

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

ỨNG DỤNG NODE.JS, REACT, POSTGRESQL
TRONG VIỆC XÂY DỰNG HỆ THỐNG QUẢN LÝ
QUÁN TRÌNH CAFE ĐÀ NẴNG

Người hướng dẫn: TS. ĐOÀN DUY BÌNH

Sinh viên thực hiện: LÊ VĂN NHẬT

Mã số sinh viên: 2151220116

Lớp: 21CT3

Đà Nẵng, 12/2025

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

**ỨNG DỤNG NODE.JS, REACT, POSTGRESQL
TRONG VIỆC XÂY DỰNG HỆ THỐNG QUẢN LÝ
QUÁN TRÌNH CAFÉ ĐÀ NẴNG**

Người hướng dẫn: TS. ĐOÀN DUY BÌNH

Sinh viên thực hiện: LÊ VĂN NHẬT

Mã số sinh viên: 2151220116

Lớp: 21CT3

Đà Nẵng, 12/2025

TÓM TẮT

Tên đề tài: Ứng dụng Node.js, React, Postgresql xây dựng hệ thống quản lý quán Trình Café Đà Nẵng.

Sinh viên thực hiện: Lê Văn Nhật

Mã số SV: 2151220116

Lớp: 21CT3

Trong bối cảnh ngành F&B (Food and Beverage) ngày càng cạnh tranh, các quán café cần những giải pháp công nghệ để nâng cao hiệu quả quản lý và tối ưu trải nghiệm khách hàng. Các hệ thống quản lý thủ công hoặc phần mềm cục bộ thường bộc lộ nhiều hạn chế, gây khó khăn trong việc mở rộng quy mô và đồng bộ dữ liệu. Do đó, việc xây dựng một hệ thống quản lý hiện đại, toàn diện, bảo mật và có khả năng mở rộng trở thành nhu cầu tất yếu.

Đề tài tập trung phát triển hệ thống quản lý quán café với các chức năng cốt lõi: quản lý menu, quản lý bàn và khu vực phục vụ, xử lý đơn hàng, thanh toán, và thống kê doanh thu. Hệ thống được thiết kế để phục vụ đa đối tượng: nhân viên (giảm tải công việc thủ công), khách hàng (đặt món và thanh toán tiện lợi), và chủ quán (báo cáo doanh thu minh bạch và phân tích hiệu quả kinh doanh).

Về công nghệ, hệ thống sử dụng **Node.js** cho backend để xử lý đồng thời nhiều yêu cầu với tốc độ cao, **React** cho frontend giúp xây dựng giao diện trực quan, hiện đại và thân thiện, và **PostgreSQL** để quản lý dữ liệu tập trung, an toàn và dễ mở rộng. Sự kết hợp này đảm bảo tính ổn định, hiệu quả và bảo mật trong suốt quá trình vận hành.

Kết quả kỳ vọng của đề tài là một hệ thống quản lý quán café hoàn chỉnh, có khả năng triển khai thực tế tại các quán vừa và nhỏ. Hệ thống giúp giảm thiểu sai sót, rút ngắn thời gian phục vụ, đồng thời nâng cao trải nghiệm khách hàng và năng suất lao động. Bên cạnh đó, hệ thống cũng cung cấp nền tảng để phát triển thêm các tính năng nâng cao trong tương lai như gợi ý món bằng AI, tích hợp thanh toán điện tử, hay quản lý khách hàng thân thiết.

Với định hướng ứng dụng thực tiễn và tính mở rộng cao, đề tài không chỉ giải quyết được nhu cầu hiện tại của các quán café mà còn đóng góp vào tiến trình chuyển đổi số trong lĩnh vực dịch vụ ăn uống, tạo lợi thế cạnh tranh bền vững cho Trình Café nói riêng và ngành F&B nói chung.

LỜI NÓI ĐẦU

Trong quá trình thực hiện đồ án tốt nghiệp với đề tài “Ứng dụng Node.js, React, PostgreSQL trong xây dựng hệ thống quản lý quán Trình Café Đà Nẵng”, em đã nhận được sự hỗ trợ từ rất nhiều phía, em xin chân thành cảm ơn đến những người đã đồng hành với em trong khoảng thời gian 4 tháng vừa qua.

Trước tiên, em xin chân thành cảm ơn TS. Đoàn Duy Bình, người đã tận tình hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện đồ án. Sự chỉ dẫn, kiến thức chuyên môn sâu sắc và những ý kiến góp ý quý báu của thầy đã giúp em định hướng đúng đắn trong quá trình nghiên cứu và phát triển hệ thống. Em rất biết ơn vì thầy đã luôn sẵn sàng giải đáp mọi thắc mắc và hỗ trợ hết mình cho em để em có thể hoàn thành tốt nhất nhiệm vụ của mình.

Em cũng xin cảm ơn Khoa Công nghệ Thông tin, Trường Đại học Kiến trúc Đà Nẵng, nơi đã cung cấp môi trường học tập và nghiên cứu tốt nhất để em phát triển. Các thầy cô trong khoa đã tạo điều kiện thuận lợi cho em trong quá trình học tập và nghiên cứu.

Với điều kiện thời gian cũng như kinh nghiệm hạn chế của sinh viên nên bài báo cáo này không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của quý thầy cô cũng như lời nhận xét của hội đồng bảo vệ đồ án để em có điều kiện bổ sung, nâng cao kiến thức của bản thân, phục vụ tốt hơn cho công tác sau này.

Em xin chân thành cảm ơn!

CAM ĐOAN

Tôi xin cam đoan rằng đồ án tốt nghiệp với đề tài "**ỨNG DỤNG NODE.JS, REACT, POSTGRESQL TRONG VIỆC XÂY DỰNG HỆ THỐNG QUẢN LÝ QUÀN TRÌNH CAFÉ ĐÀ NẴNG**" là kết quả nghiên cứu và làm việc của riêng tôi dưới sự hướng dẫn của TS. Đoàn Duy Bình.

Tất cả nội dung trong đồ án, bao gồm các phần nghiên cứu lý thuyết, phân tích hệ thống, thiết kế và triển khai đều do tôi tự thực hiện. Những tài liệu, số liệu tham khảo từ các nguồn khác đều được trích dẫn rõ ràng và đúng quy định.

Tôi hoàn toàn chịu trách nhiệm về tính trung thực và chính xác của các nội dung trong đồ án này. Nếu có bất kỳ sai phạm nào, tôi xin chịu hoàn toàn trách nhiệm trước nhà trường.

Sinh viên thực hiện

Lê Văn Nhật

DANH MỤC BẢNG

Bảng 3. 1 Bảng danh sách Use Case Khách hàng	28
Bảng 3. 2 Bảng danh sách Use Case quản trị viên	30
Bảng 3. 3 Bảng thực thể Người dùng	36
Bảng 3. 4 Bảng thực thể Bàn	37
Bảng 3. 5 Bảng thực thể danh mục món	37
Bảng 3. 6 Bảng thực thể món uống	37
Bảng 3. 7 Bảng thực thể đơn đặt	38
Bảng 3. 8 Bảng thực thể chi tiết đơn đặt	38
Bảng 3. 9 Bảng thực thể doanh thu	38
Bảng 3. 10 Bảng API người dùng.....	47
Bảng 3. 11 Bảng API danh mục	47
Bảng 3. 12 Bảng API món uống.....	47
Bảng 3. 13 Bảng API bàn	48
Bảng 3. 14 Bảng API đơn hàng	48
Bảng 3. 15 Bảng API doanh thu	48
Bảng 3. 16 Mối quan hệ giữa các bảng	51
Bảng 5. 1 Bảng kiểm thử đăng nhập	61
Bảng 5. 2 Bảng kiểm thử hiển thị menu	61
Bảng 5. 3 Bảng kiểm thử đặt món.....	61
Bảng 5. 4 Bảng kiểm thử quản lý món.....	62
Bảng 5. 5 Bảng kiểm thử quản lý đơn.....	62
Bảng 5. 6 Bảng kiểm thử thống kê doanh thu	62

DANH MỤC HÌNH ẢNH

Hình 2. 1 Tổng quan về hệ thống thông tin quản lý	7
Hình 2. 2 Visual Studio Code	10
Hình 2. 3 Tìm kiếm VSCode trong AppCenter	12
Hình 2. 4 Tiến hành cài đặt.....	12
Hình 2. 5 Tải file cài đặt cho Ubuntu	13
Hình 2. 6 Truy cập vào thư mục chứa file cài đặt bằng Terminal	14
Hình 2. 7 Tiến hành cài đặt.....	14
Hình 2. 8 Node.JS.....	15
Hình 2. 9 Cài đặt Curl.....	16
Hình 2. 10 Cài đặt Node.JS	17
Hình 2. 11 Kiểm tra phiên bản của node.JS và Npm	17
Hình 2. 12 PostgreSQL.....	18
Hình 2. 13 Cài đặt PostgreSQL	19
Hình 2. 14 Cài đặt pgadmin4.....	20
Hình 2. 15 ReactJS	21
Hình 3. 1 Sơ đồ Use Case tổng quát.....	28
Hình 3. 2 Sơ đồ Use Case phân rã Khách hàng.....	29
Hình 3. 3 Sơ đồ Use Case phân rã Quản trị viên.....	30
Hình 3. 4 Use Case quản lý menu, danh mục.....	31
Hình 3. 5 Use Case Quản lý bàn.....	32
Hình 3. 6 Sơ đồ hoạt động.....	33
Hình 3. 7 Biểu đồ tuần tự	35
Hình 3. 8 Mô hình ERD	39
Hình 3. 9 Cấu trúc thư mục	46

Hình 4. 2 Giao diện chính (khách)	53
Hình 4. 3 Giao diện đăng nhập (khách).....	54
Hình 4. 4 Giao diện giỏ hàng (khách)	55
Hình 4. 5 Giao diện lịch sử đặt đơn.....	56
Hình 4. 6 Giao diện quản lý đơn hàng (admin)	57
Hình 4. 7 Giao diện quản lý bàn (admin)	57
Hình 4. 8 Giao diện quản lý thực đơn (admin).....	58
Hình 4. 9 Giao diện quản lý menu (admin).....	58
Hình 4. 10 Giao diện quản lý danh mục (admin)	59
Hình 4. 11 Giao diện báo cáo, thống kê (admin).....	59

DANH MỤC VIẾT TẮT

Ký hiệu	Tiếng Anh	Tiếng Việt
MIS	Management Information System	Hệ thống thông tin quản lý
API	Application Programming Interface	Giao diện lập trình ứng dụng
CRUD	Create – Read – Update – Delete	Tạo – Đọc – Cập nhật – Xóa
QR	Quick Response Code	Mã phản hồi nhanh
UI	User Interface	Giao diện người dùng
UX	User Experience	Trải nghiệm người dùng
JWT	JSON Web Token	Token xác thực JSON
DB	Database	Cơ sở dữ liệu
JSON	JavaScript Object Notation	Định dạng dữ liệu JSON
REST	Representational State Transfer	Kiến trúc RESTful
HTTP	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản
CPU	Central Processing Unit	Bộ xử lý trung tâm
RAM	Random Access Memory	Bộ nhớ truy cập ngẫu nhiên
CRUD	Create – Read – Update – Delete	Thao tác cơ bản trên dữ liệu
VS Code	Visual Studio Code	Công cụ lập trình VS Code
SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc

MỤC LỤC

TÓM TẮT	i
LỜI NÓI ĐẦU.....	ii
CAM ĐOAN.....	iii
DANH MỤC BẢNG	iv
DANH MỤC HÌNH ẢNH	v
DANH MỤC VIẾT TẮT	vii
MỤC LỤC	viii
MỞ ĐẦU	1
Chương 1. ĐỐI TƯỢNG VÀ PHẠM VI	2
1.1. Tổng quan đề tài	2
1.2. Đối tượng nghiên cứu	3
1.3. Phạm vi nghiên cứu	4
1.3.1. Về chức năng	4
1.3.2. Về công nghệ	5
1.3.3. Về sản phẩm.....	5
1.3.4. Về giao diện	5
1.3.5. Về môi trường triển khai.....	5
Chương 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ	7
2.1. Tổng quan về hệ thống thông tin quản lý	7
2.1.1. Mục tiêu của hệ thống thông tin quản lý	7
2.1.2. Vai trò của MIS trong lĩnh vực quản lý quán café	8
2.1.3. Sự cần thiết của hệ thống thông tin quản lý trong việc quản lý quán café hiện nay.....	9

2.1.4. Liên hệ với đề tài.....	9
2.2. Công cụ và cài đặt.....	10
2.2.1. Visual Studio Code	10
2.2.2. Node.JS.....	15
2.2.3. PostgreSQL.....	18
2.2.4. React.....	21
Chương 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	23
3.1. Khảo sát hiện trạng quản lý quán café	23
3.2. Yêu cầu chức năng của hệ thống	24
3.2.1. Chức năng dành cho khách hàng	24
3.2.2. Chức năng dành cho quản trị viên.....	25
3.3. Yêu cầu phi chức năng	26
3.3.1. Hiệu năng (Performance Requirements)	26
3.3.2. Tính ổn định (Reliability).....	26
3.3.3. Tính bảo mật (Security Requirements).....	27
3.3.4. Khả năng mở rộng (Scalability)	27
3.3.5. Tính thân thiện với người dùng (Usability)	27
3.3.6. Tính tương thích (Compatibility).....	27
3.3.7. Khả năng bảo trì (Maintainability).....	27
3.3.8. Sao lưu và phục hồi (Backup & Recovery).....	28
3.4. Mô hình nghiệp vụ	28
3.4.1. Biểu đồ Use case.....	28
3.4.2. Biểu đồ Hoạt động (Activity Diagram).....	33
3.4.3. Biểu đồ Trình tự (Sequence Diagram)	35
3.5. Phân tích dữ liệu	36
3.5.1. Xác định thực thể và thuộc tính	36
3.5.2. Thiết kế mô hình ERD	39

3.5.3.	<i>Mô hình cơ sở dữ liệu logic và vật lý</i>	40
3.6.	Thiết kế hệ thống	44
3.6.1.	<i>Kiến trúc hệ thống tổng thể</i>	44
3.6.2.	<i>Thiết kế Frontend với React</i>	44
3.6.3.	<i>Thiết kế Backend với Node.js</i>	45
3.6.4.	<i>Thiết kế cơ sở dữ liệu với PostgreSQL</i>	49
Chương 4. XÂY DỰNG HỆ THỐNG		52
4.1.	Kiến trúc hệ thống tổng thể	52
4.1.1.	<i>Frontend (React)</i>	52
4.1.2.	<i>Backend (Node.JS)</i>	52
4.1.3.	<i>Cơ sở dữ liệu (PostgreSQL)</i>	52
4.2.	Thiết kế giao diện cho hệ thống	53
4.2.1.	<i>Giao diện chính (khách)</i>	53
4.2.2.	<i>Giao diện đăng nhập (khách)</i>	54
4.2.3.	<i>Giao diện giỏ hàng (khách)</i>	55
4.2.4.	<i>Giao diện lịch sử đơn hàng (khách)</i>	56
4.2.5.	<i>Giao diện quản lý đơn hàng (admin)</i>	57
4.2.6.	<i>Giao diện quản lý bàn (admin)</i>	57
4.2.7.	<i>Giao diện quản lý thực đơn (admin)</i>	58
4.2.8.	<i>Giao diện quản lý menu (admin)</i>	58
4.2.9.	<i>Giao diện quản lý danh mục (admin)</i>	59
4.2.10.	<i>Giao diện báo cáo, thống kê (admin)</i>	59
Chương 5. KIỂM THỬ VÀ ĐÁNH GIÁ		60
5.1.	Phương pháp kiểm thử	60
5.1.1.	<i>Kiểm thử chức năng (Functional testing)</i>	60
5.1.2.	<i>Kiểm thử đơn vị (Unit Testing)</i>	60
5.1.3.	<i>Kiểm thử tích hợp (Integration Testing)</i>	60

5.1.4.	<i>Kiểm thử giao diện (UI/UX Testing)</i>	60
5.1.5.	<i>Kiểm thử hiệu năng (Performance Testing)</i>	60
5.1.6.	<i>Kiểm thử bảo mật (Security Testing)</i>	60
5.2.	Kịch bản kiểm thử	61
5.2.1.	<i>Kiểm thử đăng nhập</i>	61
5.2.2.	<i>Kiểm thử hiển thị menu</i>	61
5.2.3.	<i>Kiểm thử đặt món</i>	61
5.2.4.	<i>Kiểm thử quản lý món</i>	62
5.2.5.	<i>Kiểm thử quản lý đơn</i>	62
5.2.6.	<i>Kiểm thử thống kê doanh thu</i>	62
5.3.	Kết quả kiểm thử	62
5.3.1.	<i>Các chức năng hoạt động ổn định</i>	62
5.3.2.	<i>Kết nối API – Database ổn định</i>	63
5.3.3.	<i>Hiệu năng</i>	63
5.3.4.	<i>Tính ổn định và bảo mật</i>	63
5.3.5.	<i>Giao diện</i>	63
5.4.	Đánh giá ưu điểm và hạn chế của hệ thống	63
5.4.1.	<i>Ưu điểm</i>	63
5.4.2.	<i>Hạn chế</i>	63
KẾT LUẬN		64
1.	Tóm tắt kết quả nghiên cứu	64
2.	Mức độ hoàn thành mục tiêu đề tài	64
3.	Đóng góp chính của hệ thống	65
4.	Hướng phát triển trong tương lai	65
TÀI LIỆU THAM KHẢO		67

MỞ ĐẦU

Trong bối cảnh công nghệ thông tin phát triển nhanh chóng, các ngành dịch vụ, đặc biệt là lĩnh vực F&B (Food and Beverage), đã và đang có những thay đổi rõ rệt trong cách thức quản lý và vận hành. Các quán cafe, nhà hàng ngày nay không chỉ tập trung vào chất lượng sản phẩm, mà còn chú trọng đến việc tối ưu hóa quy trình phục vụ, rút ngắn thời gian chờ đợi và nâng cao trải nghiệm khách hàng. Điều này thúc đẩy nhu cầu áp dụng các hệ thống quản lý hiện đại, thông minh và có tính tự động hóa cao.

Thực tế cho thấy, nhiều cơ sở kinh doanh vẫn đang sử dụng phương pháp quản lý thủ công hoặc phần mềm truyền thống với nhiều hạn chế: dữ liệu phân tán, thiếu đồng bộ, khó khăn trong việc mở rộng và tích hợp các tính năng mới. Điều này không chỉ gây lãng phí thời gian, tăng nguy cơ sai sót mà còn hạn chế khả năng phân tích và đưa ra quyết định chính xác của chủ quán. Do đó, việc nghiên cứu và xây dựng một hệ thống quản lý quán cafe dựa trên nền tảng công nghệ hiện đại là yêu cầu cấp thiết.

Đề tài "Ứng dụng Node.js, React và PostgreSQL trong xây dựng hệ thống quản lý quán Cafe (Trình Cafe)" ra đời nhằm giải quyết những bất cập đó. Hệ thống được thiết kế để cung cấp các chức năng cơ bản nhưng quan trọng như quản lý thực đơn, quản lý bàn, đặt món, xử lý thanh toán, báo cáo doanh thu và hỗ trợ phân tích dữ liệu. Không chỉ dừng lại ở đó, hệ thống còn hướng tới khả năng mở rộng trong tương lai, với các tính năng nâng cao như gợi ý món tự động, quản lý khách hàng thân thiết, hay tích hợp thanh toán điện tử.

Với việc lựa chọn các công nghệ phổ biến và mạnh mẽ hiện nay như Node.js cho backend, React cho frontend và PostgreSQL cho cơ sở dữ liệu, hệ thống kỳ vọng mang lại hiệu suất cao, ổn định, dễ bảo trì và đảm bảo an toàn dữ liệu. Đây cũng là bộ công nghệ phù hợp để phát triển các ứng dụng web hiện đại, linh hoạt và dễ triển khai thực tế.

Đề tài không chỉ có ý nghĩa thực tiễn đối với quán Trình Cafe, mà còn có thể áp dụng rộng rãi cho các cơ sở kinh doanh trong ngành F&B. Việc triển khai hệ thống sẽ góp phần thúc đẩy quá trình chuyển đổi số, nâng cao chất lượng quản lý, cải thiện dịch vụ khách hàng và hỗ trợ doanh nghiệp phát triển bền vững trong môi trường cạnh tranh ngày càng gay gắt.

CHƯƠNG 1. ĐỐI TƯỢNG VÀ PHẠM VI

1.1. Tổng quan đề tài

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ trên toàn cầu, việc ứng dụng công nghệ thông tin vào hoạt động quản lý kinh doanh đã trở thành một xu thế tất yếu, đặc biệt là trong lĩnh vực dịch vụ ăn uống – nơi yêu cầu sự linh hoạt, nhanh chóng và trải nghiệm khách hàng ngày càng được đặt lên hàng đầu. Các quán café hiện nay không chỉ đơn thuần là nơi thưởng thức đồ uống mà còn là không gian làm việc, gặp gỡ, giải trí, và do đó việc quản lý hoạt động của quán một cách hiệu quả đóng vai trò vô cùng quan trọng.

Ở Việt Nam, phần lớn các quán café vẫn đang sử dụng phương pháp quản lý thủ công hoặc các phần mềm rời rạc, thiếu tính kết nối và khả năng mở rộng. Điều này dẫn đến việc quản lý dữ liệu còn chồng chéo, tốn thời gian, dễ xảy ra sai sót khi xử lý đơn hàng, thanh toán hay thống kê doanh thu. Trong khi đó, ở các quốc gia phát triển, nhiều thương hiệu lớn như Starbucks, Costa Coffee, hay Dunkin' đã áp dụng các nền tảng quản lý tích hợp, cho phép khách hàng đặt món trực tuyến, thanh toán điện tử và tích điểm thưởng ngay trên cùng một hệ thống. Sự khác biệt này cho thấy tiềm năng to lớn trong việc ứng dụng các công nghệ web hiện đại vào hoạt động quản lý quán café tại Việt Nam.

Trước thực tiễn đó, đề tài “*Ứng dụng React, Node.js và PostgreSQL trong xây dựng hệ thống quản lý quán Café*” được hình thành với mục tiêu phát triển một hệ thống quản lý toàn diện, hiện đại và thân thiện với người dùng. s

Hệ thống được thiết kế theo mô hình Client–Server, trong đó phía frontend chịu trách nhiệm hiển thị và tương tác, còn backend đóng vai trò xử lý logic nghiệp vụ, kết nối cơ sở dữ liệu, đảm bảo trao đổi thông tin hiệu quả giữa người dùng và hệ thống. Mục tiêu của đề tài không chỉ dừng lại ở việc xây dựng một phần mềm quản lý đơn giản, mà còn hướng đến tạo ra một nền tảng có thể mở rộng cho các mô hình café đa chi nhánh, tích hợp các chức năng như thống kê doanh thu, quản lý nhân viên, theo dõi tình trạng bàn, đặt món trực tuyến và xử lý thanh toán.

Về mặt nghiên cứu, đề tài góp phần khẳng định tính ứng dụng thực tiễn của các công nghệ web hiện đại trong việc giải quyết các vấn đề thực tế của doanh nghiệp vừa

và nhỏ. Về mặt kỹ thuật, việc kết hợp React, Node.js và PostgreSQL cho phép tận dụng ưu điểm của từng công nghệ: hiệu năng cao, khả năng tái sử dụng mã nguồn, bảo mật, và dễ dàng triển khai trên nhiều môi trường khác nhau. Hệ thống được phát triển theo hướng RESTful API giúp việc mở rộng sang mobile app hoặc tích hợp với các nền tảng khác trở nên thuận tiện hơn.

Khi hoàn thiện, sản phẩm của đồ án sẽ mang lại nhiều lợi ích thực tế: giúp chủ quán dễ dàng quản lý hoạt động kinh doanh, giảm thiểu sai sót trong khâu phục vụ, tối ưu quy trình vận hành, đồng thời mang lại trải nghiệm tiện lợi cho khách hàng. Ngoài ra, hệ thống còn có thể trở thành nền tảng tham khảo cho các sinh viên, nhà phát triển hoặc doanh nghiệp nhỏ muốn ứng dụng công nghệ vào lĩnh vực quản lý dịch vụ.

Tóm lại, đề tài không chỉ có ý nghĩa trong phạm vi học thuật mà còn mang giá trị ứng dụng thực tiễn cao, đáp ứng nhu cầu chuyển đổi số trong ngành dịch vụ ăn uống, đồng thời khẳng định vai trò của lập trình web hiện đại trong việc xây dựng các giải pháp quản lý thông minh và bền vững cho tương lai.

1.2. Đối tượng nghiên cứu

Đề tài tập trung nghiên cứu và phát triển một hệ thống quản lý quán cà phê trực tuyến, trong đó xác định hai đối tượng (actor) chính tham gia vào hệ thống bao gồm: Khách hàng và Người quản trị (Quản lý quán). Mỗi đối tượng đảm nhiệm một vai trò khác nhau trong quá trình vận hành và khai thác hệ thống, nhằm đảm bảo toàn bộ quy trình đặt món, phục vụ và thanh toán diễn ra hiệu quả, nhanh chóng và chính xác.

- **Đối với khách hàng**

Khách hàng là **người sử dụng cuối (end-user)** của hệ thống, có vai trò chính là **xem và đặt món trực tuyến** thông qua giao diện web. Thông qua hệ thống, khách hàng có thể trải nghiệm việc đặt món tiện lợi tại quán hoặc từ xa, đồng thời theo dõi trạng thái phục vụ một cách trực quan. Các chức năng chính mà khách hàng có thể thực hiện bao gồm:

- Truy cập website để xem danh sách món theo danh mục.
- Đăng ký, đăng nhập để đặt món và lưu thông tin cá nhân.
- Chọn bàn hoặc quét mã QR tại bàn để đặt món.
- Thêm món vào giỏ hàng, đặt món và gửi yêu cầu phục vụ đến nhân viên.
- Theo dõi tình trạng đơn hàng (đang pha chế, đã phục vụ, đã thanh toán).

- Đánh giá chất lượng dịch vụ hoặc phản hồi sau khi sử dụng dịch vụ.
- Đối với quản trị

Người quản trị là **người chịu trách nhiệm quản lý và điều hành toàn bộ hoạt động** của hệ thống quán cà phê. Họ sử dụng giao diện quản trị riêng để theo dõi doanh thu, xử lý đơn hàng và quản lý thông tin liên quan đến quán. Các chức năng chính mà quản trị có thể thực hiện bao gồm:

- Quản lý danh mục món (thêm, sửa, xóa).
- Quản lý thông tin món ăn/đồ uống: tên món, giá, hình ảnh, mô tả, tình trạng.
- Quản lý bàn và khu vực phục vụ, theo dõi trạng thái từng bàn (rảnh, đang phục vụ, đã đặt trước).
- Xử lý và xác nhận đơn hàng, cập nhật trạng thái phục vụ và thanh toán.
- Quản lý người dùng (khách hàng và nhân viên).

Thống kê và báo cáo doanh thu, hỗ trợ ra quyết định quản lý.

1.3. Phạm vi nghiên cứu

1.3.1. Về chức năng

Đề tài tập trung vào việc phát triển các chức năng cơ bản của một hệ thống quản lý quán cà phê trực tuyến, bao gồm:

- Đăng nhập, đăng ký và quản lý tài khoản người dùng (khách hàng và quản trị viên).
- Xem menu món, chi tiết từng món kèm hình ảnh và giá bán.
- Đặt món trực tuyến hoặc tại quán thông qua mã QR của từng bàn.
- Quản lý giỏ hàng và đơn hàng: thêm, sửa, hủy món trong giỏ hàng, theo dõi trạng thái đơn (đang pha chế, đã phục vụ, đã thanh toán).
- Quản lý bàn và khu vực phục vụ cho người quản trị, bao gồm việc cập nhật tình trạng bàn (rảnh, bận, đặt trước).
- Quản lý danh mục món và sản phẩm: thêm, chỉnh sửa, xóa, cập nhật thông tin món.
- Thống kê và báo cáo doanh thu, giúp người quản trị theo dõi hiệu quả kinh doanh theo ngày, tháng, quý.

Các chức năng mở rộng như tích hợp thanh toán điện tử, chương trình khách hàng thân thiết, hoặc tính năng đặt bàn trước online có thể không được triển khai trong phạm vi hiện tại, mà chỉ được đề cập ở mức định hướng phát triển tương lai.

1.3.2. Về công nghệ

Đề tài sử dụng ReactJS cho phần frontend và Node.js kết hợp ExpressJS cho phần backend, đảm bảo khả năng phản hồi nhanh và dễ mở rộng. PostgreSQL được chọn làm hệ quản trị cơ sở dữ liệu chính vì ưu điểm về tính ổn định, bảo mật cao và khả năng xử lý dữ liệu phức tạp.

Phạm vi nghiên cứu không bao gồm việc tìm hiểu sâu về các công nghệ nâng cao như AI, Machine Learning, Big Data hay các framework ngoài phạm vi đề tài.

1.3.3. Về sản phẩm

Sản phẩm chính là một website quản lý quán café trực tuyến, bao gồm:

- Phần dành cho khách hàng: xem menu, đặt món, theo dõi đơn hàng.
- Phần dành cho quản trị viên: quản lý món, bàn, đơn hàng, người dùng và doanh thu.

Cơ sở dữ liệu được thiết kế chuẩn hóa, hỗ trợ mở rộng khi triển khai tại nhiều chi nhánh khác nhau. Sản phẩm hướng đến quán café tầm trung hoặc chuỗi nhỏ, nơi cần hệ thống quản lý linh hoạt và dễ vận hành.

1.3.4. Về giao diện

Giao diện website được thiết kế theo phong cách hiện đại, đơn giản và thân thiện, sử dụng màu sắc nhẹ nhàng, dễ nhìn, phù hợp với đặc trưng thương hiệu quán café.

Hệ thống ưu tiên khả năng hiển thị tốt trên thiết bị desktop và mobile, sử dụng Responsive Design để tự động điều chỉnh bố cục.

Các trang chính bao gồm: đơn hàng, bàn, thực đơn, quản lý menu, danh mục, báo cáo.

1.3.5. Về môi trường triển khai

Website được phát triển và chạy trên môi trường web sử dụng Node.js runtime kết hợp PostgreSQL server. Môi trường triển khai thử nghiệm bao gồm:

- Môi trường phát triển: chạy cục bộ trên localhost, Network IP.

- Hệ thống hoạt động tốt trên các trình duyệt hiện tại (Chrome, Brave, Edge).

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ

2.1. Tổng quan về hệ thống thông tin quản lý



Hình 2. 1 Tổng quan về hệ thống thông tin quản lý

Hệ thống thông tin quản lý (Management Information System – MIS) là một tập hợp được tổ chức bao gồm: **phần cứng (hardware)**, **phần mềm (software)**, **dữ liệu (data)**, **con người (people)** và **quy trình (procedures)** nhằm mục đích thu thập, xử lý, lưu trữ và phân phối thông tin để phục vụ cho công tác **quản lý, điều hành và hỗ trợ ra quyết định** trong tổ chức hoặc doanh nghiệp.

MIS đóng vai trò như một cầu nối giữa hoạt động thực tế và hệ thống quản trị của doanh nghiệp, giúp các cấp quản lý theo dõi và kiểm soát toàn bộ hoạt động một cách hệ thống, chính xác và kịp thời. Ngày nay, khi công nghệ thông tin phát triển mạnh mẽ, các hệ thống MIS không chỉ mang tính hỗ trợ mà còn trở thành **yếu tố trọng tâm** trong chiến lược phát triển doanh nghiệp.

2.1.1. Mục tiêu của hệ thống thông tin quản lý

Một hệ thống MIS được xây dựng nhằm đạt các mục tiêu chính sau:

- **Tự động hóa quy trình:** Giảm thiểu các công việc thủ công, hạn chế sai sót do con người và tăng tốc độ xử lý.

- **Quản lý dữ liệu tập trung:** Tất cả thông tin được lưu trữ trong một hệ thống thống nhất, dễ truy xuất và bảo trì.
- **Nâng cao hiệu quả quản lý:** Cung cấp dữ liệu chính xác giúp nhà quản lý đưa ra quyết định nhanh chóng và hiệu quả.
- **Tiết kiệm chi phí vận hành:** Giảm nhu cầu nhân sự ở những khâu tác vụ đơn giản, tối ưu nguồn lực.
- **Hỗ trợ phân tích hoạt động kinh doanh:** Thông qua các báo cáo, biểu đồ, biểu thống kê doanh thu, sản phẩm bán chạy...

2.1.2. Vai trò của MIS trong lĩnh vực quản lý quán café

Trong mô hình kinh doanh quán café, MIS có vai trò đặc biệt quan trọng vì khối lượng giao dịch cao, quy trình phục vụ nhiều bước và yêu cầu chính xác gần như tuyệt đối. MIS giúp:

- Quản lý menu và giá bán
 - Dễ dàng cập nhật thông tin món, giá thành, hình ảnh.
 - Phân loại món theo danh mục giúp khách hàng dễ tìm kiếm.
- Quản lý bàn và khu vực phục vụ
 - Theo dõi trạng thái bàn (Rảnh khách / đã có khách ngồi).
 - Hỗ trợ quét QR để khách hàng tự xem menu và gọi món.
- Quản lý đơn hàng theo thời gian thực
 - Đơn hàng được gửi trực tiếp đến quầy bar hoặc bếp.
 - Giảm thiểu sai sót giữa nhân viên phục vụ và pha chế.
 - Cập nhật trạng thái đơn hàng theo từng bước.
- Quản lý nhân viên và phân quyền
 - Quản lý tài khoản: admin, thu ngân, nhân viên phục vụ.
 - Theo dõi lịch làm việc, ca làm, hiệu suất làm việc.
- Báo cáo doanh thu, phân tích hoạt động kinh doanh
 - Tự động thống kê doanh thu hàng ngày, hàng tuần, hàng tháng.
 - Xác định sản phẩm bán chạy, khung giờ đông khách.
 - Hỗ trợ quản lý đưa ra quyết định điều chỉnh giá, số lượng nhân viên hoặc chính sách kinh doanh.

Nhờ MIS, toàn bộ hoạt động của quán café được **tự động hóa – ghi nhận chính xác – kiểm soát chặt chẽ**, giúp nâng cao hiệu quả vận hành và trải nghiệm khách hàng.

2.1.3. Sự cần thiết của hệ thống thông tin quản lý trong việc quản lý quán cà phê hiện nay

Trong bối cảnh chuyển đổi số, việc ứng dụng MIS trong kinh doanh quán cà phê là **xu hướng tất yếu**, xuất phát từ nhu cầu:

- **Tăng sự cạnh tranh:** Quán cà phê cần nâng cao chất lượng và tốc độ phục vụ.
- **Giảm sai sót:** Việc ghi order thủ công dễ gây nhầm lẫn, gây mất thiện cảm đối với khách hàng.
- **Theo dõi doanh thu minh bạch:** Chủ quán có thể theo dõi từ xa, tránh thất thoát.
- **Tối ưu nhân lực:** Giảm chi phí nhân viên phục vụ nhờ tự động đặt món thông qua việc tự động hoá đặt món qua QR.
- **Quản lý nhiều chi nhánh:** MIS giúp đồng bộ dữ liệu toàn hệ thống.

Với mô hình cà phê hiện đại, việc ứng dụng MIS không chỉ là lợi thế mà còn là yêu cầu bắt buộc để vận hành ổn định, chuyên nghiệp và có khả năng mở rộng.

2.1.4. Liên hệ với đề tài

Trong đề tài “*Ứng dụng React, Node.js và PostgreSQL trong xây dựng hệ thống quản lý quán Cà phê*”, việc triển khai MIS được thể hiện qua:

- **ReactJS:** Xây dựng giao diện người dùng trực quan, thân thiện.
- **Node.js:** Xử lý nghiệp vụ, quản lý luồng dữ liệu và giao tiếp với cơ sở dữ liệu.
- **PostgreSQL:** Lưu trữ tập trung, bảo mật toàn bộ dữ liệu như món, đơn hàng, bàn, nhân viên, ...

Hệ thống được thiết kế đáp ứng nhu cầu của mô hình quán cà phê thực tế, đồng thời đảm bảo khả năng mở rộng trong tương lai.

2.2. Công cụ và cài đặt

2.2.1. Visual Studio Code



Hình 2. 2 Visual Studio Code

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mạnh mẽ và hoàn toàn miễn phí, được phát triển bởi Microsoft. VS Code hỗ trợ nhiều ngôn ngữ lập trình phổ biến như HTML, CSS, JavaScript, TypeScript, Python, PHP,... và đặc biệt có khả năng mở rộng linh hoạt thông qua các tiện ích mở rộng (extensions). Điều này giúp lập trình viên có thể tùy chỉnh môi trường làm việc phù hợp với từng công nghệ sử dụng, chẳng hạn như ReactJS, Node.js, hay PostgreSQL – những công nghệ chính được áp dụng trong đề tài.

VS Code cung cấp nhiều tính năng hỗ trợ lập trình viên như gợi ý mã (IntelliSense), kiểm tra lỗi cú pháp, tích hợp Git, debug trực tiếp, và Live Server giúp xem kết quả ngay lập tức khi chỉnh sửa. Trong đề tài này, VS Code được sử dụng làm môi trường phát triển chính cho cả frontend và backend. Với giao diện thân thiện và khả năng mở rộng qua các tiện ích như ESLint, Prettier, React Developer Tools và REST Client, VS Code giúp quá trình lập trình, kiểm thử và quản lý mã nguồn trở nên nhanh chóng, hiệu quả và thuận tiện hơn.

Ưu điểm:

- **Miễn phí và nhẹ:** Dung lượng nhỏ, khởi động nhanh, hoạt động ổn định trên nhiều hệ điều hành (Windows, macOS, Linux).

- **Hỗ trợ đa ngôn ngữ:** Làm việc tốt với HTML, CSS, JavaScript, TypeScript, Python, PHP, React, Node.js,...
- **Tích hợp Git:** Cho phép commit, push, pull, và theo dõi lịch sử thay đổi mã nguồn trực tiếp trong IDE.
- **Tính năng gợi ý mã thông minh (IntelliSense):** Tự động hoàn thành, gợi ý biến, hàm, và cú pháp giúp lập trình nhanh và chính xác hơn.
- **Hỗ trợ Debug:** Cho phép chạy và kiểm tra lỗi chương trình trực tiếp trong VS Code mà không cần công cụ ngoài.
- **Hệ sinh thái mở rộng:** Cung cấp hàng nghìn plugin như ESLint, Prettier, Live Server, React Developer Tools,... hỗ trợ nhiều framework khác nhau.
- **Live Preview:** Hiện thị kết quả thay đổi giao diện ngay lập tức, giúp tối ưu quá trình phát triển frontend.
- **Giao diện thân thiện và tùy biến cao:** Có thể thay đổi chủ đề, phím tắt, và cấu hình theo phong cách cá nhân.

Nhược điểm:

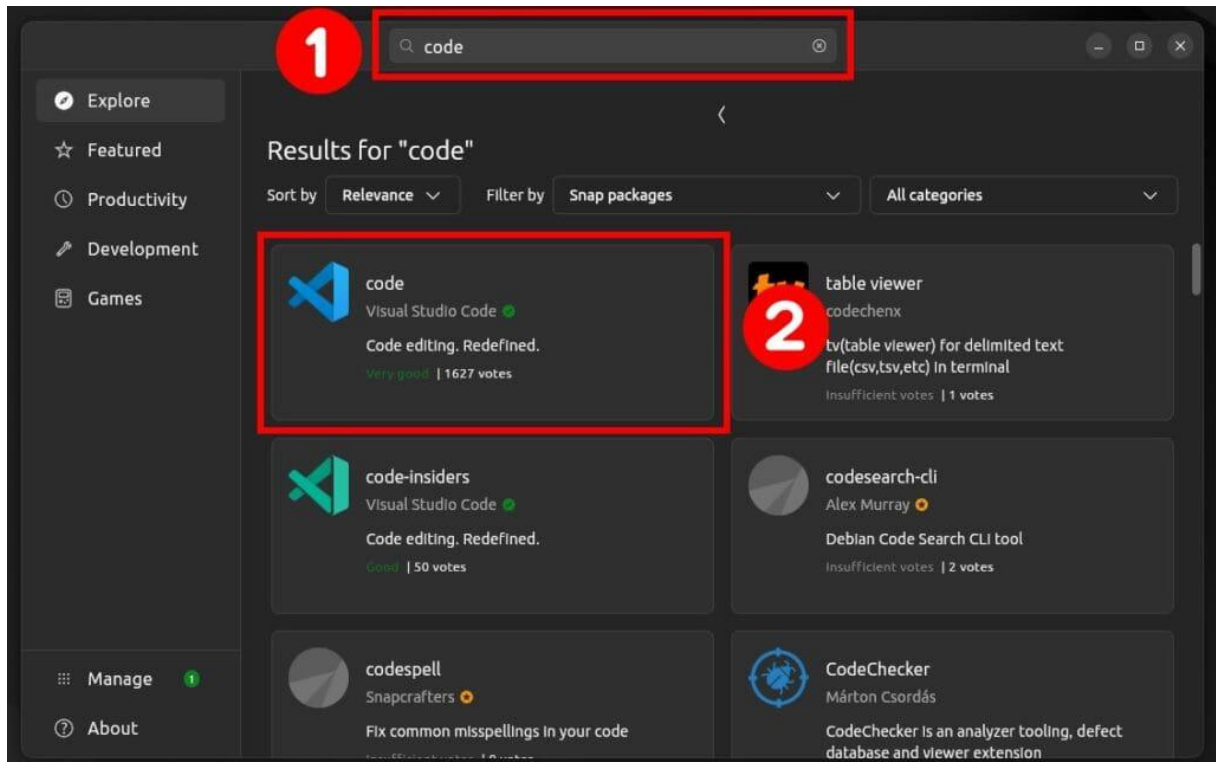
- **Phụ thuộc vào tiện ích mở rộng:** Mặc dù linh hoạt, nhưng VS Code cần cài đặt nhiều plugin để hỗ trợ đầy đủ chức năng, dễ gây xung đột hoặc làm chậm chương trình nếu cài quá nhiều.
- **Hiệu năng giảm khi mở dự án lớn:** Khi làm việc với các dự án React hoặc Node.js có nhiều tệp tin, VS Code có thể tiêu tốn nhiều RAM và CPU.
- **Cấu hình ban đầu phức tạp:** Người mới bắt đầu cần thời gian để cài đặt và cấu hình các tiện ích mở rộng cần thiết.
- **Hạn chế tính năng nâng cao so với IDE chuyên dụng:** Một số tính năng như thiết kế giao diện trực quan, quản lý cơ sở dữ liệu hoặc debug nâng cao không mạnh bằng các IDE như WebStorm hay IntelliJ.
- **Cập nhật thường xuyên:** Việc cập nhật liên tục đôi khi gây lỗi không tương thích với plugin cũ.

Các bước cài đặt:

Vì dự án được thực hiện trên Linux (cụ thể là Ubuntu) nên sẽ chỉ tập trung hướng dẫn cài đặt cho Ubuntu

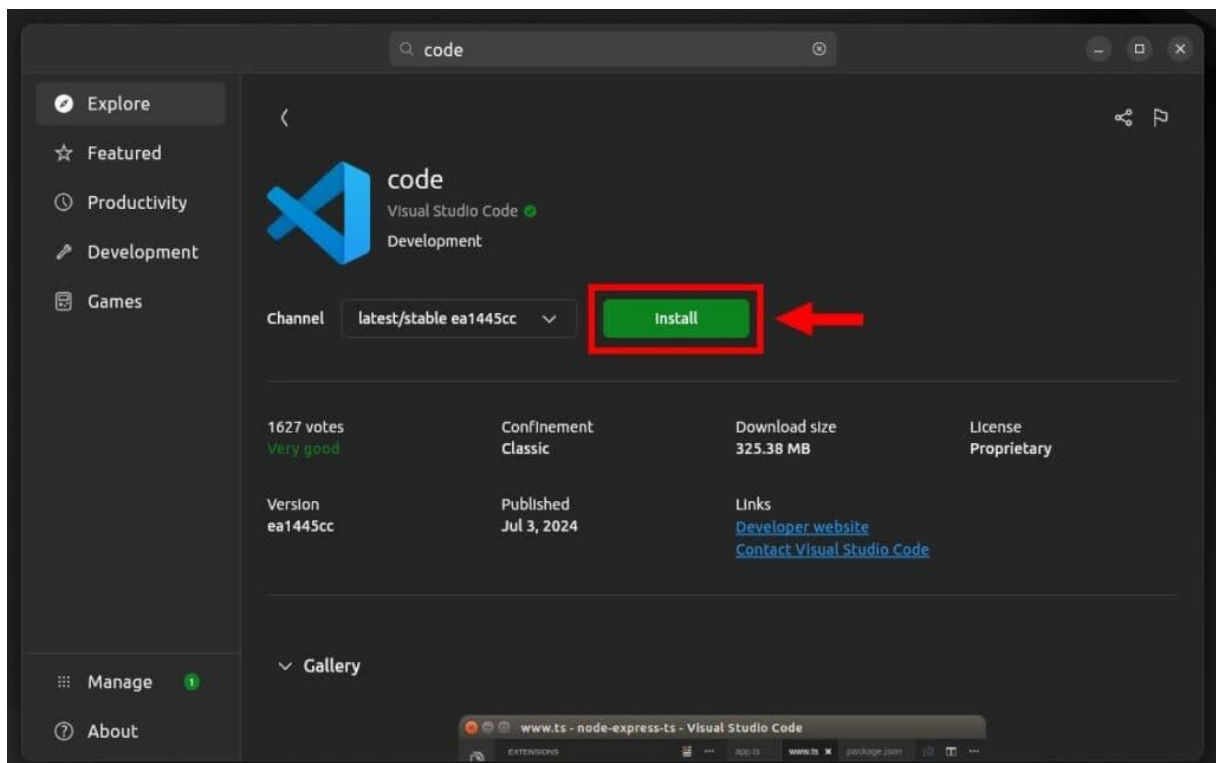
Cách 1: cài đặt VS Code trên App Center:

B1: Vào App Center, trong ô tìm kiếm gõ từ “code”



Hình 2. 3 Tìm kiếm VSCode trong AppCenter

B2: Vào ứng dụng có biểu tượng của VS Code và nhấn nút install



Hình 2. 4 Tiến hành cài đặt

Cách 2: cài đặt bằng lệnh Terminal:

B1: Vào trang web <https://code.visualstudio.com/Download>

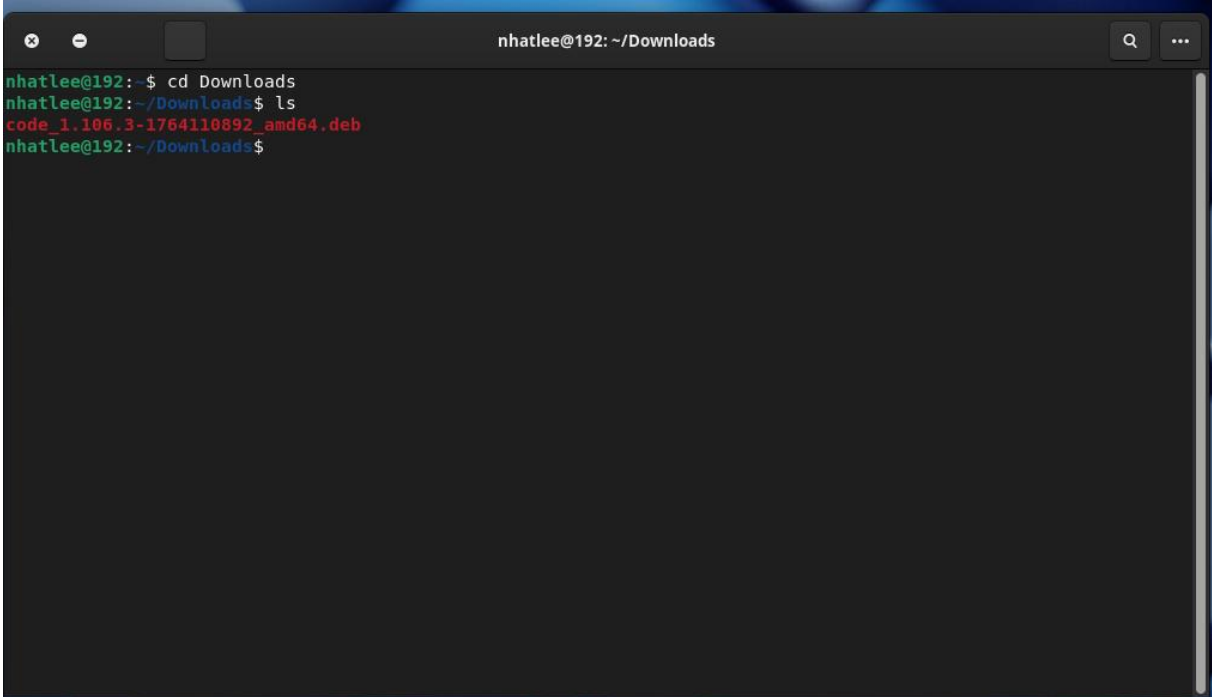
B2: Chọn vào mục .deb



Hình 2. 5 Tải file cài đặt cho Ubuntu

B3: Sau khi tải về thành công nhấn nút Super (Window), gõ tìm kiếm Terminal hoặc nhấn tổ hợp phím Ctrl + Alt + T để mở ứng dụng Terminal.

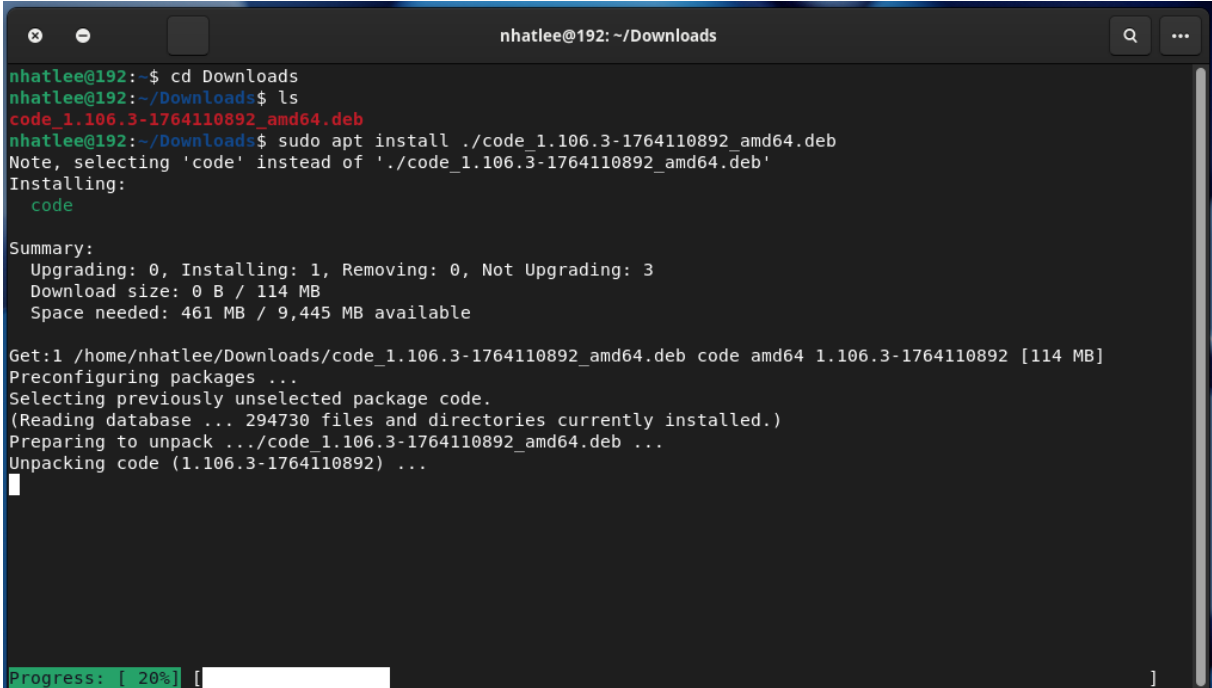
B4: Điều hướng đến thư mục chứa file vừa tải về, ở đây tôi lưu vào thư mục Downloads bằng lệnh “cd Downloads”.

A terminal window titled 'nhatlee@192: ~/Downloads' showing the following commands and output:

```
nhatlee@192:~$ cd Downloads
nhatlee@192:~/Downloads$ ls
code_1.106.3-1764110892_amd64.deb
nhatlee@192:~/Downloads$
```

Hình 2. 6 Truy cập vào thư mục chứa file cài đặt bằng Terminal

B5: Thực hiện cài đặt bằng lệnh “sudo apt install ./tên_file.deb”, nhập mật khẩu người dùng và nhấn Enter, hộp thoại yêu cầu xác nhận chọn Yes.

A terminal window titled 'nhatlee@192: ~/Downloads' showing the installation process:

```
nhatlee@192:~$ cd Downloads
nhatlee@192:~/Downloads$ ls
code_1.106.3-1764110892_amd64.deb
nhatlee@192:~/Downloads$ sudo apt install ./code_1.106.3-1764110892_amd64.deb
Note, selecting 'code' instead of './code_1.106.3-1764110892_amd64.deb'
Installing:
code

Summary:
Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 3
Download size: 0 B / 114 MB
Space needed: 461 MB / 9,445 MB available

Get:1 /home/nhatlee/Downloads/code_1.106.3-1764110892_amd64.deb code amd64 1.106.3-1764110892 [114 MB]
Preconfiguring packages ...
Selecting previously unselected package code.
(Reading database ... 294730 files and directories currently installed.)
Preparing to unpack .../code_1.106.3-1764110892_amd64.deb ...
Unpacking code (1.106.3-1764110892) ...
Progress: [ 20%] [ ]
```

Hình 2. 7 Tiến hành cài đặt

2.2.2. Node.JS



Hình 2. 8 Node.JS

Node.js là một nền tảng chạy mã JavaScript phía máy chủ (server-side), được xây dựng trên V8 Engine của Google Chrome và phát triển bởi Ryan Dahl vào năm 2009.

Khác với JavaScript chỉ chạy trên trình duyệt, Node.js cho phép lập trình viên sử dụng cùng một ngôn ngữ để phát triển cả frontend và backend, giúp tiết kiệm thời gian và dễ quản lý mã nguồn.

Trong đề tài này, Node.js được sử dụng để xây dựng phần backend của hệ thống quản lý quán café. Nó xử lý yêu cầu từ người dùng, tương tác với cơ sở dữ liệu PostgreSQL, xác thực người dùng, quản lý đơn hàng, và cung cấp API cho phần frontend ReactJS thông qua mô hình RESTful API.

Ưu điểm:

- Hiệu năng cao: Node.js sử dụng mô hình bất đồng bộ (asynchronous) và cơ chế non-blocking I/O, giúp xử lý đồng thời nhiều yêu cầu mà không bị treo server.
- Dùng chung một ngôn ngữ: Cả frontend và backend đều sử dụng JavaScript, giúp giảm độ phức tạp và chi phí học tập.
- Kho thư viện phong phú: Hệ sinh thái npm (Node Package Manager) cung cấp hàng triệu gói hỗ trợ cho việc phát triển nhanh chóng.
- Dễ mở rộng và triển khai: Phù hợp cho các ứng dụng thời gian thực như chat, đặt món, theo dõi đơn hàng, ...

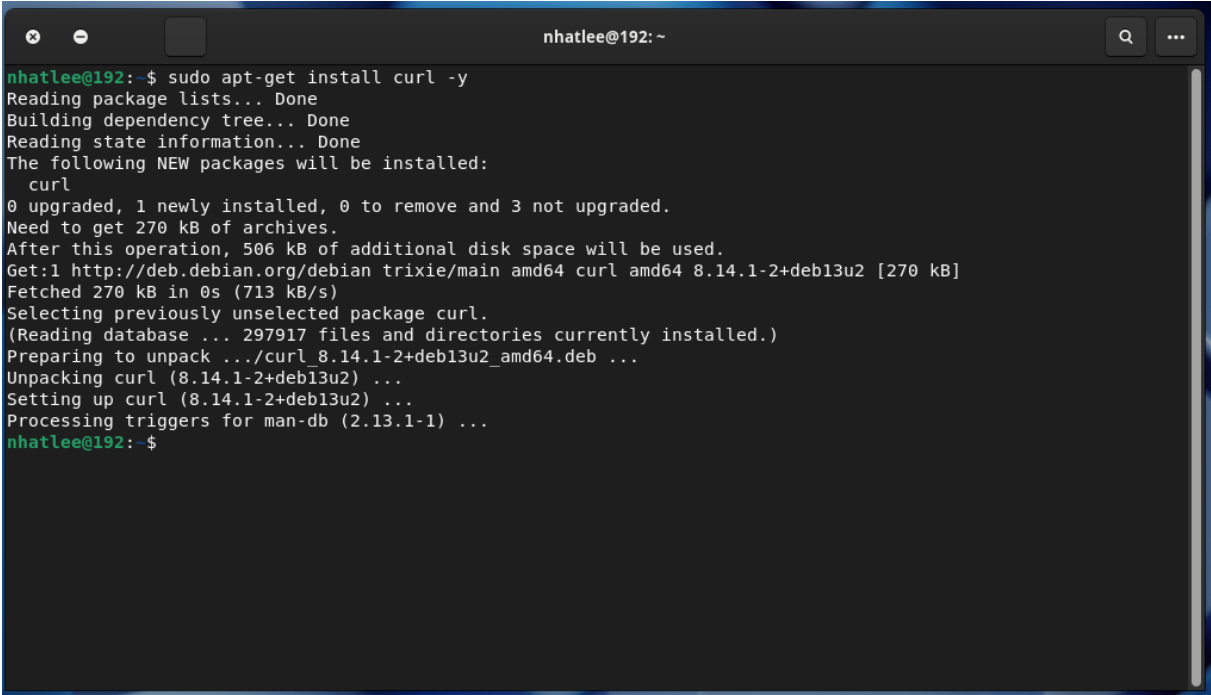
- Cộng đồng lớn: Có tài liệu và hỗ trợ đa dạng, dễ tìm kiếm giải pháp khi gặp lỗi.

Nhược điểm:

- Không phù hợp cho tác vụ tính toán nặng: Do sử dụng đơn luồng, Node.js không mạnh trong các tác vụ CPU-intensive (ví dụ: xử lý hình ảnh, thuật toán phức tạp).
- Xử lý lỗi bất đồng bộ phức tạp: Việc quản lý callback hoặc promise sai có thể dẫn đến lỗi khó kiểm soát.
- Bảo mật phụ thuộc thư viện: Một số gói npm có thể chứa lỗ hổng bảo mật nếu không được kiểm tra kỹ.
- Không có chuẩn cấu trúc cố định: Mỗi lập trình viên có thể tổ chức code khác nhau, gây khó khăn cho việc bảo trì nếu làm việc nhóm lớn.

Các bước cài đặt

B1: Cài đặt Curl bằng lệnh “sudo apt-get install curl -y”

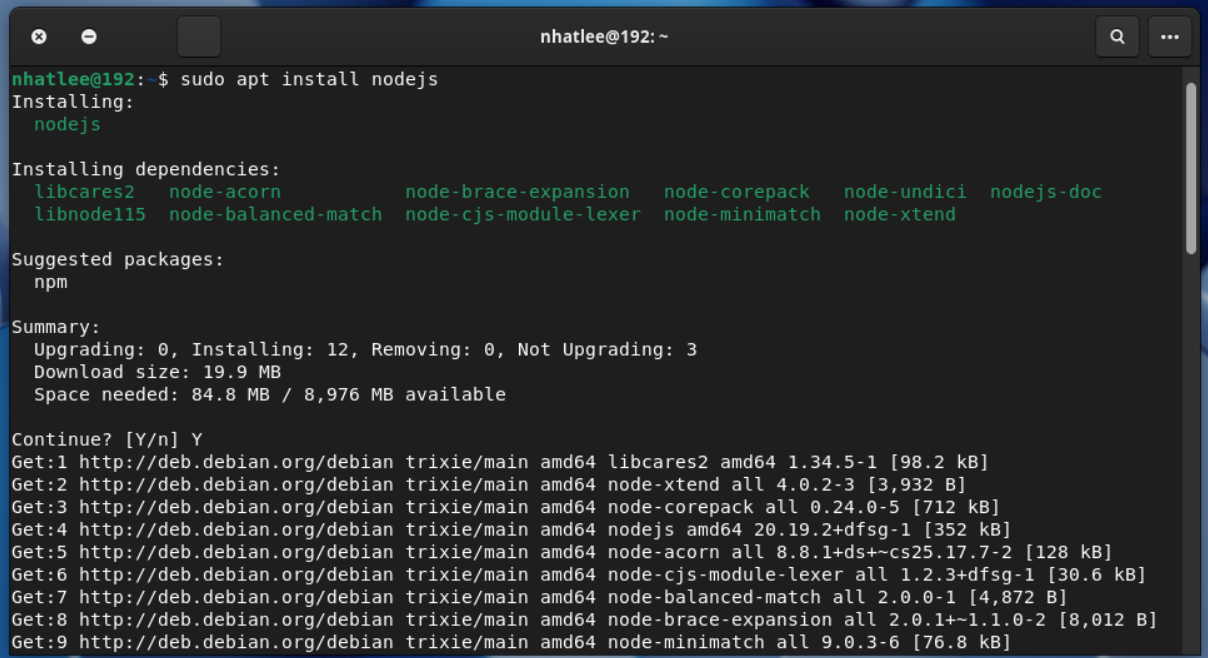


```
nhatlee@192: ~  
nhatlee@192:~$ sudo apt-get install curl -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following NEW packages will be installed:  
  curl  
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.  
Need to get 270 kB of archives.  
After this operation, 506 kB of additional disk space will be used.  
Get:1 http://deb.debian.org/debian trixie/main amd64 curl amd64 8.14.1-2+deb13u2 [270 kB]  
Fetched 270 kB in 0s (713 kB/s)  
Selecting previously unselected package curl.  
(Reading database ... 297917 files and directories currently installed.)  
Preparing to unpack .../curl_8.14.1-2+deb13u2_amd64.deb ...  
Unpacking curl (8.14.1-2+deb13u2) ...  
Setting up curl (8.14.1-2+deb13u2) ...  
Processing triggers for man-db (2.13.1-1) ...  
nhatlee@192:~$
```

Hình 2. 9 Cài đặt Curl

B2: Kích hoạt repository bằng lệnh “curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -”

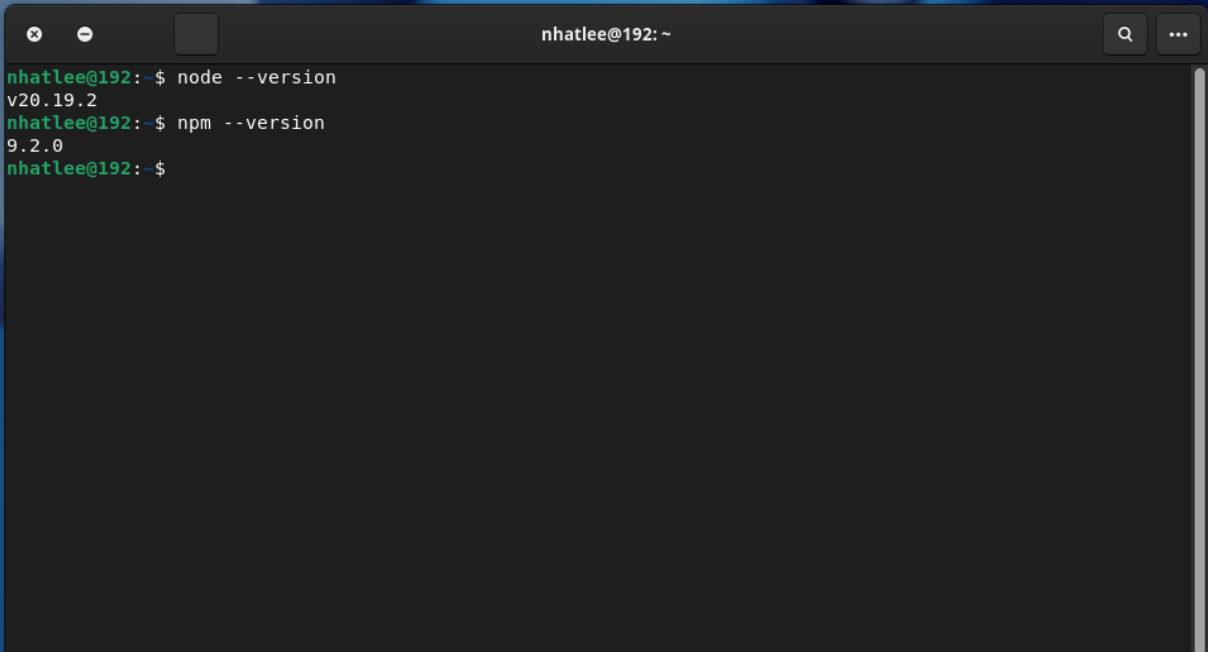
B3: Cài đặt Node.JS và Npm bằng lệnh “sudo apt install nodejs” và “sudo apt install npm”



```
nhatlee@192: ~  
nhatlee@192:~$ sudo apt install nodejs  
Installing:  
  nodejs  
  
Installing dependencies:  
  libcares2  node-acorn  node-brace-expansion  node-corepack  node-undici  nodejs-doc  
  libnode115  node-balanced-match  node-cjs-module-lexer  node-minimatch  node-xtend  
  
Suggested packages:  
  npm  
  
Summary:  
  Upgrading: 0, Installing: 12, Removing: 0, Not Upgrading: 3  
  Download size: 19.9 MB  
  Space needed: 84.8 MB / 8,976 MB available  
  
Continue? [Y/n] Y  
Get:1 http://deb.debian.org/debian trixie/main amd64 libcares2 amd64 1.34.5-1 [98.2 kB]  
Get:2 http://deb.debian.org/debian trixie/main amd64 node-xtend all 4.0.2-3 [3,932 B]  
Get:3 http://deb.debian.org/debian trixie/main amd64 node-corepack all 0.24.0-5 [712 kB]  
Get:4 http://deb.debian.org/debian trixie/main amd64 nodejs amd64 20.19.2+dfsg-1 [352 kB]  
Get:5 http://deb.debian.org/debian trixie/main amd64 node-acorn all 8.8.1+ds+~cs25.17.7-2 [128 kB]  
Get:6 http://deb.debian.org/debian trixie/main amd64 node-cjs-module-lexer all 1.2.3+dfsg-1 [30.6 kB]  
Get:7 http://deb.debian.org/debian trixie/main amd64 node-balanced-match all 2.0.0-1 [4,872 B]  
Get:8 http://deb.debian.org/debian trixie/main amd64 node-brace-expansion all 2.0.1+~1.1.0-2 [8,012 B]  
Get:9 http://deb.debian.org/debian trixie/main amd64 node-minimatch all 9.0.3-6 [76.8 kB]
```

Hình 2. 10 Cài đặt Node.JS

B4: Xác minh cài đặt Node.JS, Npm và kiểm tra phiên bản của Npm lần lượt là “node -version” và “npm --version”



```
nhatlee@192: ~  
nhatlee@192:~$ node --version  
v20.19.2  
nhatlee@192:~$ npm --version  
9.2.0  
nhatlee@192:~$
```

Hình 2. 11 Kiểm tra phiên bản của node.JS và Npm

2.2.3. PostgreSQL



Hình 2. 12 PostgreSQL

PostgreSQL là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở mạnh mẽ, được phát triển từ Đại học California tại Berkeley và duy trì bởi cộng đồng toàn cầu. PostgreSQL tuân thủ chặt chẽ chuẩn SQL, đồng thời hỗ trợ các tính năng nâng cao như trigger, stored procedure, foreign key, transaction, và JSON cho dữ liệu phi cấu trúc.

Trong đề tài này, PostgreSQL được sử dụng làm cơ sở dữ liệu chính để lưu trữ toàn bộ thông tin của hệ thống quản lý quán cà phê như: người dùng, món uống, bàn, đơn hàng, chi tiết hóa đơn, và thống kê doanh thu. Việc lựa chọn PostgreSQL giúp đảm bảo tính toàn vẹn, ổn định và bảo mật dữ liệu trong quá trình vận hành hệ thống.

Ưu điểm:

- **Mã nguồn mở, miễn phí:** Có thể sử dụng, tùy chỉnh và mở rộng mà không mất phí bản quyền.
- **Độ tin cậy cao:** Hỗ trợ **ACID (Atomicity, Consistency, Isolation, Durability)** đảm bảo tính toàn vẹn dữ liệu trong mọi giao dịch.
- **Hiệu năng mạnh mẽ:** Xử lý tốt các truy vấn phức tạp, phù hợp với hệ thống có lượng dữ liệu lớn.
- **Hỗ trợ dữ liệu phong phú:** Lưu trữ cả dữ liệu dạng **quan hệ** và **phi quan hệ (JSON, XML)**.
- **Tính mở rộng cao:** Cho phép định nghĩa hàm, kiểu dữ liệu, và chỉ mục riêng phù hợp với nhu cầu.

- **Bảo mật tốt:** Hỗ trợ mã hóa dữ liệu, phân quyền người dùng, và xác thực truy cập linh hoạt.
- **Tương thích đa nền tảng:** Hoạt động tốt trên Windows, Linux, macOS.

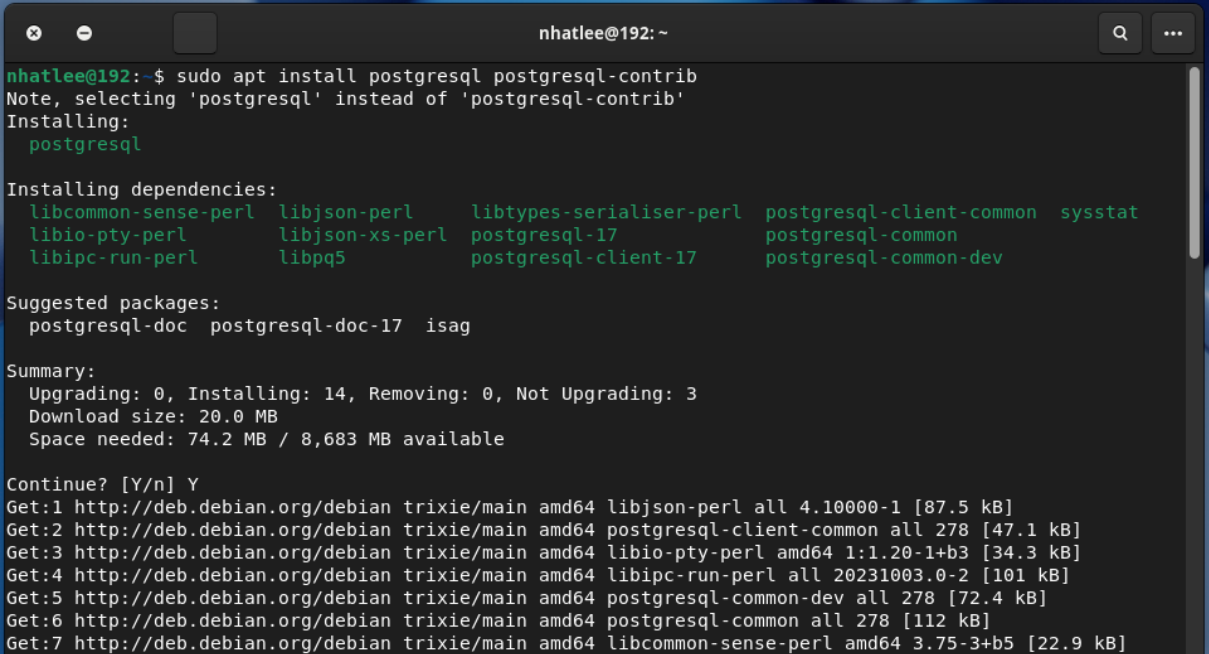
Nhược điểm:

- **Cấu hình ban đầu phức tạp:** Cần thiết lập chi tiết về quyền truy cập, cổng, và file cấu hình (postgresql.conf).
- **Hiệu năng thấp hơn MySQL trong các tác vụ nhỏ:** Do tối ưu cho hệ thống lớn, việc xử lý truy vấn đơn giản có thể chậm hơn một chút.
- **Công cụ hỗ trợ đồ họa hạn chế:** Dù có **pgAdmin 4**, nhưng chưa thân thiện bằng các công cụ thương mại như SQL Server Management Studio.
- **Yêu cầu hiểu biết về SQL nâng cao:** Khả năng mở rộng cao nhưng đòi hỏi người dùng có kỹ năng quản trị cơ sở dữ liệu tốt.

Các bước cài đặt

Cài đặt PostgreSQL:

B1: Cài đặt package Postgres cùng với một package -contrib có các tiện ích và chức năng bổ sung bằng lệnh “sudo apt install postgresql postgresql-contrib”



```
nhatlee@192: ~  
nhatlee@192:~$ sudo apt install postgresql postgresql-contrib  
Note, selecting 'postgresql' instead of 'postgresql-contrib'  
Installing:  
  postgresql  
  
Installing dependencies:  
  libcommon-sense-perl  libjson-perl  libtypes-serialiser-perl  postgresql-client-common  sysstat  
  libio-pty-perl        libjson-xs-perl  postgresql-17          postgresql-common  
  libipc-run-perl       libpq5          postgresql-client-17   postgresql-common-dev  
  
Suggested packages:  
  postgresql-doc  postgresql-doc-17  isag  
  
Summary:  
  Upgrading: 0, Installing: 14, Removing: 0, Not Upgrading: 3  
  Download size: 20.0 MB  
  Space needed: 74.2 MB / 8,683 MB available  
  
Continue? [Y/n] Y  
Get:1 http://deb.debian.org/debian trixie/main amd64 libjson-perl all 4.10000-1 [87.5 kB]  
Get:2 http://deb.debian.org/debian trixie/main amd64 postgresql-client-common all 278 [47.1 kB]  
Get:3 http://deb.debian.org/debian trixie/main amd64 libio-pty-perl amd64 1:1.20-1+b3 [34.3 kB]  
Get:4 http://deb.debian.org/debian trixie/main amd64 libipc-run-perl all 20231003.0-2 [101 kB]  
Get:5 http://deb.debian.org/debian trixie/main amd64 postgresql-common-dev all 278 [72.4 kB]  
Get:6 http://deb.debian.org/debian trixie/main amd64 postgresql-common all 278 [112 kB]  
Get:7 http://deb.debian.org/debian trixie/main amd64 libcommon-sense-perl amd64 3.75-3+b5 [22.9 kB]
```

Hình 2. 13 Cài đặt PostgreSQL

B2: Kiểm tra dịch vụ bằng lệnh “sudo systemctl start postgresql.service”

Cài đặt pgAdmin4:

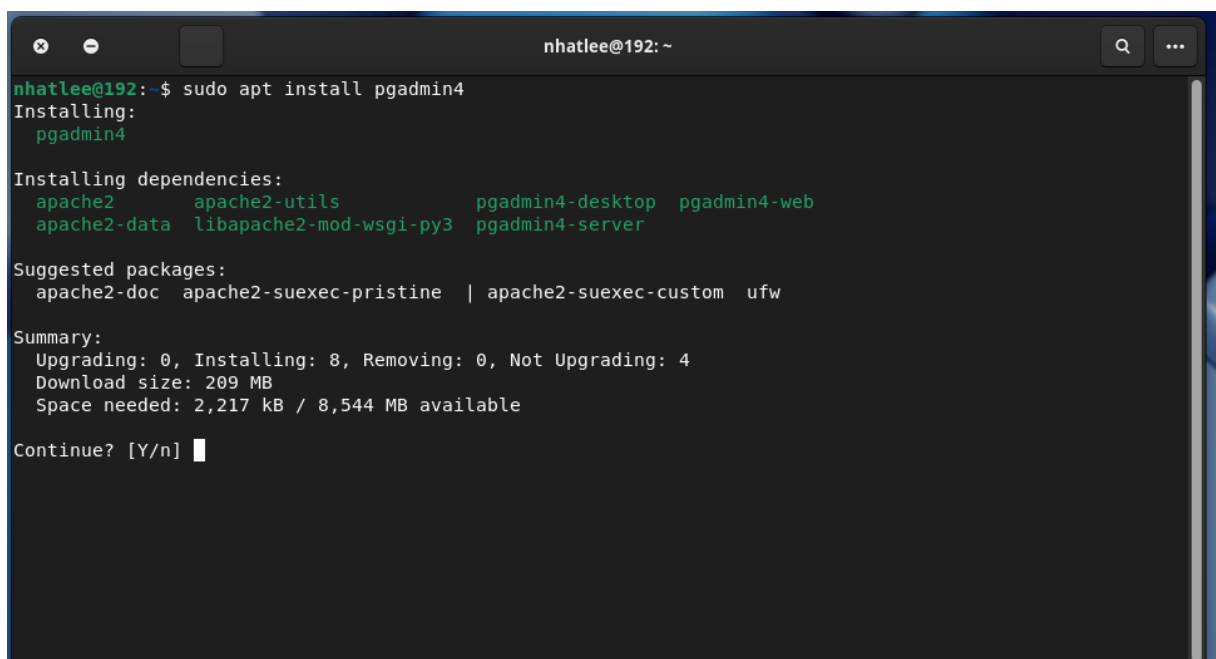
B1: Cài đặt khoá công khai cho lưu trữ (nếu chưa thực hiện trước đó): `curl -fsS https://www.pgadmin.org/static/packages_pgadmin_org.pub | sudo gpg --dearmor -o /usr/share/keyrings/packages-pgadmin-org.gpg`

B2: Tạo tệp cấu hình kho lưu trữ: `sudo sh -c 'echo "deb [signed-by=/usr/share/keyrings/packages-pgadmin-org.gpg]`

`https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/$(lsb_release -cs) pgadmin4 main" > /etc/apt/sources.list.d/pgadmin4.list && apt update'`

B3: Cài đặt pgadmin4

- Cài đặt cho cả chế độ desktop và web: `sudo apt install pgadmin4`
- Chỉ cài đặt môi trường desktop: `sudo apt install pgadmin4-desktop`
- Chỉ cài đặt cho chế độ web: `sudo apt install pgadmin4-web`

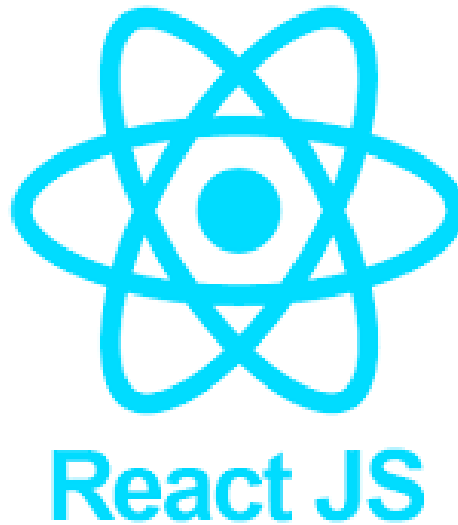


```
nhatlee@192: ~  
nhatlee@192:~$ sudo apt install pgadmin4  
Installing:  
pgadmin4  
  
Installing dependencies:  
apache2 apache2-utils pgadmin4-desktop pgadmin4-web  
apache2-data libapache2-mod-wsgi-py3 pgadmin4-server  
  
Suggested packages:  
apache2-doc apache2-suexec-pristine | apache2-suexec-custom ufw  
  
Summary:  
Upgrading: 0, Installing: 8, Removing: 0, Not Upgrading: 4  
Download size: 209 MB  
Space needed: 2,217 kB / 8,544 MB available  
  
Continue? [Y/n]
```

Hình 2. 14 Cài đặt pgadmin4

B4: Cấu hình máy chủ web, nếu bạn đã cài đặt pgadmin4-web: `sudo /usr/pgadmin4/bin/setup-web.sh`

2.2.4. React



Hình 2. 15 ReactJS

ReactJS là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook (nay là Meta), ra mắt lần đầu năm 2013. React được sử dụng để xây dựng giao diện người dùng (User Interface) cho các ứng dụng web hiện đại, đặc biệt là các hệ thống có nhiều thành phần (component) tái sử dụng và cần cập nhật dữ liệu nhanh.

Trong đề tài này, ReactJS được dùng để phát triển giao diện website quản lý quán café, bao gồm cả phần khách hàng (xem menu, đặt món, theo dõi đơn hàng) và phần quản trị viên (quản lý món, bàn, doanh thu, người dùng). React giúp tạo ra giao diện tương tác linh hoạt, phản hồi nhanh và thân thiện với người dùng.

Ưu điểm:

- **Tốc độ cao với Virtual DOM:** React chỉ cập nhật phần thay đổi trên giao diện thay vì tải lại toàn bộ trang, giúp tăng hiệu năng đáng kể.
- **Tái sử dụng component:** Cho phép chia nhỏ giao diện thành nhiều component độc lập, dễ bảo trì và mở rộng.
- **Kết hợp dễ dàng với backend:** Dễ dàng tích hợp với **Node.js RESTful API**, giúp truyền dữ liệu mượt mà giữa frontend và backend.
- **Cộng đồng lớn, tài liệu phong phú:** Có nhiều thư viện hỗ trợ (React Router, Redux, Axios, Bootstrap, TailwindCSS, v.v.).
- **Hỗ trợ SPA (Single Page Application):** Cho phép xây dựng ứng dụng web hoạt động mượt mà mà không cần tải lại trang.

- **Tương thích đa nền tảng:** Có thể phát triển tiếp thành ứng dụng di động bằng React Native.

Nhược điểm:

- **Cần học thêm công cụ hỗ trợ:** Để phát triển đầy đủ, lập trình viên cần nắm vững thêm Redux, Router, Hooks, ...
- **Thay đổi nhanh:** React cập nhật thường xuyên, có thể khiến một số thư viện hoặc cú pháp cũ không còn tương thích.
- **Không phải framework hoàn chỉnh:** React chỉ tập trung vào phần hiển thị, nên các chức năng khác như routing, quản lý trạng thái, xác thực,... phải cài thêm thư viện.
- **Khó tối ưu SEO:** Vì là ứng dụng SPA, việc tối ưu SEO cần thêm cấu hình (ví dụ: Next.js hoặc SSR).

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Khảo sát hiện trạng quản lý quán café

Qua khảo sát một số mô hình quán café truyền thống, phần lớn quy trình quản lý hiện nay vẫn còn mang tính thủ công, thiếu sự đồng bộ và dễ xảy ra sai sót. Một số vấn đề phổ biến:

- Quản lý menu và giá bán
 - Thay đổi giá thường được cập nhật thủ công.
 - Nhân viên dễ nhầm lẫn món khi menu gồm nhiều loại đồ uống.
 - Ít có công cụ theo dõi món bán chạy hay đánh giá nhu cầu khách hàng.
- Quản lý bàn và khu vực
 - Trạng thái bàn thường được theo dõi bằng cách xem bàn đó đã có khách ngồi rồi hay chưa hoặc ghi chú miệng.
 - Tình trạng bàn không được cập nhật theo thời gian thực.
- Quy trình gọi món
 - Nhân viên ghi order bằng giấy dễ dẫn đến sai sót (ghi nhầm món, nhầm bàn).
 - Khi quán đông, thời gian nhập đơn vào máy thu ngân lâu làm ảnh hưởng đến trải nghiệm khách hàng.
- Quản lý đơn hàng và thanh toán
 - Đơn hàng không được lưu trữ tập trung.
 - Việc tổng kết doanh thu cuối ngày hoặc giao ca mất nhiều thời gian, dễ thất thoát.
 - Không theo dõi được lịch sử đơn hàng của khách.
- Quản lý nhân viên
 - Chăm công, theo dõi công việc chưa có hệ thống cụ thể.
 - Khó đánh giá hiệu suất và phân quyền rõ ràng.
- Báo cáo và thống kê
 - Các báo cáo thường được lập thủ công bằng Excel.
 - Chưa có công cụ phân tích doanh thu, sản phẩm bán chạy ... theo thời gian.

Kết luận: Việc xây dựng quản lý hệ thống là cần thiết để tối ưu vận hành, giảm sai sót và nâng cao chất lượng phục vụ.

3.2. Yêu cầu chức năng của hệ thống

Dựa trên kết quả khảo sát và phân tích thực tế hoạt động tại các quán cà phê, hệ thống cần đáp ứng đầy đủ các chức năng nhằm hỗ trợ người dùng (khách hàng) và quản trị viên (chủ quán/nhân viên). Các yêu cầu chức năng được chia thành hai nhóm chính như sau:

3.2.1. Chức năng dành cho khách hàng

Khách hàng là đối tượng trực tiếp sử dụng giao diện đặt món. Hệ thống phải cung cấp các chức năng giúp khách hàng dễ dàng thao tác, nhanh chóng và chính xác.

- Đăng ký tài khoản
 - Khách hàng có thể tạo mới tài khoản.
 - Cho phép cập nhật thông tin tài khoản sau khi tạo.
- Đăng nhập hệ thống
 - Xác thực thông tin đăng nhập.
 - Mã hoá mật khẩu để đảm bảo an toàn.
- Xem danh sách menu
 - Hiển thị toàn bộ menu theo danh mục.
 - Hiện thị giá và mô tả món.
 - Tìm kiếm món ăn theo tên.
- Xem menu bằng mã QR
 - Khách hàng quét mã QR tại từng bàn.
 - Hệ thống tự xác định số bàn và trạng thái bàn.
- Thêm món vào giỏ hàng
 - Chọn số lượng món.
 - Tính tổng tiền theo thời gian thực.
 - Chỉnh sửa số lượng hoặc xoá món khỏi giỏ hàng.
- Đặt món
 - Đơn order được xác nhận và được gửi gắm hệ thống.
 - Hiện thị thông báo thành công.
- Theo dõi tình trạng đơn order
 - Theo dõi trạng thái: đang pha chế / đã phục vụ / hoàn tất.
 - Tự động cập nhật theo thời gian thực.
- Xem lịch sử đơn order

- Xem danh sách đơn đã đặt.
- Xem chi tiết từng hoá đơn.
- Cập nhật thông tin
 - Thay đổi mật khẩu, thông tin liên hệ.
 - Lưu lại lịch sử thay đổi.

3.2.2. Chức năng dành cho quản trị viên

Quản trị viên có quyền cao nhất, quản lý toàn bộ dữ liệu của hệ thống và điều hành hoạt động quán café.

- Đăng nhập quản trị viên
 - Bảo mật bằng tài khoản riêng.
 - Phân quyền: Admin – Nhân viên.
- Quản lý danh mục món
 - Thêm danh mục mới.
 - Chỉnh sửa tên danh mục.
 - Xoá danh mục thừa / không cần thiết.
 - Ràng buộc: chỉ được xoá khi danh mục không có món.
- Quản lý món uống
 - Thêm món mới.
 - Chỉnh sửa thông tin món.
 - Xoá món khỏi menu.
 - Hiển thị danh sách món theo danh mục.
- Quản lý bàn
 - Thêm bàn mới với số bàn.
 - Sinh mã QR tự động cho mỗi bàn.
 - Cập nhật trạng thái bàn (rảnh / bận).
- Quản lý đơn
 - Xem danh sách các đơn đang chờ xử lý.
 - Cập nhật trạng thái đơn hàng: đang pha chế → đang phục vụ → hoàn tất.
 - Huỷ đơn khi khách hàng yêu cầu.
- Quản lý người dùng
 - Thêm tài khoản nhân viên.
 - Phân quyền: admin, nhân viên, khách hàng.

- Cập nhật thông tin tài khoản.
- Khoá hoặc xoá tài khoản khi cần.
- Thống kê doanh thu
 - Thống kê theo ngày, tuần, tháng, quý, năm.
 - Biểu đồ trực quan doanh thu.
 - Xem sản phẩm bán chạy, doanh thu theo món.
 - Xuất file báo cáo khi cần.
- Quản lý hệ thống
 - Sao lưu và phục hồi dữ liệu.
 - Cấu hình các thông số hệ thống.
 - Giám sát hoạt động truy cập của người dùng.

3.3. Yêu cầu phi chức năng

Yêu cầu phi chức năng là các tiêu chí không trực tiếp thể hiện chức năng của hệ thống nhưng có vai trò quan trọng, đảm bảo hệ thống hoạt động ổn định, đáng tin cậy và mang lại trải nghiệm tốt cho người dùng. Dưới đây là các yêu cầu phi chức năng chính của hệ thống quản lý quán cà phê:

3.3.1. Hiệu năng (Performance Requirements)

- Hệ thống phải đảm bảo thời gian phản hồi dưới 2 giây với các tác vụ như xem menu, cập nhật giỏ hàng, quản lý đơn hàng.
- Hệ thống phải xử lý được nhiều request đồng thời từ khách hàng mà không bị treo hoặc giật lag, đặc biệt vào giờ cao điểm.
- Cơ sở dữ liệu PostgreSQL cần tối ưu truy vấn để đảm bảo hiệu suất cao khi số lượng món và đơn đặt tăng lên.

3.3.2. Tính ổn định (Reliability)

- Hệ thống phải hoạt động liên tục mà không bị gián đoạn, đảm bảo tính ổn định cho hoạt động kinh doanh.
- Thông tin đơn hàng, trạng thái bàn, và dữ liệu khách hàng phải được lưu trữ đồng bộ chính xác.
- Các lỗi phát sinh phải được hệ thống ghi lại để dễ kiểm tra và khắc phục.

3.3.3. Tính bảo mật (Security Requirements)

- Mật khẩu người dùng phải được mã hoá bằng thuật toán bcrypt hoặc tương đương.
- Phân quyền rõ ràng giữa admin – nhân viên – khách hàng.
- API phải được bảo vệ bằng JWT (JSON Web Token) tranh truy cập trái phép.
- Hệ thống phải có cơ chế kiểm soát truy cập và hạn chế quyền theo vai trò.
- Dữ liệu truyền giữa client và server nên sử dụng giao thức **HTTPS** khi triển khai thực tế.

3.3.4. Khả năng mở rộng (Scalability)

- Kiến trúc hệ thống (React – Node.js – PostgreSQL) phải dễ dàng mở rộng khi quán có thêm chi nhánh.
- Hệ thống có thể thêm mới danh mục món, bàn hoặc tính năng mà không ảnh hưởng đến cấu trúc tổng thể.
- Backend sử dụng RESTful API giúp dễ dàng tích hợp với mobile app trong tương lai.

3.3.5. Tính thân thiện với người dùng (Usability)

- Giao diện phải dễ hiểu, đơn giản, trực quan và phù hợp với nhiều đối tượng người dùng.
- Hệ thống phải hỗ trợ tiếng Việt.
- Các nút bấm, biểu tượng và màu sắc phải rõ ràng, dễ thao tác.
- Giao diện khách hàng cần hiển thị menu và giá rõ ràng, thao tác đặt món nhanh.

3.3.6. Tính tương thích (Compatibility)

- Hệ thống phải hoạt động tốt trên hầu hết các trình duyệt hiện có.
- Tương thích với tất cả các thiết bị.
- Giao diện sử dụng **Responsive Design** để tự điều chỉnh trên kích thước màn hình khác nhau.

3.3.7. Khả năng bảo trì (Maintainability)

- Mã nguồn cần được tổ chức theo cấu trúc rõ ràng, dễ hiểu, hỗ trợ mở rộng và sửa lỗi.
- Backend và frontend tách biệt giúp bảo trì từng phần dễ dàng.

- Các thành phần trong React phải được chia nhỏ thành component giúp tái sử dụng và thay đổi nhanh chóng.

3.3.8. Sao lưu và phục hồi (Backup & Recovery)

- Cơ sở dữ liệu PostgreSQL phải có cơ chế sao lưu định kỳ để tránh mất dữ liệu.
- Hệ thống cần cho phép khôi phục dữ liệu trong trường hợp xảy ra lỗi nghiêm trọng hoặc sự cố máy chủ.

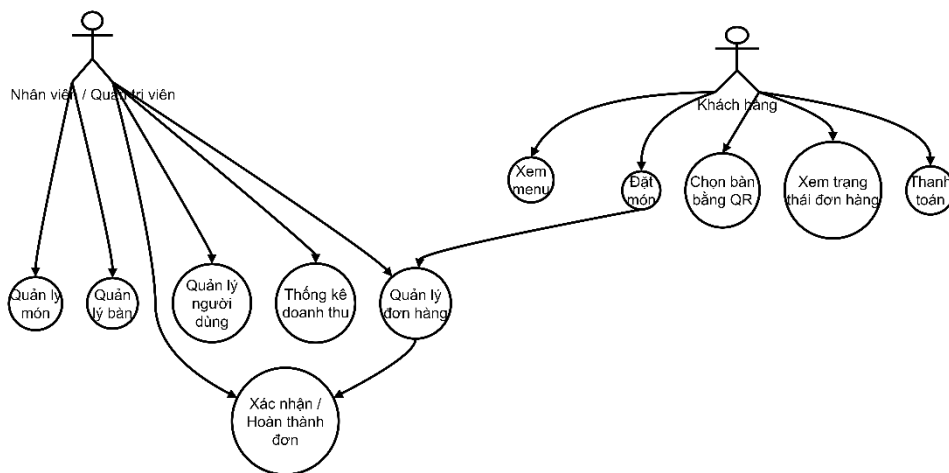
3.4. Mô hình nghiệp vụ

Mô hình nghiệp vụ giúp mô tả các chức năng chính và luồng hoạt động trong hệ thống quản lý quán café. Thông qua các biểu đồ Use Case, Activity và Sequence, ta có cái nhìn tổng quan về cách các tác nhân tương tác với hệ thống, cũng như cách dữ liệu và quy trình hoạt động được xử lý.

3.4.1. Biểu đồ Use case

- Các tác nhân (Actor)
 - Khách hàng (Customer): Người trực tiếp thao tác đặt món trên ứng dụng.

Nhân viên / Quản trị viên (Staff / Admin): Người quản lý hệ thống: món, bàn, doanh thu, đơn hàng.



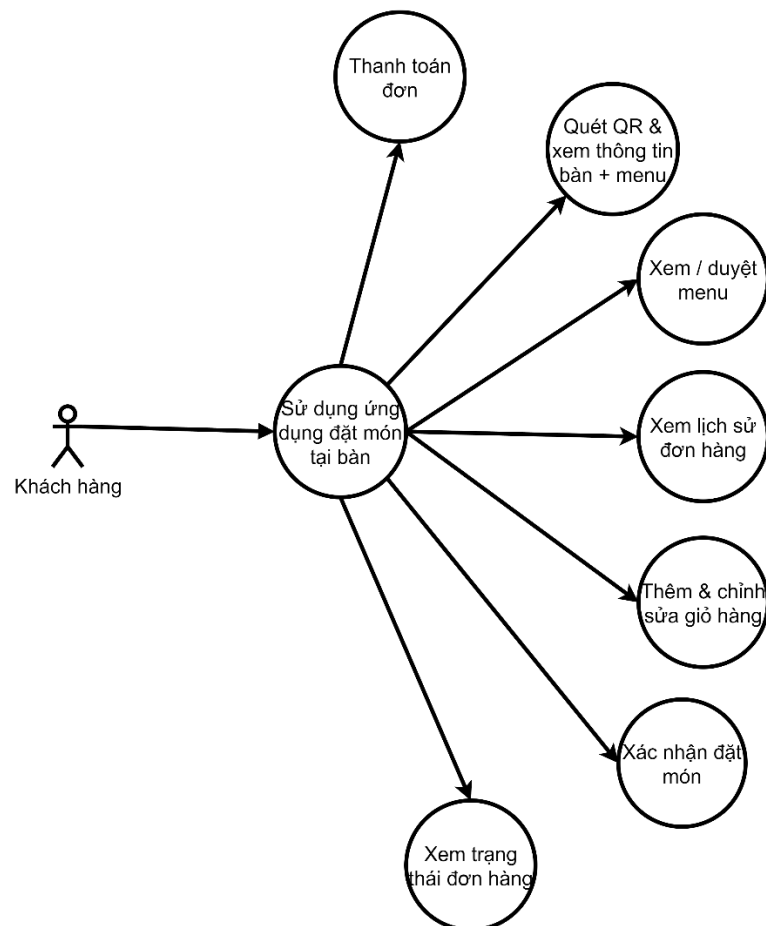
Hình 3. 1 Sơ đồ Use Case tổng quát

- Use Case dành cho khách hàng

Bảng 3. 1 Bảng danh sách Use Case Khách hàng

Use Case	Mô tả
----------	-------

Đăng ký tài khoản	Tạo tài khoản mới để đặt món
Đăng nhập	Truy cập hệ thống bằng email/mật khẩu
Xem Menu	Xem danh sách món, giá, preview món
Chọn bàn bằng QR	Quét mã QR đúng bàn đang ngồi
Thêm món vào giỏ	Chọn món, số lượng món
Đặt món	Xác nhận đặt món
Theo dõi trạng thái đơn	Xem tiến trình pha chế, phục vụ
Xem lịch sử đơn hàng	Xem lại các đơn đã đặt

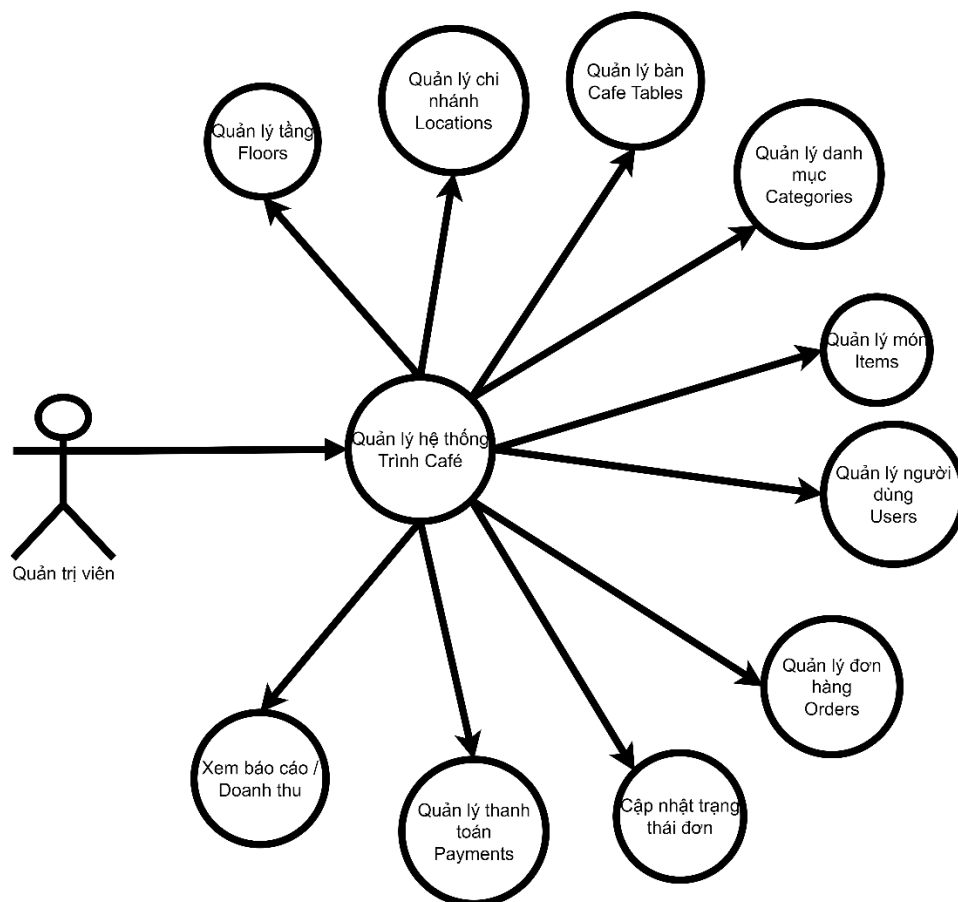


Hình 3. 2 Sơ đồ Use Case phân rã Khách hàng

- Use Case dành cho quản trị viên

Bảng 3. 2 Bảng danh sách Use Case quản trị viên

Use Case	Mô tả
Quản lý danh mục	Thêm / sửa / xoá danh mục
Quản lý món	Thêm / sửa / xoá món
Quản lý bàn	Tạo bàn, cập nhật trạng thái
Quản lý đơn hàng	Nhận đơn, xác nhận, cập nhật trạng thái
Quản lý người dùng	Thêm / sửa / xoá tài khoản nhân viên hoặc khách
Thống kê doanh thu	Xem báo cáo ngày / tuần / tháng



Hình 3. 3 Sơ đồ Use Case phân rã Quản trị viên

Mô tả Use Case bằng lời

Use Case: Đặt món (Khách hàng)

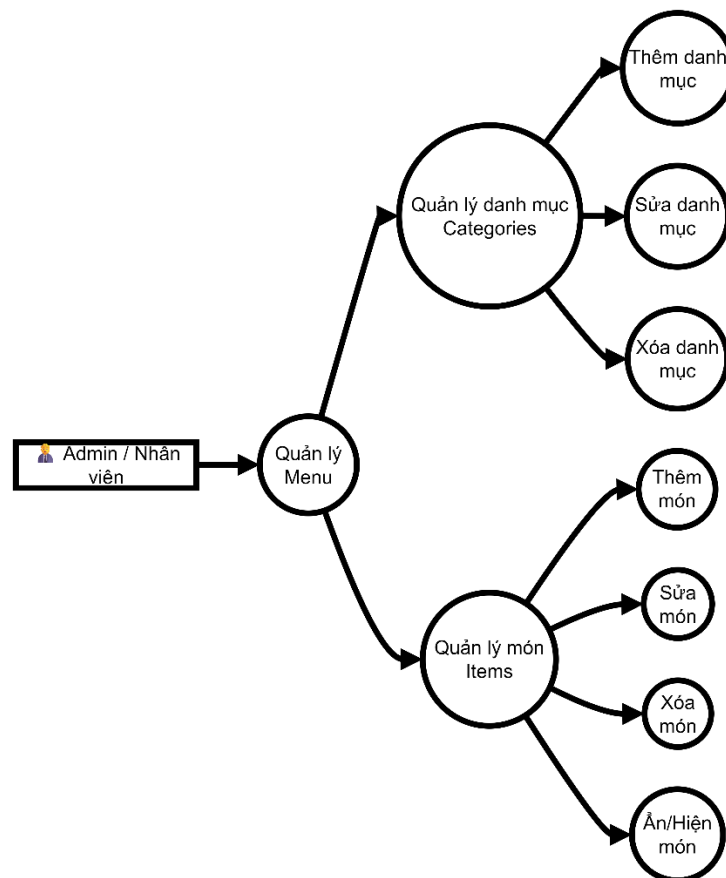
Mục tiêu: Cho phếp khách hàng đặt món qua mã QR

Luồng chính:

1. Khách quét mã QR để chọn bàn.
2. Hệ thống xác định số bàn và chuyển đến menu.
3. Khách chọn món → thêm vào giỏ hàng.
4. Khách xác nhận đơn hàng.
5. Hệ thống gửi đơn đến nhân viên pha chế.
6. Khách theo dõi trạng thái đơn.

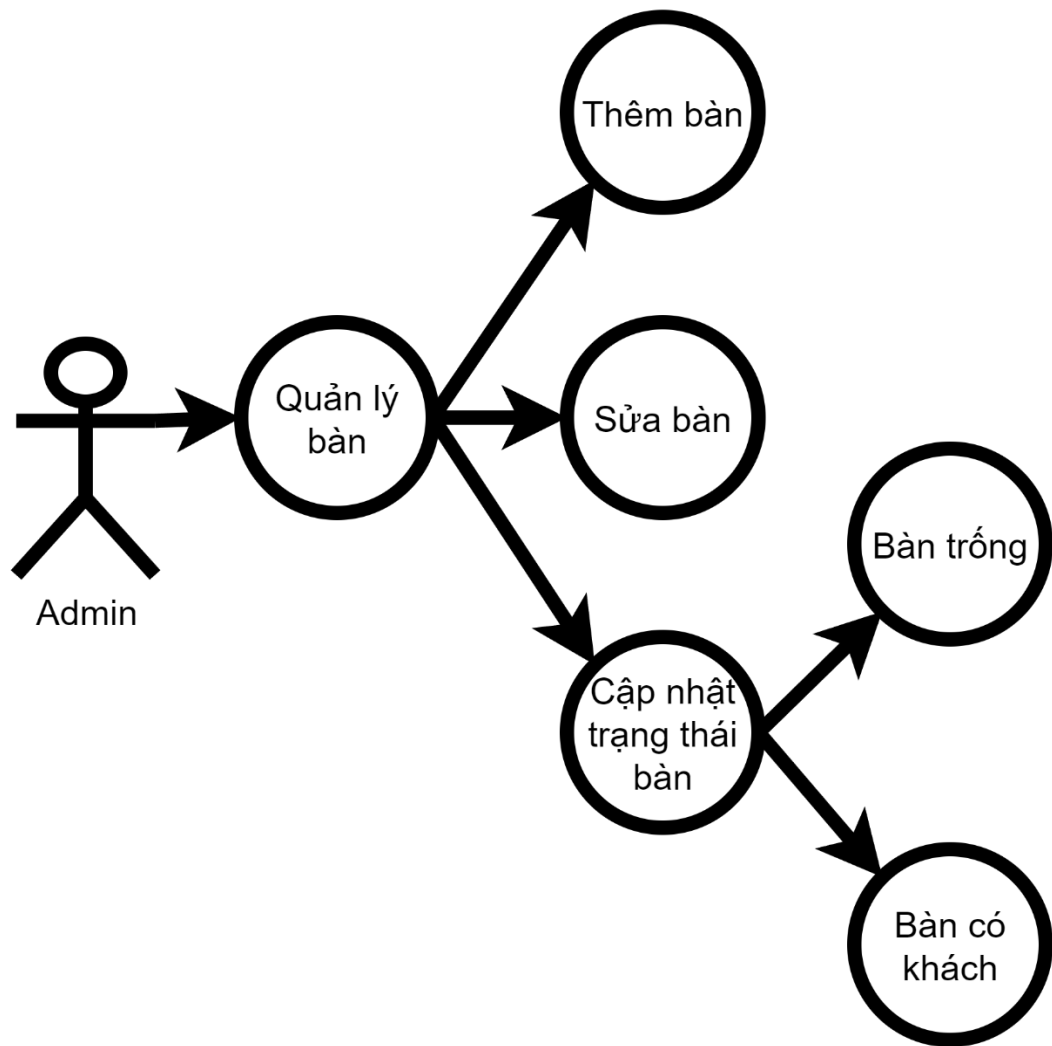
Use Case: Quản lý đơn hàng (Admin)

1. Admin xem danh sách đơn mới.
2. Cập nhật trạng thái đơn → “Đang pha chế”.
3. Sau khi hoàn tất → chuyển sang “Đã phục vụ”.
4. Khi thanh toán → chuyển sang “Hoàn tất”.



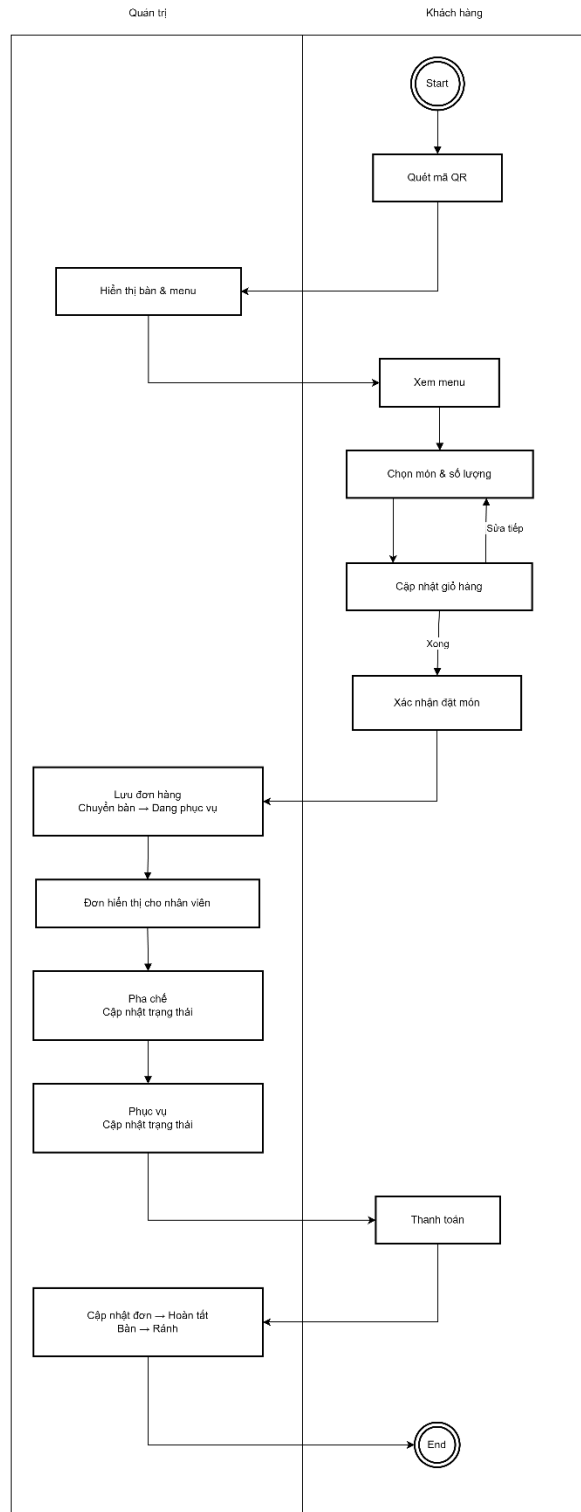
5.

Hình 3. 4 Use Case quản lý menu, danh mục



Hình 3. 5 Use Case Quản lý bàn

3.4.2. Biểu đồ Hoạt động (Activity Diagram)



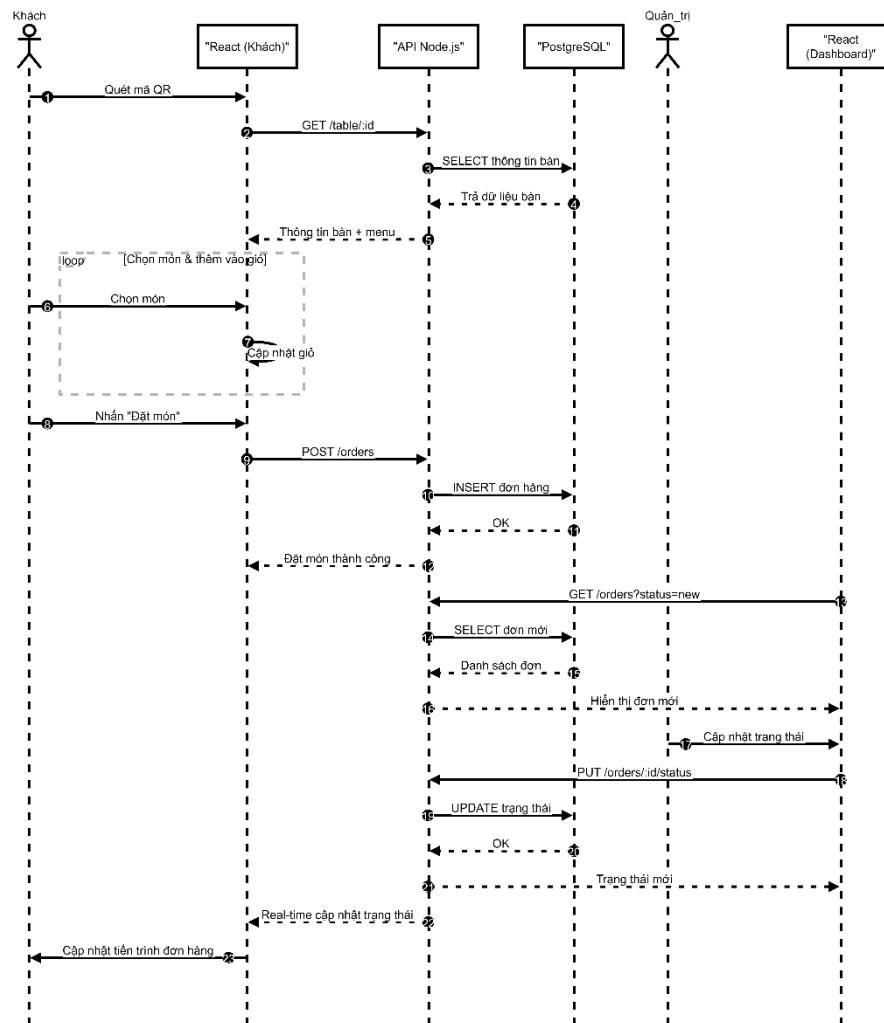
Hình 3. 6 Sơ đồ hoạt động

Dưới đây là mô tả quy trình hoạt động chính: **Đặt món trong quán café.**

Luồng hoạt động đặt món

- **Khách quét mã QR**
 - Hệ thống hiển thị thông tin bàn và menu.
- **Xem menu**
 - Khách lựa chọn danh mục → món → số lượng.
- **Thêm vào giỏ hàng**
 - Có thể chỉnh sửa số lượng hoặc xóa món.
- **Xác nhận đặt món**
 - Hệ thống ghi lại đơn hàng vào database.
 - Chuyển trạng thái bàn → “Đang phục vụ”.
- **Nhân viên nhận đơn**
 - Admin thấy đơn mới hiển thị trong dashboard.
- **Pha chế**
 - Nhân viên cập nhật trạng thái đơn → “Đang pha chế”.
- **Phục vụ**
 - Cập nhật trạng thái → “Đã phục vụ”.
- **Thanh toán**
 - Khách thanh toán tại quầy.
 - Hệ thống cập nhật trạng thái đơn → “Hoàn tất”.
 - Bàn trở về trạng thái “Rảnh”.

3.4.3. Biểu đồ Trình tự (Sequence Diagram)



Hình 3. 7 Biểu đồ tuần tự

Dưới đây là phần mô tả trình tự các bước xử lý khi khách đặt món qua QR:

Sequence: Khách đặt món → Hệ thống xử lý

Tác nhân: Khách → Giao diện React → API Node.js → PostgreSQL → Node.js → Giao diện React.

Trình tự hoạt động:

- Khách quét QR
 - React gửi request GET /table/:id
 - Node.js truy vấn PostgreSQL
 - Trả về thông tin bàn.
- Khách chọn món → thêm vào giỏ
 - React tạo danh sách tạm thời trên client.

- Khách bấm đặt món
 - React gửi POST /orders kèm thông tin bàn + món.
 - Node.js kiểm tra hợp lệ.
 - Lưu đơn hàng vào PostgreSQL.
 - Trả về trạng thái “Đặt thành công”.
- Admin nhận đơn mới
 - Bảng dashboard gọi GET /orders?status=new
 - Hiện thị danh sách đơn hàng mới.
- Admin cập nhật trạng thái đơn
 - PUT /orders/:id/status
 - Database cập nhật.
 - React (khách hàng) nhận cập nhật real-time.
- Khách xem trạng thái đơn hàng
 - GET /orders/:id mỗi vài giây hoặc websocket.
 - Hiện thị tiến trình pha chế – phục vụ.

3.5. Phân tích dữ liệu

Phân tích dữ liệu là bước quan trọng nhằm xác định các thực thể, mối quan hệ và cấu trúc dữ liệu cần thiết để hệ thống có thể lưu trữ và xử lý thông tin chính xác. Dựa trên yêu cầu hệ thống, các thực thể chính và mô hình dữ liệu được xác định như sau:

3.5.1. Xác định thực thể và thuộc tính

Dưới đây là danh sách các thực thể (entities) chính trong hệ thống cùng các thuộc tính (attributes) cần thiết để lưu trữ thông tin.

- Thực thể Người dùng (User)

Bảng 3. 3 Bảng thực thể Người dùng

Thuộc tính	Mô tả
user_id (PK)	Mã định danh người dùng
name	Tên người dùng
email	Email đăng nhập
password	Mật khẩu đã mã hóa

role	Vai trò (admin, nhân viên, khách)
phone	Số điện thoại
created_at	Ngày tạo tài khoản

- Thực thể Bàn (Table)

Bảng 3. 4 Bảng thực thể Bàn

Thuộc tính	Mô tả
table_id (PK)	Mã bàn
table_name	Tên hoặc số bàn
qr_code	Mã QR liên kết đến bàn
status	Trạng thái (rảnh, bận, đặt trước)

- Thực thể Danh mục món (Category)

Bảng 3. 5 Bảng thực thể danh mục món

Thuộc tính	Mô tả
category_id (PK)	Mã danh mục
name	Tên danh mục (cà phê, trà, nước ép...)

- Thực thể Món uống (Menu_Items)

Bảng 3. 6 Bảng thực thể món uống

Thuộc tính	Mô tả
item_id (PK)	Mã món
category_id (FK)	Mã danh mục
name	Tên món
price	Giá
description	Mô tả

image_url	Hình ảnh món
status	Trạng thái (còn – hết)

- Thực thể Đơn hàng (Order)

Bảng 3. 7 Bảng thực thể đơn đặt

Thuộc tính	Mô tả
order_id (PK)	Mã đơn hàng
user_id (FK)	Người tạo đơn
table_id (FK)	Bàn đặt món
total_price	Tổng tiền
status	Trạng thái đơn (đang pha chế, đã phục vụ, hoàn tất)
created_at	Thời gian tạo đơn

- Thực thể Chi tiết đơn hàng (Order_Items)

Bảng 3. 8 Bảng thực thể chi tiết đơn đặt

Thuộc tính	Mô tả
id (PK)	Mã dòng chi tiết
order_id (FK)	Mã đơn hàng
item_id (FK)	Mã món
quantity	Số lượng
price	Giá tại thời điểm đặt

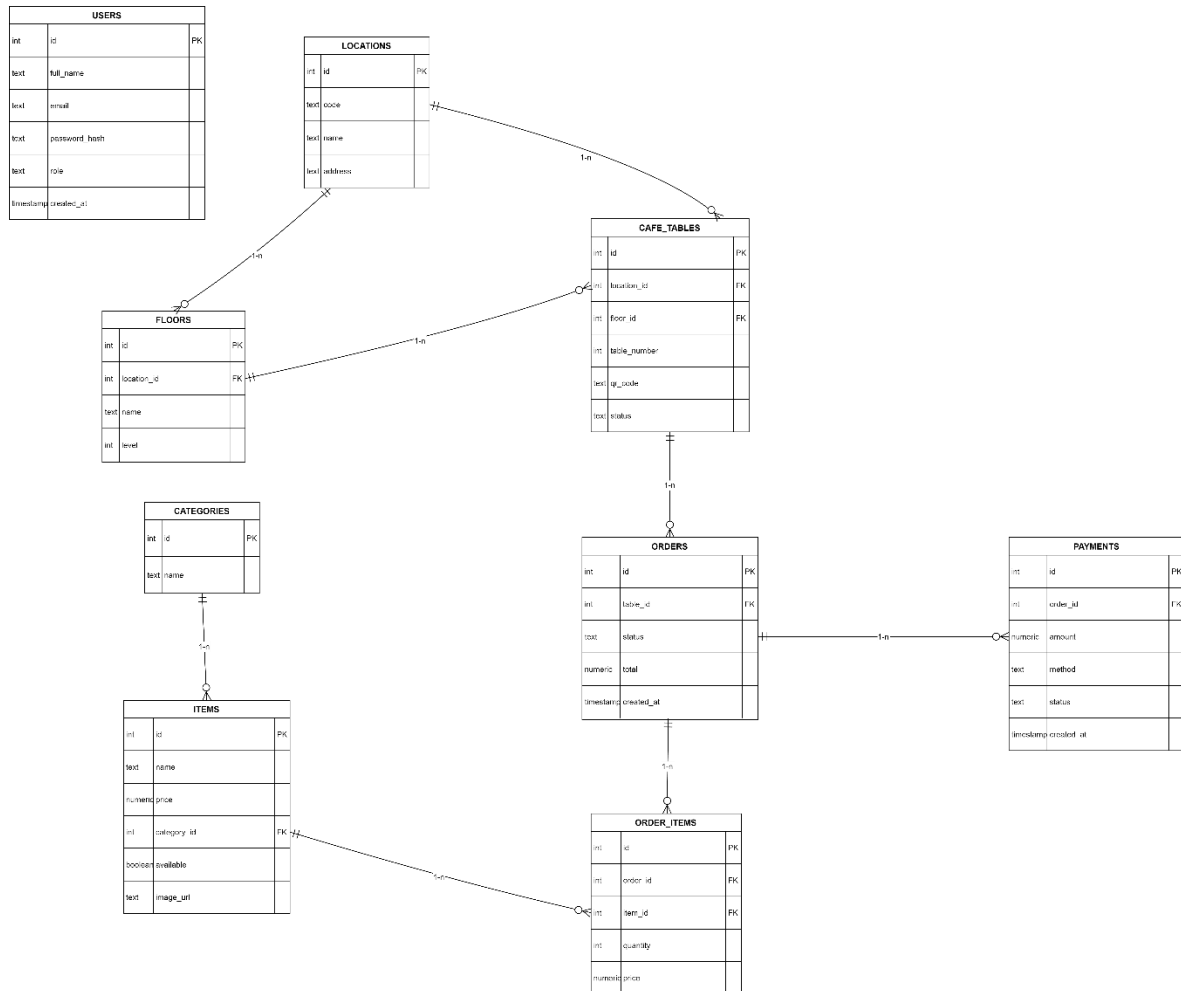
- Thực thể doanh thu (Revenue)

Bảng 3. 9 Bảng thực thể doanh thu

Thuộc tính	Mô tả
id (PK)	Mã thống kê

date	Ngày thống kê
total_income	Tổng doanh thu trong ngày

3.5.2. Thiết kế mô hình ERD



Hình 3. 8 Mô hình ERD

Dưới đây là mô tả ERD bằng lời theo các mối quan hệ:

Các mối quan hệ chính

- **Users – Orders**
1 người dùng có nhiều đơn hàng
→ Quan hệ: **1 – n**
- **Tables – Orders**
1 bàn có thể có nhiều đơn hàng (theo thời gian)
→ Quan hệ: **1 – n**

- **Categories – Menu_Items**
Mỗi danh mục chứa nhiều món
→ Quan hệ: **1 – n**
- **Menu_Items – Order_Items**
1 món có thể xuất hiện trong nhiều đơn hàng
→ Quan hệ: **1 – n**
- **Orders – Order_Items**
1 đơn hàng chứa nhiều món
→ Quan hệ: **1 - n**

3.5.3. Mô hình cơ sở dữ liệu logic và vật lý

- **Mô hình logic**

Mô hình logic xác định các bảng và ràng buộc chính (PK, FK):

Bảng Users

- PK: user_id
- Các trường chính: name, email, password, role, created_at

Bảng Tables

- PK: table_id
- Trường: table_name, qr_code, status

Bảng Categories

- PK: category_id
- Trường: name

Bảng Menu_Items

- PK: item_id
- FK: category_id
- Trường: name, price, image_url, description, status

Bảng Orders

- PK: order_id
- FK: user_id, table_id
- Trường: total_price, status, created_at

Bảng Order_Items

- PK: id
- FK: order_id, item_id
- Trường: quantity, price

Bảng Revenue

- PK: id
- Trường: date, total_income

• **Mô hình vật lý (Database Physical Design)**

Dưới đây là ví dụ cấu trúc lệnh tạo bảng trong PostgreSQL:

Bảng User:

```
CREATE TABLE IF NOT EXISTS users (  
  id SERIAL PRIMARY KEY,  
  full_name TEXT NOT NULL,  
  email TEXT UNIQUE NOT NULL,  
  password_hash TEXT NOT NULL,  
  role TEXT NOT NULL DEFAULT 'staff', -- 'admin' or 'staff'  
  created_at TIMESTAMP DEFAULT NOW()  
);
```

Bảng Tables:

```
CREATE TABLE IF NOT EXISTS cafe_tables (  
  id SERIAL PRIMARY KEY,  
  location_id INT NOT NULL REFERENCES locations(id) ON DELETE CASCADE,  
  floor_id INT NOT NULL REFERENCES floors(id) ON DELETE CASCADE,  
  table_number INT NOT NULL,  
  qr_code TEXT, -- path or content  
  status TEXT NOT NULL DEFAULT 'available' -- available, occupied, reserved  
);
```

Bảng Locations:

CREATE TABLE IF NOT EXISTS locations (

id SERIAL PRIMARY KEY,
code TEXT UNIQUE NOT NULL,
name TEXT NOT NULL,
address TEXT

);

Bảng Floors:

CREATE TABLE IF NOT EXISTS floors (

id SERIAL PRIMARY KEY,
location_id INT NOT NULL REFERENCES locations(id) ON DELETE CASCADE,
name TEXT NOT NULL,
level INT NOT NULL

);

Bảng Category:

CREATE TABLE IF NOT EXISTS categories (

id SERIAL PRIMARY KEY,
name TEXT NOT NULL

);

Bảng Items:

CREATE TABLE IF NOT EXISTS items (

id SERIAL PRIMARY KEY,
name TEXT NOT NULL,
price NUMERIC(12,2) NOT NULL,
category_id INT NOT NULL REFERENCES categories(id) ON DELETE RESTRICT,
available BOOLEAN DEFAULT TRUE,
image_url TEXT

);

Bảng Orders:

```
CREATE TABLE IF NOT EXISTS orders (  
  id SERIAL PRIMARY KEY,  
  table_id INT NOT NULL REFERENCES cafe_tables(id) ON DELETE RESTRICT,  
  status TEXT NOT NULL DEFAULT 'pending', -- pending, preparing, served, paid,  
  cancelled  
  total NUMERIC(12,2) DEFAULT 0,  
  created_at TIMESTAMP DEFAULT NOW()  
);
```

Bảng Order Items:

```
CREATE TABLE IF NOT EXISTS order_items (  
  id SERIAL PRIMARY KEY,  
  order_id INT NOT NULL REFERENCES orders(id) ON DELETE CASCADE,  
  item_id INT NOT NULL REFERENCES items(id) ON DELETE RESTRICT,  
  quantity INT NOT NULL,  
  price NUMERIC(12,2) NOT NULL  
);
```

Bảng Payment:

```
CREATE TABLE IF NOT EXISTS payments (  
  id SERIAL PRIMARY KEY,  
  order_id INT NOT NULL REFERENCES orders(id) ON DELETE RESTRICT,  
  amount NUMERIC(12,2) NOT NULL,  
  method TEXT NOT NULL, -- cash, card, ewallet  
  status TEXT NOT NULL DEFAULT 'completed',  
  created_at TIMESTAMP DEFAULT NOW()  
);
```

3.6. Thiết kế hệ thống

3.6.1. Kiến trúc hệ thống tổng thể

Hệ thống quản lý quán café được thiết kế theo mô hình **Client – Server**, trong đó các thành phần được phân tách rõ ràng nhằm đảm bảo tính linh hoạt, dễ mở rộng và thuận tiện cho việc bảo trì.

Theo mô hình này, phía **Client** chịu trách nhiệm hiển thị giao diện và tương tác với người dùng, trong khi phía **Server** đảm nhận vai trò xử lý nghiệp vụ, xác thực người dùng và quản lý dữ liệu. Hai thành phần giao tiếp với nhau thông qua các **RESTful API**.

Kiến trúc tổng thể của hệ thống bao gồm ba lớp chính:

- **Lớp giao diện (Presentation Layer):** hiển thị giao diện và tiếp nhận thao tác từ người dùng.
- **Lớp xử lý nghiệp vụ (Business Logic Layer):** xử lý các yêu cầu, kiểm tra dữ liệu và thực thi nghiệp vụ.
- **Lớp dữ liệu (Data Layer):** lưu trữ và quản lý dữ liệu trong cơ sở dữ liệu.

Việc áp dụng kiến trúc này giúp hệ thống hoạt động ổn định, dễ dàng nâng cấp và tích hợp thêm các chức năng mới trong tương lai.

3.6.2. Thiết kế Frontend với React

Frontend của hệ thống được thiết kế nhằm cung cấp giao diện trực quan, thân thiện và dễ sử dụng cho người dùng. Công nghệ ReactJS được lựa chọn để xây dựng giao diện do có khả năng tái sử dụng component và hiệu năng cao.

Giao diện hệ thống được tổ chức theo mô hình component, mỗi component đảm nhận một chức năng cụ thể như:

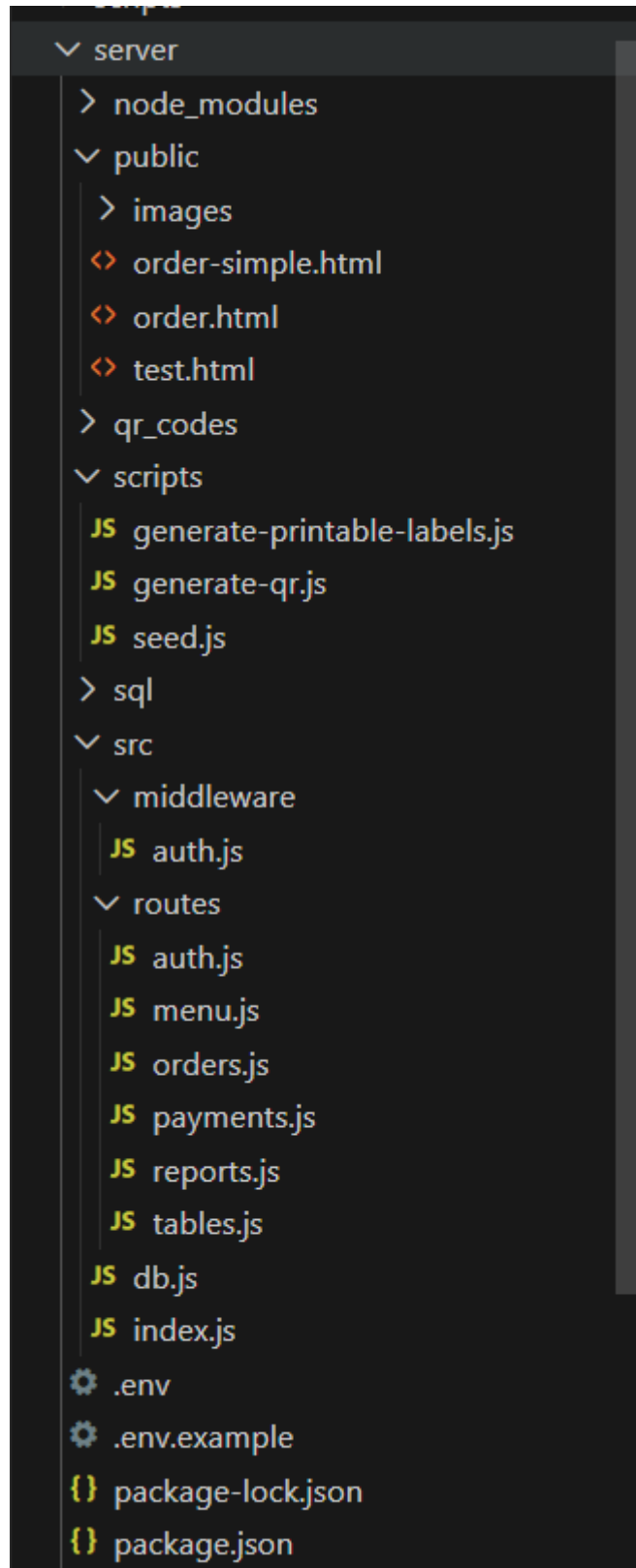
- Giao diện đăng nhập và phân quyền người dùng
- Giao diện quản lý bàn
- Giao diện quản lý menu
- Giao diện đặt món và thanh toán
- Giao diện thống kê và báo cáo

Frontend giao tiếp với Backend thông qua các API theo chuẩn REST để gửi và nhận dữ liệu, đảm bảo tính đồng bộ và nhất quán trong quá trình sử dụng hệ thống.

3.6.3. Thiết kế Backend với Node.js

Backend sử dụng ExpressJS và xây dựng theo RESTful API.

Cấu trúc thư mục:



Hình 3. 9 Cấu trúc thư mục

Danh sách các API chính

API người dùng

Bảng 3. 10 Bảng API người dùng

Method	Endpoint	Chức năng
POST	/api/auth/register	Đăng ký
POST	/api/auth/login	Đăng nhập
GET	/api/users	Lấy danh sách người dùng (admin)

API danh mục

Bảng 3. 11 Bảng API danh mục

Method	Endpoint	Chức năng
GET	/api/categories	Lấy danh mục
POST	/api/categories	Thêm
PUT	/api/categories/:id	Sửa
DELETE	/api/categories/:id	Xóa

API món uống

Bảng 3. 12 Bảng API món uống

Method	Endpoint	Chức năng
GET	/api/menu	Xem menu
POST	/api/menu	Thêm món
PUT	/api/menu/:id	Cập nhật
DELETE	/api/menu/:id	Xóa

API bàn

Bảng 3. 13 Bảng API bàn

Method	Endpoint	Chức năng
GET	/api/tables	Danh sách bàn
POST	/api/tables	Thêm bàn
PUT	/api/tables/:id	Cập nhật
DELETE	/api/tables/:id	Xóa

API đơn hàng

Bảng 3. 14 Bảng API đơn hàng

Method	Endpoint	Chức năng
POST	/api/orders	Tạo đơn
GET	/api/orders	Danh sách đơn
GET	/api/orders/:id	Chi tiết đơn
PUT	/api/orders/:id/status	Cập nhật trạng thái

API doanh thu

Bảng 3. 15 Bảng API doanh thu

Method	Endpoint	Chức năng
GET	/api/revenue/daily	Doanh thu theo ngày
GET	/api/revenue/monthly	Doanh thu theo tháng

Các module chức năng:

- **Module xử lý xác thực (Auth Module)**
 - JWT token
 - Mã hóa mật khẩu bằng bcrypt

- **Module quản lý người dùng**
 - Thêm, sửa, xóa nhân viên
 - Phân quyền: admin – staff – customer
- **Module quản lý menu**
 - CRUD danh mục và món
 - Upload hình ảnh (tùy chọn)
- **Module quản lý bàn**
 - Gán mã QR tự động
 - Cập nhật trạng thái theo đơn hàng
- **Module đơn hàng**
 - Tạo đơn hàng từ giỏ hàng
 - Chuyển trạng thái theo thời gian thực
- **Module thống kê doanh thu**
 - Tổng hợp doanh thu

Xuất dữ liệu dạng biểu đồ

3.6.4. Thiết kế cơ sở dữ liệu với PostgreSQL

Bảng dữ liệu chính (menu, order, khách hàng, doanh thu)

- **Bảng Users (tài khoản người dùng: admin, nhân viên, sau này có thể thêm khách hàng)**
 - user_id – mã người dùng (khóa chính, từ id)
 - name – họ tên (full_name)
 - email – email đăng nhập (duy nhất)
 - password – mật khẩu đã mã hóa (password_hash)
 - role – vai trò (admin, staff, ...)
 - phone – số điện thoại (bạn thêm để quản lý liên hệ)
 - created_at – ngày giờ tạo tài khoản
- **Bảng Locations (chi nhánh/quán Trình Café)**
 - location_id – mã chi nhánh (từ id)
 - code – mã code chi nhánh (ví dụ: TRINH_DN_01)

- name – tên chi nhánh (Trình Café Cơ Sở 1...)
- address – địa chỉ chi nhánh
- **Bảng Floors (các tầng trong từng chi nhánh)**
 - floor_id – mã tầng (từ id)
 - location_id – mã chi nhánh (liên kết Locations)
 - name – tên tầng (Tầng 1, Tầng 2, Khu sân thượng...)
 - level – số thứ tự tầng (1, 2, 3...)
- **Bảng Cafe_Tables (bàn trong quán)**
 - table_id – mã bàn (từ id)
 - location_id – mã chi nhánh (liên kết Locations)
 - floor_id – mã tầng (liên kết Floors)
 - table_number – số hiệu bàn (1, 2, 3...)
 - qr_code – nội dung/link/path QR code của bàn
 - status – trạng thái bàn (available, occupied, reserved)
- **Bảng Categories (nhóm món)**
 - category_id – mã danh mục món (từ id)
 - name – tên danh mục (Cà phê, Trà sữa, Nước ép...)
- **Bảng Items (món đồ uống / món ăn)**
 - item_id – mã món (từ id)
 - name – tên món (Cà phê sữa đá, Latte...)
 - price – đơn giá (NUMERIC(12,2))
 - category_id – mã danh mục (liên kết Categories)
 - available – còn bán hay không (TRUE/FALSE)
 - image_url – đường dẫn ảnh minh họa món
- **Bảng Orders (đơn gọi món tại bàn)**
 - order_id – mã đơn hàng (từ id)
 - table_id – mã bàn (liên kết Cafe_Tables)
 - status – trạng thái đơn (pending, preparing, served, paid, cancelled)
 - total – tổng tiền đơn hàng
 - created_at – thời gian tạo đơn

- **Bảng Order_Items (chi tiết món trong mỗi đơn)**

- order_item_id – mã dòng chi tiết (từ id)
- order_id – mã đơn hàng (liên kết Orders)
- item_id – mã món (liên kết Items)
- quantity – số lượng món
- price – đơn giá tại thời điểm order (để không bị lệ thuộc giá hiện tại)

- **Bảng Payments (thanh toán đơn hàng)**

- payment_id – mã thanh toán (từ id)
- order_id – mã đơn hàng (liên kết Orders)
- amount – số tiền thanh toán
- method – phương thức thanh toán (cash, card, ewallet)
- status – trạng thái thanh toán (completed, có thể mở rộng pending, failed...)
- created_at – thời gian thực hiện thanh toán

Mối quan hệ giữa các bảng

Bảng 3. 16 Mối quan hệ giữa các bảng

Quan hệ	Loại
Locations – Floors	1 : n
Locations – Cafe_Tables	1 : n
Floors – Cafe_Tables	1 : n
Categories – Items	1 : n
Cafe_Tables – Orders	1 : n
Orders – Order_Items	1 : n
Items – Order_Items	1 : n
Orders – Payments	1 : n
Users – Orders	1 : n (tùy chọn nghiệp vụ)

CHƯƠNG 4. XÂY DỰNG HỆ THỐNG

4.1. Kiến trúc hệ thống tổng thể

Hệ thống quản lý quán Café được xây dựng theo mô hình Client – Server, trong đó bao gồm 3 thành phần chính là:

4.1.1. Frontend (React)

- Đảm nhiệm việc hiển thị giao diện người dùng.
- Xử lý tương tác, điều hướng, quản lý giỏ hàng, hiển thị trạng thái đơn hàng.
- Gửi các yêu cầu (request) tới API backend và nhận dữ liệu phản hồi (response).

4.1.2. Backend (Node.JS)

- Đảm nhận xử lý nghiệp vụ:
 - Quản lý người dùng, món, bàn, đơn hàng.
 - Xử lý logic đặt món, cập nhật trạng thái, tính tổng tiền.
- Tạo RESTful API để giao tiếp với frontend.
- Trung gian kết nối với cơ sở dữ liệu PostgreSQL.

4.1.3. Cơ sở dữ liệu (PostgreSQL)

- + Lưu trữ toàn bộ dữ liệu:
 - Tài khoản người dùng
 - Menu
 - Đơn hàng
 - Chi tiết đơn hàng
 - Bàn
 - Doanh thu

Mô hình tổng thể:

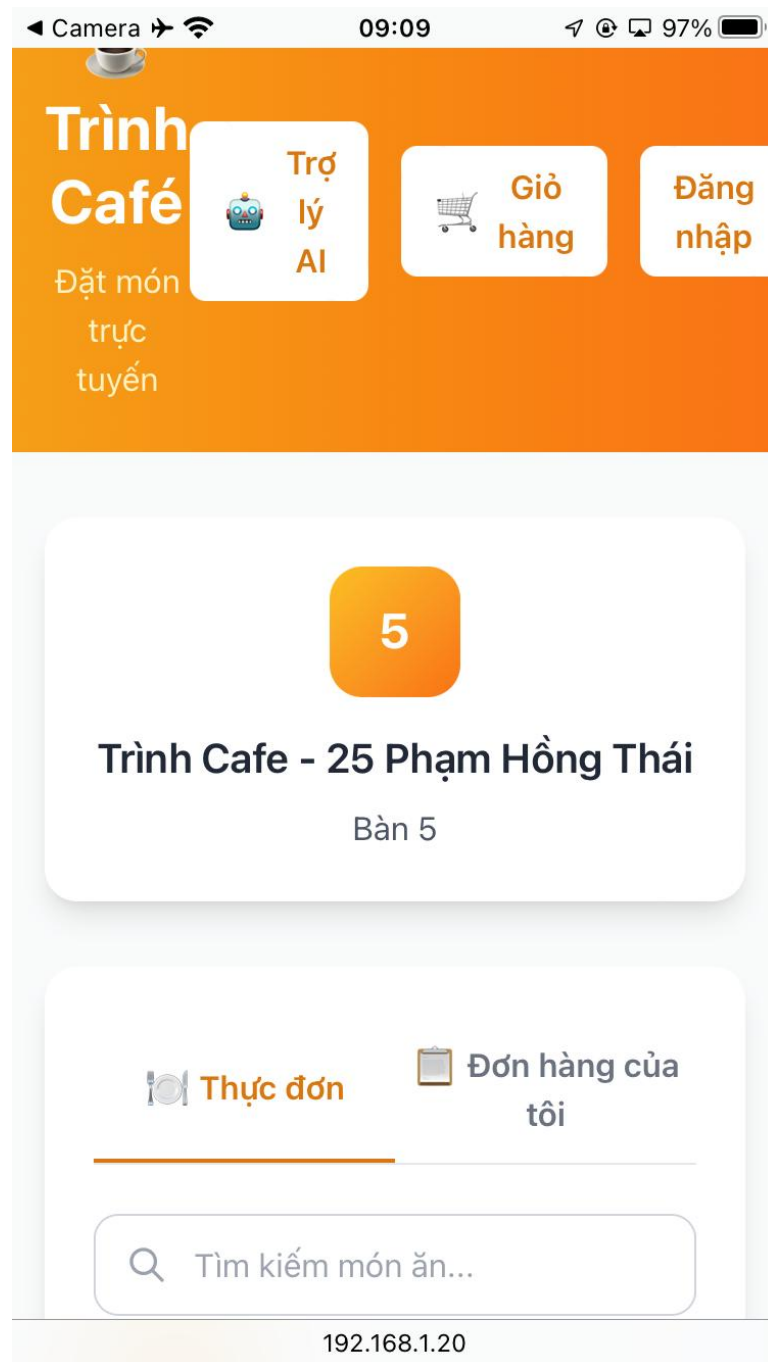
User (Browser) → ReactJS (frontend) → Node.js Express API (backend) → PostgreSQL (Database) → Node.js Express API (backend) → ReactJS (frontend) → User (Browser).

Ưu điểm kiến trúc

- + Tách biệt frontend và backend → dễ bảo trì và mở rộng.
- + API có thể sử dụng cho mobile app trong tương lai.
- + Cơ sở dữ liệu quan hệ đảm bảo tính toàn vẹn và bảo mật cao.
- + Toàn bộ hệ thống phù hợp để triển khai trên cloud như Render, Railway, Vercel.

4.2. Thiết kế giao diện cho hệ thống

4.2.1. Giao diện chính (khách)



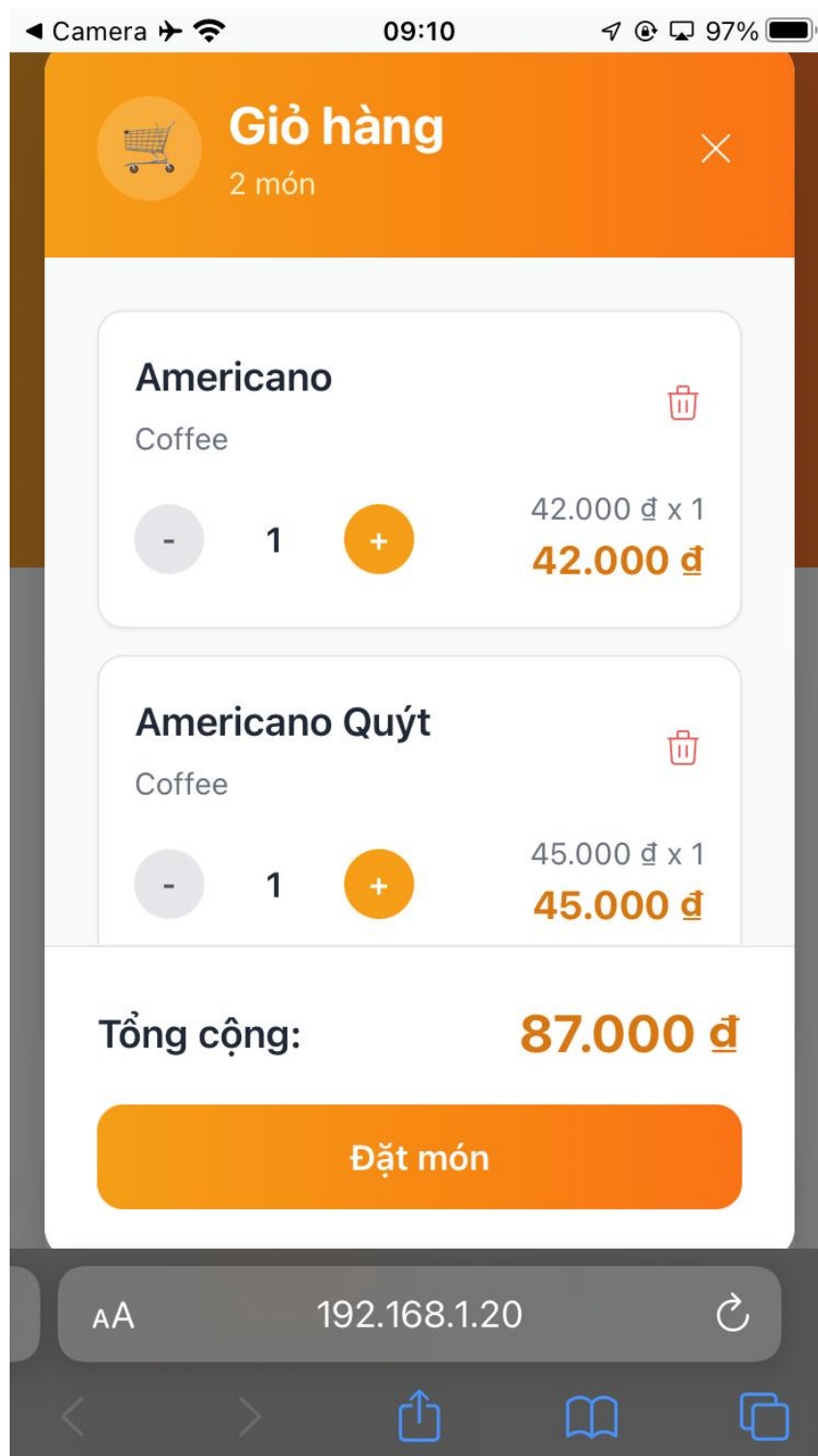
Hình 4. 1 Giao diện chính (khách)

4.2.2. Giao diện đăng nhập (khách)

The screenshot shows a mobile application interface for 'Trình Café'. At the top, there's a status bar with 'Camera', signal strength, Wi-Fi, time '09:09', and battery '97%'. Below the status bar, the app header features the name 'Trình Café' on the left and three navigation buttons: 'Trợ lý AI' (with a robot icon), 'Giỏ hàng' (with a shopping cart icon), and 'Đăng nhập' (highlighted in orange). A modal dialog titled 'Đăng nhập' is open in the center. It has a close button (X) in the top right corner. Inside the modal, there are two tabs: 'Đăng nhập' (selected and underlined in orange) and 'Đăng ký'. Below the tabs are two input fields: 'Email' and 'Mật khẩu' (Password). At the bottom of the modal is a large orange button labeled 'Đăng nhập'. Below the modal, a search bar is visible with the placeholder text 'Tìm kiếm món ăn...'. At the very bottom of the screen, the IP address '192.168.1.20' is displayed.

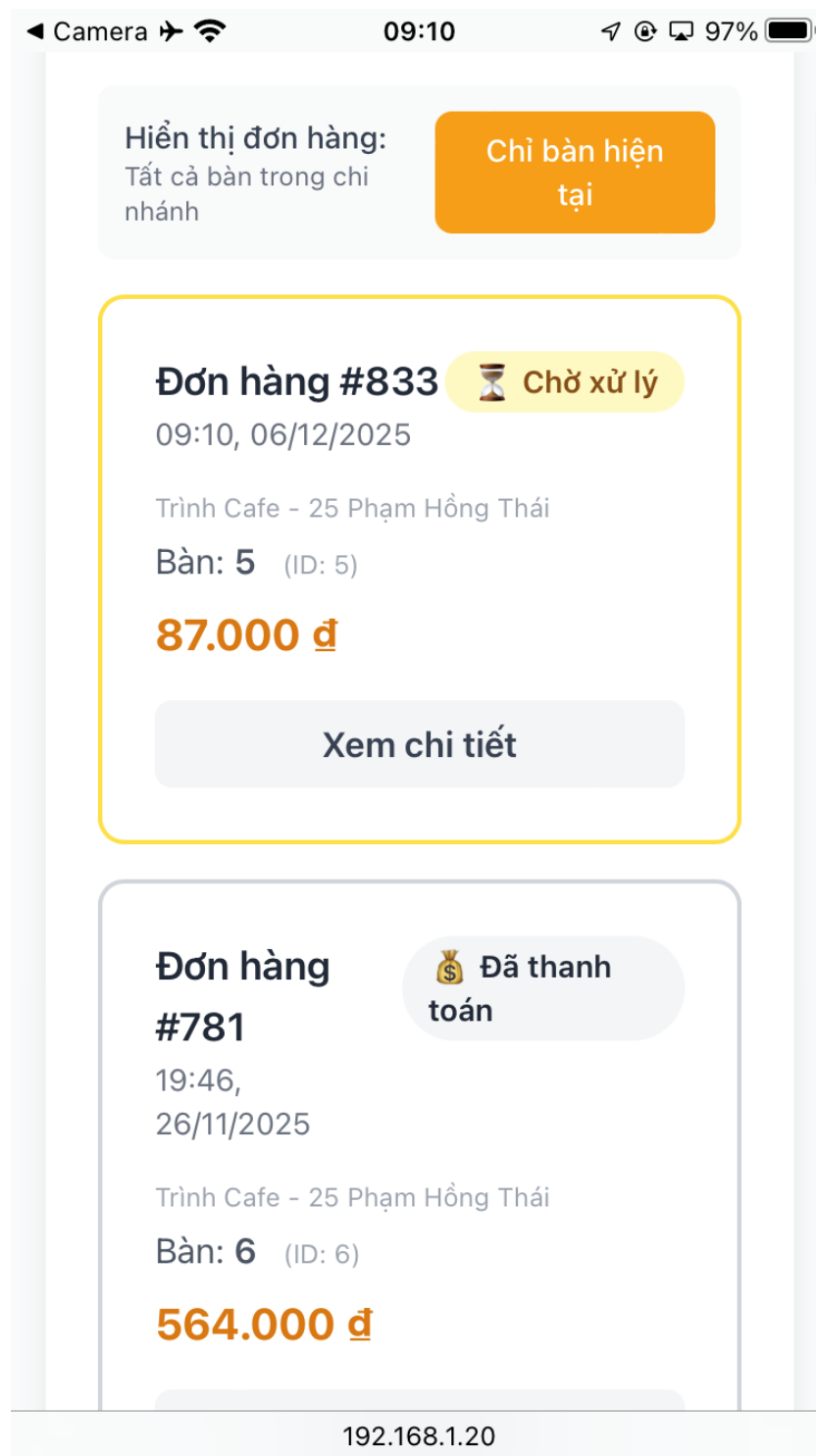
Hình 4. 2 Giao diện đăng nhập (khách)

4.2.3. Giao diện giỏ hàng (khách)



Hình 4. 3 Giao diện giỏ hàng (khách)

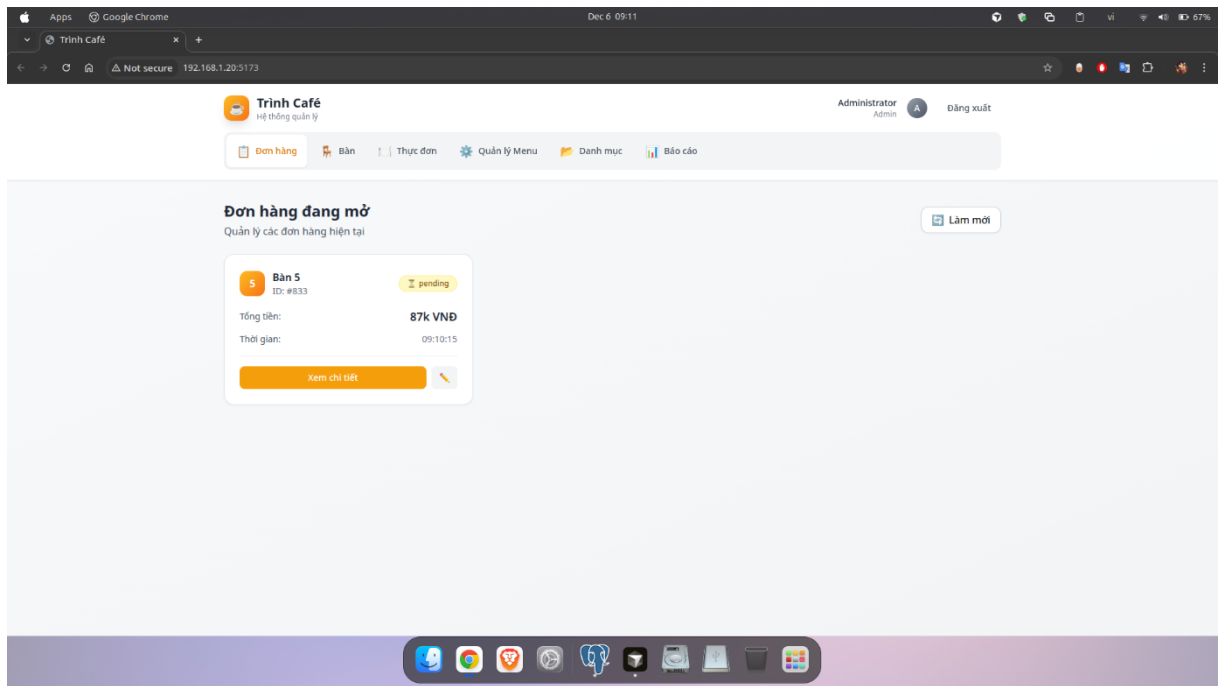
4.2.4. Giao diện lịch sử đơn hàng (khách)



Hình 4. 4 Giao diện lịch sử đặt đơn

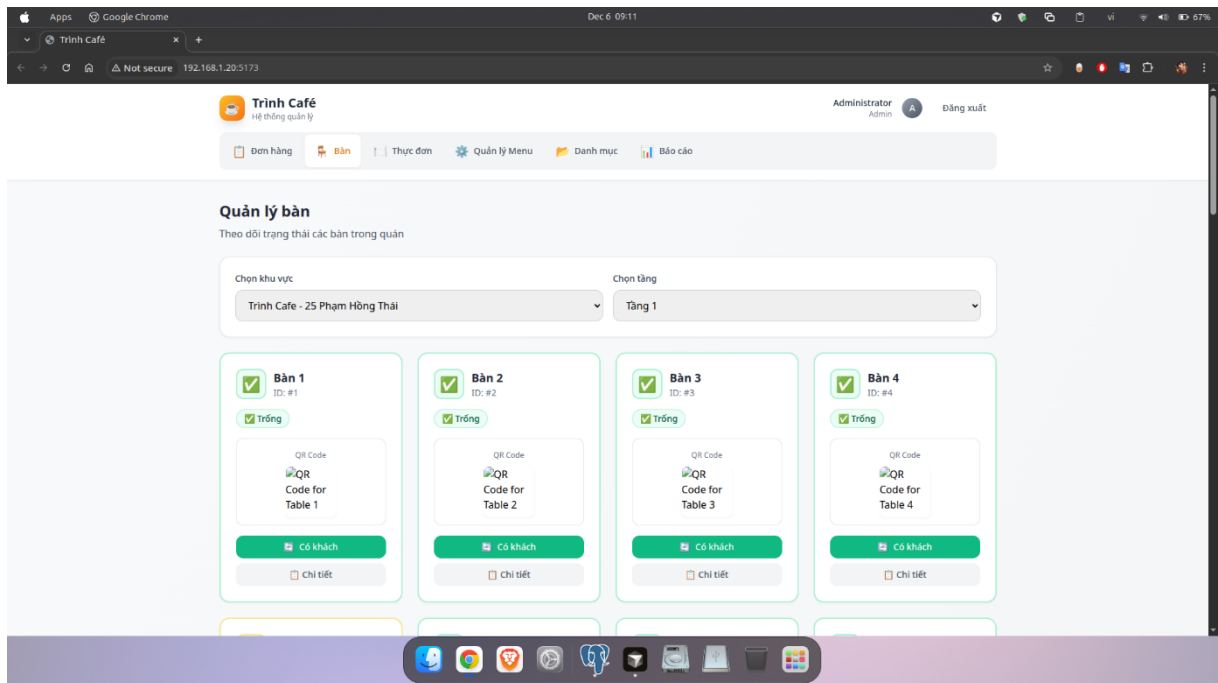
ỨNG DỤNG NODE.JS, REACT, POSTGRESQL TRONG VIỆC XÂY DỰNG HỆ THỐNG QUẢN LÝ QUÁN
TRÌNH CÀFÉ

4.2.5. Giao diện quản lý đơn hàng (admin)



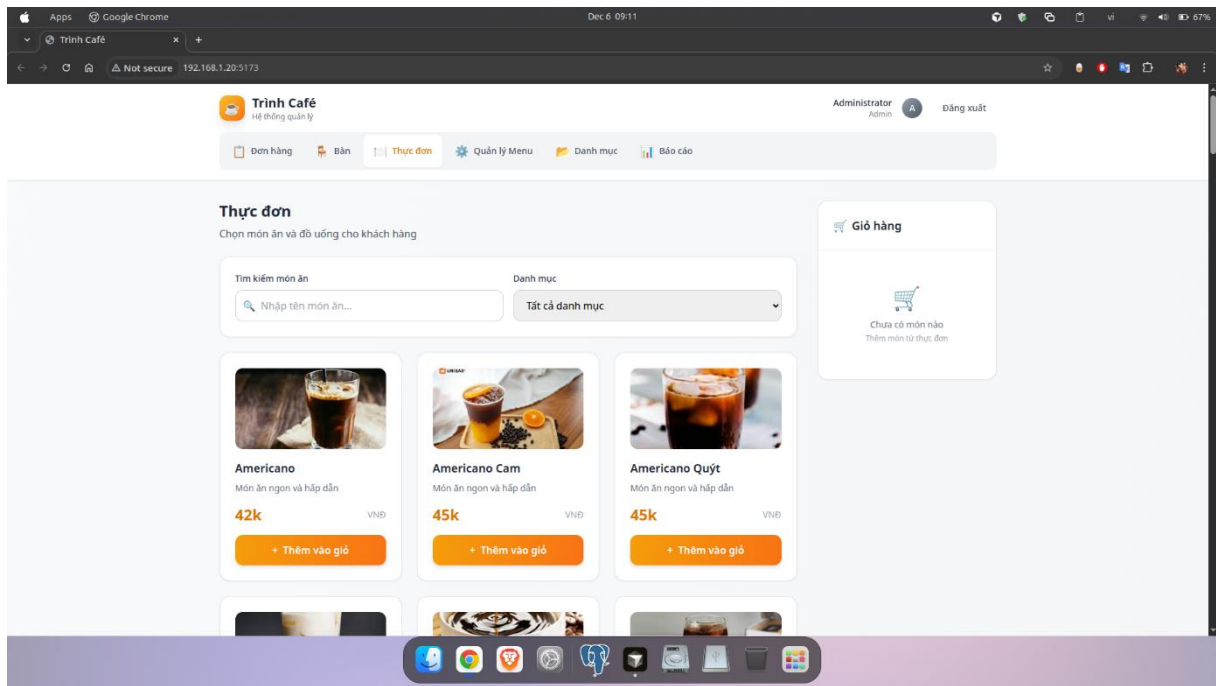
Hình 4. 5 Giao diện quản lý đơn hàng (admin)

4.2.6. Giao diện quản lý bàn (admin)



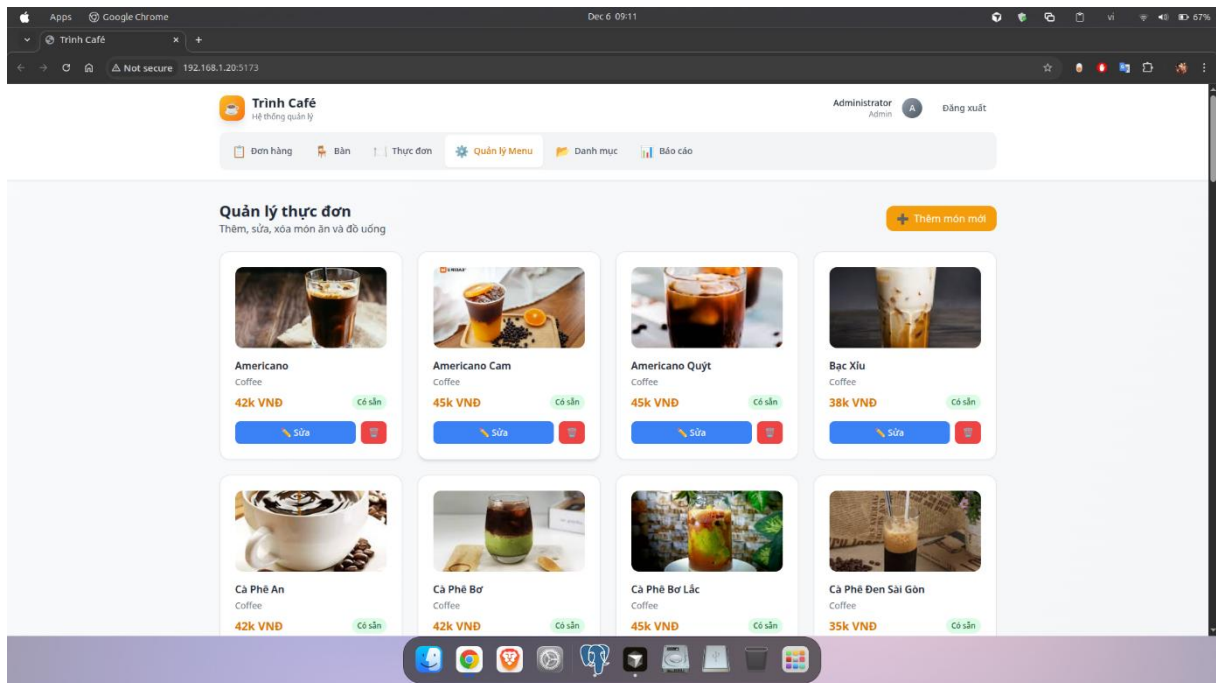
Hình 4. 6 Giao diện quản lý bàn (admin)

4.2.7. Giao diện quản lý thực đơn (admin)



Hình 4. 7 Giao diện quản lý thực đơn (admin)

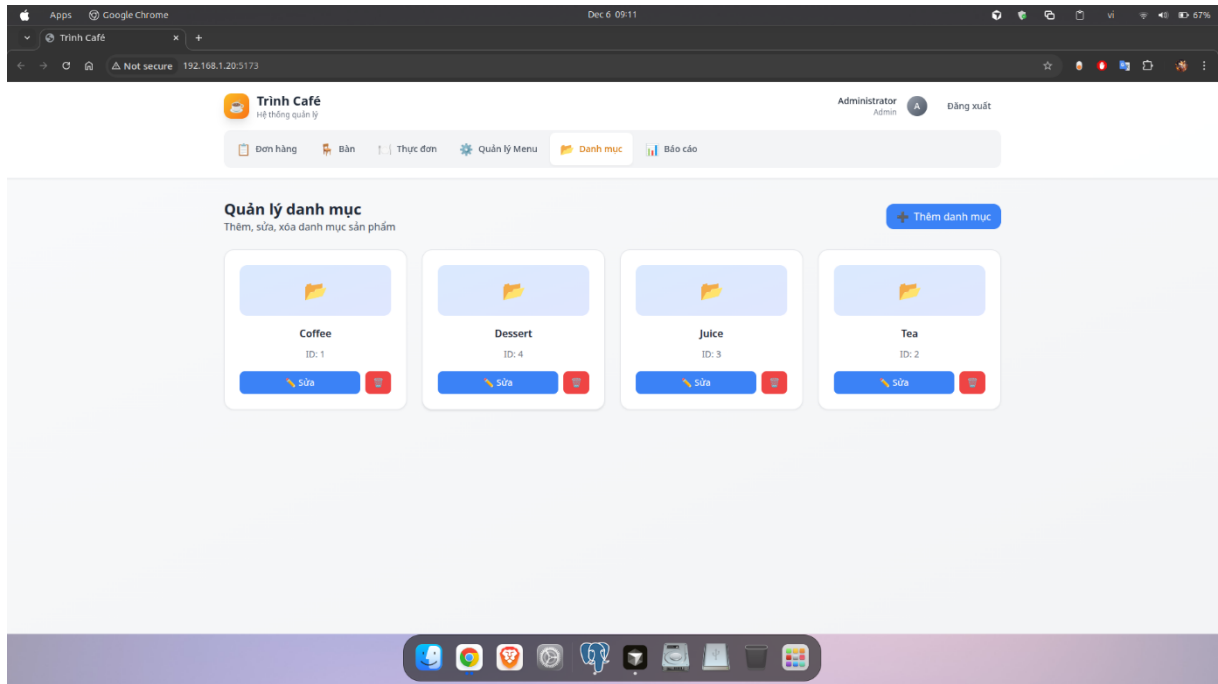
4.2.8. Giao diện quản lý menu (admin)



Hình 4. 8 Giao diện quản lý menu (admin)

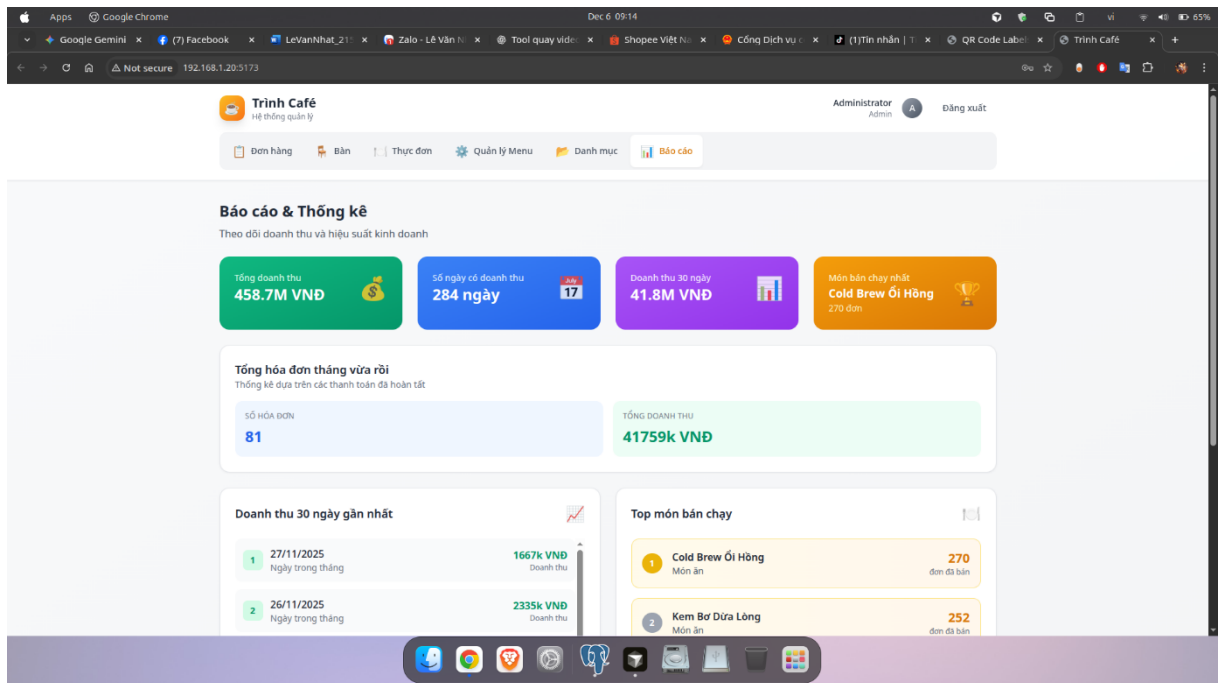
ỨNG DỤNG NODE.JS, REACT, POSTGRESQL TRONG VIỆC XÂY DỰNG HỆ THỐNG QUẢN LÝ QUÁN
TRÌNH CAFE

4.2.9. Giao diện quản lý danh mục (admin)



Hình 4. 9 Giao diện quản lý danh mục (admin)

4.2.10. Giao diện báo cáo, thống kê (admin)



Hình 4. 10 Giao diện báo cáo, thống kê (admin)

CHƯƠNG 5. KIỂM THỬ VÀ ĐÁNH GIÁ

5.1. Phương pháp kiểm thử

Để đảm bảo hệ thống hoạt động ổn định, đúng yêu cầu và đáp ứng các chức năng đã đặt ra, em tiến hành kiểm thử theo các phương pháp như sau:

5.1.1. Kiểm thử chức năng (*Functional testing*)

- Mục tiêu: kiểm tra từng chức năng có hoạt động đúng theo yêu cầu không.
- Áp dụng cho: đăng nhập, đăng ký, quản lý món, bàn, đặt món, theo dõi đơn, thống kê, ...

5.1.2. Kiểm thử đơn vị (*Unit Testing*)

- Kiểm tra logic bên trong từng module backend Node.js.
- Bao gồm: xử lý giỏ hàng, tính tổng tiền, cập nhật trạng thái đơn.

5.1.3. Kiểm thử tích hợp (*Integration Testing*)

- Kiểm tra sự tương tác giữa Frontend ReactJS với Backend API và giữa Backend với PostgreSQL.
- Đảm bảo dữ liệu truyền nhận đúng định dạng và không lỗi.

5.1.4. Kiểm thử giao diện (*UI/UX Testing*)

- Kiểm tra bố cục, độ dễ sử dụng, thao tác người dùng.
- Đánh giá thao tác đặt món, điều hướng trang, xem trạng thái đơn hàng.

5.1.5. Kiểm thử hiệu năng (*Performance Testing*)

- Kiểm tra tốc độ phản hồi của API.
- Kiểm tra khả năng chịu tải khi có nhiều người dùng truy cập cùng 1 lúc.

5.1.6. Kiểm thử bảo mật (*Security Testing*)

- Kiểm tra tính an toàn của tài khoản người dùng.
- Kiểm tra phân quyền truy cập (admin – nhân viên – khách hàng).
- Kiểm tra API có bị truy cập trái phép hay không.

5.2. Kịch bản kiểm thử

5.2.1. Kiểm thử đăng nhập

Bảng 5. 1 Bảng kiểm thử đăng nhập

STT	Chức năng	Input	Kỳ vọng	Kết quả
1	Đăng nhập hợp lệ	Email + mật khẩu đúng	Đăng nhập thành công	Pass
2	Email sai	Email không tồn tại	Thông báo lỗi	Pass
3	Mật khẩu sai	Email đúng – password sai	Thông báo lỗi	Pass

5.2.2. Kiểm thử hiển thị menu

Bảng 5. 2 Bảng kiểm thử hiển thị menu

STT	Chức năng	Input	Kỳ vọng
1	Xem menu	Không cần input	Hiển thị danh sách món đầy đủ
2	Lọc món	Chọn danh mục	Chỉ hiển thị món của danh mục đó

5.2.3. Kiểm thử đặt món

Bảng 5. 3 Bảng kiểm thử đặt món

STT	Kiểm thử	Input	Kỳ vọng
1	Thêm món vào giỏ	Chọn món → chọn số lượng	Giỏ hàng thêm món thành công
2	Xóa món	Nhấn nút xóa	Món biến mất khỏi giỏ
3	Tạo đơn hàng	Nhấn “Đặt món”	Đơn hàng được lưu vào hệ thống

5.2.4. Kiểm thử quản lý món

Bảng 5. 4 Bảng kiểm thử quản lý món

STT	Chức năng	Kỳ vọng
1	Thêm món	Món hiển thị đúng trong menu
2	Sửa thông tin món	Món cập nhật thành công
3	Xóa món	Món không còn trong danh sách

5.2.5. Kiểm thử quản lý đơn

Bảng 5. 5 Bảng kiểm thử quản lý đơn

STT	Chức năng	Kỳ vọng
1	Nhận đơn	Đơn chuyển sang trạng thái “Đang pha chế”
2	Hoàn thành	Chuyển sang “Đã phục vụ”
3	Thanh toán	Hệ thống cập nhật “Hoàn tất”

5.2.6. Kiểm thử thống kê doanh thu

Bảng 5. 6 Bảng kiểm thử thống kê doanh thu

STT	Chức năng	Kỳ vọng
1	Doanh thu từng ngày	Đúng với số đơn đã hoàn tất
2	Doanh thu theo tháng	Tổng hợp chính xác theo PostgreSQL

5.3. Kết quả kiểm thử

Sau khi tiến hành kiểm thử toàn diện, kết quả tổng hợp như sau:

5.3.1. Các chức năng hoạt động ổn định

- Đăng nhập, đăng ký hoạt động chính xác.
- Chức năng xem menu và đặt món hoạt động mượt mà.
- Giỏ hàng xử lý chính xác số lượng và tổng tiền.
- Quản trị viên sử dụng được đầy đủ chức năng quản lý món, bàn, đơn hàng.

5.3.2. Kết nối API – Database ổn định

- React gửi request đúng định dạng.
- Node.js xử lý nghiệp vụ và trả kết quả chính xác.
- PostgreSQL lưu dữ liệu đầy đủ và nhất quán.

5.3.3. Hiệu năng

- Thời gian phản hồi trung bình của các API: <1.2 giây.
- Hệ thống xử lý tốt khi kiểm thử đồng thời 20–30 request.

5.3.4. Tính ổn định và bảo mật

- Mật khẩu mã hóa thành công.
- JWT hoạt động đúng yêu cầu.
- Các API yêu cầu phân quyền đều được bảo vệ.

5.3.5. Giao diện

- Giao diện thân thiện, dễ sử dụng.
- Responsive tốt trên laptop và điện thoại.

5.4. Đánh giá ưu điểm và hạn chế của hệ thống

5.4.1. Ưu điểm

- Giao diện đẹp, dễ sử dụng, tối ưu cho khách hàng và quản trị viên.
- Quy trình đặt món nhanh, hạn chế sai sót, tiết kiệm thời gian.
- Cập nhật trạng thái đơn hàng theo thời gian thực, giúp khách theo dõi dễ dàng.
- Quản lý món, bàn, doanh thu đơn giản, dễ thao tác cho admin.
- Cấu trúc API rõ ràng, dễ mở rộng thêm chi nhánh trong tương lai.
- Cơ sở dữ liệu PostgreSQL ổn định, đảm bảo toàn vẹn dữ liệu.
- Có thể phát triển thành app mobile, do backend hỗ trợ RESTful API.

5.4.2. Hạn chế

- Chưa có tính năng thanh toán trực tuyến (VNPay, Momo).
- Chưa hỗ trợ chức năng chat trực tiếp giữa khách và nhân viên.
- Chưa có hệ thống phân tích dữ liệu nâng cao (BI, Machine Learning).
- Chưa tích hợp hệ thống in hóa đơn trực tiếp từ bếp hoặc quầy bar.
- Hệ thống thông báo real-time cần WebSocket để tối ưu hơn.

KẾT LUẬN

1. Tóm tắt kết quả nghiên cứu

Đề tài “*Xây dựng hệ thống quản lý quán Cafe sử dụng ReactJS, Node.js và PostgreSQL*” đã tập trung nghiên cứu và triển khai một giải pháp phần mềm hỗ trợ quản lý hoạt động kinh doanh trong quán cafe theo hướng hiện đại hóa và tự động hóa. Thông qua quá trình phân tích, thiết kế và xây dựng hệ thống, đề tài đã:

- Khảo sát thực tế hoạt động của mô hình quán cafe và xác định được các vấn đề tồn tại trong quản lý thủ công.
- Phân tích yêu cầu hệ thống theo hai nhóm: chức năng và phi chức năng.
- Xây dựng mô hình nghiệp vụ gồm Use Case, Activity và Sequence Diagram.
- Thiết kế cơ sở dữ liệu quan hệ với PostgreSQL đảm bảo tính toàn vẹn, nhất quán và dễ mở rộng.
- Phát triển giao diện người dùng bằng ReactJS với kiến trúc component rõ ràng, thân thiện và dễ sử dụng.
- Xây dựng backend bằng Node.js/Express theo mô hình RESTful API, kết nối chặt chẽ với cơ sở dữ liệu.
- Thực hiện kiểm thử hệ thống, đánh giá hiệu năng, bảo mật và tính ổn định để đảm bảo hệ thống đáp ứng đầy đủ yêu cầu đặt ra.

Kết quả cho thấy hệ thống hoạt động ổn định, giao diện trực quan, đầy đủ chức năng cơ bản cho cả khách hàng và quản trị viên, qua đó mang lại hiệu quả rõ rệt trong quá trình vận hành quán cafe.

2. Mức độ hoàn thành mục tiêu đề tài

Dựa trên mục tiêu đã đề ra ban đầu, đề tài đã hoàn thành hầu hết các yêu cầu cốt lõi:

✓ Hoàn thành tốt

- Xây dựng hệ thống đặt món trực tuyến qua QR.
- Quản lý món, bàn, đơn hàng, tài khoản người dùng.
- Theo dõi trạng thái đơn hàng theo thời gian thực.
- Hiện thị thống kê doanh thu theo ngày/tháng.
- Giao diện thân thiện, dễ sử dụng, responsive tốt.
- Backend Node.js hoạt động ổn định, API rõ ràng, dễ mở rộng.
- Cơ sở dữ liệu PostgreSQL thiết kế khoa học, đảm bảo toàn vẹn dữ liệu.

✓ Mức độ đáp ứng yêu cầu

- Hệ thống đạt khoảng 90–95% mục tiêu ban đầu.
- Một số tính năng nâng cao chưa triển khai do thời gian hạn chế.

3. Đóng góp chính của hệ thống

Hệ thống mang lại nhiều đóng góp thực tiễn trong hoạt động kinh doanh quán cà phê:

Đối với người dùng (Khách hàng)

- Trải nghiệm đặt món nhanh chóng, hiện đại thông qua mã QR.
- Xem trạng thái đơn hàng và lịch sử mua hàng dễ dàng.
- Hạn chế sai sót khi ghi order thủ công.

Đối với quán

- Tự động hóa toàn bộ quy trình quản lý: món, bàn, đơn, nhân viên, doanh thu.
- Theo dõi hoạt động quán theo thời gian thực.
- Tổng hợp doanh thu chính xác, chủ động trong việc đánh giá hiệu quả kinh doanh.
- Giảm chi phí nhân sự do quy trình vận hành tối ưu hơn.

Đối với lĩnh vực công nghệ

- Minh chứng rõ ràng cho khả năng kết hợp ReactJS – Node.js – PostgreSQL trong xây dựng hệ thống quản lý thực tế.
- Hệ thống dễ mở rộng sang mobile app hoặc đa chi nhánh.

4. Hướng phát triển trong tương lai

Để hệ thống hoàn thiện hơn và phù hợp với yêu cầu thực tế phức tạp, một số hướng phát triển trong tương lai gồm:

Tính năng thanh toán điện tử

- Tích hợp các cổng thanh toán như Momo, ZaloPay, VNPay, Paypal, ...
- Hỗ trợ thanh toán không chạm (contactless)

Hệ thống in hoá đơn tự động

- Kết nối với máy in hóa đơn tại quầy bar hoặc bếp.
- In order ngay khi khách đặt món.

Hỗ trợ quản lý đa chi nhánh

- Cho phép quản lý nhiều cửa hàng từ một hệ thống trung tâm.
- Báo cáo doanh thu theo từng chi nhánh.

Phân tích dữ liệu nâng cao

- Ứng dụng AI gợi ý món bán chạy.
- Phân tích hành vi khách hàng.
- Dự đoán lượng nguyên liệu cần nhập.

Phát triển ứng dụng Mobile

- Tạo app dành cho khách và admin với giao diện chuyên nghiệp.
- Đồng bộ dựa trên API backend hiện có.

Quản lý kho – nguồn cung ứng

- Quản lý nguyên vật liệu, nhập kho, xuất kho.
- Theo dõi tồn kho theo thời gian thực.

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] Nguyễn Thanh Tùng (2020). *Giáo trình Phát triển ứng dụng Web*. NXB Đại học Quốc gia TP.HCM.
- [2] Trịnh Minh Trí (2021). *Giới thiệu Node.js và ứng dụng*. Tạp chí Khoa học Công nghệ, Đại học Cần Thơ.
- [3] Đặng Văn Hùng (2019). *Tìm hiểu về ReactJS và ứng dụng trong phát triển web hiện đại*. Hội thảo CNTT & Truyền Thông Việt Nam.
- [4] Nguyễn Tấn Lợi (2020). *Cơ sở dữ liệu quan hệ và SQL nâng cao*. NXB Khoa học Kỹ thuật.
- [5] Khoa Công nghệ Thông tin – Trường Đại học Công nghệ (2022). *Bài giảng Phân tích và Thiết kế Hệ thống Thông tin*.
- [6] Phạm Minh Hiếu (2021). *Ứng dụng kiến trúc RESTful API trong phát triển Web*. Tạp chí Khoa học và Công nghệ, ĐH Đà Nẵng.
- [7] Trường Đại học Bách Khoa Hà Nội (2020). *Giáo trình Cơ sở Dữ liệu*.
- [8] Nguyễn Đình Phúc (2023). *Ứng dụng React Hooks trong xây dựng giao diện người dùng*. Bài giảng nội bộ – HUS.
- [9] Lê Văn Tuấn (2021). *Hướng dẫn lập trình website thương mại điện tử với Node.js và MySQL*. NXB Thông Tin & Truyền Thông.
- [10] Nguyễn Thị Hường (2020). *Phân tích thiết kế hệ thống thông tin quản lý trong doanh nghiệp*. Đại học Thương mại.

Tiếng Anh

- [1] Flanagan, D. (2020). *JavaScript: The Definitive Guide*. O'Reilly Media.
- [2] Freeman, A., & Robson, R. (2018). *Beginning Node.js*. Apress.
- [3] Raj, R., & Rubel, M. (2021). *Node.js Design Patterns* (Third Edition). Packt Publishing.
- [4] Banks, A., & Porcello, E. (2017). *Learning React: Functional Web Development with React and Redux*. O'Reilly Media.
- [5] Wieruch, R. (2020). *The Road to React*. Self-published.
- [6] PostgreSQL Global Development Group (2023). *PostgreSQL Documentation*. Truy cập tại: <https://www.postgresql.org/docs/>
- [7] Express.js Foundation (2023). *Express.js Guide: API Reference*. <https://expressjs.com/>

- [8] React Team – Meta (2023). *React Official Documentation*.
<https://react.dev/>
- [9] Mozilla Developer Network (MDN). *Web Technology Documentation*.
<https://developer.mozilla.org/>
- [10] Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley.
- [11] Garverick, R. (2022). *RESTful Web APIs with Node.js*. Packt Publishing.
- [12] ISO/IEC 25010:2011. *Systems and software engineering – System quality models*.