khi khởi động server, sử dụng thư viện như pickle (Python) hoặc TensorFlow.js (Node.js).
- Khi người dùng gửi yêu cầu gợi ý qua API, backend truyền dữ liệu đầu vào (calo, chất béo, chất xơ) vào mô hình.
- Mô hình trả về danh sách ba công thức phù hợp nhất, backend truy vấn cơ sở dữ liệu để lấy chi tiết công thức và gửi về frontend.

CHƯƠNG II: PHÂN TÍCH THIẾT KẾ HỆ THỐNG ỨNG DỤNG

2.1 Mô tả bằng ngôn ngữ tự nhiên

SmartBytes là một website công thức nấu ăn được thiết kế đặc biệt cho khách vãng lai – những người muốn nhanh chóng tìm kiếm và tham khảo các công thức mà không cần bất kỳ sự ràng buộc nào.

Khi truy cập SmartBytes, người dùng sẽ thấy ngay một kho tàng món ăn phong phú. Có thể dễ dàng tìm kiếm công thức theo tên món, nguyên liệu có sẵn, hoặc loại món ăn mong muốn (ví dụ: món canh, món xào, món tráng miệng). Mỗi công thức đều được trình bày rõ ràng với danh sách nguyên liệu, định lượng và các bước thực hiện chi tiết, giúp việc nấu ăn trở nên đơn giản hơn bao giờ hết.

Điểm đặc biệt nhất của SmartBytes dành cho khách vãng lai chính là hệ thống gợi ý món ăn theo nhu cầu dinh dưỡng. Người dùng không cần đăng ký hay đăng nhập. Nếu muốn ăn uống lành mạnh hơn hoặc có mục tiêu dinh dưỡng cụ thể (như cần ít calo, nhiều chất xơ), chỉ cần nhập các con số đó. Ngay lập tức, SmartBytes sẽ đưa ra ba gợi ý món ăn phù hợp nhất với yêu cầu. Điều này giúp dễ dàng lựa chọn bữa ăn khoa học mà không tốn thời gian tính toán.

Hơn nữa, nếu tìm thấy một công thức ưng ý và muốn lưu lại để dùng sau, có thể thêm vào danh sách yêu thích chỉ với một cú nhấp chuột, mà không cần tạo tài khoản. Danh sách này sẽ được lưu trữ trực tiếp trên trình duyệt, giúp dễ dàng xem lại các món ăn đã chọn mỗi khi ghé thăm lại website.

Với SmartBytes, việc tìm kiếm công thức, lập kế hoạch bữa ăn và ăn uống lành mạnh trở nên nhanh chóng, tiện lợi và không hề rườm rà, đúng như một công cụ hữu ích dành cho mọi khách vãng lai.

2.2 Yêu cầu người dùng

Tìm kiếm và Duyệt Công thức: Người dùng cần có khả năng tìm kiếm công thức nấu ăn theo nhiều tiêu chí khác nhau như tên món ăn hoặc thời gian chế biến. Hệ thống cần hiển thị kết quả tìm kiếm một cách trực quan với hình ảnh, tên món và thông tin cơ bản về từng công thức.

Xem Chi tiết Công thức Khi chọn một món ăn, người dùng cần được cung cấp thông tin đầy đủ bao gồm danh sách nguyên liệu với định lượng cụ thể, các bước

thực hiện chi tiết, thời gian chuẩn bị và nấu nướng, độ khó của món ăn, và đặc biệt là thông tin dinh dưỡng chi tiết (calo, chất béo, carbohydrate, protein, chất xơ, vitamin và khoáng chất).

Hệ thống Gợi ý Dinh dưỡng Người dùng có nhu cầu nhập các tiêu chí dinh dưỡng cụ thể như lượng calo mong muốn hàm lượng chất béo tối đa, lượng chất xơ tối thiểu, hoặc protein cần thiết. Dựa trên những thông số này, hệ thống cần đưa ra 3 gợi ý món ăn phù hợp nhất, kèm theo giải thích tại sao những món này được lựa chọn.

Lưu trữ Món ăn Yêu thích Người dùng muốn có thể lưu các công thức yêu thích để dễ dàng truy cập lại sau này mà không cần phải tạo tài khoản. Danh sách món ăn đã lưu cần được duy trì ngay cả khi đóng trình duyệt và mở lại, với khả năng thêm, xóa và sắp xếp các món ăn đã lưu.

2.3 Mô tả yêu cầu phần mềm

Hệ thống phục vụ nhóm Người dùng vắng lai là chủ yếu.

2.3.1 Usecase cho khách vắng lai

STT	UseCase	Mô tả chi tiết
1	Tìm kiếm công thức nấu ăn	Người dùng tìm kiếm các công thức nấu ăn dựa trên các tiêu chí khác nhau để tìm ra món ăn phù hợp với nhu cầu
2	Xem chi tiết công thức món ăn	Người dùng xem thông tin chi tiết của một công thức nấu ăn cụ thể
3	Nhận gợi ý món ăn theo nhu cầu dinh dưỡng	Người dùng cung cấp các chỉ số dinh dưỡng mong muốn và nhận được 3 gợi ý món ăn phù hợp

4	Lưu công thức yêu thích	Người dùng lưu trữ các công thức yêu thích để dễ dàng truy cập lại sau này
---	-------------------------	--

2.3.2 Yêu cầu chức năng

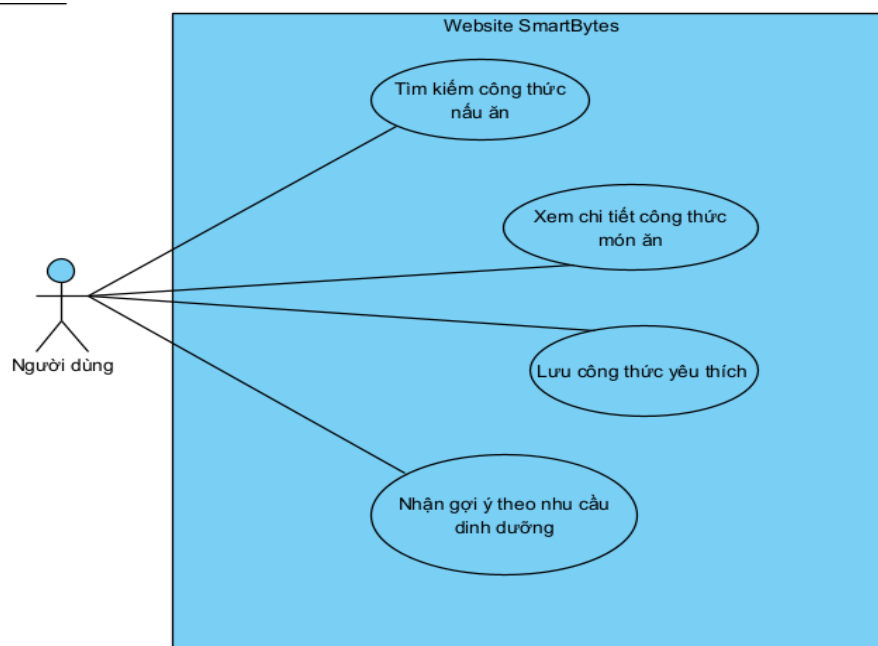
- Quản lý Công thức Nấu ăn
 - Hệ thống hiển thị danh sách các công thức nấu ăn với thông tin cơ bản (tên món, thời gian chế biến, hình ảnh).
 - Hệ thống cung cấp thông tin chi tiết công thức bao gồm nguyên liệu, cách chế biến, tên, hình ảnh, thời gian chuẩn bị, nấu.
 - Hệ thống hiển thị thông tin dinh dưỡng chi tiết như calo, chất béo, chất xơ, protein, Carbohydrate.
- Tính năng tìm kiếm
 - Hệ thống cho phép tìm kiếm công thức theo tên món ăn.
 - Hệ thống cho phép tìm kiếm theo thời gian chế biến, số lượng nguyên liệu, thời gian chuẩn bị nguyên liệu và lượng calories.
- Hệ thống gợi ý dinh dưỡng
 - Hệ thống cho phép người dùng nhập các tiêu chí dinh dưỡng (calo, chất béo, chất xơ, protein,...).
 - Hệ thống phân tích các tiêu chí đã nhập và đưa ra 3 gợi ý món ăn phù hợp nhất dựa trên thuật toán KNN.
- Quản lý món ăn yêu thích
 - Hệ thống cho phép lưu công thức vào danh sách yêu thích không cần đăng nhập.
 - Hệ thống hiển thị danh sách món ăn đã lưu với thông tin cơ bản.
 - Hệ thống cho phép xóa món ăn khỏi danh sách yêu thích.
- Điều hướng và giao diện
 - Hệ thống cung cấp menu điều hướng rõ ràng giữa các trang chính.

2.3.3 Yêu cầu phi chức năng

- Hiệu suất
 - Thời gian tải trang chủ không vượt quá 3 giây trên kết nối internet trung bình.
 - Thời gian phản hồi tìm kiếm không vượt quá 2 giây.

- Hệ thống gợi ý dinh dưỡng phải trả kết quả trong vòng 5 giây.
- Ứng dụng hoạt động mượt mà khi tải các công thức.
- Khả năng sử dụng
 - Giao diện đơn giản, trực quan, phù hợp với người dùng từ 18-65 tuổi.
 - Không yêu cầu hướng dẫn sử dụng cho các chức năng cơ bản.
 - Các thao tác chính được thực hiện không được quá phức tạp.
 - Thông báo lỗi và hướng dẫn được hiển thị rõ ràng.
- Khả năng tương thích
 - Tương thích với các trình duyệt phổ biến (Chrome, Firefox, Safari, Edge) phiên bản mới nhất.
 - Responsive design hoạt động tốt trên desktop (1920x1080), tablet (768x1024), mobile (375x667).
 - Hỗ trợ cả thao tác chuột và cảm ứng.
 - Tương thích với các hệ điều hành phổ biến (Windows, macOS, iOS, Android).
- Khả năng bảo trì
 - Code được tổ chức theo cấu trúc rõ ràng, dễ hiểu.
 - Comment đầy đủ cho các hàm và logic phức tạp.
 - Sử dụng naming convention nhất quán.
 - Tách biệt logic xử lý và giao diện để dễ dàng cập nhật.

2.3.4 Vẽ biểu đồ UC tổng quát



2.3.5 Kịch bản của các chức năng

1. Quản lý công thức nấu ăn

1.1 Hiện thị danh sách công thức nấu ăn

- Mô tả: Người dùng muốn xem danh sách các công thức nấu ăn với thông tin cơ bản.
- Kịch bản:
 - Người dùng truy cập vào mục “Công thức” trên thành điều hướng của SmartBytes.
 - Hệ thống hiện thị danh sách công thức dạng lưới, bao gồm:
 - Hình ảnh minh họa món ăn.
 - Tên món ăn.
 - Thời gian chế biến.
 - Nút lưu món ăn.
 - Người dùng cuộn trang để xem thêm công thức (sử dụng lazy loading để tải 20 công thức mỗi lần).
- Kết quả mong đợi: Người dùng thấy danh sách công thức với thông tin cơ bản, tải nhanh và dễ xem.
- Ngoại lệ: Nếu không có công thức nào trong cơ sở dữ liệu, hệ thống hiện thị thông báo “Hiện chưa có công thức nào”.

1.2 Xem thông tin chi tiết công thức

Mô tả: Người dùng muốn xem chi tiết công thức cụ thể

- Kịch bản:
 - Người dùng nhấp vào một công thức từ danh sách.
 - Hệ thống chuyển đến trang chi tiết, hiện thị:
 - Tên món ăn.
 - Hình ảnh lớn của món ăn.
 - Danh sách nguyên liệu.
 - Thời gian chuẩn bị, thời gian nấu ăn.
 - Thông tin dinh dưỡng: calo, chất béo, chất xơ, protein, carbohydrate,...
 - Các bước chế biến được liệt kê rõ ràng, dễ theo dõi.

- Người dùng có thể nhấp nút “Quay lại ” để trở về danh sách hoặc “Lưu ” để thêm vào danh sách yêu thích .
- Kết quả mong đợi: Người dùng thấy đầy đủ thông tin chi tiết của công thức, dễ hiểu và rõ ràng.
- Ngoại lệ: Nếu công thức thiếu dữ liệu, hệ thống sẽ để trống hoặc không hiện dữ liệu đó.

2. Tìm kiếm công thức nấu ăn

2.1 Tìm kiếm

- Mô tả: Người dùng muốn tìm công thức bằng cách nhập tên món ăn
- Kịch bản:
 - Người dùng vào trang “Công thức”.
 - Người dùng bấm vào mục tìm kiếm.
 - Họ nhập từ khóa vào thanh tìm kiếm, ví dụ: “chicken”.
 - Người dùng nhấn enter hoặc ấn nút “lọc”.
 - Hệ thống trả về danh sách các công thức có tên phù hợp với từ khóa đã nhập (ví dụ: Chicken masala).
 - Người dùng nhấp vào một công thức để xem chi tiết.
- Kết quả mong đợi: Hệ thống hiển thị chính xác các công thức khớp với từ khóa.
- Ngoại lệ: Nếu không tìm thấy công thức, hệ thống hiển thị “Không tìm thấy kết quả”.

2.2 Tìm kiếm theo thời gian chế biến, số lượng nguyên liệu, thời gian chuẩn bị, và lượng calories

- Mô tả: Người dùng muốn tìm công thức dựa trên các tiêu chí cụ thể.
- Kịch bản:
 - Người dùng vào trang “Recipe”.
 - Người dùng bấm vào mục tìm kiếm.
 - Ở dưới thanh tìm kiếm, sẽ có các tiêu chí.
 - Họ chọn các tiêu chí, tiêu chí theo khoảng số lượng:
 - Thời gian chế biến.
 - Số lượng nguyên liệu.
 - Thời gian chuẩn bị.
 - Số lượng calories.

- Người dùng nhấn nút “Lọc”.
- Hệ thống trả về danh sách các công thức phù hợp.
- Người dùng nhấp vào một công thức để xem chi tiết.
- Kết quả mong đợi: Hệ thống hiển thị các công thức đáp ứng đầy đủ hoặc gần đúng các tiêu chí.
- Ngoại lệ: Nếu không có công thức phù hợp, hệ thống hiển thị “Không tìm thấy công thức phù hợp”.

3. Hệ thống gợi ý dinh dưỡng

- Mô tả: Người dùng muốn nhận gợi ý món ăn dựa trên nhu cầu dinh dưỡng, sử dụng thuật toán KNN.
- Kịch bản:
 - Người dùng vào mục “Gợi ý”.
 - Họ nhập các tiêu chí dinh dưỡng: calories, fat, carbohydrates, protein, cholesterol, sodium, fiber, ingredients.
 - Người dùng nhấn nút “Gợi ý”.
 - Hệ thống sử dụng mô hình KNN để so sánh yêu cầu với dữ liệu công thức đã có sẵn trong model, model đã được train từ trước từ file csv đã được xử lý dữ liệu, chọn ra ba món phù hợp nhất.
 - Người dùng có thể nhấp vào một gợi ý công thức để xem chi tiết công thức.
- Kết quả mong đợi: Hệ thống đề xuất chính xác hoặc gần với tiêu chí của người dùng về ba món ăn phù hợp, trong vòng 5 giây.
- Ngoại lệ: Nếu tiêu chí khác xa với những dữ liệu có sẵn trong hệ thống, hệ thống có thể gợi ý 3 món có tiêu chí chênh lệch lớn với tiêu chí ban đầu của người dùng.

4. Quản lý món ăn yêu thích

- Mô tả: Người dùng muốn lưu công thức mình yêu thích.
- Kịch bản:
 - Người dùng nhấn nút lưu ở danh sách công thức hoặc nhấn nút lưu ở chi tiết một công thức.
 - Hệ thống lưu công thức vào bộ nhớ trình duyệt(local storage) và hiển thị thông báo “Đã lưu công thức”.
 - Người dùng vào trang “Lưu công thức”.

- Hệ thống hiển thị công thức mà người dùng đã lưu trên trình duyệt này. Mỗi công thức hiển thị thông tin cơ bản(tên món, hình).
- Kết quả mong đợi: Danh sách yêu thích hiển thị đầy đủ, rõ ràng, tải nhanh, sau khi ấn lưu, công thức đã lưu phải nổi bật hơn các công thức khác.
- Ngoại lệ: Nếu chưa có công thức nào được lưu, hệ thống hiển thị “Danh sách yêu thích trống”.

CHƯƠNG III: CÔNG NGHỆ SỬ DỤNG

3.1 Các công nghệ sử dụng cho Frontend

Để xây dựng một giao diện người dùng hiện đại, tương tác cao và một ứng dụng web hiệu quả, dự án sử dụng React – một thư viện JavaScript để xây dựng giao diện người dùng. Dưới đây là mô tả chi tiết về vai trò và cách áp dụng công nghệ này trong dự án.

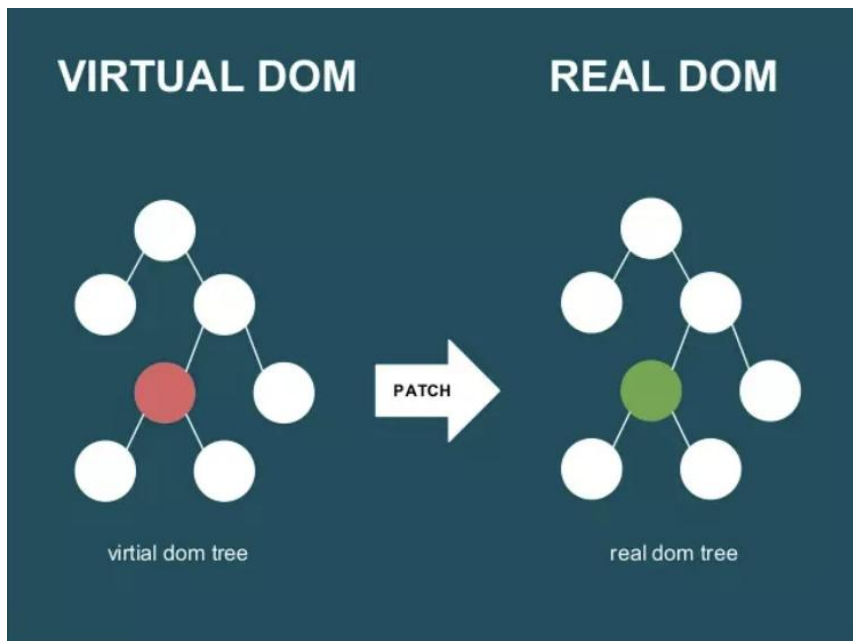
3.1.1 Giới thiệu về React

React.js là một thư viện Javascript đang nổi lên trong những năm gần đây với xu hướng Single Page Application. Trong khi những framework khác cố gắng hướng đến một mô hình MVC hoàn thiện thì React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác. Nếu như AngularJS là một Framework cho phép nhúng code javascript trong code html thông qua các attribute như ng-model, ng-repeat...thì với react là một library cho phép nhúng code html trong code javascript nhờ vào JSX, bạn có thể dễ dàng lồng các đoạn HTML vào trong JS. Tích hợp giữa javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn.

Các tính năng nổi bật của ReactJS:

1.Virtual DOM

- công nghệ DOM ảo giúp tăng hiệu năng cho ứng dụng. Việc chỉ node gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc DOM tree cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. React JS sử dụng Virtual DOM (DOM ảo) để cải thiện vấn đề này. Virtual DOM là một object Javascript, mỗi object chứa đầy đủ thông tin cần thiết để tạo ra một DOM, khi dữ liệu thay đổi nó sẽ tính toán sự thay đổi giữa object và tree thật, điều này sẽ giúp tối ưu hoá việc re-render DOM tree thật.



- React sử dụng cơ chế one-way data binding – luồng dữ liệu 1 chiều. Dữ liệu được truyền từ parent đến child thông qua props. Luồng dữ liệu đơn giản giúp chúng ta dễ dàng kiểm soát cũng như sửa lỗi.
- Với các đặc điểm ở trên, React dùng để xây dựng các ứng dụng lớn mà dữ liệu của chúng thay đổi liên tục theo thời gian. Dữ liệu thay đổi thì hầu hết kèm theo sự thay đổi về giao diện. Ví dụ như Facebook: trên Newsfeed của bạn cùng lúc sẽ có các status khác nhau và mỗi status lại có số like, share, comment liên tục thay đổi. Khi đó React sẽ rất hữu ích để sử dụng.

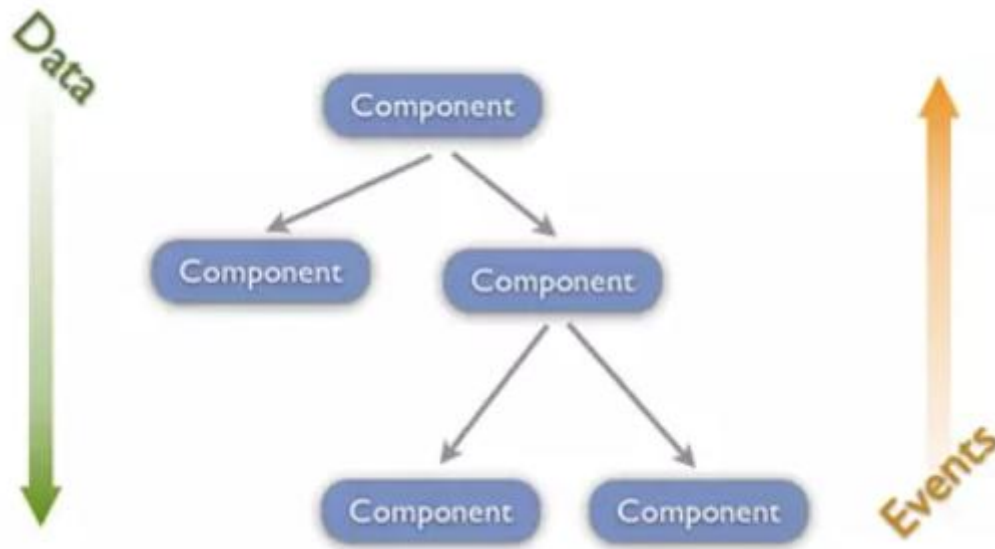
2. JSX

- JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javascript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript. Ngược với Javascript, JSX là kiểu statically-typed, nghĩa là nó được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch. Ngoài ra, nó cũng cung cấp tính năng gỡ lỗi khi biên dịch rất tốt. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascripts có thể sử dụng.

3. Component

- React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Trong React, chúng ta xây dựng trang web sử dụng những thành phần (component) nhỏ. Chúng ta có thể tái sử dụng một

component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác.



- Mỗi component trong React có một trạng thái riêng, có thể thay đổi, và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái. Mọi thứ React đều là component. Chúng giúp bảo trì mã code khi làm việc với các dự án lớn. Một react component đơn giản chỉ cần một method render. Có rất nhiều methods khả dụng khác, nhưng render là method chủ đạo.

4. Prop và State

- Props: giúp các component tương tác với nhau, component nhận input gọi là props, và trả thuộc tính mô tả những gì component con sẽ render. Prop là bất biến. State: thể hiện trạng thái của ứng dụng, khi state thay đổi thì component đồng thời render lại để cập nhật UI.

5. Redux

- Redux là một thư viện quản lý trạng thái cho các ứng dụng ReactJS. Nó giúp quản lý trạng thái của ứng dụng một cách chính xác và dễ dàng, đồng thời giúp tăng tính linh hoạt và khả năng mở rộng của ứng dụng.

Vai trò của React trong dự án:

- Tạo các thành phần giao diện tái sử dụng:
 - Mỗi công thức nấu ăn được hiển thị dưới dạng một RecipeCard component, chứa thông tin cơ bản như tên món, thời gian chế biến, và

hình ảnh. Component này được tái sử dụng trong danh sách công thức và mục yêu thích.

- Component `RecipeDetail` hiển thị chi tiết công thức, bao gồm nguyên liệu, hướng dẫn chế biến, và thông tin dinh dưỡng (calo, chất béo, chất xơ, protein, carbohydrate).
- Component `SearchBar` cho phép người dùng nhập từ khóa tìm kiếm hoặc tiêu chí dinh dưỡng.
- Quản lý trạng thái (state):
 - React sử dụng state để quản lý dữ liệu động, như danh sách công thức đã lưu trong bộ nhớ trình duyệt (local storage) hoặc các tiêu chí dinh dưỡng mà người dùng nhập vào để gợi ý món ăn.
 - Ví dụ: Khi người dùng nhấp nút “Lưu” trên một công thức, state được cập nhật để thêm công thức vào danh sách yêu thích mà không cần tải lại trang.
- Tương tác người dùng:
 - React xử lý các sự kiện như nhấp chuột (`onClick`) để lưu/xóa công thức, nhập liệu (`onChange`) trong thanh tìm kiếm, hoặc gửi yêu cầu gợi ý món ăn.
 - Ví dụ: Khi người dùng nhập “Chicken” vào thanh tìm kiếm, React gửi yêu cầu API và cập nhật giao diện với danh sách công thức phù hợp.
- Hiệu suất giao diện:
 - React sử dụng Virtual DOM để chỉ cập nhật những phần giao diện thay đổi (ví dụ: danh sách công thức mới khi cuộn trang), giúp website tải nhanh hơn, đặc biệt với danh sách 50,000 công thức.

3.1.2 Một số thư viện khác được sử dụng

- React Intersection Observer
 - là một thư viện hỗ trợ triển khai lazy loading bằng cách phát hiện khi một phần tử xuất hiện trong khung nhìn (viewport) của trình duyệt.
 - Vai trò:

- Hỗ trợ lazy loading cho danh sách 50,000 công thức, chỉ tải thêm công thức khi người dùng cuộn đến cuối trang.
- Giảm tải cho trình duyệt bằng cách chỉ render các component RecipeCard khi cần thiết.
- Ví dụ:
 - Khi người dùng cuộn xuống cuối danh sách công thức, React Intersection Observer phát hiện và kích hoạt yêu cầu API để tải thêm 20 công thức tiếp theo.
- React Router
 - là một thư viện tiêu chuẩn trong hệ sinh thái React, được sử dụng để quản lý định tuyến (routing) trong các ứng dụng React. Mặc dù Next.js có hệ thống định tuyến riêng dựa trên tệp, React Router vẫn là một lựa chọn mạnh mẽ và linh hoạt, đặc biệt nếu bạn muốn kiểm soát chi tiết hơn về cách các thành phần (components) được hiển thị dựa trên URL, hoặc nếu bạn đã quen với cách tiếp cận này từ các dự án React thuần túy.
 - Trong dự án SmartBytes, React Router được sử dụng để:
 - Tạo trải nghiệm ứng dụng đơn trang (SPA): Thay vì tải lại toàn bộ trang web mỗi khi người dùng di chuyển giữa các phần (ví dụ: từ "Trang chủ" sang "Công thức" hoặc "Món ăn yêu thích"), React Router cho phép thay đổi nội dung hiển thị mà không cần tải lại trang. Điều này mang lại một trải nghiệm mượt mà, nhanh chóng và giống như một ứng dụng máy tính để bàn hơn là một website truyền thống.
 - Quản lý các đường dẫn URL (URLs): React Router cho phép định nghĩa các ánh xạ rõ ràng giữa các đường dẫn URL và các thành phần React tương ứng. Ví dụ:
 - Điều hướng linh hoạt: Cung cấp các component như <Link> để tạo các liên kết điều hướng nội bộ mà không làm tải lại trang, và useNavigate (hoặc useHistory trong các phiên bản cũ hơn) để điều hướng chương trình (ví dụ: sau khi tìm kiếm, chuyển hướng đến trang kết quả).

- Truyền tham số qua URL: Cho phép truyền dữ liệu giữa các trang thông qua các tham số URL (ví dụ: ID của công thức để hiển thị chi tiết). Điều này rất quan trọng cho việc xem chi tiết từng công thức món ăn.

3.2 Các công nghệ sử dụng cho Backend

Dự án SmartBytes là một website công thức nấu ăn được xây dựng với frontend sử dụng React, cung cấp các tính năng như hiển thị danh sách công thức, tìm kiếm, gợi ý món ăn theo nhu cầu dinh dưỡng (calo, chất béo, chất xơ, protein, carbohydrate), và lưu công thức yêu thích mà không cần đăng nhập. Để hỗ trợ các tính năng này, backend của SmartBytes sử dụng Node.js, Express, và MongoDB. Các công nghệ này đảm bảo xử lý dữ liệu hiệu quả, tích hợp thuật toán gợi ý (KNN), và hỗ trợ quy mô lớn với 50,000 công thức. Dưới đây là mô tả chi tiết về vai trò, cách áp dụng, lợi ích, và hạn chế của từng công nghệ trong backend.

3.2.1 Giới thiệu hệ quản trị cơ sở dữ liệu MongoDB

- MongoDB là một phần mềm mã nguồn mở dùng để quản trị cơ sở dữ liệu NoSQL.
- NoSQL (Not only SQL) được sử dụng thay thế cho cơ sở dữ liệu quan hệ (Relational Database – RDB) truyền thống. Cơ sở dữ liệu NoSQL khá hữu ích trong khi làm việc với các tập dữ liệu phân tán lớn. MongoDB là một công cụ có thể quản lý thông tin hướng document cũng như lưu trữ hoặc truy xuất thông tin.
- MongoDB là một chương trình cơ sở dữ liệu mã nguồn mở được thiết kế theo kiểu hướng đối tượng trong đó các bảng được cấu trúc một cách linh hoạt cho phép các dữ liệu lưu trên bảng không cần phải tuân theo một dạng cấu trúc nhất định nào. Chính do cấu trúc linh hoạt này nên MongoDB có thể được dùng để lưu trữ các dữ liệu có cấu trúc phức tạp và đa dạng và không cố định (hay còn gọi là Big Data).
- MongoDB giúp các tổ chức lưu trữ lượng lớn dữ liệu trong khi vẫn hoạt động nhanh chóng. Ngoài lưu trữ dữ liệu, MongoDB còn được sử dụng trong các trường hợp sau:
 - Tích hợp một lượng lớn dữ liệu đa dạng.
 - Mô tả các cấu trúc dữ liệu phức tạp, biến hoá.

- Cung cấp dữ liệu cho các ứng dụng hiệu suất cao.
- Hỗ trợ các ứng dụng đám mây lai và đa đám mây.
- Hỗ trợ phương pháp phát triển Agile.
- Lợi thế của MongoDB so với các cơ sở dữ liệu dạng quan hệ (RDBMS)
 - Ít Schema hơn: MongoDB là một cơ sở dữ liệu dựa trên Document, trong đó một Collection giữ các Document khác nhau. Số trường, nội dung và kích cỡ của Document này có thể khác với Document khác.
 - Cấu trúc của một đối tượng là rõ ràng.
 - Không có các Join phức tạp.
 - Khả năng truy vấn sâu hơn. MongoDB hỗ trợ các truy vấn động trên các Document bởi sử dụng một ngôn ngữ truy vấn dựa trên Document mà mạnh mẽ như SQL.
 - MongoDB dễ dàng để mở rộng.
 - Việc chuyển đổi/ánh xạ của các đối tượng ứng dụng đến các đối tượng cơ sở dữ liệu là không cần thiết.
 - Sử dụng bộ nhớ nội tại để lưu giữ phần công việc, giúp truy cập dữ liệu nhanh hơn.
- Một số đặc điểm của MongoDB
 - Kho lưu định hướng Document: Dữ liệu được lưu trong các tài liệu kiểu JSON.
 - Lập chỉ mục trên bất kỳ thuộc tính nào.
 - Các truy vấn đa dạng.
 - Cập nhật nhanh hơn.
- So sánh hệ quản trị cơ sở dữ liệu MongoDB với cơ sở dữ liệu quan hệ khác:
 - MongoDB là lựa chọn tốt cho ứng dụng cần dữ liệu linh hoạt, dễ mở rộng, và tích hợp với JavaScript. Nó phù hợp cho các dự án web hiện đại với dữ liệu không đồng nhất, nhưng kém hiệu quả cho truy vấn phức tạp.

- MySQL phù hợp với ứng dụng cần dữ liệu có cấu trúc rõ ràng và truy vấn SQL đơn giản, nhưng khó xử lý dữ liệu linh hoạt hoặc mở rộng ngang.
- PostgreSQL là lựa chọn mạnh mẽ cho ứng dụng phức tạp, cần kết hợp dữ liệu quan hệ và JSON, nhưng yêu cầu kỹ năng cao hơn và cấu hình phức tạp.
- Firebase lý tưởng cho ứng dụng thời gian thực hoặc cần triển khai nhanh trên đám mây, nhưng phụ thuộc vào Google Cloud và hạn chế với truy vấn phức tạp.

3.2.2 Giới thiệu Nodejs

- Nodejs là một nền tảng (Platform) phát triển độc lập được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript. Node.js nổi bật với khả năng xử lý bất đồng bộ, hiệu suất cao và là lựa chọn lý tưởng cho các ứng dụng web cần khả năng mở rộng.
- 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.
- Cách NodeJS hoạt động
 - Ý tưởng chính của Node js là sử dụng non-blocking, hướng sự vào ra dữ liệu thông qua các tác vụ thời gian thực một cách nhanh chóng. Bởi vì, Node js có khả năng mở rộng nhanh chóng, khả năng xử lý một số lượng lớn các kết nối đồng thời bằng thông lượng cao.
 - Nếu như các ứng dụng web truyền thống, các request tạo ra một luồng xử lý yêu cầu mới và chiếm RAM của hệ thống thì việc tài nguyên của hệ thống sẽ được sử dụng không hiệu quả. Chính vì lẽ đó giải pháp mà Node js đưa ra là sử dụng luồng đơn (Single-Threaded), kết hợp với non-blocking I/O để thực thi các request, cho phép hỗ trợ hàng chục ngàn kết nối đồng thời.
- Tác dụng của NodeJS
 - Xây dựng API (Application Programming Interface): Node.js được sử dụng để tạo ra các API RESTful hoặc GraphQL (tùy thuộc vào thiết kế cụ thể) mà Frontend (được xây dựng bằng React/Next.js) sẽ gửi

yêu cầu để lấy dữ liệu công thức, gửi thông số tìm kiếm, hay yêu cầu gợi ý món ăn.

- Xử lý logic nghiệp vụ: Tất cả các logic phức tạp như làm sạch và chuẩn hóa dữ liệu công thức, thực thi thuật toán gợi ý KNN, quản lý việc truy vấn và thao tác với cơ sở dữ liệu MongoDB đều được viết bằng JavaScript/TypeScript và chạy trên môi trường Node.js.
- Khả năng xử lý lượng lớn kết nối đồng thời: Với kiến trúc non-blocking I/O (I/O không chặn), Node.js rất hiệu quả trong việc xử lý nhiều yêu cầu cùng lúc từ người dùng, điều này quan trọng khi có một lượng lớn công thức và nhiều người dùng truy cập.
- Vai trò của Node.Js trong dự án
 - Hỗ trợ Express:
 - Node.js là môi trường chạy các endpoint API được định nghĩa trong Express, xử lý các yêu cầu từ frontend như lấy danh sách công thức, tìm kiếm, hoặc gợi ý món ăn.
 - Ví dụ: API Route `/api/recipes` sử dụng Node.js để truy vấn MongoDB và trả về 20 công thức mỗi lần, hỗ trợ lazy loading.
 - Tích hợp thư viện backend:
 - Node.js cho phép sử dụng các thư viện như mongoose để kết nối với MongoDB hoặc sử dụng module `child_process` để chạy mô hình KNN cho gợi ý món ăn dựa trên tiêu chí dinh dưỡng.
 - Ví dụ: Khi người dùng nhập tiêu chí (400 kcal, thấp chất béo), Node.js chạy mô hình KNN trong API Route `/api/recommend` để trả về ba công thức phù hợp.
- Lợi ích của Node.Js
 - Hiệu suất bất đồng bộ: Node.js xử lý nhiều yêu cầu đồng thời, phù hợp cho các tính năng như tìm kiếm công thức hoặc gợi ý món ăn với lưu lượng truy cập lớn.
 - Tương thích với JavaScript: Vì cả frontend (React, Next.js) và backend đều dùng JavaScript, Node.js giúp đồng bộ mã nguồn, giảm độ phức tạp trong phát triển.

- Hỗ trợ thư viện mạnh mẽ: Dễ dàng tích hợp các thư viện như mongoose cho MongoDB hoặc TensorFlow.js cho KNN, phù hợp với dự án thực tập.
- Hạn chế
 - Hiệu suất với logic phức tạp: Chạy module child_process có thể chậm hơn so với các ngôn ngữ như Python, đặc biệt với mô hình phức tạp hoặc dữ liệu lớn.
 - Quản lý tài nguyên: Với 50,000 công thức, Node.js cần tối ưu hóa (cache, truy vấn hiệu quả) để tránh quá tải server.

3.2.3 Giới thiệu Express

- Express là một framework web nhẹ và linh hoạt cho Node.js, được sử dụng để xây dựng các ứng dụng web và API. Nó cung cấp các công cụ đơn giản để xử lý yêu cầu HTTP, định tuyến (routing), và tích hợp với cơ sở dữ liệu như MongoDB, giúp phát triển backend nhanh chóng và hiệu quả.
- Express đóng vai trò là nền tảng backend, quản lý các API để tương tác giữa frontend và MongoDB, nơi lưu trữ 50,000 công thức từ tập dữ liệu FoodRecSys-V1. Các ứng dụng cụ thể:
 - API Routes: Xử lý các endpoint như /api/recipes (lấy danh sách công thức), /api/recipes/:id (chi tiết công thức), /api/search (tìm kiếm theo tên/nguyên liệu), và /api/recommend (gợi ý món ăn bằng KNN dựa trên tiêu chí dinh dưỡng như calo, protein, chất béo).
 - Kết nối MongoDB: Tích hợp với thư viện mongoose để truy vấn dữ liệu công thức, hỗ trợ phân trang (lazy loading) và tìm kiếm nhanh.
 - Xử lý KNN: Chạy mô hình KNN (triển khai bằng TensorFlow.js hoặc script Python gọi qua API) trong endpoint /api/recommend để đề xuất 3 món ăn phù hợp nhất.
 - Hiệu suất: Xử lý nhiều yêu cầu đồng thời, phù hợp với các tính năng như tìm kiếm công thức hoặc gợi ý món ăn với lưu lượng truy cập lớn.
- Lợi ích của Express

- Đơn giản và dễ học: Cú pháp ngắn gọn, dễ triển khai các API như `/api/recipes`, phù hợp cho sinh viên thực tập.
- Linh hoạt: Dễ tích hợp với MongoDB (mongoose), TensorFlow.js (cho KNN), và các thư viện như cors hoặc body-parser.
- Hiệu suất cao: Tận dụng mô hình bất đồng bộ của Node.js, xử lý nhanh các yêu cầu tìm kiếm và gợi ý.
- Cộng đồng lớn: Nhiều tài liệu và plugin hỗ trợ, giúp phát triển nhanh các tính năng của SmartBytes.

3.3 Các công nghệ sử dụng cho xử lý dữ liệu và training model gợi ý món ăn

3.3.1 Mô hình K-Nearest Neighbors (KNN)

Khái niệm cơ bản về KNN

- K-Nearest Neighbors (KNN) là một thuật toán học máy thuộc nhóm học có giám sát (supervised learning), được sử dụng phổ biến trong các bài toán phân loại (classification) và hồi quy (regression). Đây là một phương pháp phi tham số (non-parametric), nghĩa là nó không giả định trước về phân phối của dữ liệu mà dựa trực tiếp vào chính dữ liệu huấn luyện để đưa ra dự đoán.
- Nguyên lý cơ bản của KNN là: "Vật hợp theo loài, người hợp theo nhóm". Khi cần dự đoán một điểm dữ liệu mới, KNN tìm kiếm K điểm dữ liệu gần nhất (neighbors) trong tập huấn luyện dựa trên một thước đo khoảng cách (thường là khoảng cách Euclidean), sau đó đưa ra kết quả dựa trên thông tin của các điểm lân cận này.
- Trong bài toán phân loại, mục tiêu là gán một nhãn lớp (label) cho một điểm dữ liệu dựa trên các đặc trưng (features) của nó. Có nhiều thuật toán phân loại, và một trong những cách phổ biến để đưa ra quyết định cuối cùng (khi có nhiều dự đoán hoặc ý kiến) là dựa trên bỏ phiếu từ các dự đoán riêng lẻ.
- Trong hồi quy, KNN cũng dùng khái niệm "k láng giềng gần nhất" tương tự như trong phân loại, nhưng thay vì chọn nhãn dựa trên "đa số phiếu" (majority vote), nó lấy trung bình (average) của giá trị các láng giềng để dự đoán.
- Sự khác biệt chính:
 - Phân loại (classification): Dự đoán nhãn rời rạc (discrete), như "mèo", "chó", "chim".

- Hồi quy (regression): Dự đoán giá trị liên tục (continuous), như 25.5, 100.3, v.v.
- KNN thuộc nhóm các mô hình "học lười" (lazy learning), tức là nó không thực sự học hay xây dựng một mô hình trong giai đoạn huấn luyện (training).
- Thay vào đó, nó chỉ lưu trữ toàn bộ tập dữ liệu huấn luyện và chờ đến khi cần dự đoán (classification hoặc regression) mới bắt đầu tính toán.
- KNN cần giữ toàn bộ tập dữ liệu huấn luyện trong bộ nhớ (memory) để so sánh với dữ liệu mới khi dự đoán.
- Do đó, nó được gọi là:
 - Instance-based: Dự đoán dựa trên các "thực thể" (instances) cụ thể trong dữ liệu huấn luyện.
 - Memory-based: Phụ thuộc nhiều vào bộ nhớ để lưu trữ dữ liệu.

Cách hoạt động của KNN

- Chuẩn bị dữ liệu
 - Tập dữ liệu huấn luyện bao gồm các đặc trưng (features) và nhãn (labels) tương ứng.
 - Các đặc trưng số (như chiều cao, cân nặng) thường được chuẩn hóa (ví dụ: bằng StandardScaler) để đảm bảo chúng đóng góp công bằng vào tính toán khoảng cách.
- Chọn giá trị K
 - K là số lượng điểm lân cận được xem xét, do người dùng chọn trước (siêu tham số - hyperparameter).
 - Giá trị K ảnh hưởng trực tiếp đến độ chính xác của mô hình: K nhỏ có thể nhạy cảm với nhiễu, K lớn có thể làm mờ ranh giới giữa các lớp.
- Tính toán khoảng cách
 - Với mỗi điểm dữ liệu mới cần dự đoán, KNN tính khoảng cách từ điểm đó đến tất cả các điểm trong tập huấn luyện.
 - Euclidean Distance (Khoảng cách Euclidean, $p=2$): Đây là thước đo khoảng cách phổ biến nhất, tính khoảng cách đường thẳng ngắn nhất giữa hai điểm trong không gian (giống như đo bằng thước kẻ).

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

- Manhattan Distance (Khoảng cách Manhattan, $p=1$): Còn gọi là "khoảng cách taxi" hoặc "khoảng cách khối thành phố", nó tính tổng giá trị tuyệt đối của chênh lệch giữa các tọa độ. Tên gọi xuất phát từ cách di chuyển trên lưới đường phố (chỉ đi ngang hoặc dọc, không đi chéo).

$$d(p, q) = \sum_{i=1}^n |p_i - q_i|$$

- Minkowski Distance (Khoảng cách Minkowski): Đây là dạng tổng quát hóa của cả Euclidean và Manhattan. Tham số p trong công thức cho phép điều chỉnh loại khoảng cách:

- $p=2$: Trở thành Euclidean.
- $p=1$: Trở thành Manhattan.

$$d(p, q) = \left(\sum_{i=1}^n |p_i - q_i|^p \right)^{\frac{1}{p}}$$

- Xác định K điểm gần nhất:
 - Sắp xếp các khoảng cách từ nhỏ đến lớn và chọn K điểm có khoảng cách nhỏ nhất.
- Dự đoán:
 - Trong bài toán phân loại: Dựa trên "đa số phiếu" (majority vote) của K điểm lân cận, nhãn nào chiếm ưu thế sẽ được chọn.
 - Trong bài toán hồi quy: Lấy trung bình (hoặc trung bình có trọng số) của giá trị đầu ra của K điểm lân cận.

3.3.2 Công nghệ xử lý dữ liệu

Pandas là một thư viện Python mã nguồn mở, được thiết kế để xử lý và phân tích dữ liệu. Nó cung cấp các cấu trúc dữ liệu mạnh mẽ như Series (dữ liệu một chiều) và DataFrame (dữ liệu hai chiều, tương tự bảng Excel), cùng với các công cụ để thao tác, làm sạch, và phân tích dữ liệu hiệu quả. Pandas được sử dụng rộng rãi trong khoa học dữ liệu, học máy, và các dự án cần xử lý dữ liệu nhỏ tới vừa.

Vai trò chính của pandas trong dự án:

- Đọc và ghi dữ liệu:
 - Đọc dữ liệu từ các tệp CSV, Excel, hoặc JSON chứa 50,000 công thức nấu ăn.
 - Xuất dữ liệu đã xử lý thành JSON để nhập vào MongoDB.
- Làm sạch dữ liệu:
 - Xử lý giá trị thiếu (missing values), ví dụ: điền giá trị trung bình cho calo hoặc loại bỏ công thức thiếu thông tin dinh dưỡng.
- Chuyển đổi và lọc dữ liệu:
 - Lọc các công thức đáp ứng tiêu chí nhất định, như công thức dưới 500 kcal hoặc có ít nhất 5g chất xơ.
 - Tạo cột mới, như tính tổng chất dinh dưỡng hoặc phân loại món ăn (thấp calo, giàu protein).
- Chuẩn bị dữ liệu cho KNN:
 - Tạo tập dữ liệu số hóa (numerical dataset) từ các đặc điểm dinh dưỡng (calo, chất béo, chất xơ, protein, carbohydrate) để sử dụng trong thuật toán KNN.
 - Chuẩn hóa (normalize) dữ liệu để đảm bảo các giá trị nằm trong cùng thang đo, cải thiện độ chính xác của KNN.

CHƯƠNG IV: XÂY DỰNG ỨNG DỤNG SMARTBYTES

4.1 Phân tích dữ liệu mong muốn

4.1.1 Mục tiêu và đặc điểm của dữ liệu

Để xây dựng một hệ thống gợi ý món ăn sử dụng mô hình KNN, dữ liệu cần được chuẩn bị sao cho phản ánh chính xác mối quan hệ giữa các thành phần nguyên liệu, lượng dinh dưỡng (nutrition) và món ăn tương ứng. Mô hình KNN hoạt động dựa trên nguyên lý tìm kiếm các "hàng xóm gần nhất" trong không gian đặc trưng, do đó dữ liệu mong muốn phải có cấu trúc số hóa, định lượng và đồng nhất nhằm hỗ trợ việc tính toán khoảng cách giữa các điểm dữ liệu (ví dụ: Euclidean distance). Trong trường hợp này, người dùng sẽ nhập các thành phần hoặc lượng thành phần mong muốn (ví dụ: protein, chất béo, carbohydrate), và hệ thống sẽ đề xuất các món ăn phù hợp nhất dựa trên dữ liệu đã huấn luyện.

Dữ liệu lý tưởng để huấn luyện mô hình KNN trong hệ thống gợi ý món ăn cần đáp ứng các yêu cầu sau:

- **Cấu Trúc Dữ Liệu**
 - Hình thức bảng (Tabular Data): Dữ liệu nên được tổ chức dưới dạng bảng với các cột rõ ràng, mỗi cột đại diện cho một đặc trưng (feature) của món ăn.
- **Định Lượng và Chuẩn Hóa**
 - Định lượng: Các thành phần dinh dưỡng cần được biểu diễn dưới dạng số (ví dụ: Protein: 15g, Fat: 8g) thay vì định tính (cao, thấp). Điều này cho phép KNN tính toán khoảng cách chính xác giữa các món ăn và yêu cầu của người dùng.
 - Chuẩn hóa (Normalization): Vì KNN nhạy cảm với độ lớn của các giá trị, dữ liệu cần được chuẩn hóa (ví dụ: Min-Max Scaling hoặc Z-score) để đảm bảo các thành phần như Protein và Calories có cùng thang đo, tránh trường hợp một đặc trưng có giá trị lớn lấn át các đặc trưng khác.
- **Tính Đầy Đủ và Sạch**

- Không có giá trị thiếu: Các cột liên quan đến thành phần dinh dưỡng phải đầy đủ dữ liệu. Nếu thiếu, cần điền giá trị bằng trung bình, trung vị hoặc loại bỏ bản ghi đó.

4.1.2 Nguồn dữ liệu

Dữ liệu là một phần của tập dữ liệu FoodRecSys-V1, được lấy từ nền tảng Kaggle. FoodRecSys-V1 là một tập dữ liệu công khai mô tả về các món ăn, bao gồm thông tin chi tiết về các công thức nấu ăn, nguyên liệu, hướng dẫn nấu, và giá trị dinh dưỡng của các món ăn.

Nguồn dữ liệu: [foodRecSys-V1](#).

Mô tả dữ liệu

- `recipe_id`: Mã định danh duy nhất cho mỗi công thức (ví dụ: 222388, 240488, 218939).
- `name`: Tên món ăn (ví dụ: "Homemade Bacon", "Pork Loin, Apples, and Sauerkraut", "Foolproof Rosemary Chicken Wings").
- `aver_rate`: Điểm đánh giá trung bình của món ăn (ví dụ: 5.0, 4.764706, 4.571429).
- `image_url`: Đường dẫn đến hình ảnh của món ăn, thường từ trang Allrecipes.
- `review_nums`: Số lượng đánh giá cho món ăn (ví dụ: 3, 29, 12).
- `ingredients`: Danh sách nguyên liệu, được liệt kê dưới dạng chuỗi văn bản (ví dụ: "belly pork, paprika kosher salt ground b ..." cho món Homemade Bacon).
- `cooking_directions`: Hướng dẫn nấu ăn, được lưu dưới dạng chuỗi JSON (ví dụ: {'directions': 'uPrep\\n5 m\\nCook\\n2 h 45 m\\nRe...'}).
- `nutrition`: Thông tin dinh dưỡng, cũng được lưu dưới dạng chuỗi JSON (ví dụ: {'niacin': 0, 'hasCompleteData': False, 'u_name...'}).
- `reviews`: Thông tin đánh giá, bao gồm số lượng người theo dõi và điểm đánh giá (ví dụ: {'rating': 5, 'followersCount': 11, ...}).

Tại sao dữ liệu phù hợp để huấn luyện mô hình KNN?

- Các thành phần trong cột Nutrition

- Thành phần dinh dưỡng có thể thấy được đầy đủ trong phần cột Nutrition, phù hợp với yêu cầu đề ra để xây dựng hệ thống.
- Tính Đa Dạng của Nguyên Liệu và Món Ăn
 - Cột ingredients chứa danh sách nguyên liệu đa dạng phản ánh nhiều loại món ăn từ các nền ẩm thực khác nhau. Điều này giúp hệ thống gợi ý có thể đáp ứng nhiều sở thích và yêu cầu của người dùng.
- Số Lượng Bản Ghi Đủ Lớn
 - Dữ liệu FoodRecSys-V1 trên Kaggle thường chứa hàng nghìn bản ghi. Số lượng lớn này đảm bảo rằng KNN có đủ "hàng xóm" để tìm kiếm và đưa ra gợi ý chính xác, tránh tình trạng thiếu dữ liệu dẫn đến kết quả không đáng tin cậy.

Hạn Chế và Cách Khắc Phục

- Mặc dù dữ liệu phù hợp, vẫn có một số hạn chế cần lưu ý:
 - Dữ liệu dinh dưỡng chưa đầy đủ: Một số bản ghi có `hasCompleteData: False` trong cột nutrition, nghĩa là thông tin dinh dưỡng không đầy đủ. Bạn cần xử lý bằng cách loại bỏ các bản ghi này hoặc điền giá trị thiếu (ví dụ: dùng trung bình của các món ăn tương tự).
 - Chuẩn hóa: Các giá trị dinh dưỡng cần được chuẩn hóa (ví dụ: Min-Max Scaling) để đảm bảo các thành phần như Protein và Calories có cùng thang đo, tránh làm sai lệch kết quả của KNN.

4.1.3 Tiền xử lí dữ liệu

4.1.3.1 Dữ liệu cho mô hình

- Phân tích cột nutrition
 - Cột nutrition chứa thông tin dinh dưỡng của một món ăn, được lưu dưới dạng một chuỗi JSON với cấu trúc phức tạp.
 - Cấu Trúc Dữ Liệu
 - `hasCompleteData`: Boolean (True/False), cho biết dữ liệu của thành phần này có đầy đủ hay không.
 - `name`: Tên của thành phần (ví dụ: "Niacin Equivalents", "Sugars").

- amount: Giá trị định lượng của thành phần (ví dụ: 15.6016 cho niacin).
- percentDailyValue: Phần trăm giá trị hàng ngày (dựa trên chế độ ăn 2000 kcal/ngày).
- displayValue: Giá trị hiển thị (có thể làm tròn hoặc định dạng khác so với amount).
- unit: Đơn vị của thành phần (ví dụ: mg, g, kcal, IU, mcg).
- Hướng tiền xử lí
 - Tập trung vào percentDailyValue, vì nó biểu thị tỷ lệ phần trăm của giá trị dinh dưỡng hàng ngày (Daily Value - DV) mà món ăn cung cấp, dựa trên chế độ ăn 2000 kcal/ngày. Đây là một chỉ số quan trọng để phân tích lượng dinh dưỡng người dùng muốn nạp và phù hợp để huấn luyện mô hình.
 - Bước 1: Tách JSON và Lấy percentDailyValue
 - Mục tiêu: Chuyển đổi dữ liệu JSON thành các cột riêng biệt, chỉ lấy giá trị percentDailyValue của các thành phần dinh dưỡng cần thiết.
 - Thành phần cần giữ: Tập trung vào các thành phần chính mà người dùng thường quan tâm: Calories, protein, fat, carbohydrates, sugars, fiber, sodium,...
 - Bước 2: Xử Lý Dữ Liệu Thiếu hoặc Không Hợp Lệ
 - Vấn đề: Một số bản ghi có thể thiếu percentDailyValue hoặc có giá trị không hợp lệ (NaN sau khi chuyển đổi).
 - Hướng xử lý:
 - Điền giá trị thiếu: Nếu một bản ghi thiếu percentDailyValue cho một thành phần chính (calories, protein, fat, carbohydrates), có thể điền bằng giá trị trung bình của cột đó trong toàn bộ tập dữ liệu.
 - Kiểm tra hasCompleteData: Dù hasCompleteData: False, percentDailyValue vẫn có giá trị (ví dụ: sodium có

percentDailyValue: '104'). Do đó, ta có thể sử dụng các giá trị này mà không cần lo lắng về hasCompleteData.

4.1.3.2 Dữ liệu cho hệ thống

Hạn chế của dữ liệu

- Cột **nutritions**, **cooking_directions**, **reviews** là chuỗi JSON, cần tách thành các trường riêng.
- **ingredients** là chuỗi, cần chuyển thành danh sách để dễ truy vấn.
- Đơn vị không đồng nhất (mg, g, kcal, IU), cần chuẩn hóa.

Quy trình tiền xử lý

- **Xử lý cột **nutritions****: Tách chuỗi JSON thành các trường riêng (calo, protein, chất béo, carbohydrate, v.v.), chuẩn hóa đơn vị (kcal cho calo, gram cho protein/chất béo), và tạo trường **normalized_nutrition** (chuẩn hóa 0-1) để hỗ trợ KNN.
- **Xử lý cột **ingredients****: Chuyển chuỗi nguyên liệu thành danh sách, chuẩn hóa thành chữ thường để dễ tìm kiếm.
- **Xử lý cột **cooking_directions****: Tách chuỗi JSON thành danh sách các bước nấu ăn, loại bỏ khoảng trắng thừa.
- **Tạo cấu trúc JSON**: Tổ chức dữ liệu thành định dạng JSON với các trường như **recipe_id**, **recipe_name**, **ingredients**, **nutritions**, **normalized_nutrition**, **image_url**, **average_rating**.
- **Nhập vào MongoDB**: Nhập JSON vào MongoDB qua **mongoimport** hoặc script Node.js, lưu vào collection **recipes**.
- **Tạo chỉ mục**: Tạo chỉ mục trên **recipe_id**, **recipe_name**, **ingredients**, và **nutritions** để tăng tốc truy vấn.

Mô tả chi tiết các trường dữ liệu sau khi được xử lý và đưa vào mongo

- **id (ObjectId)**: Trường ID mặc định của MongoDB, được tự động tạo ra.
- **recipe_id (Number)**: Mã định danh gốc của công thức từ tập dữ liệu Kaggle.
- **recipe_name (String)**: Tên đầy đủ của món ăn.
- **average_rating (Number)**: Điểm đánh giá trung bình của món ăn, đã được chuyển đổi thành kiểu số.
- **image_url (String)**: Đường dẫn URL đến hình ảnh của món ăn.

- **review_count (Number):** Tổng số lượt đánh giá món ăn, đã được chuyển đổi thành kiểu số.
- **ingredients (Array of Strings):** Danh sách các nguyên liệu cần thiết cho món ăn. Mỗi nguyên liệu đã được tách riêng thành một chuỗi trong mảng.
- **cooking_directions (Array of Strings):** Các bước hướng dẫn chế biến món ăn. Mỗi bước đã được tách thành một chuỗi riêng biệt trong mảng, giúp dễ dàng hiển thị tuần tự.
- **preparation_time_minutes (Number):** Thời gian chuẩn bị nguyên liệu, được trích xuất từ **cooking_directions** và chuyển đổi thống nhất về đơn vị phút.
- **cooking_time_minutes (Number):** Thời gian nấu, được trích xuất từ **cooking_directions** và chuyển đổi thống nhất về đơn vị phút.
- **total_time_minutes (Number):** Tổng thời gian hoàn thành món ăn (Ready In), được trích xuất từ **cooking_directions** và chuyển đổi thống nhất về đơn vị phút.
- **nutritions (Object):** Một đối tượng chứa thông tin dinh dưỡng chi tiết của món ăn. Mỗi thuộc tính trong đối tượng này đại diện cho một loại dinh dưỡng (ví dụ: calories, protein, fat, fiber).
 - Mỗi loại dinh dưỡng (ví dụ: niacin, sugars, calories, fat, protein, fiber, v.v.) là một đối tượng con, chứa:
 - **amount (Number):** Giá trị định lượng của thành phần dinh dưỡng.
 - **unit (String):** Đơn vị của giá trị (mg, g, kcal, IU, mcg).
 - **percentDailyValue (Number/Null):** Phần trăm giá trị hàng ngày (DV). Các giá trị < 1 được làm tròn xuống 0, và "-" được chuyển thành null.

4.2 Phân tích mô hình KNN trong hệ thống gợi ý món ăn

4.2.1 Biểu Diễn Văn Bản(TfidfVectorizer)

- Trong bài toán tìm kiếm món ăn tương tự dựa trên nguyên liệu, việc xử lý dữ liệu đầu vào đóng vai trò quan trọng trong việc đảm bảo hiệu quả của mô hình học máy. Danh sách nguyên liệu (**ingredients_list**) là dữ liệu văn bản tự do (free-text), không thể sử dụng trực tiếp trong các thuật toán học máy. Vì

vậy, cần có một phương pháp để biến đổi văn bản thành các đặc trưng số mà mô hình có thể hiểu được.

- Để thực hiện điều này, ta sử dụng TF-IDF Vectorizer, một kỹ thuật mạnh mẽ giúp chuyển đổi văn bản thành các vector số thông qua việc đánh trọng số các từ quan trọng. Điều này giúp giảm nhiễu từ những nguyên liệu phổ biến và nhấn mạnh những nguyên liệu mang tính đặc trưng cao.
- Mục đích
 - Biến các chuỗi văn bản thành các vector số. So sánh mức độ tương đồng giữa các từ. Làm sạch dữ liệu và chuẩn hóa để chuẩn bị cho các thuật toán học máy.
- Cách hoạt động
 - TF (Term Frequency): Đo lường số lần một từ xuất hiện trong tài liệu chia cho tổng số từ trong tài liệu.
 - IDF (Inverse Document Frequency): Đo lường mức độ quan trọng của từ. Từ càng xuất hiện nhiều trong các tài liệu khác nhau, thì trọng số của nó càng thấp.
 - TF-IDF: Kết hợp cả hai yếu tố trên để cho ra một trọng số đại diện cho tầm quan trọng của từ trong tài liệu cụ thể.
- Lý do sử dụng TF-IDF Vectorizer
 - Chuyển đổi văn bản thành vector số:
 - Máy tính không thể trực tiếp hiểu được văn bản, nhưng lại làm việc rất tốt với các con số.
 - TF-IDF Vectorizer chuyển đổi các nguyên liệu món ăn (dạng văn bản) thành vector đặc trưng số.
 - Giảm thiểu ảnh hưởng của từ thông dụng:
 - Trong các danh sách nguyên liệu, một số từ như "salt", "pepper", "water" xuất hiện ở hầu hết các công thức.
 - Nếu bạn sử dụng phương pháp Bag of Words (BoW) hoặc Count Vectorizer, những từ thông dụng này có thể khiến kết quả bị nhiễu.

- TF-IDF giải quyết vấn đề này bằng cách tính toán trọng số thấp cho các từ phổ biến, vì chúng ít có ý nghĩa trong việc phân biệt giữa các công thức.
- Nhấn mạnh từ đặc trưng:
 - Những từ ít phổ biến nhưng có ý nghĩa phân loại cao như "pork belly", "smoked paprika" sẽ được TF-IDF gán trọng số cao.
 - Điều này giúp mô hình dễ dàng nhận diện đặc điểm nổi bật của mỗi công thức.
- Áp dụng vào nguyên liệu(ingredient_list)
 - Bước 1: xây dựng từ vựng
 - Quét toàn bộ văn bản trong cột ingredient_list.
 - Tạo từ vựng duy nhất: TfidfVectorizer thu thập tất cả các từ (nguyên liệu) duy nhất xuất hiện trong toàn bộ tập dữ liệu và gán mỗi từ một chỉ số (index). Ví dụ:
 - Từ vựng: {"flour": 0, "sugar": 1, "eggs": 2, "chicken": 3, "rice": 4, "soy sauce": 5}.
 - Mỗi từ (nguyên liệu) tương ứng với một cột trong ma trận kết quả.
 - Bước 2: Chuyển đổi thành ma trận
 - Sau khi có từ vựng, TfidfVectorizer sẽ biểu diễn mỗi công thức dưới dạng một vector số:
 - Số cột của ma trận bằng số lượng từ trong từ vựng (số nguyên liệu duy nhất).
 - Số hàng của ma trận bằng số lượng công thức (số mẫu trong recipe_df).
 - Với mỗi công thức:
 - Nếu một nguyên liệu trong từ vựng xuất hiện trong công thức, cột tương ứng sẽ có giá trị là điểm TF-IDF của nguyên liệu đó.
 - Nếu nguyên liệu không xuất hiện, giá trị ở cột đó sẽ là 0.

- Bước 3: Tính điểm TF-IDF

- TF (Term Frequency): Tần suất xuất hiện của một nguyên liệu trong công thức (thường được chuẩn hóa theo độ dài của công thức).
- IDF (Inverse Document Frequency): Đo lường mức độ "hiếm" của nguyên liệu trong toàn bộ tập dữ liệu. Nguyên liệu xuất hiện trong nhiều công thức (như "salt") sẽ có IDF thấp, còn nguyên liệu hiếm (như "saffron") sẽ có IDF cao.
- Giá trị $TF-IDF = TF \times IDF$.
- Từ đó tạo ra ma trận $X_{ingredients}$, trong đó:
- Mỗi hàng là một công thức.
- Mỗi cột là một nguyên liệu trong từ vựng (tất cả nguyên liệu duy nhất trong tập dữ liệu).
- Giá trị là điểm TF-IDF của nguyên liệu đó trong công thức.

4.2.2 Chuẩn hóa dữ liệu (StandardScaler)

- Bên cạnh đó, các thành phần khác của món ăn được biểu diễn dưới dạng số (ví dụ: lượng calo, chất đạm, chất béo...) có thể có quy mô (scale) khác nhau. Để chuẩn hóa dữ liệu và đảm bảo rằng mọi đặc trưng đều được xử lý công bằng bởi mô hình, ta áp dụng Standard Scaler.
- Mục đích
 - dùng để chuẩn hóa dữ liệu số (numerical data) bằng cách đưa dữ liệu về cùng một thang đo chuẩn (standard scale).
- Cách hoạt động
 - StandardScaler chuẩn hóa dữ liệu theo công thức:

$$X_{scaled} = \frac{X - \mu}{\sigma}$$

- Trong đó:
 - X là giá trị ban đầu của dữ liệu.
 - μ là giá trị trung bình (mean) của đặc trưng đó trong tập dữ liệu.

- σ là độ lệch chuẩn (standard deviation) của đặc trưng đó trong tập dữ liệu.
- Sau khi chuẩn hóa:
 - Dữ liệu sẽ có giá trị trung bình = 0 và độ lệch chuẩn = 1.
 - Tất cả các đặc trưng sẽ nằm trên cùng một thang đo, giúp cải thiện hiệu quả của các mô hình học máy.
- Tại sao cần StandardScaler?
 - Đảm bảo công bằng giữa các đặc trưng:
 - Các cột dữ liệu số của bạn như: calories, fat, carbohydrates, protein, cholesterol, sodium, fiber có quy mô khác nhau. Ví dụ:
 - calories có thể có giá trị hàng trăm hoặc hàng ngàn.
 - fiber có thể chỉ có giá trị từ 0 đến 10.
 - Nếu không chuẩn hóa, các đặc trưng có giá trị lớn sẽ chi phối hoàn toàn mô hình, làm cho các đặc trưng nhỏ hơn trở nên không có ý nghĩa.
 - Cải thiện hiệu quả của các thuật toán học máy:
 - Các thuật toán như K-Nearest Neighbors (KNN) sử dụng khoảng cách (distance-based algorithms). Nếu không chuẩn hóa, các đặc trưng lớn sẽ gây sai lệch trong tính toán khoảng cách.
 - Một số thuật toán khác như PCA (Principal Component Analysis), SVM (Support Vector Machine), hay Gradient Descent cũng yêu cầu dữ liệu được chuẩn hóa để đạt hiệu quả cao nhất.
- Áp dụng vào các thành phần dinh dưỡng
 - Các thành phần dinh dưỡng có thang đo rất khác nhau:
 - calories và sodium có giá trị lớn (hàng trăm đến hàng nghìn).
 - fat, carbohydrates, protein, cholesterol có giá trị trung bình (hàng chục).
 - fiber có giá trị nhỏ (dưới 10).

- Nếu không chuẩn hóa, các cột như calories hoặc sodium sẽ chi phối khoảng cách Euclidean trong mô hình KNN, làm giảm tầm quan trọng của các cột như fiber.
- Yêu cầu của KNN: KNN dựa trên khoảng cách Euclidean:

$$\text{distance} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

- Nếu các đặc trưng không cùng thang đo, khoảng cách sẽ bị lệch về phía các đặc trưng có giá trị lớn. Chuẩn hóa đảm bảo rằng tất cả các đặc trưng đóng góp công bằng vào khoảng cách.

4.2.3 Huấn luyện mô hình

- Tiền xử lý nguyên liệu: Sử dụng TfidfVectorizer để chuyển recipe_df['ingredients_list'] thành ma trận X_ingredients (vector TF-IDF), với mỗi cột là một nguyên liệu duy nhất.
- Chuẩn hóa thành phần dinh dưỡng: Dùng StandardScaler để chuẩn hóa các cột số (calories, fat, carbohydrates, protein, cholesterol, sodium, fiber) thành X_numerical, đảm bảo trung bình = 0, độ lệch chuẩn = 1.
- Kết hợp đặc trưng: Ghép X_numerical và X_ingredients thành ma trận X_combined bằng np.hstack.
- Huấn luyện KNN: Khởi tạo NearestNeighbors với n_neighbors=3, metric='euclidean', và lưu trữ X_combined để tìm hàng xóm gần nhất.

4.2.4 Đưa dữ liệu đầu vào và cho ra kết quả

Đầu vào: (ví dụ: input_features = [15, 36, 1, 42, 21, 81, 2, 'pork belly, smoked paprika, kosher salt'])
(7 giá trị dinh dưỡng + danh sách nguyên liệu).

- Tiền xử lý:
 - Chuẩn hóa 7 giá trị dinh dưỡng bằng scaler.transform thành vector số.
 - Chuyển nguyên liệu thành vector TF-IDF bằng vectorizer.transform.
 - Ghép hai vector thành input_combined.

- Dự đoán: Dùng `knn.kneighbors` để tính khoảng cách Euclidean từ `input_combined` đến tất cả công thức trong `X_combined`, chọn 3 công thức có khoảng cách nhỏ nhất.
- Kết quả: Trả về DataFrame chứa `recipe_name`, `ingredients_list`, `image_url` của 3 công thức gần nhất (ví dụ: "Spicy Pork Stew", "Grilled Pork", "Beef Stir Fry").

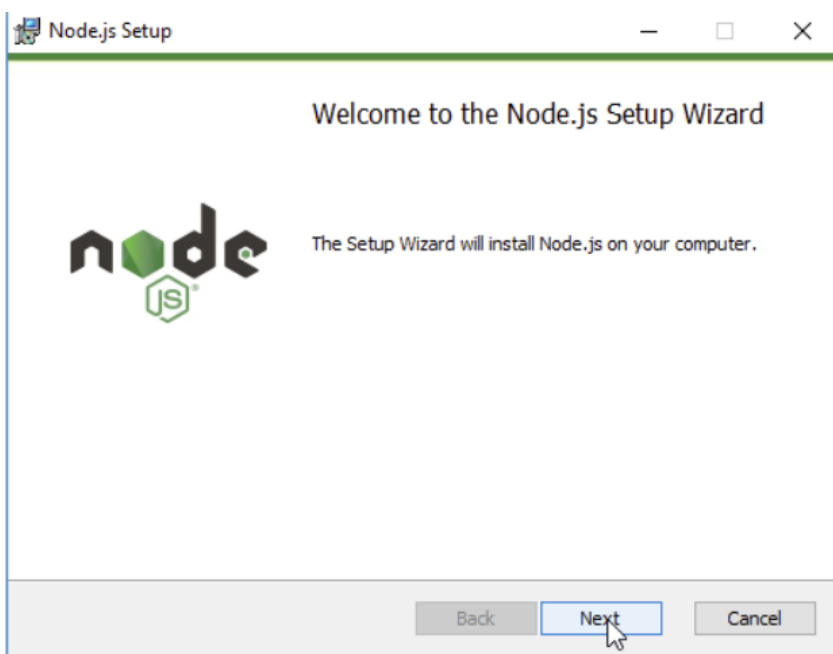
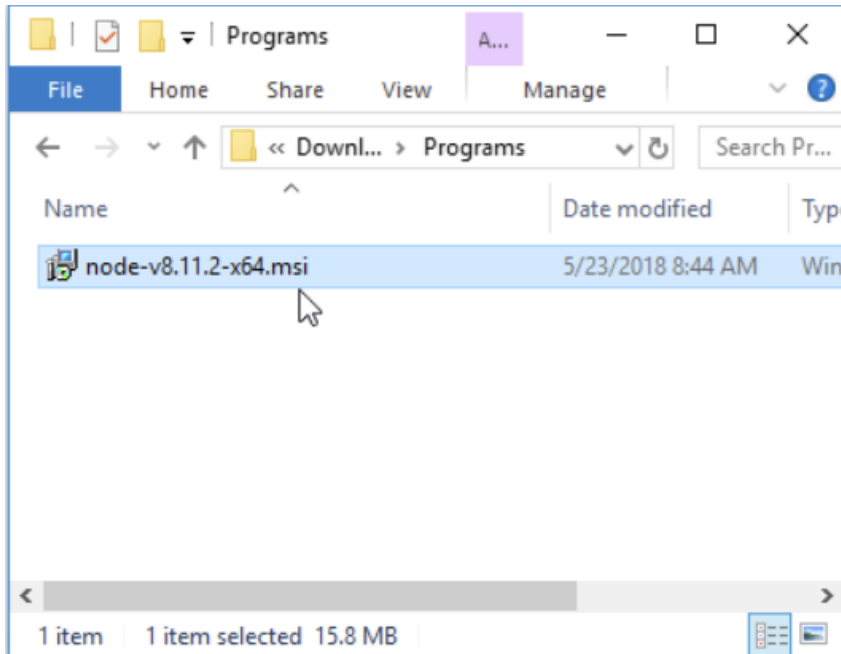
CHƯƠNG V: CÀI ĐẶT VÀ THỰC NGHIỆM

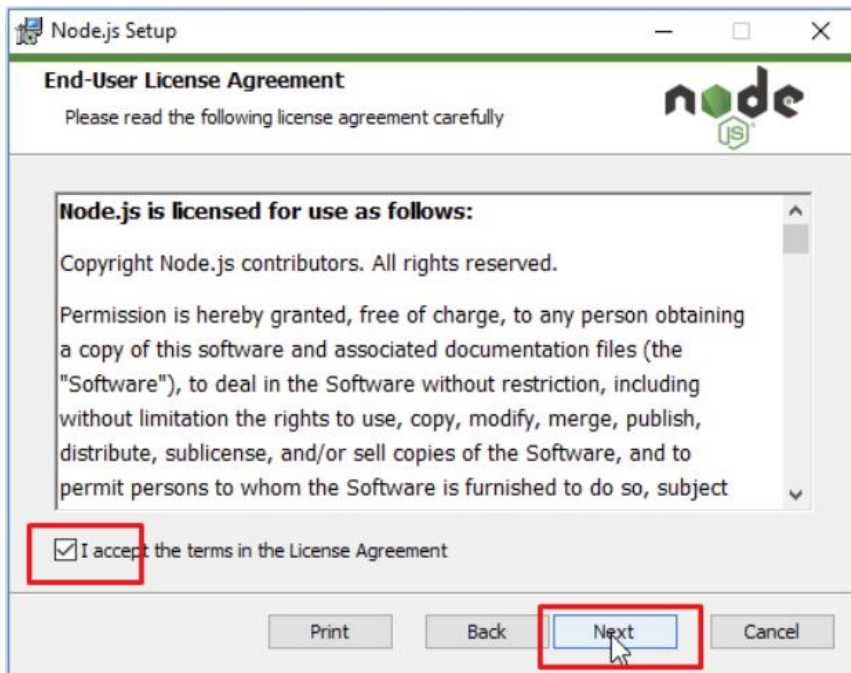
5.1 Cài đặt

5.1.1 Cài đặt NodeJS

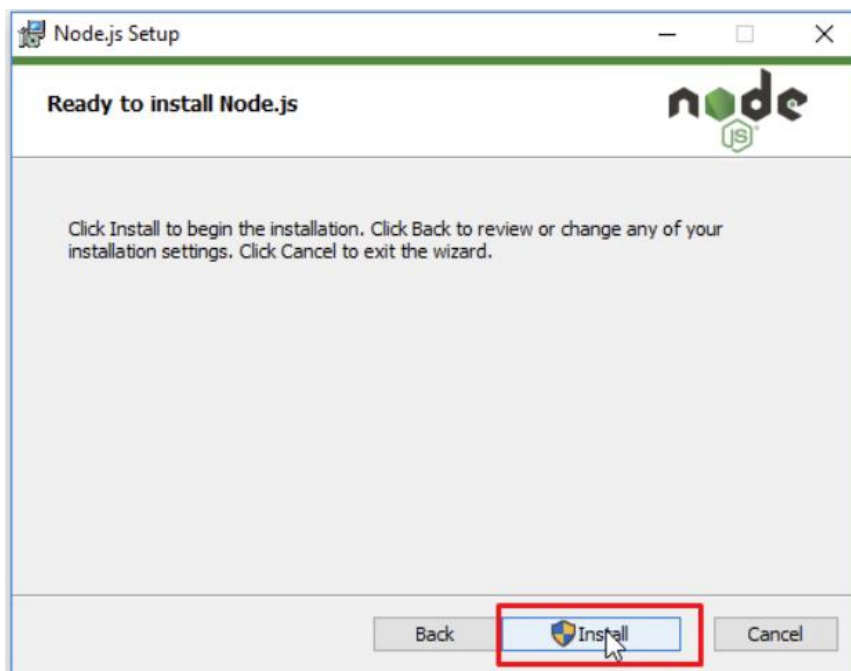
Truy cập trang <https://nodejs.org/en/download/> để download nodejs.

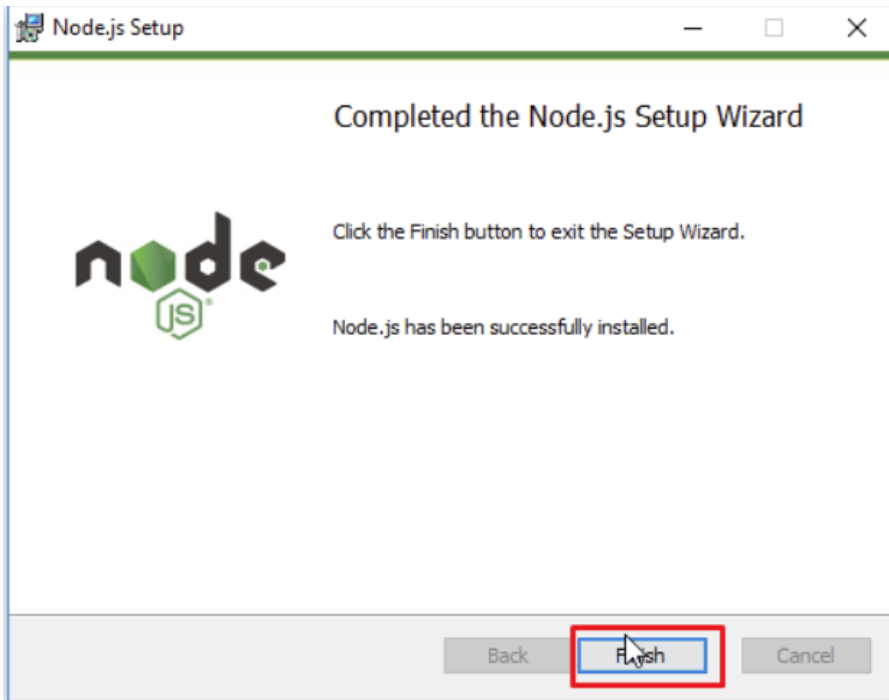
Click đúp vào file .msi vừa tải về để tiến hành cài đặt.





Chọn thư mục để cài đặt nodeJS và ấn next hai lần.

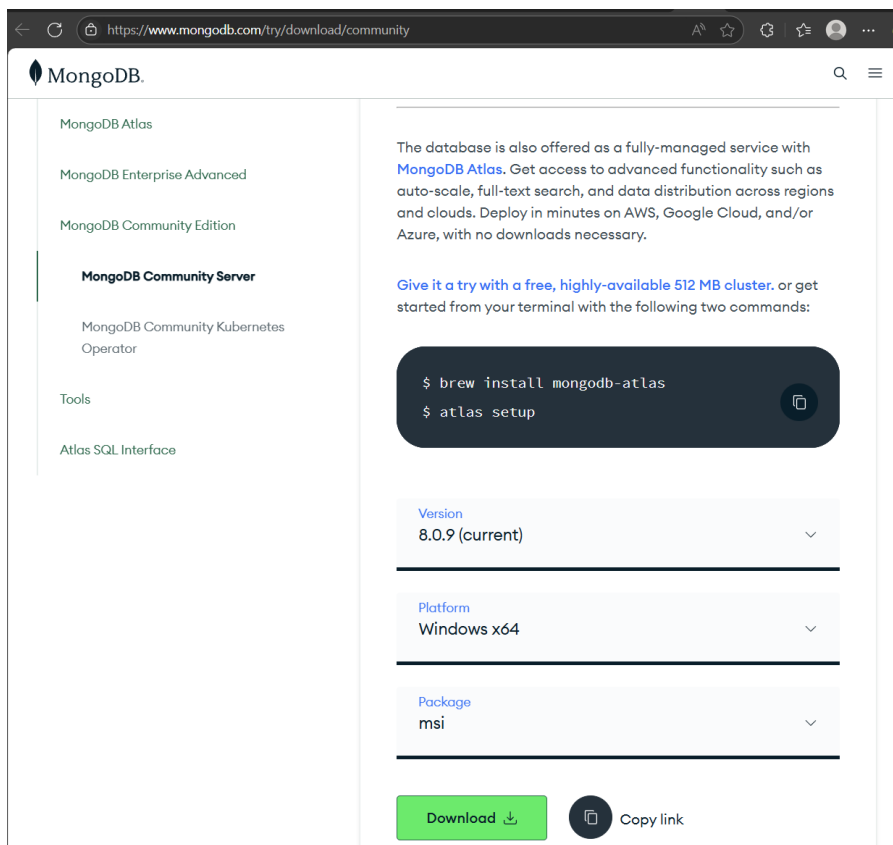




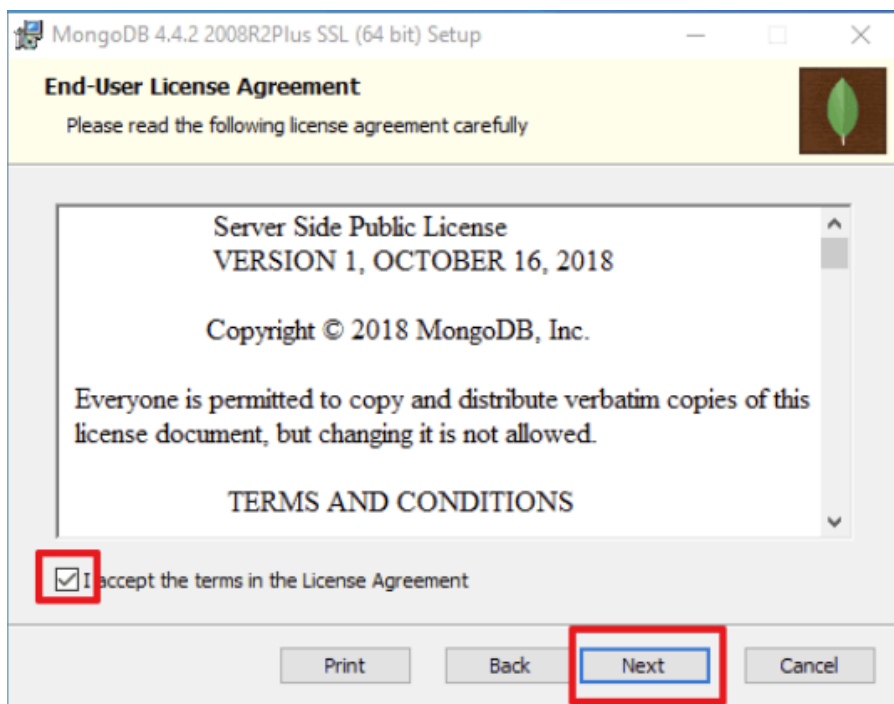
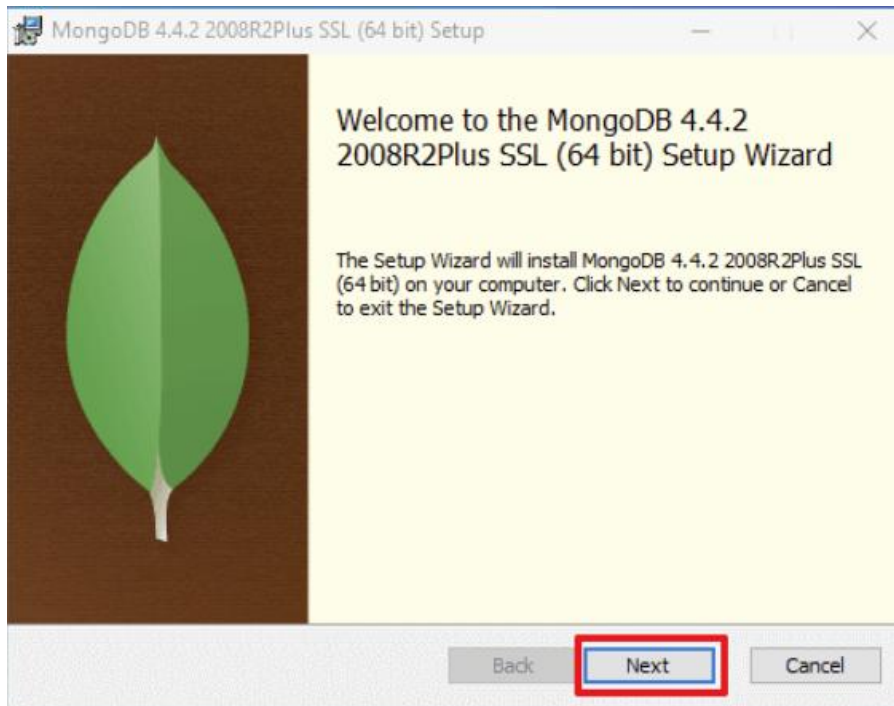
Bây giờ ta cần kiểm tra version của nodejs và npm vừa cài đặt bằng cách mở cửa sổ cmd và nhập 2 lệnh sau:

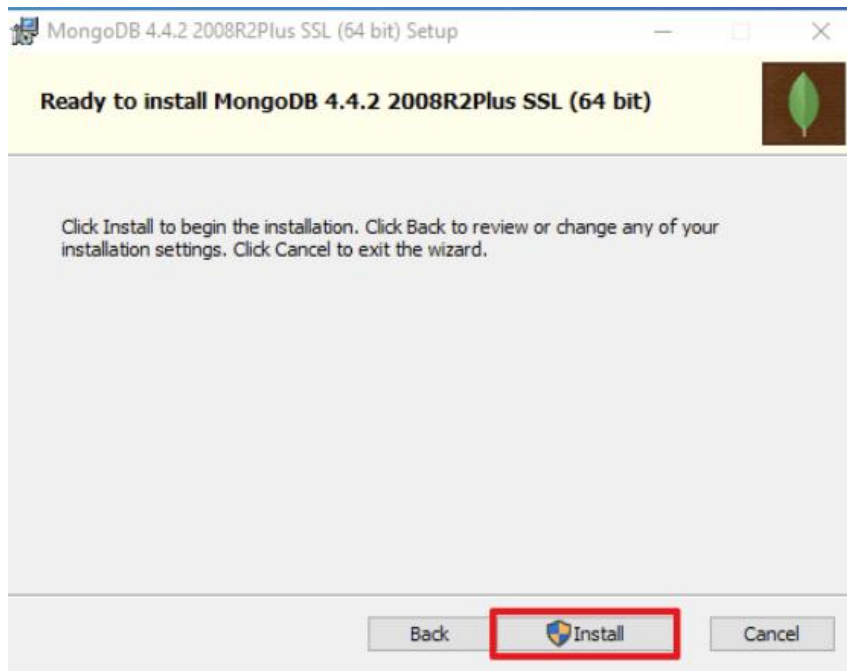
- `node -v` (kiểm tra version nodejs).
- `npm -v` (kiểm tra version npm).

5.1.2 cài đặt MongoDB (Window)



Nháy đúp vào file để bắt đầu quá trình cài đặt.





Sau khi cài đặt xong, bấm vào finish để hoàn tất quá trình cài đặt.



5.1.3 Môi trường để tiền xử lí và huấn luyện mô hình.

```
1 pandas==2.2.3
2 numpy==2.2.5
3 scikit-learn==1.6.1
4 pymongo==4.12.0
5 scipy==1.15.3
6 joblib==1.5.1
7 python-dateutil==2.9.0.post0
8 pytz==2025.2
```

5.1.4 Cấu hình BackEnd

- Khởi tạo dự án back end.

```
PS D:\Worker\SmartByte test> npm init -y
Wrote to D:\Worker\SmartByte test\package.json:

{
  "name": "smartbyte-test",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}
```

- Lệnh -y tự động tạo package.json với cấu hình mặc định.

- Cài đặt Express và các thư viện chính cho backend SmartBytes:

```
PS D:\Worker\SmartByte test> npm install express mongoose cors dotenv
added 86 packages, and audited 87 packages in 5s

16 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

- Cài đặt nodemon để tự động khởi động lại server khi thay đổi mã.

```
PS D:\Worker\SmartByte test> npm install --save-dev nodemon
added 27 packages, and audited 114 packages in 3s

20 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

- Cấu hình package.json.

```
{
  "dependencies": {
    "cors": "^2.8.5",
    "express": "^5.1.0",
    "mongoose": "^8.14.0",
    "nodemon": "^3.1.9"
  }
}
```

- Cấu hình MongoDB trong NodeJS.

```
const mongoose = require('mongoose');

module.exports.connect = async () => {
  try {
    await mongoose.connect('mongodb://localhost:27017/recipe_database');
    console.log('Connected to MongoDB');
  } catch (error) {
    console.error('Error connecting to MongoDB:', error);
  }
};
```

- Chạy local.

```
PS D:\Worker\SmartBytes\source code> npm start

> source-code@1.0.0 start
> node --inspect index.js

Debugger listening on ws://127.0.0.1:9229/56048155-6266-4d41-87ec-9cf2008dd374
For help, see: https://nodejs.org/en/docs/inspector
Server running at http://localhost:3000
Connected to MongoDB
```

5.1.5 Cấu hình Frontend

- Khởi tạo dự án.

```
PS D:\Worker\smartbytes test> npx create-react-app smartbyte-frontend

Creating a new React app in D:\Worker\smartbytes test\smartbyte-frontend.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

added 1324 packages in 1m

269 packages are looking for funding
  run `npm fund` for details

Initialized a git repository.

Installing template dependencies using npm...

added 18 packages, and changed 1 package in 7s

269 packages are looking for funding
  run `npm fund` for details
Removing template package using npm...

removed 1 package, and audited 1342 packages in 4s

269 packages are looking for funding
  run `npm fund` for details

8 vulnerabilities (2 moderate, 6 high)
```

- Cài đặt các thư viện chính:

```
PS D:\Worker\smartbytes test> npm install react-router-dom axios

added 30 packages in 2s

6 packages are looking for funding
  run `npm fund` for details
```

- Chạy môi trường.

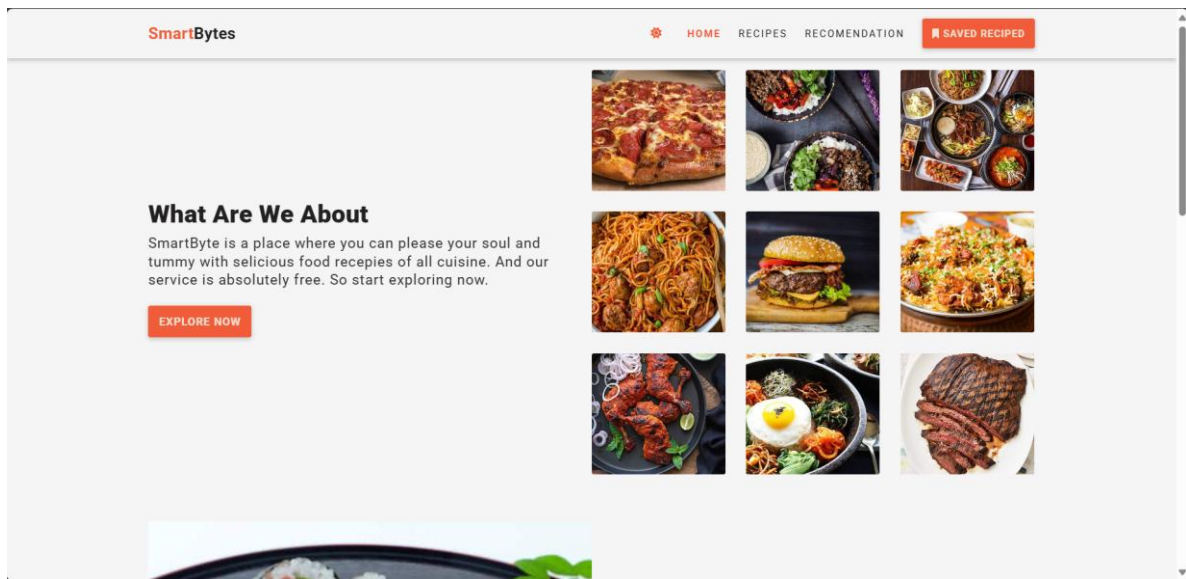
```
PS D:\Worker\smartbytes test\smartbyte-frontend> npm start

> smartbyte-frontend@0.1.0 start
> react-scripts start

(node:13456) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE]
upMiddlewares' option.
(Use `node --trace-deprecation ...` to show where the warning wa
(node:13456) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE]
etupMiddlewares' option.
Starting the development server...
Compiled successfully!
```

5.2 Kết quả thực nghiệm

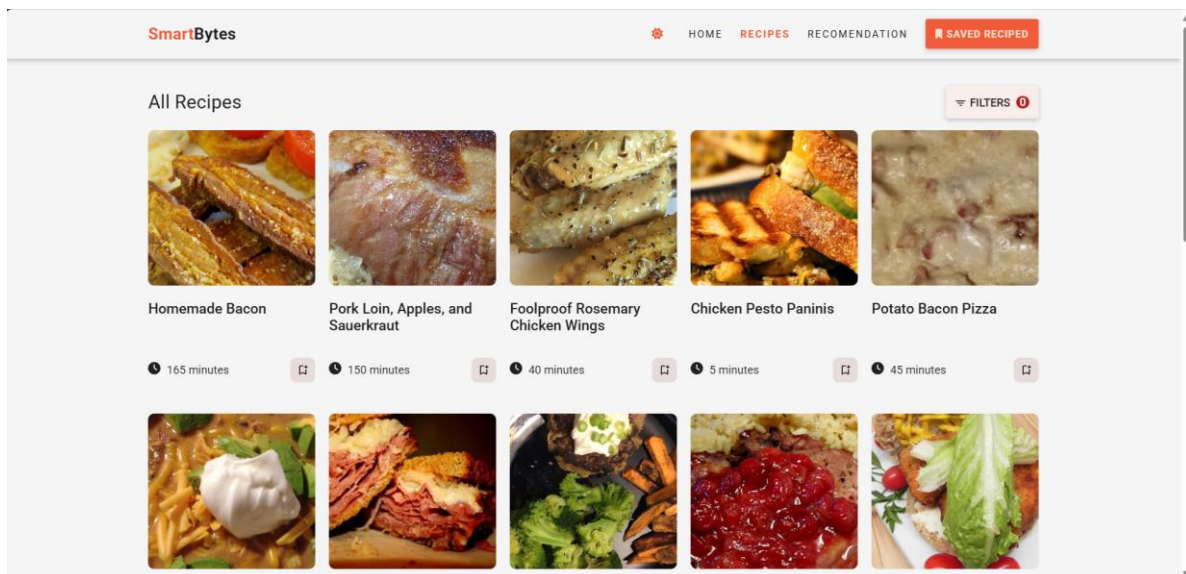
5.2.1 Trang chủ



Trang chủ <http://localhost:3001/>.

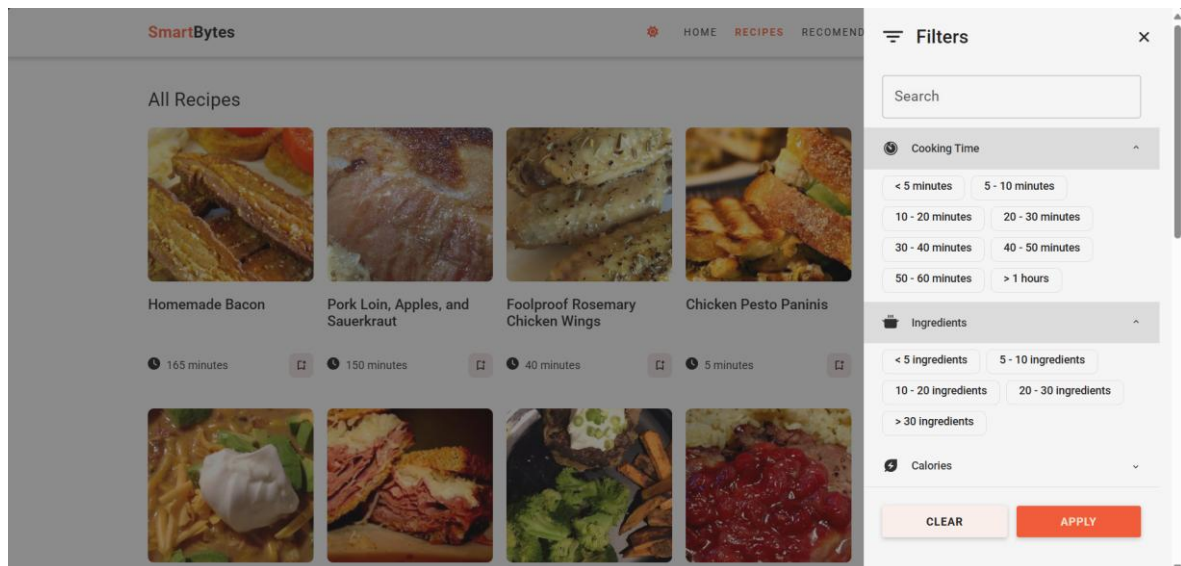
Đây là giao diện người dùng chính của trang web. Trang có tác dụng gây ấn tượng ban đầu cho người dùng.

5.2.2 Trang công thức

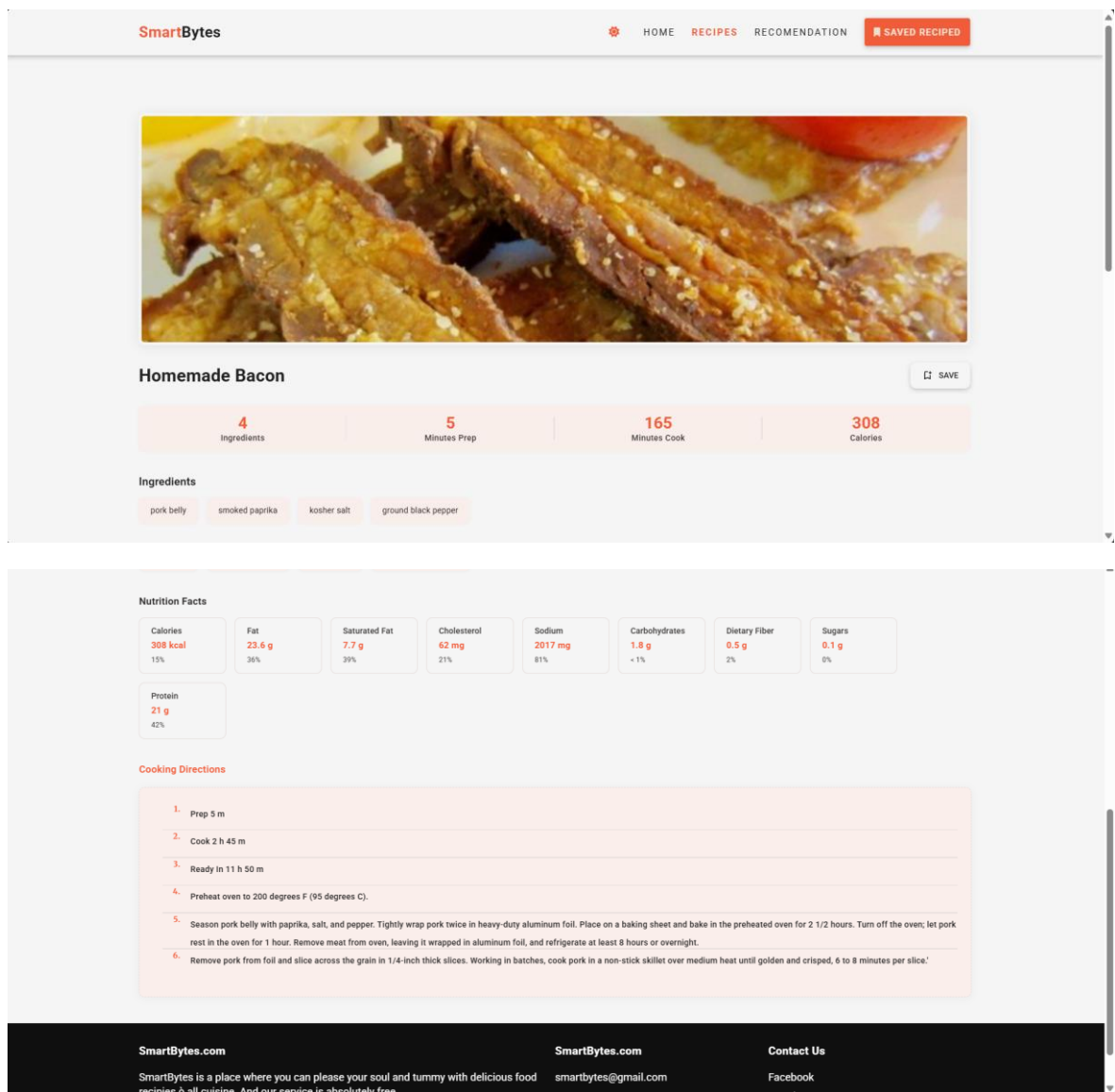


Trang công thức thể hiện những công thức món ăn với lazy loading.

Người dùng có thể tìm kiếm tên công thức, lọc theo thời gian nấu, số nguyên liệu, tổng lượng calo để lựa chọn món ăn mình mong muốn.



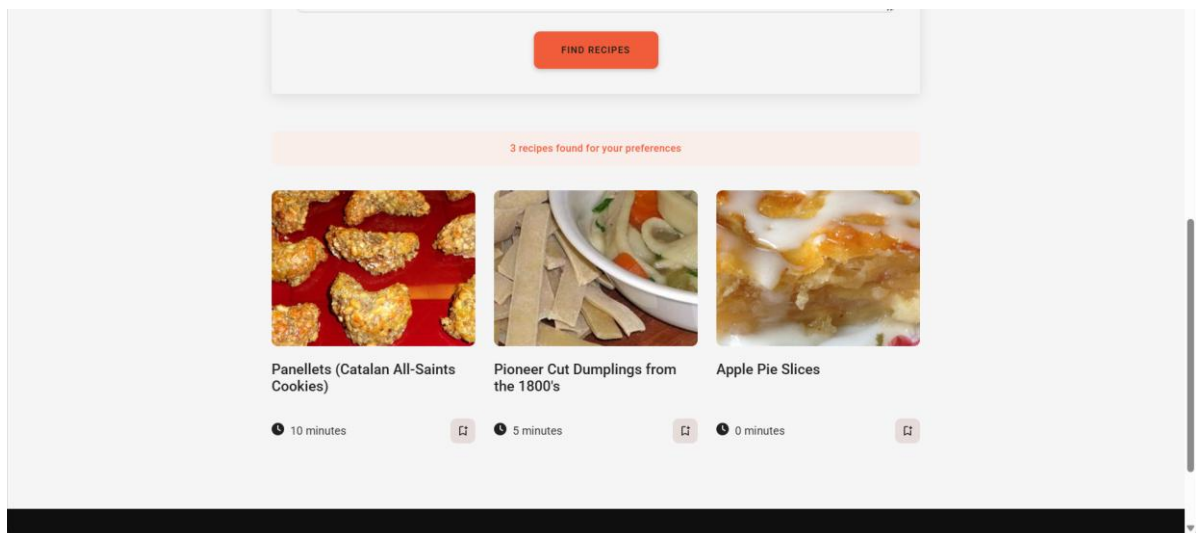
5.2.3 Trang Thông tin công thức



Trang chi tiết công thức hiển thị các thông tin của công thức bao gồm hình ảnh minh họa, nguyên liệu, dinh dưỡng, hướng dẫn nấu ăn.

5.2.4 Trang gợi ý món ăn

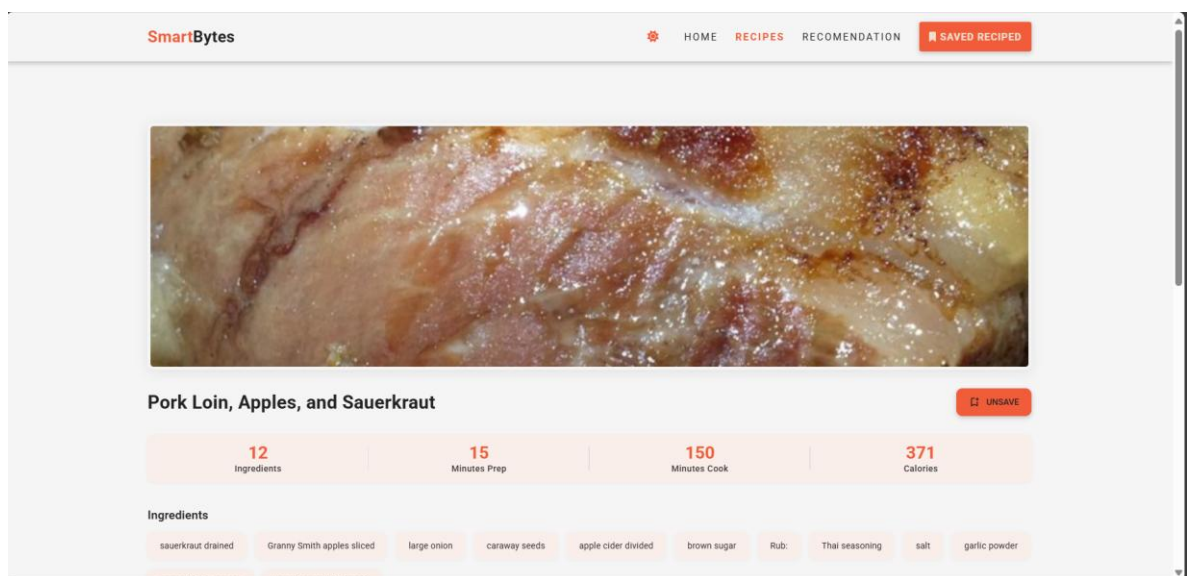
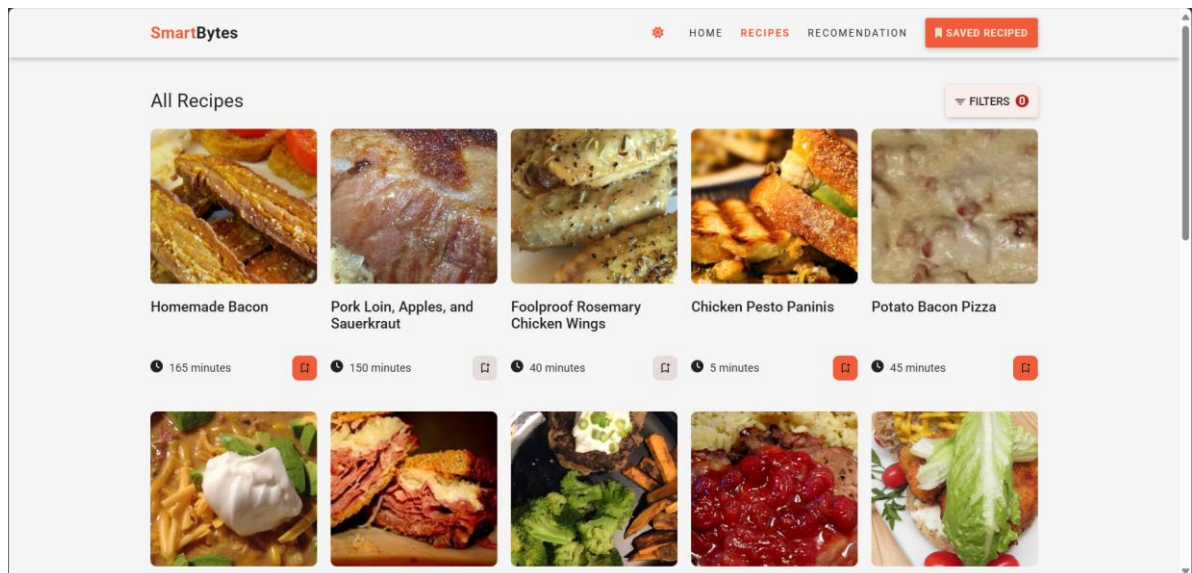
Người dùng nhập các thành phần dinh dưỡng và nguyên liệu muốn có cho bữa ăn của mình.



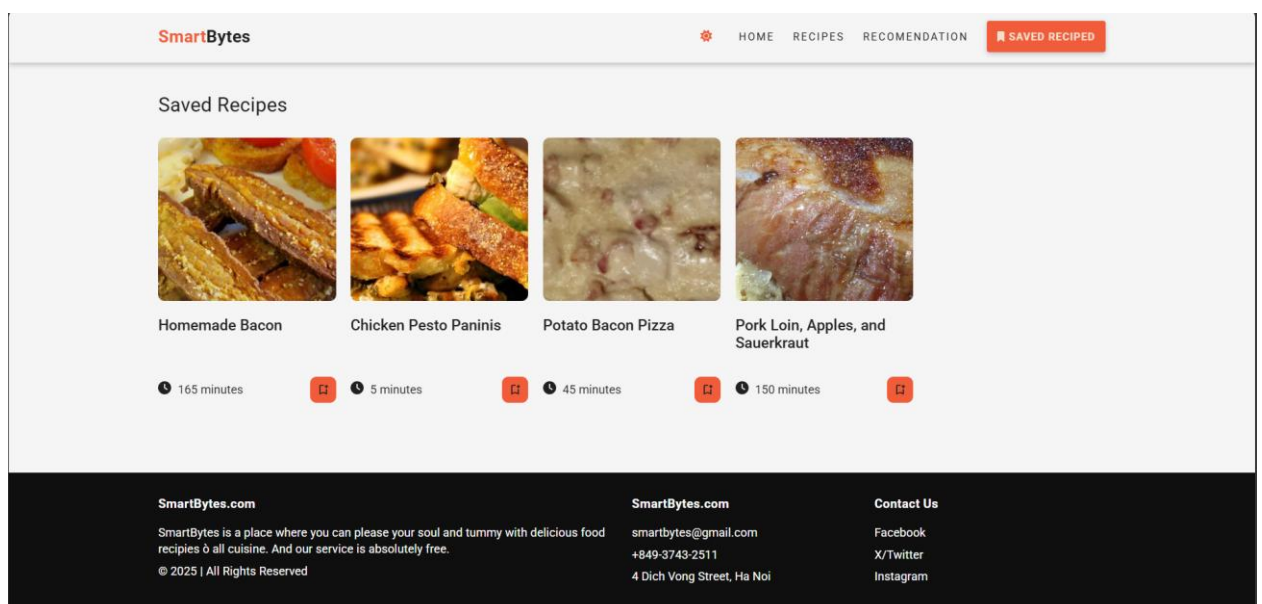
Hệ thống sẽ trả ra 3 món ăn phù hợp với tiêu chí của bạn.

5.2.5 Trang lưu món ăn

Người dùng có thể ấn vào biểu tượng lưu để lưu món ăn mình muốn.



Trang lưu món ăn sẽ hiện những món ăn người dùng đã lưu.



CHƯƠNG VI: ĐÁNH GIÁ KẾT QUẢ

6.1 Đánh giá kết quả

Dự án SmartBytes đã đạt được những kết quả đáng kể trong việc hiện thực hóa một website công thức nấu ăn có khả năng gợi ý món ăn theo nhu cầu dinh dưỡng cho khách vãng lai.

Các kết quả đạt được:

- Website hoạt động ổn định và hiệu quả: Hệ thống đã được triển khai thành công, cho phép người dùng tìm kiếm, xem chi tiết và lưu trữ công thức nấu ăn một cách mượt mà. Tính năng lazy loading đã chứng minh hiệu quả khi xử lý hàng chục nghìn công thức, đảm bảo thời gian tải trang nhanh chóng và trải nghiệm người dùng không bị gián đoạn.
- Hệ thống gợi ý dinh dưỡng hoạt động: Thuật toán KNN đã được tích hợp và hoạt động như mong đợi, cung cấp 3 gợi ý món ăn phù hợp nhất dựa trên các chỉ số dinh dưỡng mà người dùng nhập vào. Điều này tạo ra giá trị khác biệt cho SmartBytes so với các website công thức truyền thống, giúp người dùng dễ dàng theo dõi và điều chỉnh chế độ ăn uống của mình.
- Quản lý dữ liệu mạnh mẽ: Việc tiền xử lý và chuẩn hóa dữ liệu từ nguồn Kaggle, sau đó lưu trữ có cấu trúc trong MongoDB, đã tạo ra một nền tảng dữ liệu vững chắc. Dữ liệu được tổ chức tối ưu cho cả việc hiển thị trên giao diện người dùng lẫn phục vụ cho thuật toán gợi ý, đảm bảo tính nhất quán và khả năng truy vấn nhanh chóng.
- Kiến trúc công nghệ tích hợp: Sự kết hợp giữa React cho giao diện, Node.js cho xử lý Backend, cùng MongoDB cho cơ sở dữ liệu, đã tạo nên một hệ thống đồng bộ, dễ bảo trì và có tiềm năng mở rộng lớn.
- Tăng cường trải nghiệm người dùng vãng lai: Chức năng lưu món ăn yêu thích không cần đăng nhập là một điểm thành công lớn, loại bỏ rào cản cho người dùng mới và nâng cao sự tiện lợi.

6.2 Hạn chế

- Phạm vi dữ liệu công thức: Cơ sở dữ liệu có số lượng bản ghi tương đối nhỏ (khoảng 50 nghìn công thức món ăn). Số lượng công thức ban đầu được cung cấp có thể chưa thực sự phong phú và đa dạng, ngoài ra không sở hữu

các món ăn Việt như các website lớn chuyên về ẩm thực. Việc bổ sung thêm dữ liệu đòi hỏi thời gian và nguồn lực.

- Độ chính xác của thông tin dinh dưỡng: Dữ liệu dinh dưỡng cho mỗi món ăn được thu thập từ các nguồn công khai và có thể có sai số nhất định do phương pháp tính toán hoặc sự khác biệt trong nguyên liệu thực tế. Hệ thống chưa có khả năng tự động cập nhật hoặc xác thực độ chính xác tuyệt đối của các chỉ số dinh dưỡng khi người dùng thay đổi nguyên liệu.
- Lưu trữ dữ liệu cục bộ không đăng nhập: Việc lưu trữ món ăn yêu thích cục bộ trên trình duyệt người dùng (LocalStorage/SessionStorage) đồng nghĩa với việc danh sách món ăn yêu thích sẽ bị mất nếu người dùng xóa bộ nhớ cache của trình duyệt hoặc sử dụng thiết bị khác. Dự án hiện chưa có tính năng đồng bộ hóa dữ liệu này qua tài khoản người dùng.
- Không có khả năng tương tác nâng cao: Website hiện tại chỉ tập trung vào việc cung cấp thông tin và gợi ý. Các tính năng tương tác hai chiều như bình luận, đánh giá công thức, hay chia sẻ món ăn với người khác chưa được triển khai.
- Chưa tối ưu cho mọi thiết bị: Mặc dù đã có sự cân nhắc, giao diện có thể chưa được tối ưu hoàn hảo trên tất cả các loại thiết bị và kích thước màn hình khác nhau.
- Không có hệ thống quản trị người dùng: Dự án hiện tại không bao gồm hệ thống quản lý tài khoản người dùng, đăng nhập/đăng ký. Tất cả các tính năng đều có thể sử dụng mà không cần xác thực danh tính.

6.3 Hướng phát triển

Dự án SmartBytes, với những kết quả đã đạt được, có nhiều tiềm năng để tiếp tục phát triển và mở rộng, nâng cao giá trị mang lại cho người dùng. Dưới đây là một số hướng phát triển chính có thể được xem xét trong tương lai:

- Nâng cao hệ thống gợi ý, mở rộng tiêu chí, hỗ trợ chế độ ăn đặc biệt, cá nhân hóa theo sở thích.
- Thêm tính năng đăng nhập, giúp người dùng có một profile chế độ ăn riêng.
- Tăng cường nguồn dữ liệu, ước tính giá thành công thức.

TÀI LIỆU THAM KHẢO

- [1]. Hệ thống gợi ý [Content-based Filtering: Recipe Recommender](#).
- [2]. Gọi python script từ Node.js [Kết Nối Ứng Dụng Nodejs Với Python](#).
- [3]. Thuật toán KNN [Machine Learning cơ bản](#).
- [4]. Template frontend [Build a Recipe App with Vanilla JavaScript](#).
- [5]. ReactJS [Giới thiệu về ReactJS - Phần I \(Các khái niệm cơ bản\)](#).
- [6]. NodeJS [NodeJS là gì? Tổng quan kiến thức về Node.JS cho người mới](#).
- [7]. MongoDB Documentation(2025) [MongoDB Documentation](#).
- [8]. Pandas Documentation <https://pandas.pydata.org/docs>.
- [9]. Scikit-learn Documentation(2025) [scikit-learn: machine learning in Python — scikit-learn 1.6.1 documentation](#).
- [10]. FoodRecSys-V1 Dataset(2023) [foodRecSys-V1](#).