BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC ĐẠI NAM

--- 🕮 ---

ĐỒ ÁN TỐT NGHIỆP

SỬ DỤNG REACT NATIVE VÀ NESTJS ĐỂ THIẾT KẾ VÀ TRIỂN KHAI ỨNG DỤNG GIỚI THIỆU ĐẶC SẢN THEO VÙNG MIỀN Ở VIỆT NAM

SINH VIÊN THỰC HIỆN : NGUYỄN VĂN TẠO

MÃ SINH VIÊN : 1451020203

KHOA : CÔNG NGHỆ THÔNG TIN

HÀ NỘI - 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC ĐẠI NAM

NGUYỄN VĂN TẠO

SỬ DỤNG REACT NATIVE VÀ NESTJS ĐỂ THIẾT KẾ VÀ TRIỂN KHAI ỨNG DỤNG GIỚI THIỆU ĐẶC SẢN THEO VÙNG MIỀN Ở VIỆT NAM

CHUYÊN NGÀNH : **CÔNG NGHỆ THÔNG TIN**

MÃ SỐ : **74.80.201**

NGƯỜI HƯỚNG DẪN: TS. NGUYỄN NGỌC TÂN

HÀ NỘI - 2024

LỜI CAM ĐOAN

Kính thưa Quý thầy cô giáo,

Tên em là Nguyễn Văn Tạo, mã số sinh viên: 1451020203.

Là sinh viên lớp CNTT 14 – 04 khoa Công nghệ thông tin trường Đại học Đại Nam khóa 2020 – 2024.

Em xin cam đoan bài báo cáo đồ án tốt nghiệp này là công trình nghiên cứu của bản thân em dưới sự hướng dẫn của thầy TS. Nguyễn Ngọc Tân. Các số liệu, kết quả, phân tích và đánh giá trong bài báo cáo là trung thực, được thu thập từ quá trình thực tập và nghiên cứu của cá nhân em, không sao chép hoặc sử dụng kết quả nghiên cứu của người khác.

Em cũng đã tuân thủ quy định về bảo mật thông tin và quyền sở hữu trí tuệ. Mọi trích dẫn, sử dụng tài liệu, dữ liệu từ các nguồn khác đều đã được ghi rõ trong tài liệu tham khảo.

Em xin cam kết các kết quả trong bài báo cáo đồ án tốt nghiệp này chưa từng được công bố trong bất kỳ bài báo cáo nào khác.

Hà Nội, ngày 24 tháng 05 năm 2024

Sinh viên thực hiện

LỜI CẢM ƠN

Sau thời gian học tập tại trường, được sự cho phép của Khoa Công nghệ thông tin trường Đại học Đại Nam và sự tiếp nhận hướng dẫn khoa học tận tình của thầy TS. Nguyễn Ngọc Tân, em đã hoàn thành đồ án tốt nghiệp của mình.

Em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy TS. Nguyễn Ngọc Tân, người đã không chỉ truyền đạt kiến thức chuyên môn mà còn luôn động viên, khích lệ em trong suốt quá trình thực hiện đồ án. Sự chỉ bảo tận tâm của thầy đã giúp em vượt qua những khó khăn, hoàn thiện bản thân và hoàn thành đồ án này một cách tốt nhất.

Em cũng xin bày tỏ lòng biết ơn đến quý thầy cô trong Khoa Công nghệ Thông tin, những người đã trang bị cho em nền tảng kiến thức vững chắc, tạo tiền đề cho em thực hiện đồ án tốt nghiệp.

Vì bài báo cáo đồ án tốt nghiệp được thực hiện trong thời gian ngắn và hạn chế về mặt kiến thức chuyên môn, do đó có thể bài báo cáo của em sẽ không thể tránh khỏi những sai sót nhất định. Đồng thời bài báo cáo là kết quả của một quá trình tổng kết, thu thập kết quả từ thực tế, những bài học đúc rút lại được trong quá trình làm đồ án của em. Em rất mong có được những ý kiến đóng góp của thầy, cô để bài báo cáo của em có thể hoàn thiện hơn.

Qua bài báo cáo này, em xin cảm ơn thầy Nguyễn Ngọc Tân, giảng viên đã trực tiếp hướng dẫn em trong quá trình thực hiện đồ án tốt nghiệp. Đồng thời, em cũng xin trân trọng cảm ơn tất cả mọi người đã đồng hành và hỗ trợ em trong suốt thời gian qua.

Em xin chân thành cảm ơn!

LỜI NÓI ĐẦU

Quá trình học tập tại trường đã cho mỗi sinh viên lượng kiến thức lý thuyết về chuyên ngành. Tuy nhiên lượng kiến thức ấy sẽ không bao giờ là đủ nếu như không được đi đôi với thực hành, đối với xã hội ngày càng phát triển hiện nay thì việc cọ sát thực tế cùng với những kiến thức mà sinh viên đã học được ở trên trường thực sự rất cần thiết. Việc này sẽ giúp sinh viên vận dụng được tốt hơn những kiến thức mà mình đã tiếp thu được ở trên trường cũng như biết được môi trường làm việc mà sau này mình sẽ phải tham gia sẽ như thế nào.

Trong bối cảnh sự phát triển mạnh mẽ của công nghệ thông tin, không thể phủ nhận vai trò quan trọng của nó đối với đời sống con người. Các lĩnh vực trong xã hội ngày càng sử dụng công nghệ thông tin và nhiều phần mềm đã mang lại hiệu quả rõ ràng. Tuy nhiên, với sự đổi mới không ngừng, ngành công nghệ thông tin càng phải phát triển để đáp ứng những thách thức mới. Trong bối cảnh này, việc ứng dụng công nghệ thông tin và tin học hóa trở thành quyết định quan trọng đối với chính phủ, tổ chức, và doanh nghiệp.

Ngày nay, ứng dụng công nghệ thông tin không chỉ là một yếu tố quyết định mà còn là một cơ hội để tạo nên những bước phát triển mạnh mẽ. Trong lĩnh vực giải trí, trò chơi trực tuyến đóng vai trò quan trọng, mang lại không chỉ giây phút giải trí mà còn là cơ hội để thách thức bản thân và kết nối với cộng đồng.

NHẬN XÉT

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Từ viết đầy đủ
API	Application Programming Interface
CLI	Command Line Interface
CSDL	Cơ sở dữ liệu
UI	User Interface
SDK	Software Development Kit

DANH MỤC HÌNH ẢNH

STT	Mô tả
Hình 3.1	*Mô hình thực thể - quan hệ (Entity – Relationship Diagram)*
Hình 3.2	*Biểu đồ lớp (Class Diagram)*
Hình 3.3	*Biểu đồ tuần tự (Sequence Diagram)*
Hình 3.4	*Biểu đồ phân cấp chức năng (Functional Decomposition Diagram)*
Hình 3.5	*Dữ liệu của bảng Regions*
Hình 3.6	*Dữ liệu của bảng Province*
Hình 3.7	*Dữ liệu của bảng Specialty*
Hình 3.8	*Dữ liệu của bảng SpecialtyDetails*
Hình 3.9	*Dữ liệu của bảng Account*
Hình 3.10	*Dữ liệu của bảng Profile*
Hình 3.11	*Giao diện trang đăng nhập và đăng ký tài khoản*
Hình 3.12	*Giao diện trang thêm thông tin cá nhân trước khi đăng ký tài khoản*
Hình 3.13	*Giao diện trang chủ khi đã đăng nhập vào ứng dụng*
Hình 3.14	*Giao diện trang đặc sản của một tỉnh*
Hình 3.15	*Giao diện thông báo khi người dùng thêm đặc sản vào danh sách yêu thích thành công*
Hình 3.16	*Giao diện trang danh sách đặc sản yêu thích*

Hình 3.17	*Giao diện trang chi tiết đặc sản của một tỉnh*
Hình 3.18	*Giao diện trang hồ sơ thông tin cá nhân của người dùng*
Hình 3.19	*Giao diện trang sửa thông tin cá nhân người dùng*

DANH MỤC BẢNG BIỂU

STT	Mô tả
Bảng 3.1	Bảng Regions
Bảng 3.2	Bảng Province
Bảng 3.3	Bảng Specialty
Bảng 3.4	Bảng Specialty Details
Bảng 3.5	Bảng Favorites
Bảng 3.6	Bảng Account
Bảng 3.7	Bảng Profile
Bảng 3.8	Bảng Notification
Bảng 3.9	Bảng Account_favorite_specialty
Bảng 3.10	Bảng Specialty_favorite_account

MỤC LỤC

Chương 1 1

GIỚI THIỆU KHÁI QUÁT VỀ ĐỀ TÀI 1

1.1. Lý do chọn đề tài 1

1.2. Vấn đề nghiên cứu 2

1.3. Phương pháp nghiên cứu 2

1.4. Bố cục luận văn 3

Chương 2 5

CƠ SỞ LÝ THUYẾT 5

2.1. Tổng quan về React Native 5

2.1.1. Khái niệm cơ bản về React Native 5

2.1.2. Các tính năng cơ bản 6

2.1.3. Cách hoạt động 7

2.1.4. Các thành phần chính 8

2.2. TypeScript 9

2.2.1. TypeScript là gì? 9

2.2.2. Lý do nên sử dụng TypeScript 10

2.2.3. Chức năng của TypeScript 11

2.2.4. Ưu điểm và hạn chế của TypeScript 13

2.3. Tổng quan về NestJs 14

2.3.1. Decorator trong NestJS 14

***2.3.2.* Dependency Injection trong NestJs** 15

2.3.3. Hỗ trợ công nghệ và tích hợp CSDL 16

2.4. Tổng quan về Docker và DBeaver MySql 17

2.4.1. Docker 17

2.4.2. DBeaver MySql 20

2.5. Forward port và chế độ bảo mật 22

2.5.1. Lý do cần forward port 22

2.5.2. Ý nghĩa 23

2.5.3. Hạn chế 24

2.5.4. Chế độ bảo mật 25

Chương 3 27

THIẾT KẾ HỆ THỐNG VÀ KẾT QUẢ ĐẠT ĐƯỢC 27

3.1. Sơ đồ usecase 27

3.1.1. Mô hình thực thể - quan hệ (Entity - Relationship Diagram) 27

3.1.2. Biểu đồ lớp (Class Diagram) 28

3.1.3. Biểu đồ tuần tự (Sequence Diagram) 30

3.1.4. Biểu đồ phân cấp chức năng (Functional Decomposition Diagram) 32

3.2. Thiết kế cơ sở dữ liệu 34

3.2.1. Bảng Regions 34

3.2.2. Bảng Province 34

3.2.3. Bảng Specialty 35

3.2.4. Bảng Specialty Details 35

3.2.5. Bảng Favorites 36

3.2.6. Bảng Account 36

3.2.7. Bảng Profile 37

3.2.8. Bảng Notification 38

3.2.9. Bảng Account_favorite_specialty 39

3.2.10. Bảng Specialty_favorite_account 39

3.3. Dữ liệu của một số bảng 40

3.4. Giao diện và chức năng của ứng dụng.....43

3.4.1. Giao diện của ứng dụng.....43

3.4.2. Một vài chức năng.....51

KẾT LUẬN 53

DANH MỤC TÀI LIỆU THAM KHẢO 55

Chương 1

GIỚI THIỆU KHÁI QUÁT VỀ ĐỀ TÀI

1.1. Lý do chọn đề tài

Việt Nam tự hào sở hữu một nền văn hóa ẩm thực vô cùng phong phú và đa dạng. Mỗi vùng miền trên dải đất hình chữ S đều có những món ăn đặc sản mang đậm hương vị và bản sắc riêng, tạo nên một bức tranh ẩm thực đầy màu sắc và hấp dẫn. Nhận thấy tiềm năng to lớn từ sự đa dạng này, việc thiết kế và phát triển một ứng dụng giới thiệu đặc sản theo vùng miền sẽ không chỉ là một công cụ hữu ích cho du khách khám phá văn hóa ẩm thực địa phương mà còn là cầu nối quảng bá những giá trị truyền thống đến với đông đảo người dân trong nước.

Ứng dụng này sẽ đóng vai trò như một "cẩm nang ẩm thực", cung cấp thông tin chi tiết về nguồn gốc, cách chế biến, hương vị đặc trưng và địa chỉ thưởng thức các món ăn đặc sản ở từng vùng miền. Thông qua ứng dụng, du khách có thể dễ dàng tìm kiếm và lựa chọn những món ăn phù hợp với sở thích, đồng thời khám phá những nét văn hóa độc đáo ẩn chứa trong từng món ăn.

Việc giới thiệu đặc sản theo vùng miền không chỉ góp phần thúc đẩy ngành du lịch nước nhà phát triển mà còn mang lại nhiều lợi ích kinh tế và xã hội to lớn. Khi du khách biết đến và tìm đến những món ăn đặc sản, điều này sẽ tạo ra nhu cầu tiêu thụ, từ đó kích thích sản xuất và kinh doanh các sản phẩm địa phương. Đồng thời, việc quảng bá đặc sản còn giúp bảo tồn và phát huy những giá trị văn hóa truyền thống, tạo động lực cho người dân địa phương gìn giữ và phát triển nghề truyền thống, góp phần bảo vệ môi trường và phát triển bền vững các nguồn lợi thế địa phương.

Đặc biệt, đối với những khu vực nhỏ và vùng sâu vùng xa, ứng dụng này sẽ mở ra cơ hội tiếp cận thị trường rộng lớn hơn, giúp người dân địa phương quảng bá sản phẩm và tăng thu nhập. Qua đó, ứng dụng không chỉ là công cụ giới thiệu ẩm thực mà còn là đòn bẩy thúc đẩy phát triển kinh tế, nâng cao đời sống người dân, đồng thời thu hẹp khoảng cách phát triển giữa các vùng miền.

1.2. Vấn đề nghiên cứu

Nghiên cứu toàn diện và chuyên sâu về các đặc sản ẩm thực và thực phẩm chế biến đặc trưng của từng vùng miền trên khắp lãnh thổ Việt Nam thông qua các nguồn tài liệu trực tuyến. Điều này bao gồm việc tổng hợp thông tin chi tiết về nguồn gốc, quy trình sản xuất, thành phần nguyên liệu, hương vị đặc trưng và giá trị dinh dưỡng của từng sản phẩm từ các trang web, blog ẩm thực, sách điện tử và các nguồn đáng tin cậy khác.

Phân tích các yếu tố lịch sử, văn hóa, địa lý và nguồn nguyên liệu đã tác động và định hình nên sự phát triển của các đặc sản ẩm thực này dựa trên các nghiên cứu và bài viết trực tuyến. Tìm hiểu về những câu chuyện lịch sử, truyền thống văn hóa, điều kiện địa lý và sự đa dạng sinh học đã góp phần tạo nên những nét độc đáo và riêng biệt của từng món ăn, từ đó làm nổi bật giá trị văn hóa và lịch sử của chúng thông qua các nguồn tài liệu lịch sử, địa lý trực tuyến.

Thực hiện nghiên cứu về xu hướng sử dụng ứng dụng di động trong lĩnh vực du lịch và ẩm thực tại Việt Nam thông qua các báo cáo thị trường, bài viết phân tích và số liệu thống kê trực tuyến. Từ đó, đánh giá tiềm năng và cơ hội phát triển của ứng dụng giới thiệu đặc sản vùng miền trong bối cảnh công nghệ ngày càng phát triển như hiện nay.

Thu thập thông tin từ các diễn đàn, mạng xã hội, và các trang đánh giá trực tuyến để đánh giá nhu cầu và mong muốn của cả du khách và người tiêu dùng địa phương về thông tin liên quan đến đặc sản vùng miền. Xác định những thông tin cần thiết, hình thức trình bày phù hợp và các tính năng bổ sung mà người dùng mong muốn có trong ứng dụng thông qua việc phân tích các bình luận, phản hồi và đánh giá trực tuyến. Điều này giúp đảm bảo rằng ứng dụng được thiết kế và phát triển đáp ứng tối đa nhu cầu của người dùng, mang lại trải nghiệm tốt nhất và góp phần quảng bá hiệu quả các đặc sản vùng miền Việt Nam.

1.3. Phương pháp nghiên cứu

Thực hiện tìm hiểu trên internet để nắm bắt nhu cầu và mong muốn của người dùng tiềm năng đối với ứng dụng giới thiệu đặc sản vùng miền. Tìm kiếm

và phân tích các bài viết, thảo luận trên các diễn đàn, mạng xã hội, blog du lịch và ẩm thực để hiểu rõ hơn về những mong đợi, kỳ vọng và trải nghiệm mà người dùng mong muốn có được từ một ứng dụng như vậy. Thu thập thông tin từ các cuộc khảo sát trực tuyến, đánh giá của người dùng về các ứng dụng tương tự, cũng như các xu hướng tìm kiếm liên quan đến đặc sản và du lịch ẩm thực trên các công cụ tìm kiếm phổ biến. Từ đó, xây dựng một bức tranh tổng quan về nhu cầu thị trường và định hướng phát triển sản phẩm phù hợp.

Song song với việc tìm hiểu ý kiến người dùng, tiến hành thu thập dữ liệu về đặc sản và thông tin vùng miền từ các nguồn đáng tin cậy trên internet. Khai thác triệt để các trang web chính thức của các địa phương, các tổ chức du lịch, các blog ẩm thực uy tín, các trang thương mại điện tử chuyên về đặc sản và các nguồn thông tin trực tuyến khác để có được những thông tin chính xác, đầy đủ và cập nhật nhất về các đặc sản vùng miền trên khắp cả nước. Đảm bảo tính xác thực và chất lượng của dữ liệu thông qua quá trình kiểm chứng và đối chiếu từ nhiều nguồn khác nhau, từ đó xây dựng một cơ sở dữ liệu phong phú và đa dạng về đặc sản ẩm thực Việt Nam, làm nền tảng vững chắc cho việc phát triển ứng dụng.

Sử dụng React Native để phát triển ứng dụng di động đa nền tảng với giao diện người dùng linh hoạt và hiệu suất cao.

Sử dụng NestJS để xây dựng các API và quản lý dữ liệu, đảm bảo tính bảo mật và hiệu suất cho ứng dụng.

1.4. Bố cục luận văn

Chương 1: Giới thiệu khái quát về đề tài

1.1. Lý do chọn đề tài

1.2. Vấn đề nghiên cứu

1.3. Phương pháp nghiên cứu

1.4. Bố cục luận văn

Chương 2: Cơ sở lý thuyết

2.1. Tổng quan về React Native

2.2. Tổng quan về TypeScript

2.3. Tổng quan về NestJs

2.4. Tổng quan về Docker và Dbeaver MySql

2.5. Forward port và chế độ bảo mật

Chương 3: Thiết kế hệ thống và kết quả đạt được

3.1. Sơ đồ usecase

3.2. Thiết kế cơ dở dữ liệu

3.3. Dữ liệu của một số bảng

3.4. Giao diện và chức năng của ứng dụng

Kết luận

Tài liệu tham khảo

Tiểu kết:

Chương 1 đã giới thiệu tổng quan về đề tài, bao gồm lý do chọn đề tài, vấn đề nghiên cứu, phương pháp nghiên cứu và bố cục của luận văn.

Đề tài tập trung vào việc thiết kế và phát triển ứng dụng di động giới thiệu đặc sản vùng miền ở Việt Nam, nhằm tận dụng tiềm năng du lịch và ẩm thực phong phú của đất nước.

Các vấn đề nghiên cứu đã được xác định, bao gồm việc thu thập và phân tích thông tin về đặc sản vùng miền, đánh giá nhu cầu người dùng và xu hướng sử dụng ứng dụng di động.

Phương pháp nghiên cứu kết hợp giữa khảo sát người dùng, thu thập dữ liệu từ các nguồn đáng tin cậy và sử dụng công nghệ React Native và NestJS để phát triển ứng dụng.

Chương 2

CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về React Native

2.1.1. Khái niệm cơ bản về React Native

React Native, một framework phát triển ứng dụng di động đa nền tảng, đã tạo nên một cuộc cách mạng trong cách các nhà phát triển tiếp cận việc xây dựng ứng dụng cho cả iOS và Android. Được Facebook giới thiệu vào năm 2015, React Native đã nhanh chóng trở thành một công cụ phổ biến và được ưa chuộng, giải quyết nhiều thách thức tồn tại từ lâu trong lĩnh vực phát triển ứng dụng di động.

Điểm cốt lõi của React Native là khả năng sử dụng JavaScript, một ngôn ngữ lập trình phổ biến và linh hoạt, để xây dựng các ứng dụng có giao diện người dùng (UI) gần như tương tự với các ứng dụng native. Điều này đạt được thông qua việc sử dụng các thành phần UI gốc của từng nền tảng, được kết nối với mã JavaScript thông qua một cầu nối (bridge). Nhờ đó, ứng dụng React Native có thể mang lại trải nghiệm người dùng mượt mà và tự nhiên, đồng thời tận dụng tối đa các tính năng và khả năng của từng hệ điều hành.

Một trong những lợi ích lớn nhất của React Native là khả năng tái sử dụng mã giữa các nền tảng. Điều này đồng nghĩa với việc các nhà phát triển không cần phải viết hai bộ mã riêng biệt cho iOS và Android, giúp tiết kiệm đáng kể thời gian và công sức. Ngoài ra, việc sử dụng một ngôn ngữ lập trình duy nhất giúp giảm thiểu rủi ro lỗi và đảm bảo tính nhất quán của ứng dụng trên các nền tảng khác nhau.

React Native còn được trang bị nhiều tính năng hữu ích, như Hot Reloading, cho phép các nhà phát triển xem ngay lập tức kết quả của những thay đổi trong mã mà không cần phải biên dịch lại toàn bộ ứng dụng. Điều này giúp tăng tốc độ phát triển và tạo điều kiện thuận lợi cho việc thử nghiệm và tinh chỉnh ứng dụng.

Cộng đồng React Native cũng là một yếu tố quan trọng đóng góp vào sự thành công của framework này. Với một cộng đồng lớn mạnh và tích cực, các nhà

phát triển có thể dễ dàng tìm thấy sự hỗ trợ, tài liệu và các thư viện mã nguồn mở để giải quyết các vấn đề phát sinh trong quá trình phát triển.

React Native là một framework mạnh mẽ và linh hoạt, mang đến một giải pháp hiệu quả cho việc xây dựng ứng dụng di động đa nền tảng. Sự kết hợp giữa khả năng tái sử dụng mã, hiệu suất cao, tính linh hoạt và cộng đồng hỗ trợ lớn mạnh đã giúp React Native trở thành một công cụ không thể thiếu trong bộ công cụ của các nhà phát triển ứng dụng di động hiện đại.

2.1.2. Các tính năng cơ bản

React Native, một framework đa nền tảng mạnh mẽ, được xây dựng dựa trên sự kết hợp hài hòa giữa hai thành phần chính: React và Native. React, một thư viện JavaScript do Facebook phát triển, cung cấp một cách tiếp cận linh hoạt và hiệu quả để xây dựng giao diện người dùng. Trong khi đó, Native là các bộ công cụ phát triển phần mềm (SDK) của hai nền tảng di động hàng đầu là iOS và Android. Sự kết hợp này tạo nên sức mạnh của React Native, cho phép các nhà phát triển tận dụng những ưu điểm của cả hai thế giới.

Về mặt kiến trúc, React Native sử dụng hai loại thành phần cơ bản: "Native Components" và "React Components". Native Components là các thành phần giao diện người dùng được cung cấp sẵn bởi hệ điều hành của từng nền tảng, ví dụ như View, Text, Image, và TextInput. Các thành phần này được hiển thị và quản lý trực tiếp bởi hệ điều hành, đảm bảo hiệu suất và trải nghiệm người dùng tối ưu.

React Components, mặt khác, là các thành phần React thuần túy được viết bằng JavaScript. Chúng đóng vai trò là các khối xây dựng cơ bản cho giao diện người dùng của ứng dụng React Native. Các React Components này có thể được kết hợp với nhau để tạo thành các thành phần phức tạp hơn, giúp tổ chức và quản lý mã nguồn một cách hiệu quả.

React Native cung cấp hai cách tiếp cận chính để xây dựng ứng dụng: Expo CLI và React Native CLI. Expo CLI là một công cụ dòng lệnh đơn giản và dễ sử dụng, cung cấp một môi trường phát triển tích hợp sẵn với nhiều tính năng tiện

lợi. Expo CLI phù hợp cho những dự án nhỏ và trung bình, hoặc khi muốn nhanh chóng tạo một ứng dụng mẫu để thử nghiệm ý tưởng.

React Native CLI cung cấp sự kiểm soát và tùy chỉnh tối đa đối với dự án, có thể dễ dàng tích hợp các thư viện Native Modules của bên thứ ba, hoặc thậm chí viết các thành phần Native Modules của riêng mình để mở rộng khả năng của React Native. React Native CLI là lựa chọn phù hợp cho những dự án lớn và phức tạp, yêu cầu sự linh hoạt và khả năng tùy biến cao.

2.1.3. Cách hoạt động

React Native sử dụng JavaScript để tạo ra các thành phần giao diện người dùng và kết nối chúng với mã native thông qua "bridge" (cầu nối), Virtual DOM, React Component và Native Component, Expo CLI và React Naitve CLI, JavaScript Thread và Native Thread:

Trong React Native, "Bridge" là thành phần quan trọng nhất, hoạt động như một cầu nối giữa mã JavaScript và native code. Khi một ứng dụng React Native chạy, mã JavaScript sẽ chạy trên một thread riêng, trong khi UI (giao diện người dùng) được xử lý trên thread native. Bridge chịu trách nhiệm truyền thông tin giữa hai bên này. Khi muốn hiển thị hoặc thay đổi gì đó trên giao diện người dùng, React Native sẽ gửi một thông điệp qua Bridge đến native code, yêu cầu thực hiện thay đổi đó.

React sử dụng Virtual DOM để tối ưu hóa việc cập nhật giao diện người dùng. Thay vì cập nhật trực tiếp trên DOM thực, React áp dụng các thay đổi trên Virtual DOM. Sau đó, một thuật toán diff sẽ so sánh sự khác biệt giữa Virtual DOM và DOM thực để xác định những thay đổi tối thiểu cần thực hiện. Trong React Native, quá trình này tương tự nhưng thay vì cập nhật DOM, nó sẽ yêu cầu native code cập nhật UI.

React Components: Là các thành phần được xây dựng sử dụng JSX (một cú pháp mở rộng của JavaScript), cho phép miêu tả giao diện người dùng một cách khai báo; React Components xử lý logic và dữ liệu của ứng dụng. Native Components: React Native sẽ dịch các React Components thành các Native

Components tương ứng của iOS và Android. Điều này đảm bảo rằng ứng dụng cuối cùng sẽ có hiệu suất và cảm giác như một ứng dụng native.

Expo CLI: Là một bộ công cụ và một framework cho phép xây dựng ứng dụng React Native mà không cần cài đặt và cấu hình môi trường phát triển native. Expo giúp việc phát triển ứng dụng trở nên dễ dàng hơn nhưng có những hạn chế về việc truy cập trực tiếp vào các API và thành phần native; React Native CLI: Cung cấp sự linh hoạt cao hơn, cho phép bạn cấu hình toàn bộ môi trường phát triển native, tạo điều kiện thuận lợi cho việc tích hợp sâu các tính năng native và thư viện bên thứ ba.

Mã JavaScript của ứng dụng chạy trên JavaScript thread, trong khi giao diện người dùng chạy trên UI thread native. Bridge đóng vai trò truyền tải các lệnh và sự kiện giữa hai thread này, giúp ứng dụng hoạt động mượt mà và hiệu quả.

2.1.4. Các thành phần chính

JSX (JavaScript XML) là một phần mở rộng của ngôn ngữ JavaScript, được sử dụng chủ yếu trong các thư viện và framework JavaScript như React. JSX giúp tạo ra cú pháp giống HTML để mô tả cấu trúc và giao diện người dùng trong các ứng dụng web. JSX cho phép việc viết mã JavaScript và HTML trong một file duy nhất, tạo ra mã nguồn dễ đọc và hiểu hơn.

React Component là một thư viện JavaScript được sử dụng để thiết lập giao diện người dùng. Component trong React sẽ giúp phân chia giao diện người dùng (UI) thành các thành phần nhỏ hơn để việc quản lý và tái sử dụng trở nên đơn giản. Mỗi Component sẽ đảm nhiệm từng phần hiển thị khác nhau của giao diện.

React Lifecycle là một tập hợp các phương thức được gọi trong quá trình một component React được tạo ra, cập nhật và xóa bỏ. Điều này cho phép bạn quản lý và thay đổi trạng thái của component, tương tác với DOM, cũng như gọi API và các tác vụ khác.

React Hooks là một tính năng mạnh mẽ trong React cho phép quản lý trạng thái và hiệu suất một cách dễ dàng trong các component functional. Với Hooks, bạn có thể sử dụng trạng thái, hiệu suất, và các tác vụ phụ thuộc vào trạng thái một cách tự nhiên, tạo ra mã ngắn gọn và dễ đọc hơn.

React Context là một cách để quản lý trạng thái trên toàn cầu. Nó có thể được sử dụng cùng với useState Hook để chia sẻ trạng thái giữa các thành phần lồng vào nhau dễ dàng hơn so với useState một mình.

Redux là một predictable state management tool cho các ứng dụng Javascript. Nó giúp bạn viết các ứng dụng hoạt động một cách nhất quán, chạy trong các môi trường khác nhau (client, server, and native) và dễ dàng để test.

2.2. TypeScript

2.2.1. TypeScript là gì?

TypeScript, một ngôn ngữ lập trình mã nguồn mở được phát triển và duy trì bởi Microsoft, đã nhanh chóng trở thành một công cụ không thể thiếu trong việc xây dựng các ứng dụng JavaScript quy mô lớn và phức tạp. Được giới thiệu vào năm 2012, TypeScript đã giải quyết một trong những điểm yếu lớn nhất của JavaScript: thiếu hệ thống kiểu tĩnh.

Về bản chất, TypeScript là một siêu tập của JavaScript, có nghĩa là mọi mã JavaScript hợp lệ đều có thể chạy trong TypeScript. Tuy nhiên, TypeScript bổ sung thêm một lớp quan trọng: hệ thống kiểu tĩnh. Điều này cho phép các nhà phát triển khai báo kiểu dữ liệu cho các biến, hàm và tham số, giúp phát hiện lỗi tiềm ẩn ngay trong quá trình biên dịch, trước khi mã được thực thi.

TypeScript còn hỗ trợ nhiều tính năng hướng đối tượng mạnh mẽ như lớp (class), giao diện (interface), kế thừa (inheritance) và generic. Các tính năng này giúp tổ chức và quản lý mã nguồn một cách hiệu quả, tăng tính tái sử dụng và dễ bảo trì.

Một điểm mạnh khác của TypeScript là khả năng tương thích ngược với JavaScript. TypeScript có thể sử dụng tất cả các tính năng của ECMAScript 2015

(ES6) và các phiên bản mới hơn, bao gồm các tính năng như class, module, arrow function và destructuring assignment. Điều này giúp các nhà phát triển JavaScript dễ dàng chuyển đổi sang TypeScript mà không cần phải học lại từ đầu.

Trước TypeScript, đã có một số nỗ lực nhằm cải thiện JavaScript, như CoffeeScript và Dart. Tuy nhiên, cả hai ngôn ngữ này đều sử dụng cú pháp hoàn toàn mới, khác biệt đáng kể so với JavaScript. Điều này gây khó khăn cho việc học và chuyển đổi, khiến chúng không được cộng đồng đón nhận rộng rãi.

TypeScript, ngược lại, đã chọn một hướng đi khác. Bằng cách giữ nguyên cú pháp của JavaScript và chỉ bổ sung thêm các tính năng mới, TypeScript đã tạo ra một môi trường thân thiện và dễ tiếp cận cho các nhà phát triển JavaScript. Điều này giúp TypeScript nhanh chóng trở nên phổ biến và được sử dụng rộng rãi trong các dự án lớn nhỏ.

Sự ra đời của TypeScript đã đánh dấu một bước ngoặt quan trọng trong lịch sử phát triển của JavaScript. Với hệ thống kiểu tĩnh, các tính năng hướng đối tượng mạnh mẽ và khả năng tương thích ngược với JavaScript, TypeScript đã mang đến một giải pháp hiệu quả để giải quyết các vấn đề về khả năng mở rộng, bảo trì và độ tin cậy của mã JavaScript.

Ngày nay, TypeScript đã trở thành một phần không thể thiếu trong bộ công cụ của nhiều nhà phát triển web và ứng dụng, đóng góp quan trọng vào việc xây dựng các ứng dụng JavaScript hiện đại, an toàn và hiệu quả.

2.2.2. Lý do nên sử dụng TypeScript

TypeScript không chỉ là một ngôn ngữ lập trình, mà còn là một công cụ đắc lực giúp các nhà phát triển JavaScript vượt qua những thách thức trong việc xây dựng và duy trì các dự án lớn. Với khả năng kiểm tra kiểu tĩnh, phát hiện lỗi sớm và hỗ trợ các tính năng hướng đối tượng, TypeScript đã trở thành một lựa chọn ưu tiên cho nhiều dự án phát triển phần mềm hiện đại.

Một trong những lợi ích quan trọng nhất của TypeScript là khả năng phát triển dự án lớn một cách dễ dàng hơn. Trong các dự án JavaScript quy mô lớn,

việc quản lý và tổ chức mã nguồn trở nên phức tạp hơn bao giờ hết. TypeScript, với hệ thống kiểu tĩnh, giúp các nhà phát triển xác định rõ ràng kiểu dữ liệu của các biến, hàm và tham số. Điều này giúp phát hiện các lỗi tiềm ẩn ngay trong quá trình biên dịch, trước khi mã được thực thi, tiết kiệm thời gian và công sức gỡ lỗi sau này.

TypeScript không chỉ được các nhà phát triển độc lập ưa chuộng mà còn được nhiều framework JavaScript hàng đầu như Angular và Ionic khuyến khích sử dụng. Việc các framework này tích hợp chặt chẽ với TypeScript giúp tạo ra một môi trường phát triển thống nhất và hiệu quả, đồng thời tận dụng tối đa các tính năng và lợi ích của TypeScript.

Một ưu điểm khác của TypeScript là khả năng hỗ trợ các tính năng mới nhất của JavaScript. TypeScript luôn cập nhật và tích hợp các tính năng mới của JavaScript, bao gồm cả những tính năng chưa được hỗ trợ rộng rãi trên các trình duyệt. Điều này giúp các nhà phát triển tận dụng tối đa sức mạnh của JavaScript mà không phải lo lắng về vấn đề tương thích.

TypeScript là một dự án mã nguồn mở, được phát triển và duy trì bởi Microsoft và cộng đồng. Điều này đồng nghĩa với việc có thể sử dụng TypeScript hoàn toàn miễn phí và đóng góp vào sự phát triển của nó. Hơn nữa, Microsoft cung cấp một bộ công cụ phát triển mạnh mẽ cho TypeScript, bao gồm trình biên dịch, trình gỡ lỗi và tích hợp với các IDE phổ biến như Visual Studio Code, giúp đơn giản hóa quá trình phát triển và cải thiện trải nghiệm của nhà phát triển.

Một điểm cộng lớn của TypeScript là khả năng tương thích với JavaScript. TypeScript được biên dịch thành JavaScript, do đó có thể chạy mã TypeScript trên bất kỳ môi trường nào hỗ trợ JavaScript, cũng có thể sử dụng kết hợp cú pháp của JavaScript và TypeScript trong cùng một dự án, giúp việc chuyển đổi từ JavaScript sang TypeScript trở nên dễ dàng và linh hoạt hơn.

2.2.3. Chức năng của TypeScript

TypeScript, một siêu tập của JavaScript, đã mang đến một cuộc cách mạng trong cách các nhà phát triển tiếp cận việc xây dựng ứng dụng web và di động.

Với hệ thống kiểu tĩnh và các tính năng hướng đối tượng mạnh mẽ, TypeScript không chỉ giúp cải thiện chất lượng mã nguồn mà còn tăng cường hiệu suất làm việc và giảm thiểu lỗi tiềm ẩn.

Một trong những chức năng cốt lõi và quan trọng nhất của TypeScript là Static Typing (kiểu tĩnh). Khác với JavaScript, nơi các biến có thể thay đổi kiểu dữ liệu một cách linh hoạt, TypeScript yêu cầu các nhà phát triển khai báo rõ ràng kiểu dữ liệu cho các biến, hàm và tham số. Điều này giúp TypeScript phát hiện các lỗi liên quan đến kiểu dữ liệu ngay trong quá trình biên dịch, trước khi mã được thực thi. Nhờ đó, TypeScript giúp ngăn chặn các lỗi tiềm ẩn và đảm bảo tính ổn định của ứng dụng.

Interfaces là một tính năng quan trọng khác của TypeScript, giúp định nghĩa cấu trúc của các đối tượng một cách rõ ràng và chặt chẽ. Với Interfaces, các nhà phát triển có thể xác định các thuộc tính và phương thức mà một đối tượng phải có, bao gồm cả kiểu dữ liệu và các ràng buộc khác. Điều này không chỉ giúp tăng tính dễ đọc và dễ hiểu của mã nguồn mà còn hỗ trợ kiểm tra kiểu dữ liệu một cách tự động, giảm thiểu nguy cơ lỗi do sử dụng sai kiểu dữ liệu.

TypeScript cũng hỗ trợ hệ thống lớp (Classes) tương tự như các ngôn ngữ lập trình hướng đối tượng khác như C# và Java. Với Classes, các nhà phát triển có thể định nghĩa các đối tượng với các thuộc tính và phương thức, đồng thời áp dụng các khái niệm như kế thừa, đa hình và đóng gói. Điều này giúp tổ chức mã nguồn một cách logic và dễ hiểu, tăng tính tái sử dụng và khả năng bảo trì của ứng dụng.

Module là một tính năng quan trọng khác của TypeScript, cho phép chia nhỏ mã nguồn thành các module độc lập, mỗi module có thể chứa các lớp, hàm, biến và các thành phần khác. Việc sử dụng module giúp tổ chức mã nguồn một cách rõ ràng và dễ quản lý, đồng thời tránh xung đột tên và tăng khả năng tái sử dụng mã.

Generics là một tính năng mạnh mẽ của TypeScript, cho phép viết các hàm và lớp có thể hoạt động với nhiều kiểu dữ liệu khác nhau. Generics giúp tăng tính linh hoạt và tái sử dụng của mã nguồn, đồng thời đảm bảo tính an toàn kiểu dữ

liệu. Với Generics, ta có thể viết các hàm và lớp tổng quát, có thể hoạt động với nhiều kiểu dữ liệu khác nhau mà không cần phải viết lại mã cho từng kiểu dữ liệu cụ thể.

TypeScript với những tính năng ưu việt như Static Typing, Interfaces, Classes, Modules và Generics, đã trở thành một công cụ không thể thiếu cho các nhà phát triển JavaScript hiện đại. Sự kết hợp giữa sức mạnh của JavaScript và tính an toàn của kiểu tĩnh đã giúp TypeScript trở thành một lựa chọn hàng đầu cho việc xây dựng các ứng dụng web và di động phức tạp, đáng tin cậy và dễ bảo trì.

2.2.4. Ưu điểm và hạn chế của TypeScript

Ưu điểm của TypeScript:

Tính thuận tiện và hoàn toàn miễn phí giúp các nhà phát triển tiết kiệm chi phí và dễ dàng tiếp cận.

Cung cấp cú pháp rõ ràng và gợi ý thông tin, giúp cho việc viết mã nhanh chóng và dễ hiểu hơn.

Hỗ trợ các tính năng của lập trình hướng đối tượng như lớp và giao diện, giúp tái sử dụng mã nguồn và tạo ra các đối tượng mô-đun.

Với kiểu dữ liệu tĩnh và kiểm tra lỗi trong quá trình biên dịch, TypeScript giúp giảm tỷ lệ mắc lỗi và phát hiện lỗi sớm hơn trong quá trình phát triển.

TypeScript giúp giảm boilerplate code (mã lặp lại mà không có giá trị thêm) bằng cách sử dụng các tính năng như khai báo kiểu dữ liệu và generics.

Cho phép hợp nhất mã nguồn JavaScript hiện có vào dự án TypeScript, giúp tăng tính tương thích và dễ dàng chuyển đổi từ JavaScript sang TypeScript.

Cung cấp các tính năng như autocomplete, kiểm tra lỗi và gợi ý thông tin, giúp tăng hiệu suất làm việc và giảm thời gian phát triển.

Hạn chế của TypeScript:

Yêu cầu quá trình biên dịch từ TypeScript sang JavaScript trước khi chạy được trên trình duyệt hoặc máy chủ. Điều này tạo ra bước công việc phụ và tăng thời gian phát triển.

Đòi hỏi một số thiết lập ban đầu như cấu hình tệp tsconfig.json và cài đặt trình biên dịch TypeScript. Quá trình này có thể làm phiền một số nhà phát triển.

Chỉ là phần ngôn ngữ mở rộng hỗ trợ: TypeScript là một phần mở rộng của JavaScript, điều này có nghĩa là các tính năng trong TypeScript chỉ có thể được sử dụng trong môi trường hỗ trợ TypeScript. Trình duyệt và một số môi trường chạy JavaScript không thể hiểu và chạy trực tiếp mã TypeScript.

2.3. Tổng quan về NestJs

2.3.1. Decorator trong NestJS

NestJS là một framework Node.js được xây dựng dựa trên nguyên tắc của Angular để xây dựng các ứng dụng server-side hiệu quả và dễ bảo trì. Một trong những điểm mạnh của NestJS là sự hỗ trợ của nó cho Decorator, một tính năng mạnh mẽ được kế thừa từ TypeScript.

Decorator trong NestJS cho phép thêm metadata và chức năng vào các thành phần như class, method, hoặc property. Bằng cách này, ta có thể dễ dàng định dạng và mở rộng ứng dụng của mình.

Một số Decorator phổ biến trong NestJs:

@Module: Được sử dụng để định nghĩa một module trong ứng dụng NestJS. Module là cách chính để tổ chức code trong NestJS.

@Controller: Sử dụng để định nghĩa một controller. Controllers là các thành phần chịu trách nhiệm điều hướng các yêu cầu HTTP đến các endpoint tương ứng.

@Injectable: Được sử dụng để đánh dấu một service trong NestJS. Services là các thành phần chịu trách nhiệm xử lý logic kinh doanh của ứng dụng.

@Inject: Sử dụng để chú thích các dependency và yêu cầu NestJS cung cấp chúng khi tạo instance của class.

@Get, @Post, @Put, @Delete: Được sử dụng để định nghĩa các endpoint HTTP và xác định phương thức HTTP mà endpoint đó xử lý.

@Param, @Body, @Query: Sử dụng để truy cập các tham số của request như URL parameters, request body, và query parameters.

@Guard, @Interceptor, @Middleware: Được sử dụng để thêm các logic phổ quát vào các request, chẳng hạn như xác thực, kiểm soát truy cập, hay logging.

@Res, @Req: Được sử dụng để truy cập các đối tượng Response và Request của Express trong các endpoint của NestJS.

2.3.2. Dependency Injection trong NestJs

Trong NestJS, Dependency Injection (DI) là một trong những tính năng quan trọng nhất, giúp quản lý và tổ chức các thành phần của ứng dụng một cách dễ dàng và linh hoạt. DI là một mô hình thiết kế phổ biến trong lập trình phần mềm, giúp giảm sự phụ thuộc giữa các thành phần của ứng dụng và tạo điều kiện cho việc tái sử dụng code và kiểm thử.

Injectors: Trong NestJS, có một injector tự động quản lý các dependency. Injector này có thể tìm và cung cấp các dependency mà một class hoặc một thành phần cần.

Providers: Trong NestJS, một provider là một class có thể được inject và sử dụng ở nhiều nơi khác nhau trong ứng dụng. Các providers có thể là các service, các class thực hiện logic kinh doanh, hoặc các dependency bên ngoài như các thư viện.

Injection Tokens: Mỗi provider trong NestJS được định danh bằng một Injection Token. Injection Token là một giá trị độc nhất được sử dụng để xác định provider và inject nó vào các thành phần khác.

Module Dependencies: Trong NestJS, các dependency được khai báo và quản lý trong các module. Mỗi module định nghĩa một phạm vi cho các dependency và chỉ có thể sử dụng các dependency được khai báo trong cùng một module hoặc trong các module con của nó.

Scope của Dependency: NestJS hỗ trợ các scope khác nhau cho các dependency như Singleton, Request, và Transient. Singleton là một instance duy

nhất được sử dụng trong toàn bộ ứng dụng, Request là một instance mới được tạo cho mỗi request HTTP, và Transient là một instance mới được tạo mỗi khi được inject.

Custom Providers: Bạn có thể định nghĩa các custom provider để cung cấp các dependency tùy chỉnh hoặc để thay thế các dependency mặc định của NestJS.

2.3.3. Hỗ trợ công nghệ và tích hợp CSDL

NestJS là một framework Node.js mạnh mẽ và linh hoạt, cung cấp một loạt các tính năng và công nghệ để phát triển ứng dụng server-side hiệu quả.

Hỗ trợ công nghệ:

TypeScript: NestJS được xây dựng hoàn toàn bằng TypeScript, cho phép sử dụng các tính năng nâng cao của TypeScript như kiểu dữ liệu tĩnh, interface, decorators, ...

Express.js: NestJS sử dụng Express.js làm framework HTTP server mặc định, cho phép sử dụng các middleware và plugin của Express trong ứng dụng của mình.

WebSockets: NestJS hỗ trợ WebSockets thông qua WebSocket Gateway, cho phép xây dựng các ứng dụng thời gian thực như chat hoặc game.

GraphQL: NestJS cung cấp hỗ trợ tích hợp GraphQL thông qua @nestjs/graphql, cho phép bạn xây dựng API dạng GraphQL một cách dễ dàng và mạnh mẽ.

Microservices: NestJS hỗ trợ việc xây dựng và triển khai các ứng dụng microservices thông qua Nest Microservices.

Testing: NestJS cung cấp một framework testing tích hợp để viết và chạy các unit test và end-to-end test cho ứng dụng.

Tích hợp CSDL:

TypeORM: TypeORM là một ORM (Object-Relational Mapping) cho TypeScript và JavaScript, được NestJS tích hợp sẵn. TypeORM cho phép tương tác với CSDL quan hệ một cách dễ dàng và linh hoạt.

Sequelize: NestJS cũng hỗ trợ tích hợp Sequelize, một ORM phổ biến khác, cho phép làm việc với CSDL quan hệ.

Mongoose: Đối với MongoDB, NestJS cung cấp hỗ trợ tích hợp Mongoose, một công cụ tuyệt vời cho việc tương tác với MongoDB trong ứng dụng Node.js.

Prisma: Prisma là một ORM và công cụ tạo truy vấn được yêu thích trong cộng đồng, có thể tích hợp nó vào NestJS để làm việc với CSDL quan hệ.

2.4. Tổng quan về Docker và DBeaver MySql

2.4.1. Docker

Docker, một công nghệ tiên phong trong lĩnh vực công nghệ thông tin, đã tạo nên một cuộc cách mạng trong cách chúng ta xây dựng, đóng gói và triển khai phần mềm. Với khả năng tạo ra các "container" - những môi trường độc lập và cô lập, Docker giúp đơn giản hóa quá trình triển khai ứng dụng trên nhiều hệ thống và nền tảng khác nhau.

Điểm đặc biệt của Docker nằm ở khả năng đóng gói toàn bộ ứng dụng và các thành phần phụ thuộc của nó vào một container duy nhất. Mỗi container hoạt động như một "hộp chứa" riêng biệt, chứa đựng mọi thứ cần thiết để ứng dụng chạy được, từ mã nguồn, thư viện đến các công cụ hệ thống và môi trường thực thi. Nhờ đó, ứng dụng có thể chạy một cách nhất quán và đáng tin cậy trên bất kỳ máy chủ nào có cài đặt Docker, bất kể cấu hình phần cứng và hệ điều hành của máy chủ đó.

Docker Engine, trái tim của Docker, đóng vai trò như một bộ máy quản lý container. Nó chịu trách nhiệm tạo, khởi động, dừng, xóa và giám sát các container, đồng thời cung cấp các công cụ và giao diện để người dùng tương tác với container một cách dễ dàng. Docker Engine sử dụng công nghệ ảo hóa ở cấp

độ hệ điều hành để tạo ra các container, giúp tối ưu hóa hiệu suất và giảm thiểu tài nguyên sử dụng.

Docker Image là một thành phần quan trọng khác của Docker. Nó là một bản thiết kế (blueprint) để tạo ra các container. Image chứa các lớp (layer) hệ thống tập tin, mỗi lớp đại diện cho một thay đổi đối với image. Khi chạy một image, Docker Engine sẽ tạo ra một container từ image đó. Các image có thể được xây dựng từ đầu hoặc lấy từ các Docker Registry - kho lưu trữ và chia sẻ image công cộng hoặc riêng tư.

Docker Compose là một công cụ hữu ích giúp đơn giản hóa việc quản lý các ứng dụng đa container. Với Docker Compose, có thể sử dụng một tệp YAML để định nghĩa và cấu hình các dịch vụ (service) trong ứng dụng của mình, mỗi dịch vụ tương ứng với một container. Docker Compose sẽ tự động tạo và quản lý các container này, giúp bạn dễ dàng khởi động, dừng và mở rộng ứng dụng một cách nhanh chóng và hiệu quả.

Ưu điểm của Docker:

Tính nhất quán: Đảm bảo ứng dụng hoạt động giống nhau trên mọi môi trường (máy tính cá nhân, máy chủ, đám mây) mà không gặp vấn đề tương thích.

Tính cô lập: Các container hoạt động độc lập, không ảnh hưởng lẫn nhau, giúp quản lý và bảo trì dễ dàng hơn.

Tính linh hoạt: Dễ dàng mở rộng, sao chép và di chuyển ứng dụng giữa các môi trường. Tiết kiệm tài nguyên: Các container chia sẻ tài nguyên hệ điều hành, giúp tiết kiệm bộ nhớ và CPU.

Nhược điểm của Docker:

Docker có thể khá phức tạp đối với người mới bắt đầu, đặc biệt là khi làm việc với các ứng dụng phức tạp hoặc các cấu hình nâng cao. Việc tìm hiểu và làm quen với các khái niệm như image, container, Dockerfile, Docker Compose có thể mất thời gian.

Mặc dù Docker đã được cải thiện đáng kể về hiệu suất, nhưng việc chạy ứng dụng trong container vẫn có thể chậm hơn so với chạy trực tiếp trên hệ điều hành. Điều này là do lớp ảo hóa của Docker có thể gây ra một số chi phí về hiệu suất.

Nếu không được cấu hình và quản lý đúng cách, Docker có thể tiềm ẩn các lỗ hổng bảo mật. Ví dụ, nếu sử dụng một image không đáng tin cậy hoặc không cập nhật các bản vá bảo mật, container của bạn có thể bị tấn công.

Việc quản lý dữ liệu trong Docker có thể phức tạp hơn so với quản lý dữ liệu trên hệ thống tập tin thông thường vì cần phải tìm hiểu cách sử dụng các volume Docker để lưu trữ dữ liệu một cách liên tục.

Ứng dụng của Docker:

Trong giai đoạn phát triển ứng dụng, Docker cho phép các lập trình viên tạo ra môi trường phát triển đồng nhất, loại bỏ tình trạng "trên máy tôi chạy được". Mỗi thành viên trong nhóm có thể làm việc trên một container Docker riêng biệt, chứa đựng toàn bộ môi trường cần thiết để ứng dụng hoạt động. Điều này đảm bảo tính nhất quán giữa các môi trường làm việc, giảm thiểu các xung đột và lỗi phát sinh do sự khác biệt về cấu hình phần cứng hay phần mềm.

Khi đến giai đoạn triển khai, Docker đóng vai trò quan trọng trong việc đóng gói ứng dụng và các thành phần phụ thuộc vào một container duy nhất. Container này có thể được dễ dàng chuyển giao và chạy trên bất kỳ máy chủ nào có cài đặt Docker, bất kể môi trường hay hệ điều hành. Điều này giúp giảm thiểu thời gian và công sức cần thiết để cấu hình môi trường triển khai, đồng thời đảm bảo tính nhất quán và độ tin cậy của ứng dụng trên các môi trường khác nhau.

Trong quá trình kiểm thử, Docker cho phép tạo ra các môi trường kiểm thử cô lập và an toàn. Các container Docker có thể được cấu hình với các phiên bản phần mềm và hệ điều hành khác nhau, mô phỏng các môi trường thực tế để kiểm tra ứng dụng. Điều này giúp đảm bảo rằng ứng dụng hoạt động chính xác và ổn định trong mọi tình huống, đồng thời phát hiện và khắc phục sớm các lỗi tiềm ẩn.

Khi ứng dụng cần mở rộng để đáp ứng nhu cầu tăng cao, Docker cung cấp khả năng nhân rộng linh hoạt và hiệu quả. Bạn có thể dễ dàng tạo thêm các container để chia sẻ tải và tăng khả năng xử lý của ứng dụng. Việc mở rộng này có thể được thực hiện một cách nhanh chóng và tự động, giúp đảm bảo tính sẵn sàng và hiệu suất của ứng dụng trong mọi tình huống.

Không chỉ dừng lại ở việc phát triển, triển khai, kiểm thử và mở rộng ứng dụng, Docker còn được ứng dụng rộng rãi trong nhiều lĩnh vực khác như xây dựng môi trường Continuous Integration/Continuous Delivery (CI/CD), phát triển các ứng dụng microservices và tạo các môi trường ảo hóa cho mục đích học tập và đào tạo. Với những lợi ích vượt trội và khả năng ứng dụng đa dạng, Docker đã trở thành một công cụ không thể thiếu trong thế giới công nghệ thông tin hiện đại, giúp các tổ chức và cá nhân khai thác tối đa tiềm năng của công nghệ container.

2.4.2. DBeaver MySql

DBeaver là một công cụ quản lý cơ sở dữ liệu đa nền tảng, mã nguồn mở và miễn phí. Nó hỗ trợ nhiều loại cơ sở dữ liệu khác nhau, trong đó có MySQL. DBeaver cung cấp giao diện trực quan và nhiều tính năng mạnh mẽ giúp bạn làm việc với cơ sở dữ liệu một cách dễ dàng và hiệu quả.

Kết nối đến nhiều loại cơ sở dữ liệu: DBeaver không chỉ hỗ trợ MySQL mà còn có thể kết nối đến nhiều loại cơ sở dữ liệu khác nhau như PostgreSQL, Oracle, SQL Server, SQLite, MongoDB, và nhiều loại khác. Điều này giúp quản lý nhiều cơ sở dữ liệu khác nhau trong cùng một công cụ.

Trình soạn thảo SQL mạnh mẽ: DBeaver cung cấp một trình soạn thảo SQL mạnh mẽ với các tính năng như tô sáng cú pháp, tự động hoàn thành, định dạng mã, và kiểm tra cú pháp. Điều này giúp viết và thực thi các câu truy vấn SQL một cách nhanh chóng và chính xác.

Trình quản lý đối tượng cơ sở dữ liệu: DBeaver cung cấp một trình quản lý đối tượng trực quan, cho phép dễ dàng duyệt, tạo, chỉnh sửa và xóa các đối tượng cơ sở dữ liệu như bảng, view, thủ tục lưu trữ, hàm, trigger, và các đối tượng khác.

Trình quản lý dữ liệu: DBeaver cung cấp một trình quản lý dữ liệu mạnh mẽ, cho phép bạn xem, chỉnh sửa, thêm, xóa và lọc dữ liệu trong các bảng và cũng có thể xuất dữ liệu sang các định dạng khác nhau như CSV, Excel, XML, JSON, và HTML.

Trình quản lý người dùng và phân quyền: DBeaver cho phép tạo, chỉnh sửa và xóa người dùng, cũng như cấp và thu hồi các quyền truy cập vào các đối tượng cơ sở dữ liệu.

Ưu điểm của DBeaver:

Giao diện thân thiện: DBeaver có giao diện trực quan, dễ sử dụng, giúp bạn nhanh chóng làm quen và sử dụng các tính năng của nó.

Đa nền tảng: DBeaver hoạt động trên Windows, macOS và Linux, cho phép bạn sử dụng trên nhiều hệ điều hành khác nhau.

Hỗ trợ nhiều loại cơ sở dữ liệu: DBeaver hỗ trợ nhiều loại cơ sở dữ liệu phổ biến như MySQL, PostgreSQL, Oracle, SQL Server, SQLite, MongoDB, và nhiều loại khác.

Nhiều tính năng mạnh mẽ: DBeaver cung cấp nhiều tính năng hữu ích như tạo và chỉnh sửa bảng, truy vấn dữ liệu, xuất dữ liệu, so sánh dữ liệu, quản lý người dùng, và nhiều tính năng khác.

Nhược điểm của Dbeaver:

DBeaver là một ứng dụng Java, do đó có thể tiêu tốn khá nhiều bộ nhớ và CPU, đặc biệt là khi làm việc với các cơ sở dữ liệu lớn hoặc thực hiện các truy vấn phức tạp.

Mặc dù DBeaver cung cấp nhiều tính năng, nhưng khả năng tùy biến của nó có thể hạn chế so với các công cụ khác.

Mặc dù DBeaver là một dự án mã nguồn mở, nhưng cộng đồng hỗ trợ của nó có thể không lớn bằng các công cụ thương mại khác dẫn tới việc có thể khó tìm được sự trợ giúp khi gặp vấn đề.

Ứng dụng của DBeaver:

Quản lý cơ sở dữ liệu MySQL: DBeaver giúp bạn dễ dàng kết nối, quản lý và thao tác với cơ sở dữ liệu MySQL.

Thiết kế cơ sở dữ liệu: DBeaver cung cấp các công cụ trực quan để bạn thiết kế mô hình dữ liệu, tạo bảng, tạo mối quan hệ giữa các bảng.

Truy vấn và phân tích dữ liệu: DBeaver hỗ trợ viết và thực thi các câu truy vấn SQL phức tạp, giúp bạn phân tích và trích xuất thông tin từ cơ sở dữ liệu.

Quản trị cơ sở dữ liệu: DBeaver cung cấp các công cụ để bạn quản lý người dùng, phân quyền, sao lưu và phục hồi cơ sở dữ liệu.

Tiểu kết:

Chương 2 đã trình bày một cách tổng quan về các công nghệ và kiến thức nền tảng cần thiết cho việc phát triển ứng dụng giới thiệu đặc sản vùng miền.

React Native được lựa chọn làm framework chính để xây dựng ứng dụng di động đa nền tảng, tận dụng khả năng tái sử dụng mã và hiệu suất gần native.

TypeScript được sử dụng để tăng cường tính an toàn và khả năng bảo trì của mã nguồn JavaScript.

NestJS cung cấp một kiến trúc mạnh mẽ và các tính năng hữu ích để xây dựng backend của ứng dụng.

Cuối cùng, Docker và DBeaver MySQL được sử dụng để tạo môi trường phát triển và quản lý cơ sở dữ liệu một cách hiệu quả.

2.5. Forward port và chế độ bảo mật

2.5.1. Lý do cần forward port

Forward port 8080 hay bất kỳ cổng nào khác là một kỹ thuật mạng quan trọng cho phép truy cập vào các ứng dụng và dịch vụ chạy trên máy tính từ xa. Trong trường hợp của NestJS việc forward port 8080 là cách để mở một "cánh cửa" trên router cho phép các yêu cầu từ bên ngoài mạng nội bộ có thể tiếp cận ứng dụng.

Một trong những lý do chính để forward port 8080 là để truy cập ứng dụng NestJS từ xa. Khi phát triển ứng dụng trên máy tính cá nhân ứng dụng thường chạy trên một cổng cụ thể ví dụ như 8080. Tuy nhiên, cổng này chỉ có thể được truy cập từ trong mạng nội bộ. Bằng cách forward port 8080, ta có thể truy cập ứng dụng NestJS từ bất kỳ đâu có kết nối internet, ví dụ như từ điện thoại hoặc một máy tính khác.

Ngoài ra, forward port 8080 còn giúp vượt qua những hạn chế của tường lửa. Trong một số trường hợp, tường lửa của mạng có thể chặn các cổng cụ thể bao gồm cả cổng 8080. Việc forward port cho phép chuyển tiếp các yêu cầu đến một cổng khác không bị chặn từ đó có thể truy cập ứng dụng NestJS một cách bình thường.

Forward port 8080 cũng là một cách hữu ích để chia sẻ ứng dụng NestJS với người khác. Bằng cách cung cấp địa chỉ IP công cộng của mình và cổng đã được forward, bạn bè hoặc đồng nghiệp có thể truy cập và trải nghiệm ứng dụng từ xa.

Khi triển khai ứng dụng NestJS lên một máy chủ, việc forward port 8080 là một bước quan trọng để cho phép người dùng truy cập ứng dụng từ internet. Bằng cách forward port 8080 (hoặc cổng mà ứng dụng đang chạy) trên máy chủ có thể đảm bảo rằng ứng dụng của mình có thể được truy cập từ bất kỳ đâu, mở rộng phạm vi tiếp cận của ứng dụng đến với người dùng toàn cầu.

2.5.2. Ý nghĩa

Việc lựa chọn cổng 8080 cho dự án NestJS không phải là ngẫu nhiên mà ẩn chứa nhiều ý nghĩa và lợi ích thiết thực, góp phần tối ưu hóa quá trình phát triển và triển khai ứng dụng.

Đầu tiên, cổng 8080 đã trở thành một lựa chọn quen thuộc và phổ biến trong cộng đồng phát triển web. Nhiều ứng dụng và framework không chỉ riêng NestJS thường sử dụng cổng này trong quá trình phát triển. Điều này tạo nên sự thuận tiện cho các lập trình viên giúp họ dễ dàng ghi nhớ và truy cập vào ứng dụng một cách nhanh chóng mà không cần phải loay hoay với những con số phức tạp.

Thứ hai, việc sử dụng cổng 8080 giúp tránh xung đột với các dịch vụ web khác. Thông thường, cổng 80 là cổng mặc định cho giao thức HTTP được sử dụng rộng rãi bởi các máy chủ web. Bằng cách chọn cổng 8080 cho ứng dụng NestJS, đảm bảo rằng ứng dụng của mình có thể hoạt động độc lập không bị ảnh hưởng hay xung đột với các dịch vụ web khác đang chạy trên cùng một máy chủ.

Một ưu điểm khác của việc sử dụng cổng 8080 là tính linh hoạt và khả năng tùy biến cao. Mặc dù là một lựa chọn phổ biến, hoàn toàn có thể thay đổi cổng này nếu cần thiết. NestJS cho phép nhà phát triển dễ dàng cập nhật cấu hình dự án để sử dụng một cổng khác hoặc sử dụng biến môi trường để linh hoạt thay đổi cổng trong quá trình triển khai ứng dụng. Điều này mang lại sự tự do và khả năng điều chỉnh ứng dụng phù hợp với các yêu cầu cụ thể của dự án.

Hơn nữa, việc sử dụng cổng 8080 trong quá trình phát triển còn giúp phân biệt rõ ràng giữa môi trường phát triển và môi trường production (sản phẩm). Trong môi trường production có thể sử dụng cổng 80 (HTTP) hoặc 443 (HTTPS) để ứng dụng có thể được truy cập trực tiếp từ internet. Sự phân biệt này giúp tránh nhầm lẫn và đảm bảo tính an toàn, ổn định của ứng dụng khi triển khai.

Cuối cùng, cổng 8080 mang lại sự tiện lợi cho việc debug và kiểm thử ứng dụng. Trong quá trình phát triển có thể dễ dàng kiểm tra và gỡ lỗi ứng dụng bằng cách truy cập vào địa chỉ http://localhost:8080 trên trình duyệt. Điều này giúp nhanh chóng xem kết quả của những thay đổi trong mã nguồn và phát hiện các lỗi tiềm ẩn một cách thuận tiện.

2.5.3. Hạn chế

Mặc dù cổng 8080 mang lại nhiều lợi ích trong quá trình phát triển ứng dụng NestJS, việc sử dụng nó không phải là không có những hạn chế nhất định. Nhận thức rõ những hạn chế này sẽ giúp các nhà phát triển đưa ra quyết định sáng suốt và có những biện pháp phòng ngừa thích hợp.

Một trong những hạn chế đáng chú ý nhất của việc sử dụng cổng 8080 là nó không phải là cổng tiêu chuẩn cho giao thức HTTP. Thông thường, người dùng chỉ cần nhập tên miền của trang web để truy cập, trong khi cổng 80 được mặc

định sử dụng. Tuy nhiên, với cổng 8080 người dùng sẽ phải nhập thêm ":8080" vào cuối địa chỉ URL. Điều này có thể gây ra sự bất tiện và khó khăn cho những người dùng không quen thuộc với việc nhập cổng thủ công.

Hạn chế tiếp theo liên quan đến vấn đề bảo mật. Việc forward port 8080, nếu không được cấu hình đúng cách, có thể tạo ra một "lỗ hổng" trên hệ thống của bạn, cho phép các truy cập trái phép từ bên ngoài. Tường lửa có thể không tự động bảo vệ cổng 8080 như cách nó làm với cổng 80. Do đó, cần phải thực hiện các biện pháp bảo mật bổ sung như sử dụng tường lửa, xác thực và mã hóa mạnh mẽ để đảm bảo an toàn cho ứng dụng và dữ liệu của mình.

Một vấn đề khác có thể phát sinh khi có nhiều ứng dụng chạy trên cùng một máy tính. Việc sử dụng các cổng khác nhau cho mỗi ứng dụng, bao gồm cả cổng 8080, có thể gây nhầm lẫn và khó nhớ, sẽ phải ghi nhớ cổng của từng ứng dụng để truy cập vào chúng, gây ra sự bất tiện và mất thời gian.

Ngoài ra, việc triển khai ứng dụng sử dụng cổng 8080 trên các môi trường chia sẻ cũng có thể gặp khó khăn. Một số dịch vụ hosting giá rẻ có thể không cho phép forward port 8080, gây trở ngại cho việc triển khai ứng dụng NestJS và yêu cầu phải tìm các giải pháp thay thế khác.

Việc sử dụng cổng 8080 có thể ảnh hưởng đến khả năng SEO của trang web. Các công cụ tìm kiếm như Google thường ưu tiên các trang web sử dụng cổng 80 (HTTP) hoặc 443 (HTTPS) làm cổng mặc định. Việc sử dụng cổng 8080 có thể khiến trang web khó được tìm thấy trên các công cụ tìm kiếm gây ảnh hưởng đến lượng truy cập và hiệu quả tiếp cận người dùng.

2.5.4. Chế độ bảo mật

Việc forward port 8080, hay bất kỳ cổng nào khác để truy cập ứng dụng NestJS từ xa mang lại nhiều tiện ích, nhưng cũng tiềm ẩn những rủi ro bảo mật nếu không được thực hiện đúng cách. Để đảm bảo an toàn cho ứng dụng và dữ liệu việc áp dụng các biện pháp bảo mật là vô cùng quan trọng.

Đầu tiên sử dụng HTTPS để mã hóa lưu lượng truy cập giữa máy khách và máy chủ. HTTPS sử dụng giao thức SSL/TLS để mã hóa dữ liệu ngăn chặn việc nghe lén và giả mạo thông tin. Bằng cách này có thể bảo vệ dữ liệu nhạy cảm của người dùng và đảm bảo tính toàn vẹn của thông tin được truyền tải.

Tiếp theo, cấu hình tường lửa trên router và máy chủ là một bước không thể bỏ qua. Tường lửa đóng vai trò như một lớp bảo vệ, kiểm soát chặt chẽ lưu lượng truy cập vào và ra khỏi mạng. Hãy thiết lập các quy tắc tường lửa để chỉ cho phép các truy cập đến từ các địa chỉ IP và cổng cụ thể, đồng thời chặn các truy cập không mong muốn từ bên ngoài.

Bên cạnh đó, việc xác thực và phân quyền chặt chẽ cũng rất quan trọng. Chỉ những người dùng được ủy quyền mới nên có quyền truy cập vào các tài nguyên và chức năng nhạy cảm của ứng dụng. Điều này giúp ngăn chặn việc truy cập trái phép và bảo vệ dữ liệu khỏi những kẻ xấu.

Xử lý lỗi đầu vào cũng là một yếu tố quan trọng trong việc đảm bảo an ninh cho ứng dụng. Luôn kiểm tra và xác thực tất cả các dữ liệu đầu vào từ người dùng để ngăn chặn các cuộc tấn công như SQL injection, cross-site scripting (XSS) và các hình thức tấn công khác.

Đồng thời, việc cập nhật thường xuyên các thư viện và framework mà ứng dụng sử dụng cũng rất cần thiết. Các bản cập nhật thường chứa các bản vá lỗi bảo mật quan trọng giúp bảo vệ ứng dụng của bạn khỏi các lỗ hổng bảo mật mới nhất.

Việc giám sát hoạt động và ghi nhật ký chi tiết giúp phát hiện sớm các hoạt động bất thường hoặc các dấu hiệu của cuộc tấn công. Điều này cho phép bạn phản ứng kịp thời và ngăn chặn các cuộc tấn công trước khi chúng gây ra thiệt hại nghiêm trọng.

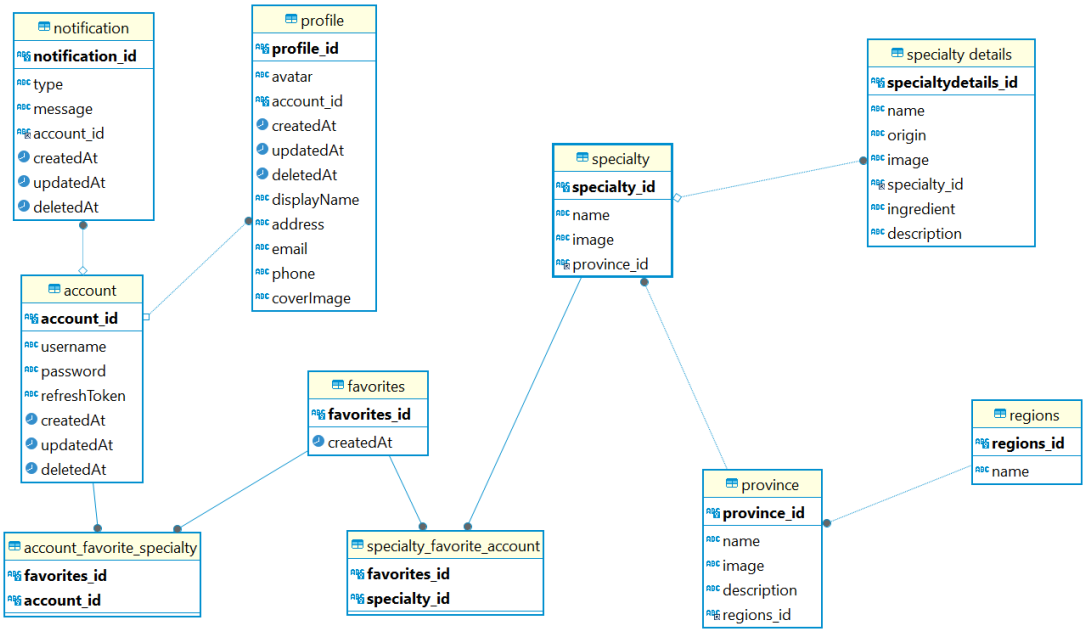
Đặc biệt lưu ý rằng, không nên forward cổng 8080 trực tiếp từ internet vào máy tính cá nhân để tránh nguy cơ bị tấn công. Thay vào đó, có thể sử dụng các biện pháp bảo mật bổ sung như VPN hoặc SSH tunneling để tạo ra một đường hầm an toàn giữa máy tính và máy chủ.

Chương 3

THIẾT KẾ HỆ THỐNG VÀ KẾT QUẢ ĐẠT ĐƯỢC

3.1. Sơ đồ usecase

3.1.1. Mô hình thực thể - quan hệ (Entity - Relationship Diagram)



Hình 3.1. Mô hình thực thể - quan hệ (Entity – Relationship Diagram)

Các thực thể (Entities):

Account: Lưu trữ các thông tin về người dùng (username, password, refreshToken, createdAt, updatedAt, deletedAt).

Profile: Chứa thông tin chi tiết về hồ sơ người dùng (avatar, displayName, address, email, phone, coverImage).

Specialty: Thông tin về đặc sản (name, image, province_id).

SpecialtyDetails: Thông tin chi tiết về đặc sản (origin, image, ingredients, description).

Province: Thông tin về tỉnh thành (name, image, description, regions_id). Regions: Thông tin về vùng miền (name).

Favorites: Lưu trữ thông tin về đặc sản yêu thích của người dùng.

Notification: Thông tin về các thông báo gửi đến người dùng.

Các mối quan hệ (Relationships):

Account - Profile: Mối quan hệ 1-1. Mỗi tài khoản có một hồ sơ và ngược lại một hồ sơ tương ứng với một tài khoản.

Account - Favorites: Mối quan hệ 1-N. Một tài khoản có thể có nhiều đặc sản yêu thích.

Specialty - SpecialtyDetails: Mối quan hệ 1-1. Mỗi đặc sản có một thông tin chi tiết và ngược lại.

Specialty - Province: Mối quan hệ N-1. Một đặc sản thuộc về một tỉnh thành.

Province - Regions: Mối quan hệ N-1. Một tỉnh thành thuộc về một vùng miền.

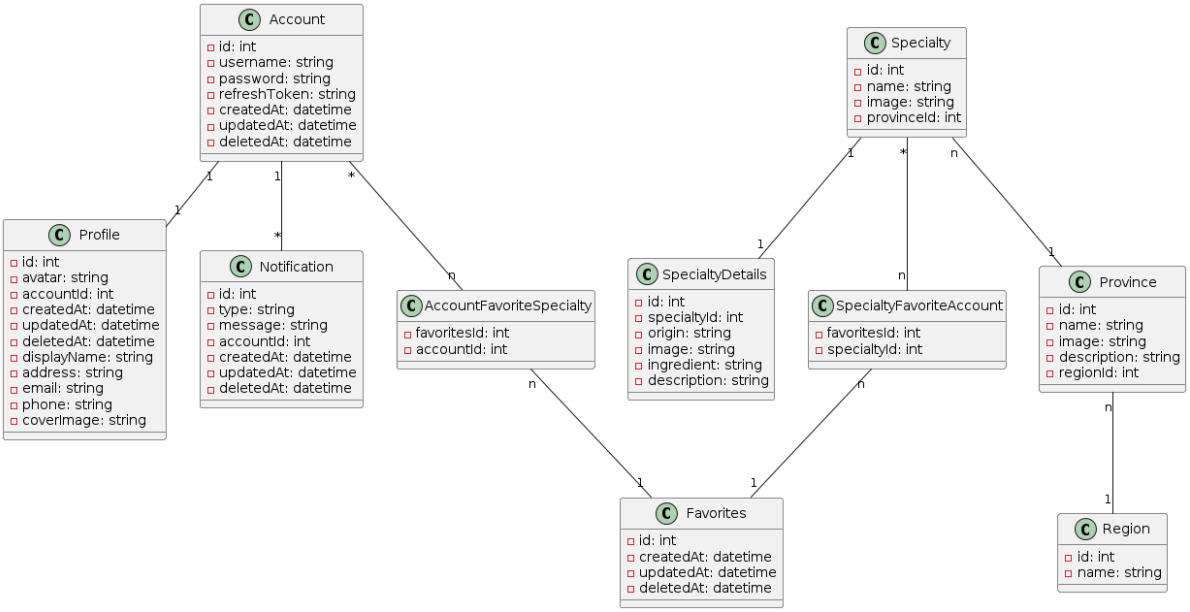
Account - Notification: Mối quan hệ 1-N. Một tài khoản có thể nhận nhiều thông báo.

Các bảng trung gian (Junction tables):

AccountFavoriteSpecialty: Lưu trữ mối quan hệ giữa Account và Favorites.

SpecialtyFavoriteAccount: Lưu trữ mối quan hệ giữa Specialty và Favorites.

3.1.2. Biểu đồ lớp (Class Diagram)



Hình 3.2. Biểu đồ lớp (Class Diagram)

Các lớp (Classes):

Account: Đại diện cho thông tin tài khoản người dùng, bao gồm: Thuộc tính: id, username, password, refreshToken, createdAt, updatedAt, deletedAt.

Profile: Chứa thông tin chi tiết về hồ sơ người dùng: Thuộc tính: id, avatar, accountId, createdAt, updatedAt, deletedAt, displayName, address, email, phone, coverImage.

Specialty: Đại diện cho thông tin về đặc sản: Thuộc tính: id, name, image, provinceId.

SpecialtyDetails: Chứa thông tin chi tiết về đặc sản: Thuộc tính: id, specialtyId, origin, image, ingredients, description.

Province: Đại diện cho thông tin về tỉnh thành: Thuộc tính: id, name, image, description, regionId.

Region: Đại diện cho thông tin về vùng miền: Thuộc tính: id, name.

Favorites: Lưu trữ thông tin về đặc sản yêu thích của người dùng: Thuộc tính: id, createdAt, updatedAt, deletedAt.

Notification: Đại diện cho thông tin về các thông báo gửi đến người dùng: Thuộc tính: id, type, message, accountId, createdAt, updatedAt, deletedAt.

Các mối quan hệ (Relationships):

Account - Profile: Mối quan hệ 1-1 (một tài khoản chỉ có một hồ sơ và ngược lại).

Account - Favorites: Mối quan hệ 1-N (một tài khoản có thể có nhiều đặc sản yêu thích).

Specialty - SpecialtyDetails: Mối quan hệ 1-1 (một đặc sản chỉ có một thông tin chi tiết và ngược lại).

Specialty - Province: Mối quan hệ N-1 (một đặc sản thuộc về một tỉnh thành).

Province - Region: Mối quan hệ N-1 (một tỉnh thành thuộc về một vùng miền).

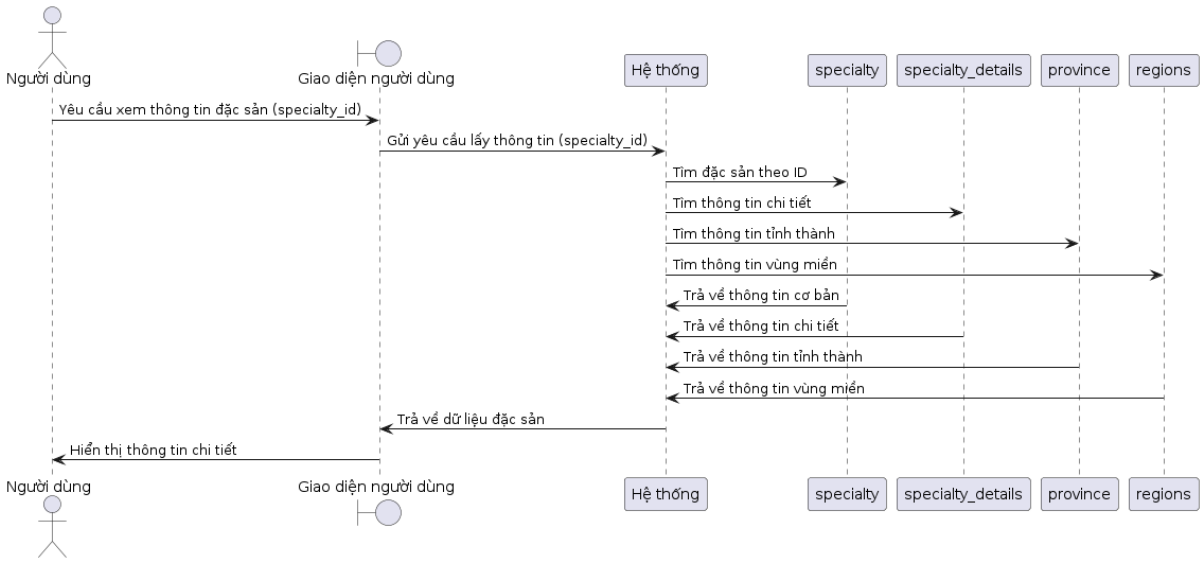
Account - Notification: Mối quan hệ 1-N (một tài khoản có thể nhận nhiều thông báo).

Các bảng trung gian (Junction tables):

AccountFavoriteSpecialty: Lưu trữ mối quan hệ giữa Account và Favorites. Thuộc tính: id, favoriteId, accountId.

SpecialtyFavoriteAccount: Lưu trữ mối quan hệ giữa Specialty và Favorites. Thuộc tính: id, favoriteId, specialtyId.

3.1.3. Biểu đồ tuần tự (Sequence Diagram)



Hình 3.3. Biểu đồ tuần tự (Sequence Diagram)

Người dùng yêu cầu xem thông tin đặc sản: Người dùng tương tác với giao diện người dùng (UI) để chọn một đặc sản cụ thể mà họ muốn xem thông tin.

UI gửi yêu cầu lấy thông tin: Giao diện người dùng gửi yêu cầu đến hệ thống, cung cấp ID của đặc sản (specialtyId) mà người dùng đã chọn.

Hệ thống tìm đặc sản theo ID: Hệ thống nhận yêu cầu và tìm kiếm thông tin về đặc sản tương ứng với specialtyId trong cơ sở dữ liệu.

Hệ thống tìm thông tin chi tiết, tỉnh thành, vùng miền: Sau khi tìm thấy đặc sản, hệ thống tiếp tục tìm kiếm thông tin chi tiết của đặc sản (specialtyDetails), thông tin về tỉnh thành (province) mà đặc sản thuộc về, và thông tin về vùng miền (region) mà tỉnh thành đó thuộc về.

Hệ thống trả về thông tin: Hệ thống trả về các thông tin đã tìm được (thông tin cơ bản của đặc sản, thông tin chi tiết, thông tin tỉnh thành, thông tin vùng miền) cho giao diện người dùng.

UI hiển thị thông tin chi tiết: Giao diện người dùng nhận dữ liệu từ hệ thống và hiển thị thông tin chi tiết của đặc sản, bao gồm cả thông tin về tỉnh thành và vùng miền liên quan, cho người dùng xem.

Các thành phần trong biểu đồ:

Người dùng: Người sử dụng ứng dụng.

Giao diện người dùng: Phần giao diện mà người dùng tương tác để xem thông tin đặc sản.

Hệ thống: Phần xử lý logic và truy xuất dữ liệu của ứng dụng.

Các thực thể (specialty, specialtyDetails, province, regions): Các bảng trong cơ sở dữ liệu lưu trữ thông tin về đặc sản, chi tiết đặc sản, tỉnh thành và vùng miền.

3.1.4. Biểu đồ phân cấp chức năng (Functional Decomposition Diagram)

Hình 3.4. Biểu đồ phân cấp chức năng (Functional Decomposition Diagram)

Quản lý hệ thống:

Đây là chức năng cốt lõi, chịu trách nhiệm quản lý tổng thể ứng dụng. Các chức năng con bao gồm:

Quản lý tài khoản: Quản lý thông tin người dùng, đăng ký, đăng nhập.

Quản lý đặc sản: Thêm, sửa, xóa, cập nhật thông tin về đặc sản.

Quản lý tỉnh thành: Thêm, sửa, xóa, cập nhật thông tin về tỉnh thành.

Quản lý vùng miền: Thêm, sửa, xóa, cập nhật thông tin về vùng miền.

Quản lý thông báo: Gửi thông báo đến người dùng về các sự kiện, khuyến mãi, cập nhật ứng dụng.

Quản lý người dùng:

Chức năng này tập trung vào việc quản lý thông tin và hoạt động của người dùng trên ứng dụng. Các chức năng con bao gồm:

Xem thông tin cá nhân: Người dùng có thể xem và chỉnh sửa thông tin cá nhân của mình.

Xem danh sách đặc sản yêu thích: Người dùng có thể xem danh sách các đặc sản đã lưu vào mục yêu thích.

Xem danh sách thông báo: Người dùng có thể xem các thông báo được hiển thị từ hệ thống.

Quản lý đặc sản:

Chức năng này tập trung vào việc quản lý thông tin về đặc sản trên ứng dụng. Các chức năng con bao gồm:

Xem danh sách đặc sản: Người dùng có thể xem danh sách tất cả các đặc sản có trên ứng dụng.

Xem chi tiết đặc sản: Người dùng có thể xem thông tin chi tiết về một đặc sản cụ thể (nguồn gốc, hình ảnh, thành phần, cách chế biến, ...).

Tìm kiếm đặc sản: Người dùng có thể tìm kiếm đặc sản theo tên, vùng miền, tỉnh thành.

Quản lý tỉnh thành: Chức năng này tập trung vào việc quản lý thông tin về tỉnh thành trên ứng dụng.

Các chức năng con bao gồm: Xem danh sách tỉnh thành: Người dùng có thể xem danh sách tất cả các tỉnh thành có trên ứng dụng.

Xem danh sách đặc sản theo tỉnh thành: Người dùng có thể xem danh sách các đặc sản thuộc một tỉnh thành cụ thể.

Quản lý vùng miền: Chức năng này tập trung vào việc quản lý thông tin về vùng miền trên ứng dụng.

Các chức năng con bao gồm: Xem danh sách vùng miền: Người dùng có thể xem danh sách tất cả các vùng miền có trên ứng dụng. Xem danh sách tỉnh thành theo vùng miền: Người dùng có thể xem danh sách các tỉnh thành thuộc

một vùng miền cụ thể. Xem danh sách đặc sản theo vùng miền: Người dùng có thể xem danh sách các đặc sản thuộc một vùng miền cụ thể.

3.2. Thiết kế cơ sở dữ liệu

3.2.1. Bảng Regions

Bảng 3.1.Bảng Regions

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	Regions_id	Varchar(36)	Id vùng miền (Primary Key)
2	Name	Varchar(255)	Tên vùng miền

Bảng Regions chứa các thông tin về Id vùng miền là khóa chính được sinh ra ngẫu nhiên trong quá trình thêm mới ngoài ra còn có tên vùng miền, ở đây sẽ là 3 miền cơ bản của Việt Nam bao gồm Miền Bắc, Miền Trung và Miền Nam.

3.2.2. Bảng Province

Bảng 3.2.Bảng Province

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	province_id	Varchar(36)	Id tỉnh (Primary Key)
2	Name	Varchar(255)	Tên tỉnh
3	Image	Varchar(255)	Hình ảnh tỉnh
4	Description	Varchar(255)	Mô tả tỉnh
5	Regions_id	Varchar(36)	Id vùng miền (Foreign Key)

Bảng Province chứa các thông tin về Id tỉnh là khóa chính được sinh ra ngẫu nhiên trong quá trình thêm mới. Ngoài ra còn có tên của 63 tỉnh thành của Việt Nam, hình ảnh đặc trưng của mỗi tỉnh, mô tả một vài điểm nổi bật về tỉnh đó. Regions_id là khóa ngoại liên kết tới bảng Regions.

3.2.3. Bảng Specialty

Bảng 3.3.Bảng Specialty

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	Specialty_id	Varchar(36)	Id đặc sản (Primary Key)
2	Name	Varchar(255)	Tên đặc sản
3	Image	Varchar(255)	Hình ảnh đặc sản
4	Province_id	Varchar(36)	Id tỉnh (Foreign Key)

Bảng Specialty chứa các thông tin về Id đặc sản là khóa chính được sinh ra trong quá trình thêm mới đặc sản. Bảng còn chứa các thông tin như là tên của đặc sản, hình ảnh về đặc sản và province_id là khóa ngoại liên kết tới bảng Province.

3.2.4. Bảng Specialty Details

Bảng 3.4.Bảng Specialty Details

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	Specialtydetails_id	Varchar(36)	Id chi tiết đặc sản (Primary Key)
2	Name	Varchar(255)	Tên chi tiết đặc sản
3	Image	Varchar(255)	Hình ảnh chi tiết đặc sản

4	Ingredient	longtext	Nguyên liệu làm nên đặc sản
5	Origin	Varchar(255)	Xuất xứ của đặc sản
6	Description	longtext	Mô tả chi tiết đặc sản
7	Specialty_id	Varchar(36)	Id đặc sản (Foreign Key)

Bảng Specialty Details chứa thông tin về: Id chi tiết đặc sản là khóa chính được sinh ra ngẫu nhiên trong quá trình thêm mới, tên của đặc sản, hình ảnh của đặc sản, xuất xứ, nguyên liệu, cách chế biến và mô tả chi tiết cho một đặc sản, specialty_id là khóa phụ liên kết tới bảng specialty.

3.2.5. Bảng Favorites

Bảng 3.5. Bảng Favorites

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	Favorites_id	Varchar(36)	Id Yêu thích (Primary Key)
2	createAt	Datetime	Thời gian tạo mục yêu thích

Bảng Favorites chứa các thông tin là Id yêu thích được sinh ra ngẫu nhiên khi người dùng yêu thích đặc sản và lưu lại thời gian người dùng nhấn vào yêu thích đặc sản nào đó.

3.2.6. Bảng Account

Bảng 3.6.Bảng Account

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	account_id	Varchar(36)	Id tài khoản (Primary Key)
2	Username	Varchar(255)	Tên đăng nhập
3	Password	Varchar(255)	Mật khẩu
4	refreshToken	Varchar(255)	Mã token làm mới
5	createAt	Datetime	Thời gian tạo tài khoản
6	updateAt	Datetime	Thời gian cập nhật tài khoản
7	deleteAt	Datetime	Thời gian xóa tài khoản (null nếu đang là chưa xóa)

Bảng Account chứa các thông tin: Id tài khoản là khóa chính được sinh ra ngẫu nhiên trong quá trình tạo mới tài khoản, username và password là 2 trường giúp người dùng có thể lấy đó làm cái đăng nhập vào ứng dụng, refreshToken là mã Token được làm mới sinh ra ngẫu nhiên, createAt và updateAt là 2 trường thời gian tạo và cập nhật tài khoản, deleteAt là thời gian xóa tài khoản nếu tài khoản vẫn còn trạng thái hoạt động thì sẽ ở dạng null còn ở trạng thái đã xóa thì sẽ hiện thời gian xóa tài khoản.

3.2.7. Bảng Profile

Bảng 3.7.Bảng Profile

STT	Tên trường	Kiểu dữ liệu	Mô tả

1	profile_id	Varchar(36)	Id hồ sơ người dùng (Primary Key)
2	displayName	Varchar(255)	Tên người dùng
3	Address	Varchar(255)	Địa chỉ
4	Email	Varchar(255)	Địa chỉ email
5	Phone	Varchar(255)	Số điện thoại
6	Avatar	Varchar(255)	Hình ảnh người dùng
7	coverImage	Varchar(255)	Ảnh bìa người dùng
8	Account_id	Varchar(36)	Id tài khoản (Foreign Key)

Bảng Profile chứa các thông tin: Id hồ sơ người dùng là khóa chính được sinh ra ngẫu nhiên từ quá trình người dùng tạo tài khoản và nhập các thông tin cá nhân. Các trường displayName, Address, Email, Phone, Avatar, coverImage là các trường lưu trữ thông tin cá nhân của người dùng khi tạo tài khoản nhập vào. Account_id là khóa phụ liên kết tới bảng Account.

3.2.8. Bảng Notification

Bảng 3.8.Bảng Notification

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	notification_id	Varchar(36)	Id thông báo (Primary Key)
2	Type	Varchar(255)	Loại thông báo
3	Message	Varchar(255)	Nội dung thông báo
4	createAt	Datetime	Thời gian tạo thông báo

5	updateAt	Datetime	Thời gian cập nhật thông báo
6	deleteAt	Datetime	Thời gian xóa thông báo (null nếu đang là chưa xóa)
5	Account_id	Varchar(36)	Id tài khoản (Foreign Key)

Bảng Notification chứa các trường thông tin: Id thông báo là khóa chính được sinh ra ngẫu nhiên trong quá trình thêm mới thông báo. Type là loại thông báo mà ứng dụng sẽ gửi tới người dùng ví dụ như việc người dùng thêm hoặc xóa đặc sản yêu thích thành công hoặc thất bại, sự kiện tại địa phương đã đang và sẽ diễn ra mà người dùng quan tâm, message là trường thông tin chứa nội dung thông báo mà ứng dụng gửi tới người dùng. Ngoài ra còn có thời gian tạo, cập nhạt và xóa thông báo, account_id là khóa phụ liên kết tới bảng account.

3.2.9. Bảng Account_favorite_specialty

Bảng 3.9.Bảng Account_favorite_specialty

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	favorites_id	Varchar(36)	Id yêu thích (Foreign Key)
2	Account_id	Varchar(36)	Id tài khoản (Foreign Key)

Bảng Account_favorite_specialty là bảng trung gian giữa 2 bảng account và favorites chứa các thông tin về Id yêu thích và Id account.

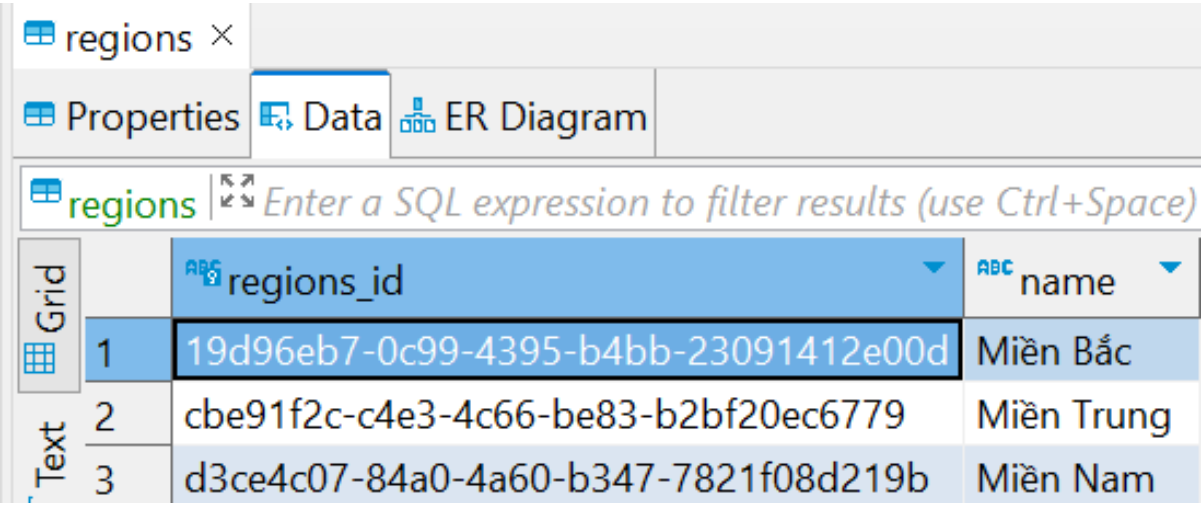
3.2.10. Bảng Specialty_favorite_account

Bảng 3.10.Bảng Specialty_favorite_account

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	favorites_id	Varchar(36)	Id yêu thích (Foreign Key)
2	Specialty_id	Varchar(36)	Id đặc sản (Foreign Key)

Bảng Specialty_favorite_account là bảng trung gian giữa 2 bảng specialty và bảng favorites chứa các thông tin về Id yêu thích và Id đặc sản.

3.3. Dữ liệu của một số bảng



Hình 3.5. Dữ liệu của bảng Regions

province ×

Properties | Data | ER Diagram — specialty_database | Databases | specialty_databas

province — Enter a SQL expression to filter results (use Ctrl+Space)

	province_id	name	image		regions_id
1	b30aa4b9-87e9-40d6-af78-86e057b5b9a7	abab	s	saa	19d96eb7-0c99-4395-b4bb-23091412e00d
2	95734619-fd71-48d1-9551-afd2bff79dd1	An Giang	https://imgur.com/on4KkDT.png	Ché	d3ce4c07-84a0-4a60-b347-7821f08d219b
3	1ed1fdb4-a2e3-4aae-959a-ce77ccf446f0	Bà Rịa-Vũng Tàu	https://imgur.com/QjkyUOG.png	Bà	d3ce4c07-84a0-4a60-b347-7821f08d219b
4	3e2c7f10-fba9-4aeb-bb05-04db6c1cf704	Bắc Giang	https://imgur.com/w64D2cl.png	.	19d96eb7-0c99-4395-b4bb-23091412e00d
5	ea071147-2e85-46e3-ae67-7359ec43366d	Bắc Kạn	https://imgur.com/oIOMXk6.png	.	19d96eb7-0c99-4395-b4bb-23091412e00d
6	e34e5682-0d84-46fd-af41-84cf07c45f66	Bạc Liêu	https://imgur.com/5qQVfaq.png	.	d3ce4c07-84a0-4a60-b347-7821f08d219b
7	e80fa517-09e5-4a5c-8ffa-65b8ac6eb3df	Bắc Ninh	https://imgur.com/K02kkio.png	.	19d96eb7-0c99-4395-b4bb-23091412e00d
8	404422ca-55db-4a73-b0fa-c1bbb144d6c6	Bến Tre	https://imgur.com/491mE2v.png	.	d3ce4c07-84a0-4a60-b347-7821f08d219b
9	261d253e-4829-4d39-a7da-be517ec37b59	Bình Định	https://imgur.com/tXLja3k.png	.	cbe91f2c-c4e3-4c66-be83-b2bf20ec6779
10	6613776d-dd33-415b-bea7-41fe224d77cb	Bình Dương	https://imgur.com/KREvRZl.png	.	d3ce4c07-84a0-4a60-b347-7821f08d219b
11	a1083e24-1fd7-45d3-ae74-723222f684fc	Bình Phước	https://imgur.com/EzVgdPo.png	.	d3ce4c07-84a0-4a60-b347-7821f08d219b
12	05b562a8-6747-4094-9c76-fbc79206a661	Bình Thuận	https://imgur.com/nEwXJlG.png	.	cbe91f2c-c4e3-4c66-be83-b2bf20ec6779
13	df241a07-f5c0-4c55-827f-a3bd9346df4c	Cà Mau	https://imgur.com/DdWjIS5.png	.	d3ce4c07-84a0-4a60-b347-7821f08d219b
14	660dc43a-2e9a-4e1d-9e91-03517f4c3c55	Cần Thơ	https://imgur.com/8D8vuQe.png	.	d3ce4c07-84a0-4a60-b347-7821f08d219b
15	e6b880d0-bcf8-411c-9906-74bbc1870dbb	Cao Bằng	https://imgur.com/8WyDp3k.png	.	19d96eb7-0c99-4395-b4bb-23091412e00d
16	497072d5-5a29-4bb2-88db-e40c280263df	Đà Nẵng	https://imgur.com/pgxSkuy.png	.	cbe91f2c-c4e3-4c66-be83-b2bf20ec6779
17	5eacfe87-359b-41e2-83f2-96fb08a7cdd3	Đắk Lắk	https://imgur.com/lNNE54k.png	.	cbe91f2c-c4e3-4c66-be83-b2bf20ec6779
18	12a93011-72c6-4a54-9858-225ec0aed518	Đắk Nông	https://imgur.com/VHnuauV.png	.	cbe91f2c-c4e3-4c66-be83-b2bf20ec6779
19	6f374468-75a0-43ed-88aa-fbe0f4c2ad96	Điện Biên	https://imgur.com/NvBqQRK.png	.	19d96eb7-0c99-4395-b4bb-23091412e00d
20	5d45f534-1923-4e15-81d0-cebd6db44ca9	Đồng Nai	https://imgur.com/3MruCte.png	.	d3ce4c07-84a0-4a60-b347-7821f08d219b
21	7d4b13ef-9181-4d45-b364-a61c167254c4	Đồng Tháp	https://imgur.com/U5H5NZk.png	.	d3ce4c07-84a0-4a60-b347-7821f08d219b
22	95bb44d0-8342-41b1-bd7f-e5d5d3f8e7fb	Gia Lai	https://imgur.com/9IYPs2u.png	.	cbe91f2c-c4e3-4c66-be83-b2bf20ec6779
23	e0b9dcb8-b8b9-48e5-b53b-0893837b605c	Hà Giang	https://imgur.com/QhCkuO2.png	.	19d96eb7-0c99-4395-b4bb-23091412e00d
24	c6f786c6-2ea1-4268-8936-6639eb0b5c48	Hà Nam	https://imgur.com/KbAJF5j.png	.	19d96eb7-0c99-4395-b4bb-23091412e00d

Hình 3.6. Dữ liệu của bảng Province

specialty ×

Properties | Data | ER Diagram — specialty_database | Databases | specialty_database

specialty — Enter a SQL expression to filter results (use Ctrl+Space)

	name	image	specialty_id	province_id
1	Cháo ám	https://imgur.com/LtsGFFV.png	00877354-c597-43d0-b2c8-54c0cc707480	2be492e1-b71c-4468-b6e0-de500d98c7ca
2	Chả cá Nha Trang	https://imgur.com/9Nq79hE.png	00e464d5-c67c-40e4-80c1-00f695cdd9ec	5b72c28f-5e6a-4f92-951c-60524d0334b7
3	Bánh ít lá gai	https://imgur.com/X4AQR3r.png	02023cd4-2abf-416a-bbfd-3c64c427d970	261d253e-4829-4d39-a7da-be517ec37b59
4	Bánh khọt Vũng Tàu	https://imgur.com/VRcokIM.png	030c3e8f-750a-43c6-addf-58051fd1ece7	1ed1fdb4-a2e3-4aae-959a-ce77ccf446f0
5	Bánh xèo Cao Lãnh	https://imgur.com/2KeuhwZ.png	0389d641-7640-486d-bdeb-a94db53fd549	7d4b13ef-9181-4d45-b364-a61c167254c4
6	Hủ tiếu mực	https://imgur.com/17E2wcZ.png	04290bba-92ae-4272-80c2-bf71c279f477	1ed1fdb4-a2e3-4aae-959a-ce77ccf446f0
7	Nộm rau dớn	https://imgur.com/opwVZXc.png	0483a991-a478-441a-b453-9b6fa0fd1a0d	abb1fb46-70c7-4772-9eee-2da67001c200
8	Thịt nai rừng	https://imgur.com/QGySPD3.png	048884ac-5003-4562-b8cf-5421cda3784a	5eacfe87-359b-41e2-83f2-96fb08a7cdd3
9	Cá bống vùi tro	https://imgur.com/081hPcS.png	05b7a217-e56d-4f21-b800-36913fbdd504	abb1fb46-70c7-4772-9eee-2da67001c200
10	Xôi tím Lai Châu	https://imgur.com/HTqaNJd.png	0601b724-4c55-43c5-8471-4611bf080434	abb1fb46-70c7-4772-9eee-2da67001c200
11	Kẹo cu đơ	https://imgur.com/rlAsR9U.png	062b4048-d547-4f2c-a095-8fe8f2a008f2	5f3df5fd-9d32-4cf2-ab57-3b7bc1c2591f
12	Cơm hến	https://imgur.com/WcX2gAf.png	065f9b92-23cb-4b58-aa48-d0f8a094d743	77a7b3ec-7673-4035-a772-248978be6509
13	Phá lấu	https://imgur.com/hdIzamj.png	075e908a-2987-49d9-a40e-ced355dd01f9	c3017168-0c36-44b3-9d69-1094901eabea
14	Chả Trẻ rơm	https://imgur.com/kdkaQ33.png	07724096-03c5-4d7d-95ed-74fa5358cf01	261d253e-4829-4d39-a7da-be517ec37b59
15	Bánh tằm ngan dừa	https://imgur.com/r3dWXQy.png	082c374c-3304-4c91-b5d8-2171571d32df	e34e5682-0d84-46fd-af41-84cf07c45f66
16	Thịt chuột Đình Bảng	https://imgur.com/Z0pvR1Y.png	0aad22b9-ce48-497b-8e72-b089a7ca7b69	e80fa517-09e5-4a5c-8ffa-65b8ac6eb3df
17	Cá kèo nướng ống sậy	https://imgur.com/rdFnoth.png	0b946e5d-010b-409a-95f0-3ae7e44e6133	37244b18-36d0-4231-b471-e8ff83dc2995
18	Lẩu thả	https://imgur.com/ndgdaC9.png	0bcc10aa-6c1b-4308-9ee4-e0580d53367f	05b562a8-6747-4094-9c76-fbc79206a661
19	Ốc len xào dừa	https://imgur.com/2F5Txcs.png	0be8b8cb-1f67-49a4-9016-34fa2d7cef65	96943a97-e3a4-476b-8327-f8a937ed6f93
20	Mủ trôm	https://imgur.com/3JtZQBW.png	0dd92f20-d313-4b35-82df-ba61d2045c53	05b562a8-6747-4094-9c76-fbc79206a661
21	Cá lóc đồng nướng trui	https://imgur.com/OI52MDJ.png	0f2bd87d-5094-40c7-be6d-ace097a1f06b	96943a97-e3a4-476b-8327-f8a937ed6f93
22	Lẩu lá rau rừng	https://imgur.com/JmkfXPu.png	12db338e-00b2-42e7-aa91-8396b33e6887	95bb44d0-8342-41b1-bd7f-e5d5d3f8e7fb
23	Su su Tam Đảo	https://imgur.com/Ju5bUUr.png	133c0ca7-9f2b-4628-a27d-2c11bf62a314	39574761-f30f-47f1-b874-cb47c5a92a99
24	Cốm hộc Phan Thiết	https://imgur.com/GkRcg19.png	13467dfe-126d-4d9f-aaf1-d674aae64212	05b562a8-6747-4094-9c76-fbc79206a661

Hình 3.7. Dữ liệu của bảng Specialty

specialty details

Properties | Data | ER Diagram — specialty_database | Databases | specialty_database | Tables | specialt

specialty details — Enter a SQL expression to filter results (use Ctrl+Space)

	specialtydetails_id	name	origin	image	specialty_id	ingredient	description
1	02a77a17-b96a-4aee-b	Mỳ Quảng	Quảng Nam	https://imgur.com/	6f657d18-ae62-	Nguyên liệu: Mì Quảng (làm từ bó	Mì Quảng có nước dùng đậm
2	02b310d9-0ec2-4c35-b	Bánh tét 3 nhân	Vĩnh Long	https://imgur.com/	ff26c223-d9ef-4	Nguyên liệu:¶Gạo nếp¶Đậu xanh (	Bánh tét 3 nhân là món ăn tru
3	0319663d-25b7-487c-b	Bánh canh bột xắt	Bến Tre	https://imgur.com/	cb8b53d2-44fa-	Nguyên liệu:¶¶Bột gạo¶Nước¶Mu	Bánh canh bột xắt là món ăn
4	03764452-cd18-4858-b	Tôm chua	Bắc Kạn	https://imgur.com/	a437fe04-5fbb-	Nguyên liệu:¶¶Tôm sú¶Muối¶Gạo'	Tôm chua là món ăn đặc sản
5	03de553f-c996-41d8-a	Rượu cần	Đắk Lắk	https://imgur.com/	3d7bbfb2-92a1	Nguyên liệu:¶¶Gạo nếp: 2 kg¶Mer	Rượu cần là thức uống truyền
6	044ebd7a-7738-4841-9	Phở khô Gia Lai	Gia Lai	https://imgur.com/	638dfb25-1660	Nguyên liệu:¶¶Bánh phở: 500g¶Th	Phở khô Gia Lai là món ăn đặ
7	053c4a5b-aa14-4e67-8	Hồng giòn Đà lạt	Lâm Đồng	https://imgur.com/	70867299-d386	Nguyên liệu: Hồng chín, nước vôi	Hồng giòn Đà Lạt có vị ngọt
8	069ddc6a-3a6a-4866-b	Bắp cải cuốn nhót xanh	Điện Biên	https://imgur.com/	4a24f4b7-4c24-	Nguyên liệu:¶¶Bắp cải: 1 cái¶Nhót	Bắp cải cuốn nhót xanh là mó
9	07c7324e-b09a-40e0-a	Canh thụt đọt mây	Đắk Nông	https://imgur.com/	6c28f57e-0e40-	Nguyên liệu:¶¶Đọt mây: 500g¶Cá	Canh thụt đọt mây là món ăn
10	07f4b4fe-bcc7-4602-b0	Cốm hộc Phan Thiết	Bình Thuận	https://imgur.com/	13467dfe-126d-	Nguyên liệu:¶¶Lúa nếp non¶Muối'	Cốm hộc Phan Thiết là món ă
11	084e7ceb-f2a9-4f38-98	Bún vịt nấu tiêu	Sóc Trăng	https://imgur.com/	14c1c295-eade-	Nguyên liệu: Vịt xiêm, bún tươi, ti	Bún vịt nấu tiêu có hương vị
12	09cf969f-f655-4354-bc	Bánh tráng cuốn thịt heo	Quảng Nam	https://imgur.com/	c6822c10-6790-	Nguyên liệu: Bánh tráng gạo, thịt	Bánh tráng cuốn thịt heo là m
13	09d9922f-6217-4fb7-b	Cá mòi kho	Hải Phòng	https://imgur.com/	d001ff93-a510-	Nguyên liệu:¶¶Cá mòi: 500g¶Mỡ l	Cá mòi kho Kiến Thụy là món
14	09e6eeb8-f240-43b6-b	Bánh rế	Bình Thuận	https://imgur.com/	73036d4c-7d5c	Nguyên liệu:¶¶Bột gạo¶Nước cốt	Bánh rế là món ăn vặt được y
15	0a94daa9-ce8a-438c-9	Mắm còng	Tiền Giang	https://imgur.com/	82cb0e7e-3539-	Nguyên liệu: Còng, muối hạt, thín	Mắm còng có vị mặn mà, cay
16	0b3b0c00-7275-4cec-b	Lẩu mẻ	Hậu Giang	https://imgur.com/	b6c96356-b645	Nguyên liệu:¶¶Cá basa/lóc/viên¶M	Lẩu mẻ mang vị chua thanh c
17	0c084f7d-5522-45f1-b	Bò tơ hấp cuốn bánh tráng	Bắc Ninh	https://imgur.com/	e8419aa7-8a04-	Nguyên liệu:¶¶Thịt bò tơ¶Bánh trá	Bò tơ hấp cuốn bánh tráng là
18	0d9d3764-f681-4dbe-a	Bánh khảo	Tuyên Quang	https://imgur.com/	23eca31c-11de-	Nguyên liệu: Bột gạo nếp, đường	Bánh khảo là món ăn truyền t
19	0eeaa447-6008-4089-a	Muồm muỗm rang	Yên Bái	https://imgur.com/	98d7bdea-2e4f-	Nguyên liệu: Muồm muỗm (một	Muồm muỗm rang có vị béo
20	0fd330a3-7921-4928-b	Chuối nếp nướng	Cần Thơ	https://imgur.com/	635cb9c8-6088-	Nguyên liệu:¶¶Chuối nếp: 1 kg¶N	Chuối nếp nướng là món ăn
21	0fffb9eb-7af7-444a-9e	Bọ xít chiên giòn	Yên Bái	https://imgur.com/	c2708df9-e7d9-	Nguyên liệu: Bọ xít, muối, ớt, dầu	Bọ xít chiên giòn có vị béo ng
22	10c640dc-9f9c-4f4a-84	Bánh cáy	Thái Bình	https://imgur.com/	ae0a3a9b-4811-	Nguyên liệu: Gạo nếp cái hoa vàn	Bánh cáy có vị ngọt thanh, dẻ
23	1160ff89-756e-40fa-b1	Bánh xoài	Khánh Hòa	https://imgur.com/	7f99e906-630b-	Nguyên liệu:¶¶Bột nếp¶Xoài chín¶	Bánh xoài có vị ngọt thơm củ
24	165b9d6d-0fa0-4b51-9	Cá chua	Kon Tum	https://imgur.com/	60bc9b11-1f8d-	Nguyên liệu: Cá chép, cơm mẻ, riề	Cá chua Kon Tum có vị chua

Hình 3.8. Dữ liệu của bảng SpecialtyDetails

account

Properties | Data | ER Diagram — specialty_database | Databases | specialty_database | Tables | accc

account — Enter a SQL expression to filter results (use Ctrl+Space)

	account_id	username	password	refreshToken	createdAt	updatedAt	deletedAt
1	0f82b2ed-59ee-4486-aff	nguyenvantao123456789	$2b$10$1WVQwUgnJri3V	eyJhbGciOiJIUzI1NiIsInR	2024-05-28 04:44:36.783444	2024-05-28 04:46:51	[NULL]
2	12fa6464-0839-4a8a-a24	Haha	$2b$10$3LFBbbEZ2MimY	[NULL]	2024-05-11 10:35:53.592232	2024-05-18 15:53:59	[NULL]
3	1a58662f-6b47-4e89-a76	abcdef	$2b$10$1pJ/Sp4IovRvkP4	eyJhbGciOiJIUzI1NiIsInR	2024-05-29 07:08:41.383387	2024-06-10 16:10:38	[NULL]
4	2cc61491-5cfb-4f7a-906	testtesst	$2b$10$i3ozQJjiyTU5lq4R	eyJhbGciOiJIUzI1NiIsInR	2024-05-29 03:04:18.149693	2024-05-29 03:04:18	[NULL]
5	342d4b70-7c04-4a4c-99	Nguyenvantao01	$2b$10$zFaYvs5Fd1Hog3	eyJhbGciOiJIUzI1NiIsInR	2024-06-12 07:09:41.565282	2024-06-12 07:09:55	[NULL]
6	433af546-e647-44a4-805	tranhoangnam	$2b$10$2hkLCLGsekPw/v	eyJhbGciOiJIUzI1NiIsInR	2024-05-14 17:09:38.465439	2024-06-09 14:28:15	[NULL]
7	4fb27842-2ebf-4505-b67	nguyenphamnhamquy	$2b$10$/rgRhrU1h5IsWd	eyJhbGciOiJIUzI1NiIsInR	2024-05-09 18:51:03.355883	2024-06-12 05:43:44	[NULL]
8	5250124a-8625-4fca-bec	dsadsdsa	$2b$10$wfQopMMLBULk	eyJhbGciOiJIUzI1NiIsInR	2024-05-29 05:58:52.078516	2024-05-29 05:59:23	[NULL]
9	75adcdae-cc6a-4db5-8c9	vuducbang	$2b$10$I5Bcnnrgc9ZuePH	eyJhbGciOiJIUzI1NiIsInR	2024-05-09 18:47:15.837378	2024-06-11 03:08:35	[NULL]
10	914ad0b1-5240-4c93-a2	phamquangthanh	$2b$10$5cEMsIjn8OTkaF4	eyJhbGciOiJIUzI1NiIsInR	2024-05-09 18:48:05.457851	2024-06-09 14:25:26	[NULL]
11	b0fcf7ab-0368-4283-bba	tktest	$2b$10$FTxxluHNDHgETv	eyJhbGciOiJIUzI1NiIsInR	2024-05-09 18:55:01.541599	2024-06-15 15:25:52	[NULL]
12	bae08388-f079-48f8-adc	dautrungduc	$2b$10$r.cSWgboA7cUFY	eyJhbGciOiJIUzI1NiIsInR	2024-05-09 18:52:12.849967	2024-06-12 05:43:10	[NULL]
13	bff35134-261a-48b0-afd	vuanhquan	$2b$10$w18NLFx14O9v.o	eyJhbGciOiJIUzI1NiIsInR	2024-05-09 18:49:56.839237	2024-05-14 20:32:03	[NULL]
14	d00c8766-6743-40e5-90	nguyentheanh	$2b$10$Cw3JxgYnf4TZzO	eyJhbGciOiJIUzI1NiIsInR	2024-05-09 18:46:38.624643	2024-06-09 14:27:39	[NULL]
15	e7f8683c-713b-4bf5-b4a	Taohattieu	$2b$10$ip8wJls21fO2RyZ	eyJhbGciOiJIUzI1NiIsInR	2024-06-07 20:54:43.566794	2024-06-10 15:36:25	[NULL]
16	eb37fe5a-aa67-4ac2-9b0	tranquanganh	$2b$10$WC6/rwsCbfHA3	eyJhbGciOiJIUzI1NiIsInR	2024-05-09 18:48:43.521152	2024-06-11 13:04:05	[NULL]
17	fc83743d-2bc9-49ef-9efa	nguyenvantao	$2b$10$.7NjxLAucbZW2F	eyJhbGciOiJIUzI1NiIsInR	2024-05-09 18:45:05.015552	2024-06-15 15:28:20	[NULL]

Hình 3.9. Dữ liệu của bảng Account

profile

Properties | Data | ER Diagram | specialty_database | Databases | specialty_database | Tables | pro

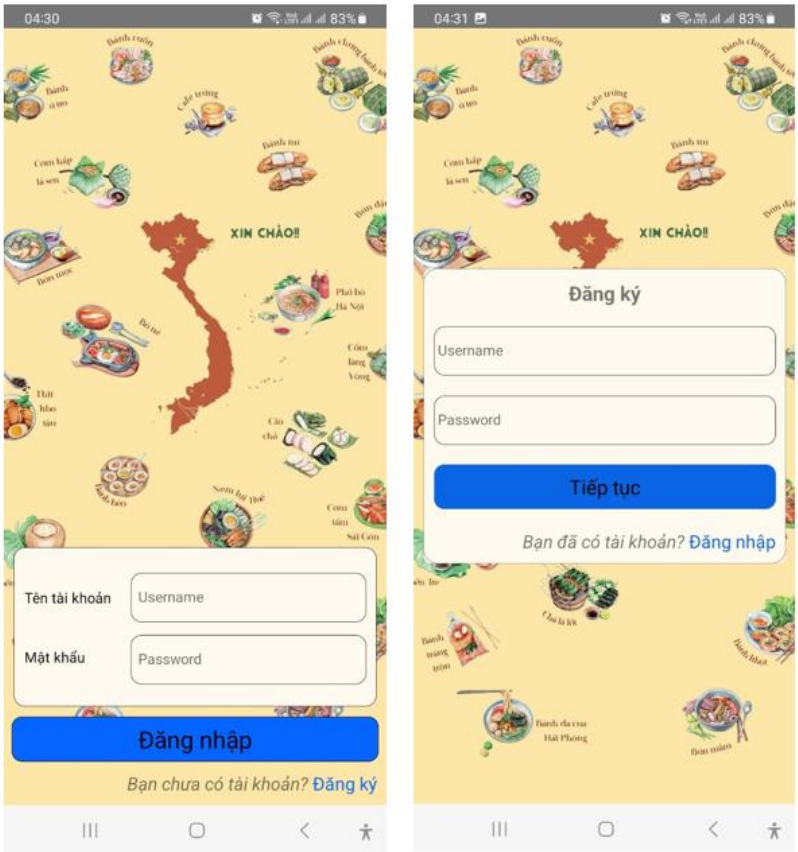
profile | Enter a SQL expression to filter results (use Ctrl+Space)

	profile_i	avatar	account_id	createdAt	updatedA	deletedAt	displayName	address	email	phone	coverImage
1	3f4f2e64-63b	https://imgu	914ad0b1-52a9	18:48:05.491	14 19:06:09.177	[NULL]	Phạm Quang Than	Thái Bình	thanh@gma	0388825955	https://imgur.com
2	442a932f-56	https://imgu	4fb27842-2eb9	18:51:03.378	14 19:06:56.216	[NULL]	Nguyễn Phạm Nhá	Hà Tĩnh	nhamquy@	0943067038	https://imgur.com
3	4fd80ea4-d8	https://imgu	1a58662f-6b49	07:08:41.448	-06-10 16:12:23	[NULL]	abcdef	abcdef	abcdef@gm	083748242	https://imgur.com
4	569b9309-e4	https://imgu	12fa6464-083	10:35:53.679	14 17:37:31.687	[NULL]	Haha Haha	Haha			
5	5bee2065-fc7		5250124a-862	05:58:52.121	29 05:58:52.121	[NULL]					
6	6748a16b-b5	https://imgu	433af546-e64	17:09:38.491	14 19:07:33.713	[NULL]	Trần Hoàng Nam	Hưng Yên	hoangnam@	0969527450	https://imgur.com
7	7de56b07-34	https://imgu	bae08388-f07	18:52:12.871	14 19:08:01.966	[NULL]	Đậu Trung Đức	Hải Phòng	trungduc@	0918634386	https://imgur.com
8	7fd75a45-a0	https://imgu	b0fcf7ab-0360	18:55:01.564	14 20:34:09.926	[NULL]	Não Cá Vàng	Love You Fore	test@gmail.	0293283232	https://imgur.com
9	92a5a32b-c9	https://imgu	342d4b70-7c	07:09:41.617	-06-12 07:12:58	[NULL]	Tạo	Đại nam	vantaodz1@	09276385	https://imgur.com
10	9fabe57a-81	https://imgu	e7f8683c-713	20:54:43.652	-06-10 15:52:42	[NULL]	Tạo Hạt Tiêu	Vĩnh Ninh	Vantaodz1@	0327948814	https://imgur.com
11	ab160544-aa	https://imgu	d00c8766-674	18:46:38.654	14 19:09:10.123	[NULL]	Nguyễn Thế Anh	Bắc Ninh	theanh@gm	0333216128	https://imgur.com
12	b0e65827-54	https://imgu	bff35134-261	18:49:56.866	14 19:10:08.831	[NULL]	Vũ Anh Quân	Bắc Giang	anhquan@g	0343412968	https://imgur.com
13	b707c0b9-d2		2cc61491-5cf	03:04:18.192	29 03:04:18.192	[NULL]	Tao Nguyen		example@g	0866372912	
14	d28a15e0-f7	https://imgu	fc83743d-2bc	18:45:05.036	-06-12 02:02:38	[NULL]	Nguyễn Văn Tạo	Vĩnh Phúc	vantaodz1@	08663729125	https://imgur.com
15	d625e4a5-64	https://imgu	75adcdae-cc6	18:47:15.855	14 19:11:27.574	[NULL]	Vũ Đức Bằng	Hà Nội	ducbang@	0357838285	https://imgur.com
16	d6316cb1-42	https://imgu	eb37fe5a-aa6	18:48:43.542	14 19:11:27.598	[NULL]	Trần Quang Anh	Hà Nội	quanganh@	0328835115	https://imgur.com
17	f940412f-4cc		0f82b2ed-59e	04:44:36.854	28 04:44:36.854	[NULL]	Tao Nguyen		example@g	0866372912	

Hình 3.10. Dữ liệu của bảng Profile

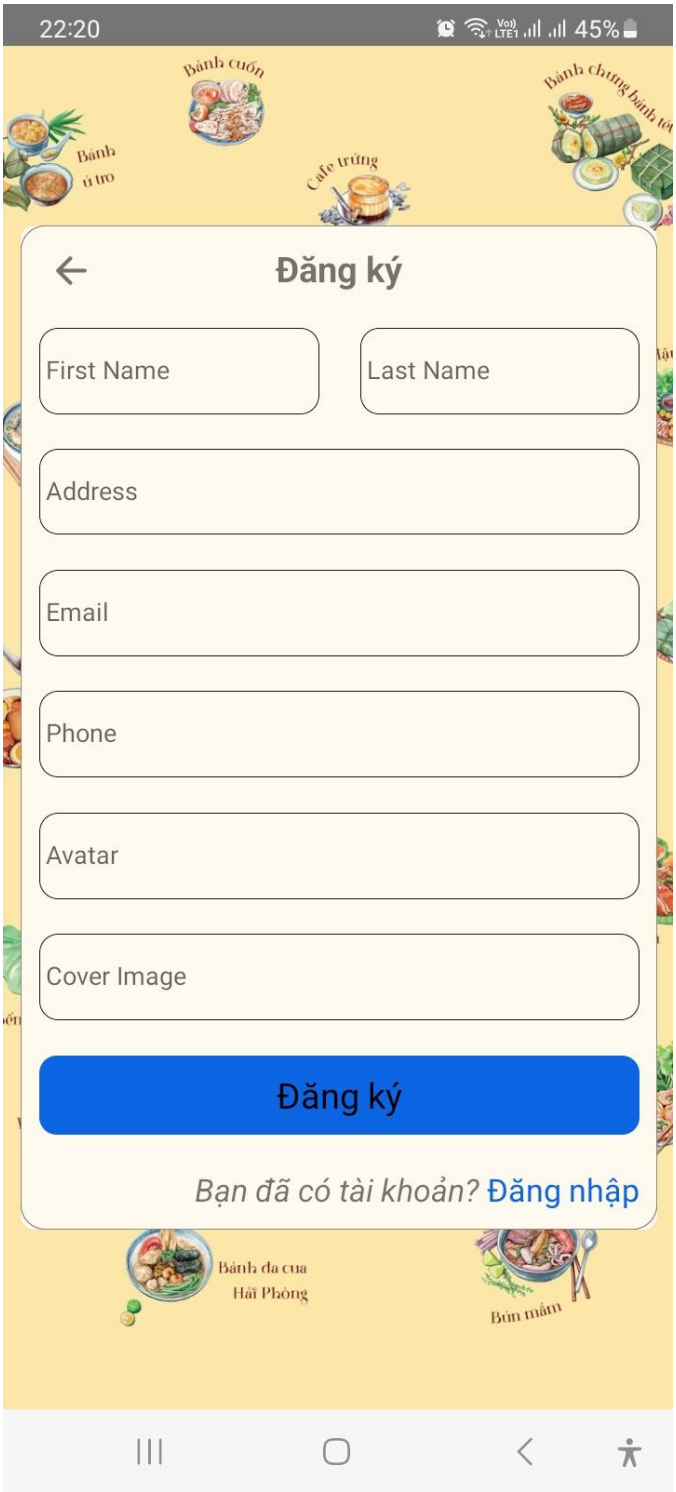
3.4. Giao diện và chức năng của ứng dụng

3.4.1. Giao diện của ứng dụng



Hình 3.11. Giao diện trang đăng nhập và đăng ký tài khoản

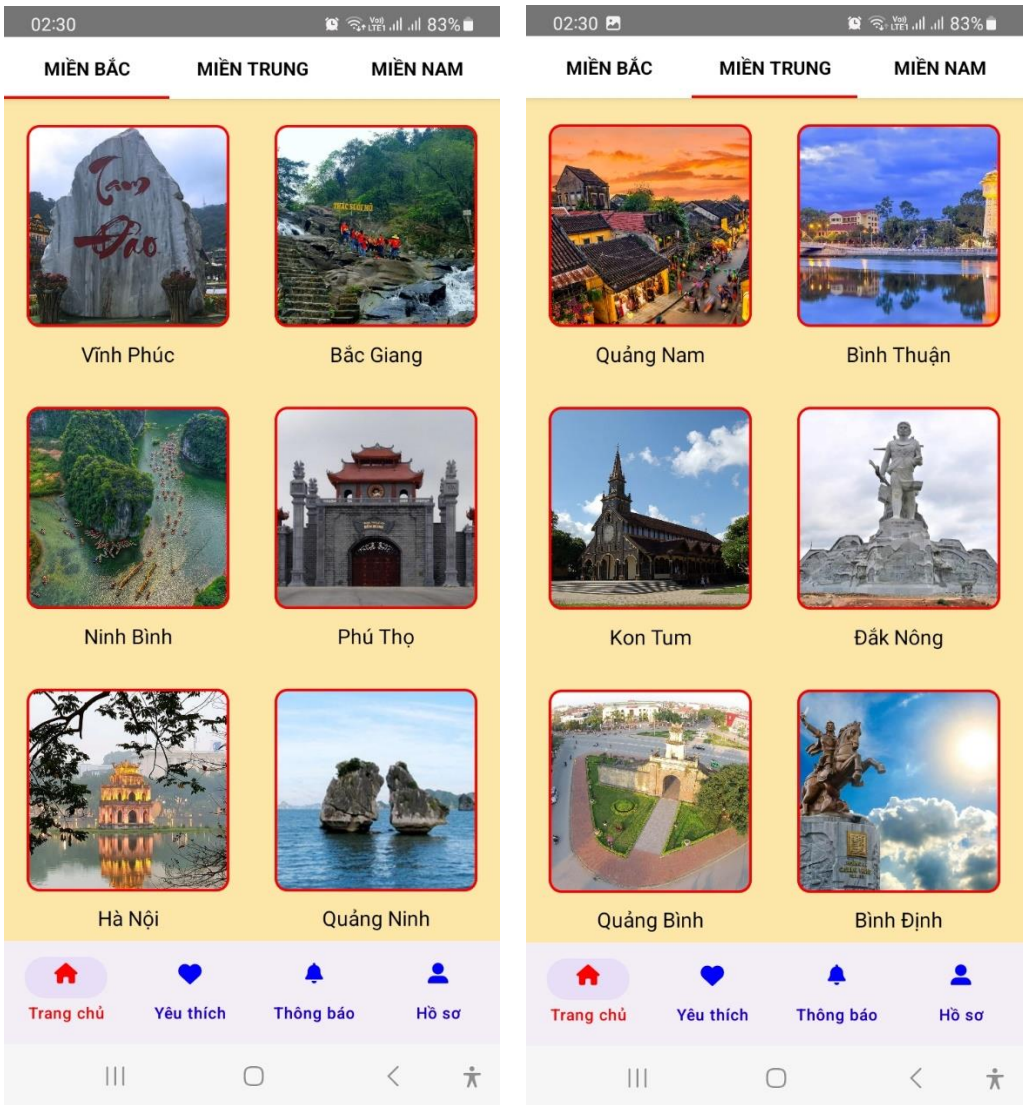
Khi bắt đầu vào ứng dụng, người dùng sẽ phải đăng ký một tài khoản cá nhân để có thể sử dụng ứng dụng, sau khi đăng ký thành công người dùng sẽ sử dụng tài khoản đó để đăng nhập vào app và xem các thông tin cần thiết.



Hình 3.12. Giao diện trang thêm thông tin cá nhân trước khi đăng ký tài khoản

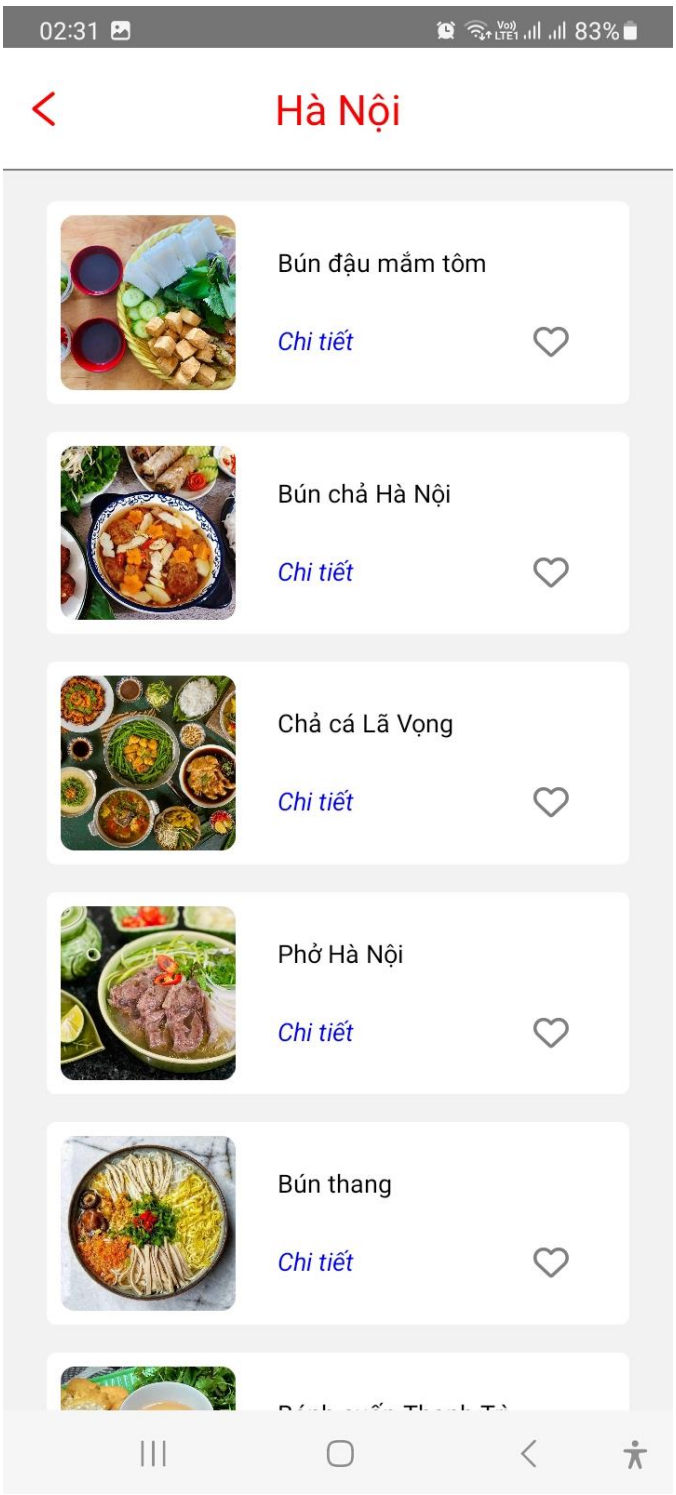
Khi người dùng đăng ký tài khoản, sau khi đã nhập xong username và

password thì người dùng sẽ được chuyển tới trang nhập các thông tin cá nhân, người dùng có thể không bắt buộc nhập và có thể nhấn đăng ký tài khoản luôn và cập nhật thông tin cá nhân sau khi vào trong ứng dụng.



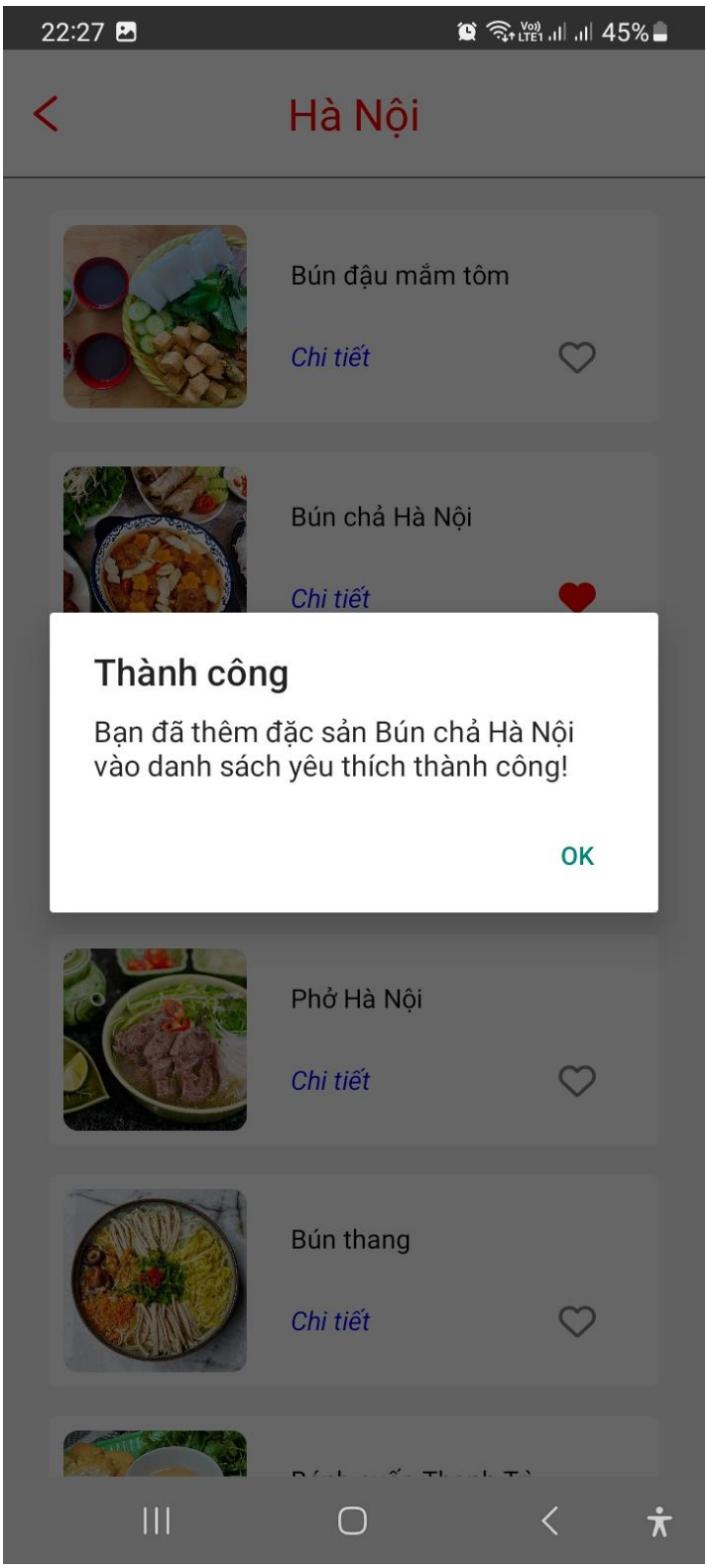
Hình 3.13. Giao diện trang chủ khi đã đăng nhập vào ứng dụng

Khi người dùng đã đăng ký tài khoản và đăng nhập vào ứng dụng, ở trang chủ sẽ hiển thị các thông tin về vùng miền, các tỉnh tương ứng với vùng miền đó, ngoài ra bên cạnh còn có các trang như yêu thích, thông báo đây là nơi lưu trữ và hiển thị các đặc sản yêu thích của người dùng, các thông báo của ứng dụng.



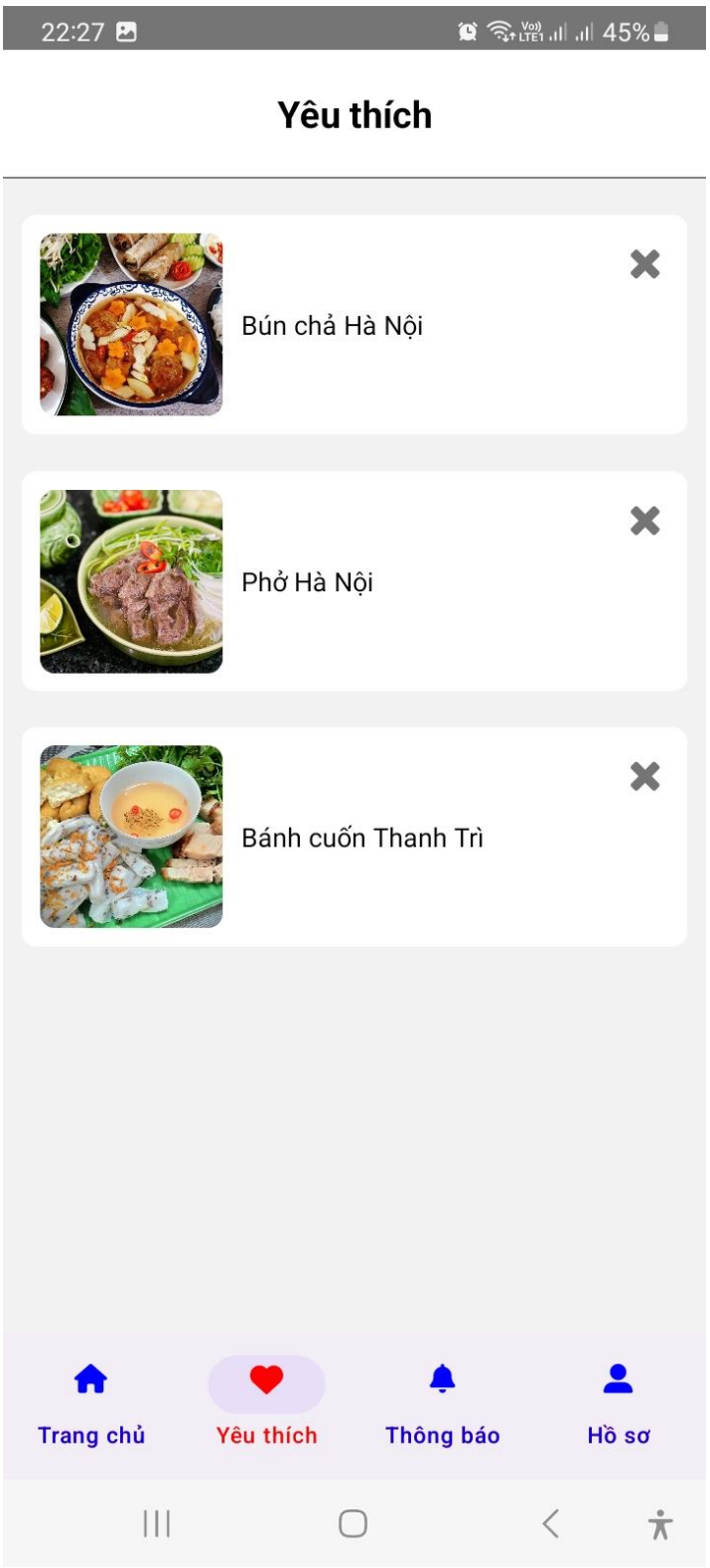
Hình 3.14. Giao diện trang đặc sản của một tỉnh

Khi nhấn vào một tỉnh của một vùng miền nào đó, ứng dụng sẽ hiển thị cho người dùng thông tin về các đặc sản thuộc tỉnh đó gồm các thông tin như hình ảnh, tên đặc sản,… Ngoài ra có thể nhấn vào yêu thích để quản lý trong trang yêu thích và xem chi tiết để biết các thông tin chi tiết của đặc sản đó.



Hình 3.15. Giao diện thông báo khi người dùng thêm đặc sản vào danh sách yêu thích thành công

Khi người dùng nhấn vào yêu thích một đặc sản nào đó ứng dụng sẽ hiển thị lên một thông báo cho biết rằng người dùng đã thêm thành công đặc sản đó vào danh sách yêu thích.

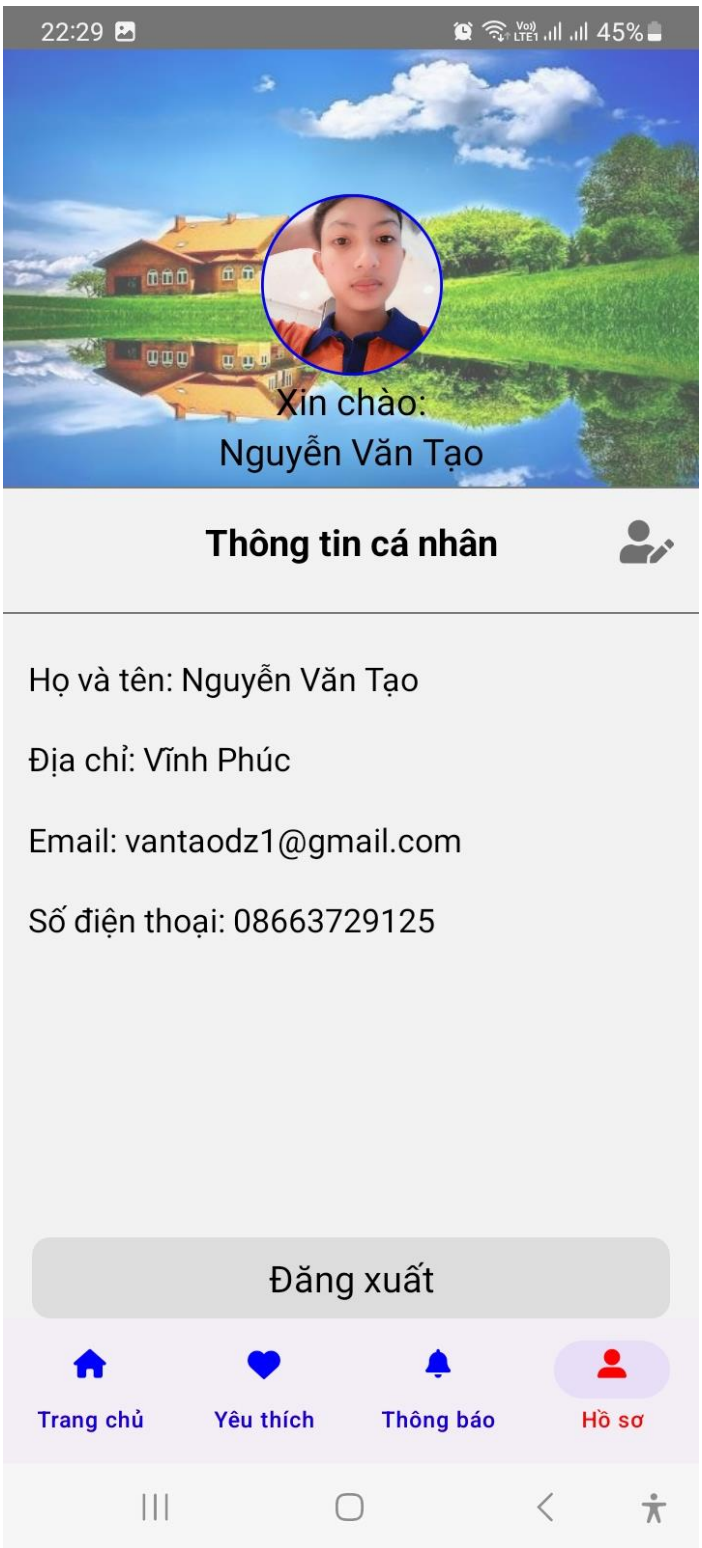


Hình 3.16. Giao diện trang danh sách đặc sản yêu thích

Khi người dùng nhấn vào thêm đặc sản vào danh sách yêu thích thành công, đặc sản sẽ được lưu và hiển thị tại trang danh sách yêu thích, người dùng cũng có thể từ danh sách này nhấn vào xem chi tiết đặc sản đó hoặc có thể xóa bỏ nó khỏi danh sách yêu thích.

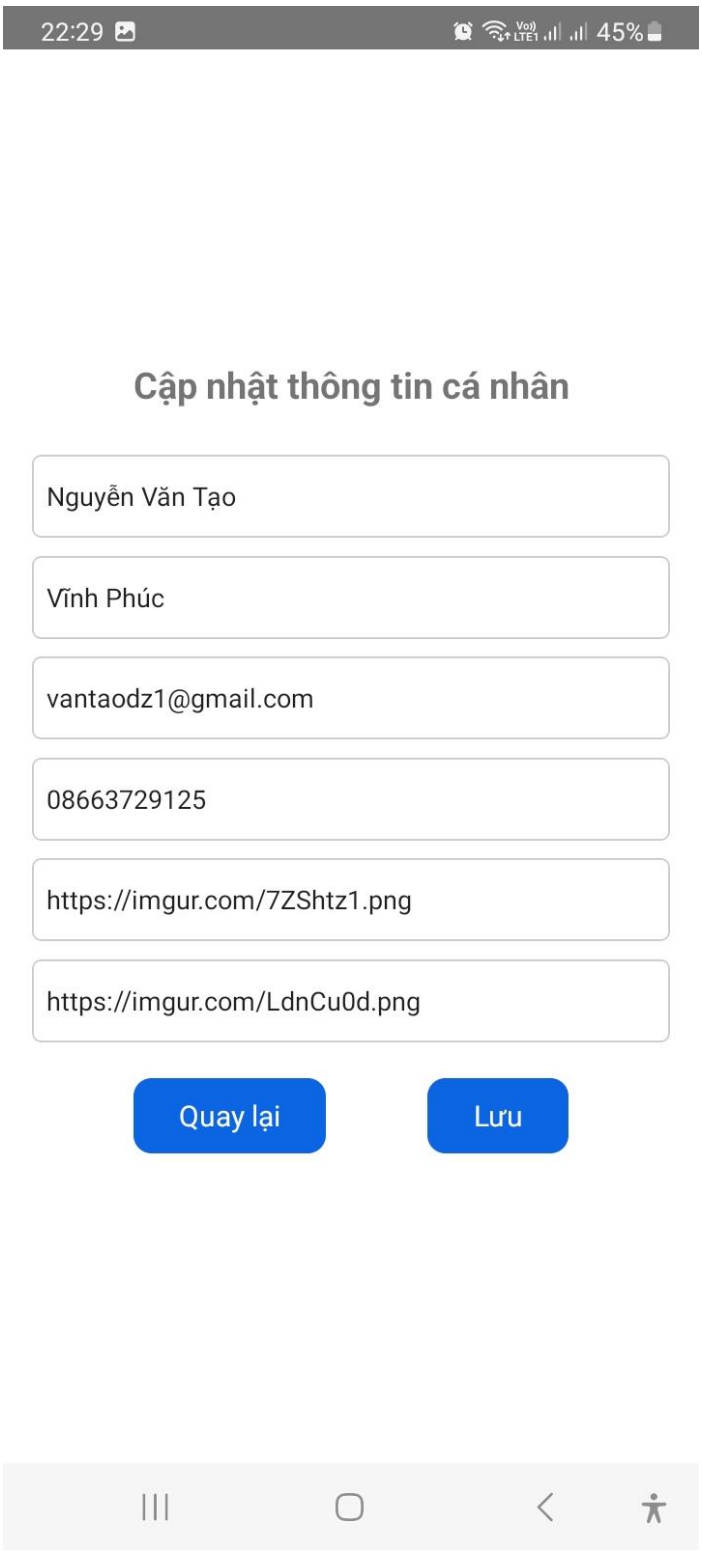
Hình 3.17. Giao diện trang chi tiết đặc sản của một tỉnh

Khi nhấn vào xem chi tiết một đặc sản, thông tin sẽ hiển thị đến với người dùng chi tiết của đặc sản đó bao gồm như tên, hình ảnh, nguyên liệu, cách chế biến, xuất xứ, mô tả,… của đặc sản đó.



Hình 3.18. Giao diện trang hồ sơ thông tin cá nhân của người dùng

Ở trang hồ sơ này sẽ hiển thị các thông tin cá nhân mà khi đăng ký tài khoản người dùng đã đăng ký.



Hình 3.19. Giao diện trang sửa thông tin cá nhân người dùng

Ở giao diện này, người dùng có thể cá nhân hóa thông tin của mình bằng cách cập nhật, chỉnh sửa các thông tin cá nhân của mình.

3.4.2. Một vài chức năng

Ứng dụng hiện tại đã có một vài chức năng cơ bản như sau:

Đăng nhập đăng ký: Người dùng sẽ đăng ký tài khoản sau đó dùng tài khoản đó để đăng nhập vào ứng dụng để có thể lưu trữ được các đặc sản mà người dùng trước đó đã nhấn yêu thích.

Hiển thị các thông tin về vùng miền, tỉnh, các đặc sản, chi tiết đặc sản (nguyên liệu, cách chế biến, hình ảnh, nguồn gốc, mô tả,…): Người dùng có thể xem được thông tin về vừng miền, các tỉnh, các đặc sản bằng việc đăng nhập vào ứng dụng.

Hiển thị danh sách các đặc sản yêu thích của người dùng: Sau khi đăng nhập vào ứng dụng, người dùng có thể nhấn vào yêu thích một hoặc nhiều đặc sản sau đó sẽ được lưu trữ và hiển thị tại trang yêu thích giúp người dùng có thể quản lý các đặc sản đã quan tâm trước đó mà không phải tìm kiếm lại từ đầu sau khi thoát khỏi ứng dụng.

Hiển thị các thông báo về các sự kiện tại địa phương: Người dùng đăng nhập vào ứng dụng sẽ có 1 trang thông báo, nơi sẽ hiển thị các thông báo từ ứng dụng được gửi tới người dùng như các sự kiện tại địa phương nào đó đã, đang và sẽ diễn ra.

Tiểu kết:

Chương 3 đã trình bày chi tiết quá trình thiết kế và cài đặt chương trình.Các bảng cơ sở dữ liệu được thiết kế cẩn thận để lưu trữ hiệu quả thông tin quan trọng, đồng thời đảm bảo tính toàn vẹn và nhất quán dữ liệu thông qua các mối quan hệ rõ ràng.

Sơ đồ Use Case đã phác họa một cách trực quan các tương tác giữa người dùng và hệ thống, làm nổi bật các chức năng chính và cách thức người dùng sẽ tương tác với chương trình.

Giao diện người dùng được thiết kế với mục tiêu tối ưu hóa trải nghiệm người dùng, chú trọng vào tính dễ sử dụng, trực quan và thân thiện.

KẾT LUẬN

Qua quá trình thực hiện đề tài, em đã đạt được một số kết quả. Trước hết, giao diện của ứng dụng đã được thiết kế đơn giản, thân thiện và dễ sử dụng, mang lại hiệu quả trải nghiệm tốt cho người dùng, phù hợp với nhiều độ tuổi khác nhau. Nhờ khả năng tối ưu hiệu năng của React Native, ứng dụng hoạt động mượt mà và có thời gian phản hồi nhanh.

Hệ thống API được xây dựng bằng NestJs đã mang lại khả năng quản lý dữ liệu và bảo mật cao. Hệ thống này tích hợp với cơ sở dữ liệu MySql và có khả năng mở rộng trong tương lai. Ứng dụng cũng cung cấp nhiều chức năng hữu ích như lọc thông tin về vùng miền và đặc sản, giúp người dùng dễ dàng tìm kiếm thông tin mong muốn. Ngoài ra, ứng dụng còn hiển thị các thông báo về các sự kiện tại địa phương và cho phép lưu trữ đặc sản yêu thích để tiện tìm kiếm sau này.

Bên cạnh những thành tựu đạt được, em cũng gặp phải một số hạn chế trong quá trình thực hiện đề tài. Ứng dụng gặp vấn đề về hiệu năng trên các thiết bị cũ, đặc biệt khi dữ liệu ứng dụng ngày càng tăng. Hơn nữa, việc chưa tích hợp hệ thống thanh toán đã làm giảm khả năng mở rộng các chức năng liên quan đến thương mại điện tử.

Ứng dụng phụ thuộc vào kết nối Internet, gây trải nghiệm không tốt đối với người dùng ở những khu vực có mạng kém. Hiện tại, ứng dụng chưa có phiên bản dành cho hệ điều hành IOS và cũng chưa tích hợp tính năng đề xuất đặc sản dựa trên sở thích và lịch sử tìm kiếm của người dùng. Ngoài ra, ứng dụng chưa hỗ trợ đa ngôn ngữ và tính năng đặt phòng, vé máy bay, tàu xe cũng chưa được phát triển.

Từ những kết quả đạt được và hạn chế còn tồn đọng, em đã đề ra những giải pháp và hướng phát triển trong tương lai. Trước hết, cần tiếp tục tối ưu hiệu năng trên các thiết bị cấu hình thấp và phát triển phiên bản dành cho hệ điều hành IOS. Em cũng dự định triển khai các kỹ thuật tối ưu cơ sở dữ liệu như sharding và replication để xử lý dữ liệu hiệu quả hơn.

Việc áp dụng các kỹ thuật machine learning sẽ giúp cá nhân hóa trải nghiệm người dùng. Các chức năng như giỏ hàng, quản lý đơn hàng, theo dõi vận chuyển và tích hợp hệ thống thanh toán sẽ được phát triển thêm. Để mở rộng đối tượng người dùng, ứng dụng sẽ hỗ trợ nhiều ngôn ngữ và cho phép tùy chỉnh giao diện theo sở thích cá nhân.

Tích hợp Google Map sẽ giúp người dùng dễ dàng định vị và khám phá các địa điểm đặc sản. Các chức năng mạng xã hội như đánh giá, bình luận, chia sẻ cũng sẽ được phát triển để tạo nên một cộng đồng sôi nổi. Nghiên cứu và phát triển tính năng đề xuất đặc sản dựa trên sở thích và lịch sử tìm kiếm, cũng như liên kết với các dịch vụ du lịch, sẽ giúp kích cầu du lịch tốt hơn.

Em cũng dự định phát triển tính năng ngoại tuyến để người dùng có thể truy cập ứng dụng mà không cần kết nối Internet, và phát triển phiên bản web của ứng dụng để tiếp cận nhiều đối tượng hơn. Cuối cùng, em sẽ xây dựng chiến lược marketing để quảng bá ứng dụng cả trong và ngoài nước, đẩy mạnh việc xây dựng cộng đồng người dùng thông qua các sự kiện và hợp tác với các doanh nghiệp địa phương.

DANH MỤC TÀI LIỆU THAM KHẢO

Tiếng Việt:

[1]. Nguyễn Hữu Hưng, Nguyễn Thành Nam. React Native: Xây dựng ứng dụng di động đa nền tảng. Nhà xuất bản Thông tin và Truyền thông, Hà Nội, 2021.

[2]. Phạm Quang Huy, Trần Văn Long. NestJS: Khởi tạo backend mạnh mẽ và linh hoạt. Nhà xuất bản Khoa học và Kỹ thuật, Hà Nội, 2022.

[3]. Trần Đức Minh, Lê Thị Thu Thủy. Đặc sản Việt Nam: Hành trình khám phá ẩm thực vùng miền. Nhà xuất bản Văn hóa - Thông tin, Hà Nội, 2019.

Tiếng Anh:

[1]. Azoff, E. M. (2020). Progressive web apps with React. Manning Publications Co.

[2]. Bonobo, J. (2021). NestJS in action. Manning Publications Co.

[3]. Nguyen, T. T. T., & Nguyen, T. M. (2018). Vietnamese cuisine: A journey through the culinary traditions of Vietnam. Thế Giới Publishers.

Danh mục các Website tham khảo:

[1]. https://www.w3schools.com/

[2]. https://reactnative.dev/

[3]. https://nestjs.com/

[4]. https://www.docker.com/