

**TRƯỜNG ĐẠI HỌC KỸ THUẬT – CÔNG NGHỆ CẦN THƠ**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN HỌC PHẦN 3**

**TÊN ĐỀ TÀI**

**TÌM HIỂU VỀ MERN STACK VÀ XÂY DỰNG ỨNG DỤNG MINH HOẠ**

**Giảng Viên Hướng Dẫn**

Th.S Trần Thị Kim Khánh

**Sinh Viên Thực Hiện**

Nguyễn Lưu Ngọc Yên - 1800777

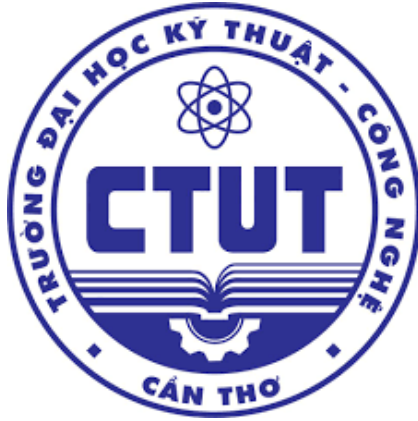
Huỳnh Thị Hồng Gấm - 1800196

Nguyễn Hùng Cường - 1800553

Ngành: Hệ Thống Thông Tin

**Cần Thơ 2022**

**TRƯỜNG ĐẠI HỌC KỸ THUẬT – CÔNG NGHỆ CẦN THƠ**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN HỌC PHẦN 3**

**TÊN ĐỀ TÀI**

**TÌM HIỂU VỀ MERN STACK VÀ XÂY DỰNG ỨNG DỤNG MINH HOẠ**

**Giảng Viên Hướng Dẫn**

Th.S Trần Thị Kim Khánh

**Sinh Viên Thực Hiện**

Nguyễn Lưu Ngọc Yến - 1800777

Huỳnh Thị Hồng Gấm - 1800196

Nguyễn Hùng Cường - 1800553

Ngành: Hệ Thống Thông Tin

[illegible]

[illegible]

## LỜI CAM ĐOAN

Chúng em xin cam đoan đề tài: “*Đồ án học phần 3 Tìm hiểu về MERN stack và xây dựng ứng dụng minh họa*” là một công trình nghiên cứu độc lập của nhóm dưới sự hướng dẫn của giáo viên hướng dẫn: Th.S Trần Thị Kim Khánh. Ngoài ra không có bất cứ sự sao chép của người khác. Đề tài, nội dung báo cáo đồ án là sản phẩm mà chúng em đã nỗ lực nghiên cứu trong quá trình học tập tại trường và nghiên cứu thêm từ nhiều nguồn khác nhau. Các số liệu, kết quả trình bày trong đồ án là hoàn toàn trung thực, chúng em xin chịu hoàn toàn trách nhiệm, kỷ luật của bộ môn và nhà trường đề ra nếu như có vấn đề xảy ra.

## LỜI CẢM ƠN

Để hoàn thành đồ án này, chúng em xin tỏ lòng biết ơn sâu sắc đến Th.S Trần Thị Kim Khánh đã tận tình hướng dẫn chúng em trong suốt quá trình làm bài. Bên cạnh đó chúng em xin chân thành cảm ơn các thầy, các cô trong Khoa Công Nghệ Thông Tin đã tạo điều kiện cho chúng em có một môn học hay và bổ ích để có thể mở rộng kiến thức, giúp chúng em hoàn thiện hơn trong quá trình học tập. Tuy nhiên quá trình nghiên cứu đồ án, kiến thức chuyên ngành còn hạn chế nên chúng em vẫn còn nhiều thiếu sót tìm hiểu, đánh giá và trình bày về đồ án. Rất mong nhận được sự quan tâm, góp ý của các thầy/ cô giảng viên bộ môn để đồ án của chúng em được đầy đủ và hoàn chỉnh. Xin chân thành cảm ơn.

*Chúng em xin chân thành cảm ơn.*

## TÓM TẮT

Với sự phát triển mạnh mẽ của công nghệ thông tin như hiện nay, cùng với các mô hình, ứng dụng, ngày càng phát triển, có thể nói khối lượng người sử dụng công nghệ thông tin để phục vụ cho công việc, đời sống ngày một tăng, cùng với đó là khối lượng truy cập dữ liệu lưu trữ ngày một lớn và phức tạp, các mô hình, ứng dụng ra đời phải đảm bảo đủ các yếu tố bảo mật, nhanh chóng, tiện lợi, đồng bộ và dễ dàng nâng cấp để phục vụ tối ưu nhất cho các nhu cầu của người sử dụng.

Thông qua sự tìm hiểu và nghiên cứu cho thấy MongoDB, Express, ReactJS và NodeJS kết hợp với nhau được cho là khá phù hợp với các nhu cầu như khối lượng dữ liệu lớn, dễ dàng nâng cấp, hỗ trợ đồng bộ tốt, nhanh nhất và dễ dàng nâng cấp bảo trì hệ thống, ứng dụng.

Chúng em lựa chọn đề tài *"Tìm hiểu về MERN stack và xây dựng ứng dụng minh họa"*, với mong muốn là nghiên cứu những công nghệ mới và xây dựng một ứng dụng đặt thức ăn với các chức năng cơ bản như: Thêm cửa hàng, thêm menu món ăn, thêm và cập nhật món ăn vào giỏ hàng.

### **BẢNG KÝ HIỆU VIẾT TẮT**

NoSQL	Not Only SQL
CSDL	Cơ sở dữ liệu
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
JS	JavaScript



# MỤC LỤC

<b>LỜI MỞ ĐẦU</b>	1
<b>CHƯƠNG I: TỔNG QUAN</b>	2
1.1 Giới thiệu tổng quan đề tài:	2
1.2 Mục tiêu đề tài:	2
1.3 Phương pháp nghiên cứu:	2
1.4 Phạm vi nghiên cứu:	2
1.5 Ý nghĩa đề tài:	3
<b>CHƯƠNG II: LÝ THUYẾT</b>	4
2.1 Giới thiệu MERN stack:	4
2.2 Giới thiệu MongoDB:	4
2.2.1 Ưu điểm:	5
2.2.2 Nhược điểm:	5
2.2.3 Hệ cơ sở dữ liệu MongoDB nên dùng trong trường hợp:	5
2.3 Giới thiệu về Express.JS:	5
2.4 Giới thiệu về ReactJS:	6
2.4.1 Các thành phần trong React:	7
2.5 Giới thiệu về NodeJS:	7
2.6 Hướng dẫn cài đặt MERN Stack:	8
2.6.1 Cài đặt NodeJS:	8
2.6.2 Thiết lập cơ sở dữ liệu MongoDB	10
2.6.3 Cài đặt ReactJS	11
<b>CHƯƠNG III: KHẢO SÁT VÀ ĐẶT TẢ WEBSITE</b>	12
3.1 Khảo sát thực trạng về chủ thể đề tài:	12
3.2 Các chức năng nghiệp vụ website:	12
<b>CHƯƠNG IV: PHÂN TÍCH THIẾT KẾ HỆ THỐNG</b>	13
4.1 Sơ đồ usecase:	13
4.1.1 Usecase tổng quát:	13

4.1.2 Usecase đăng nhập:.....	13
4.1.3 Usecase Admin: .....	14
4.1.4 Usecase Người Bán:.....	14
4.1.5 Usecase Người Mua:.....	15
4.2 Sơ đồ class: .....	15
<b>CHƯƠNG V : HƯỚNG DẪN SỬ DỤNG PHẦN MỀM</b> .....	16
5.1 Đăng ký và đăng nhập vào hệ thống.....	16
5.2 Giao diện của hệ thống .....	18
5.3 Chức năng nghiệp vụ theo từng vai trò.....	19
5.3.1 Quản trị viên (Admin).....	19
5.3.2 Người bán hàng.....	21
5.3.3 Khách hàng .....	23
<b>KẾT LUẬN</b> .....	25
<b>TÀI LIỆU THAM KHẢO</b> .....	26

## MỤC LỤC HÌNH ẢNH

Hình 1 Mô hình kiến trúc của MERN Stack.....	4
Hình 2 Sơ đồ hoạt động của Express .....	6
Hình 3 Trang chủ của NodeJS .....	8
Hình 4 Kiểm tra phiên bản NodeJS .....	9
Hình 5 Khởi tạo Server cho ứng dụng .....	9
Hình 6 Cài đặt Express và các thư viện chức năng.....	9
Hình 7 Thiết lập cơ sở dữ liệu MongoDB .....	10
Hình 8 Cài đặt ReactJS .....	11
Hình 9 Chạy ứng dụng ReactJS .....	11
Hình 10 : Sơ đồ Usecase tổng quát website đặt thức ăn .....	13
Hình 11 Sơ đồ Usecase đăng nhập website .....	13
Hình 12 Sơ đồ usecase chức năng Admin .....	14
Hình 13 Sơ đồ usecase chức năng Người Bán.....	14
Hình 14 Sơ đồ usecase chức năng Người Mua.....	15
Hình 15 Sơ đồ class .....	15
Hình 16 Giao diện đăng ký .....	16
Hình 17 Giao diện đăng nhập .....	17
Hình 18 Giao diện chính của hệ thống.....	18
Hình 19 Giao diện của quản trị viên .....	18
Hình 20 Giao diện của người bán hàng .....	19
Hình 21 Chỉnh sửa thông tin tài khoản người dùng.....	19
Hình 22 Cập nhật lại thông tin muốn chỉnh sửa .....	20
Hình 23 Chỉnh sửa thành công.....	20
Hình 24 Hệ thống thông báo cập nhật lại thành công.....	20
Hình 25 Nhập thông tin của cửa hàng .....	21
Hình 26 Thêm thông tin cửa hàng thành công.....	21
Hình 27 Nhập thông tin món ăn.....	22
Hình 28 Thêm món ăn thành công.....	22
Hình 29 Cập nhật lại món ăn cần chỉnh sửa .....	23
Hình 30 Cập nhật thành công.....	23
Hình 31 Giao diện các món ăn của cửa hàng.....	24
Hình 32 Chọn món ăn muốn thêm vào giỏ hàng .....	24
Hình 33 Thêm vào giỏ hàng thành công .....	24

## LỜI MỞ ĐẦU

Có thể thấy được hiện nay với nhu cầu sử dụng internet và giải quyết các vấn đề của con người trên chiếc điện thoại thông minh, máy tính là ngày càng cao, với các hệ thống hay phần mềm thì cần đảm bảo được các yêu cầu về việc đồng bộ, tính nhanh chóng và dễ dàng nâng cấp bảo trì để có thể kịp thời cập nhật theo thời đại và thị trường.

Với các hệ thống bán hàng lớn, hoặc các dự án công việc có nhiều truy cập sẽ rất khó khăn khi phải sử dụng cùng một máy chủ, khó để giải quyết các vấn đề về đa nền tảng, dữ liệu cứng nhắc và khó để cập nhật.

Trong quá trình tìm hiểu và học tập các công nghệ cải tiến, chúng em nhận thấy MERN stack là một mô hình sử dụng Open Source với các công nghệ liên quan đến Javascript như: MongoDB, ExpressJS, React/React Native, NodeJS.

MERN stack thật sự rất phù hợp với các dự án có khối lượng truy cập cao, dữ liệu phức tạp và có nhu cầu cập nhật mới liên tục.

Dưới sự hướng dẫn của giảng viên Trần Thị Kim Khánh đã giúp đỡ để chúng em có thể hoàn thành đồ án này. Tuy nhiên do là lần đầu tìm hiểu với MERN stack và kiến thức chuyên ngành còn hạn hẹp sẽ không thể tránh khỏi những thiếu sót, chúng em rất mong nhận được sự thông cảm và đóng góp từ quý thầy cô.

*Chúng em xin chân thành cảm ơn.*

# CHƯƠNG I: TỔNG QUAN

## 1.1 Giới thiệu tổng quan đề tài:

Hiện nay với sự phát triển của công nghệ thông tin thì tất cả mọi công việc đều có thể giải quyết bằng một cú click chuột, nhu cầu của con người mỗi ngày đều được nâng cao và phát triển để phục vụ cho cuộc sống như đi lại, chi trả, ăn uống,...

Nhắc đến ăn uống hiện nay có rất nhiều loại hình kinh doanh phát triển để tiếp cận với khách hàng và người bán, cũng như việc ảnh hưởng của đại dịch Covid 19 mọi người càng có nhu cầu ít tiếp xúc, thêm vào đó là sự bận rộn nên nhịp sống cần sự tiện lợi và nhanh chóng, từ đó các trang web đặt thức ăn online ra đời giải quyết được vấn đề di chuyển hoặc tiếp xúc của mọi người.

Trang web đặt thức ăn giúp mọi người có thể theo dõi được gian hàng, giá bán... dễ dàng thao tác và lựa chọn và tìm kiếm các đồ ăn, thức uống mà mình yêu thích. Người bán cũng có thể dễ dàng tiếp cận với khách hàng thông qua trang web.

Ứng dụng MERN stack vào hệ thống website đặt thức ăn giúp đồng bộ dữ liệu, phổ biến ở đa nền tảng và giúp hệ thống dễ dàng hơn trong việc bảo hành, sửa chữa, nâng cấp.

## 1.2 Mục tiêu đề tài:

Tìm hiểu, phân tích và xây dựng ứng dụng minh họa dựa trên MERN stack và hiểu được cách hoạt động của nó.

## 1.3 Phương pháp nghiên cứu:

Trong quá trình học tập tại trường Đại Học Kỹ Thuật – Công Nghệ Cần Thơ đã được các giảng viên trao dồi các kiến thức bổ ích liên quan đến các môn như: Phân Tích Thiết Kế Hệ Thống Thông Tin, NoSQL, Lập Trình Web, ... đặc biệt là các ngôn ngữ lập trình như:JS, CSS, HTML,...để có nền tảng kiến thức vững chắc thực hiện, thiết kế website **Đặt thức ăn** này.

## 1.4 Phạm vi nghiên cứu:

Website này ra đời với ý nghĩa và phạm vi phục vụ cho quá trình nghiên cứu MERN Stack

Website quan tâm chủ yếu đến vấn đề thêm, bớt sản phẩm vào giỏ hàng thành công và xem được giá tiền hiện tại của đơn hàng và không quan tâm đến đặt hàng.

## 1.5 Ý nghĩa đề tài:

*Ý nghĩa khoa học:*

Đề tài này mang ý nghĩa to lớn đối với cá nhân nhóm tạo ra, mặc dù còn có nhiều thiếu sót nhưng đây là đề tài đầu tiên tự nghiên cứu và tìm hiểu, đây cũng là một kinh nghiệm tích lũy trong quá trình học tập tại trường để sau này giúp ích cho công việc sau này.

*Ý nghĩa thực tiễn:*

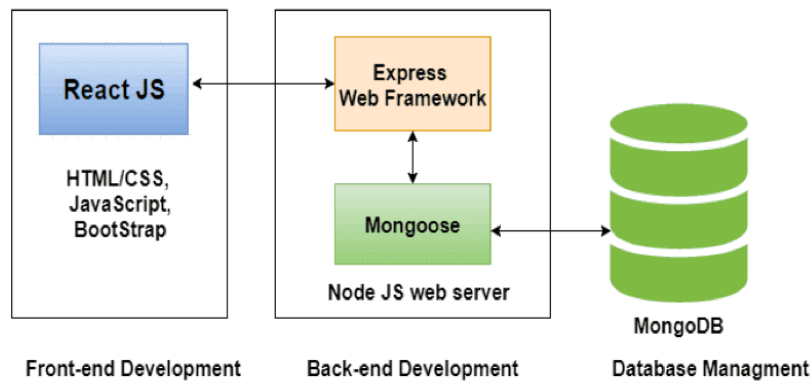
Đối với ý nghĩa cộng đồng, đề tài hỗ trợ cho những website bán hàng có thể hoạt động nhanh chóng, chính xác, đồng bộ khi có lượng truy cập, thao tác lớn và lưu trữ dữ liệu một cách hiệu quả.

## CHƯƠNG II: LÝ THUYẾT

### 2.1 Giới thiệu MERN stack:

MERN là viết tắt của MongoDB, Express, React, Node, theo tên bốn công nghệ chính tạo nên ngăn xếp.

- MongoDB - cơ sở dữ liệu tài liệu
- Express (.js) - Khung web Node.js
- React (.js) - một khung JavaScript phía máy khách
- Node (.js) - máy chủ web JavaScript hàng đầu



Hình 1 Mô hình kiến trúc của MERN Stack

MERN là một full-stack, tuân theo mô hình kiến trúc 3 tầng truyền thống, bao gồm tầng hiển thị front-end (React.js), tầng ứng dụng (Express.js và Node.js) và tầng cơ sở dữ liệu (MongoDB).

### 2.2 Giới thiệu MongoDB:

Cơ sở dữ liệu tài liệu ở gốc của ngăn xếp MERN. MongoDB được thiết kế để lưu trữ dữ liệu JSON nguyên bản (về mặt kỹ thuật, nó sử dụng phiên bản nhị phân của JSON được gọi là BSON) và mọi thứ từ giao diện dòng lệnh đến ngôn ngữ truy vấn của nó (MQL hoặc MongoDB Query Language) đều được xây dựng trên JSON và JavaScript.

MongoDB hoạt động cực kỳ hiệu quả với Node.js và làm cho việc lưu trữ, thao tác và biểu diễn dữ liệu JSON ở mọi cấp ứng dụng của bạn trở nên vô cùng dễ dàng. Đối với các ứng dụng gốc đám mây, MongoDB Atlas thậm chí còn làm điều đó dễ dàng hơn, bằng cách

cung cấp cho bạn một cụm MongoDB tự động mở rộng quy mô trên nhà cung cấp đám mây mà bạn chọn, dễ dàng chỉ bằng một vài cú nhấp chuột.

MongoDB hoạt động dựa vào các khái niệm Collection và Document. MongoDB là một dạng phần mềm cơ sở dữ liệu sử dụng mã nguồn mở NoSQL. Dữ liệu trong MongoDB được lưu bằng định dạng kiểu JSON.

MongoDB là hệ CSDL không ràng buộc, mã nguồn mở, khả năng mở rộng theo chiều ngang, độ chịu tải cao, yêu cầu tài nguyên thấp.

### **2.2.1 Ưu điểm:**

Linh hoạt trong lưu trữ các kích cỡ dữ liệu khác nhau. Tiết kiệm thời gian trong việc kiểm tra sự tương thích về cấu trúc khi thêm, xóa hoặc cập nhật dữ liệu. Tốc độ truy vấn của MongoDB nhanh hơn so với RDBMS do toàn bộ dữ liệu truy vấn đã được ghi đệm lên bộ nhớ RAM.

### **2.2.2 Nhược điểm:**

Một nhược điểm của NoSQL là hầu hết các giải pháp đều không tuân thủ ACID mạnh mẽ (Atomic, Consistency, Isolation, Durability) như các hệ thống RDBMS được thiết lập tốt hơn.

Giao dịch phức tạp không có chức năng hoặc thủ tục lưu trữ tồn tại liên kết logic.

Không phù hợp với hệ thống giao dịch cao và hệ thống kết hợp chặt chẽ hoặc nơi mô hình dữ liệu được thiết kế trước.

### **2.2.3 Hệ cơ sở dữ liệu MongoDB nên dùng trong trường hợp:**

Khi website cần có tính năng thêm dữ liệu thường xuyên, bởi MongoDB có thể ghi với tốc độ cao và cực kỳ bảo mật. Điều này giúp trang web không bị downtime khi cập nhật dữ liệu. Khi có hệ thống dữ liệu với dung lượng lớn và cần truy vấn nhanh. Khi muốn tìm kiếm thông tin trong bộ nhớ nhanh chóng vì MongoDB có tốc độ tìm kiếm thông tin liên quan cực kỳ nhanh.

## **2.3 Giới thiệu về Express.JS:**

Express.js là một framework được xây dựng trên nền tảng của Node.js. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Express.js hỗ trợ các method HTTP và middleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

Express.js tương đối tối thiểu với nhiều tính năng có sẵn dưới dạng plugin. Express là thành phần back-end của các ngăn xếp phát triển phổ biến như ngăn xếp MEAN, MERN

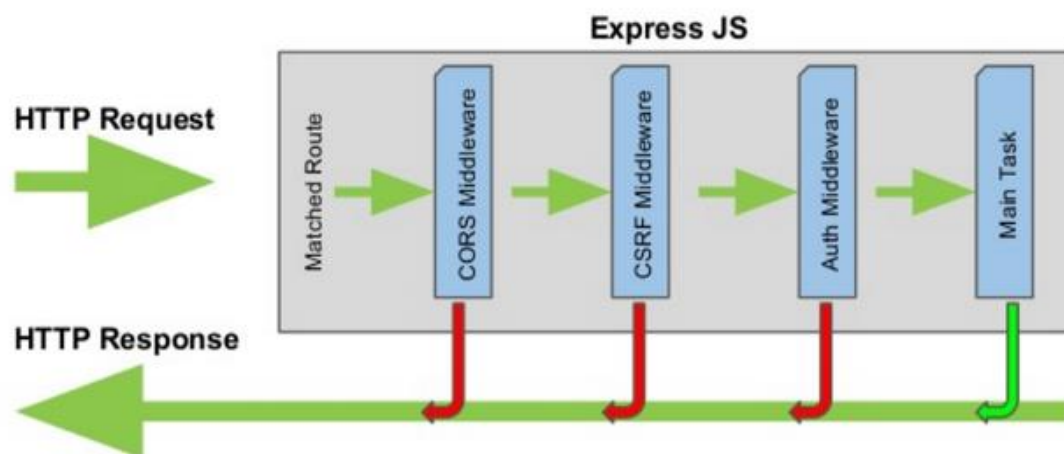


hoặc MEVN, cùng với phần mềm cơ sở dữ liệu MongoDB và khung hoặc thư viện JavaScript front-end.

Express là một khung ứng dụng web Node.js tối thiểu và linh hoạt. Nó tạo điều kiện cho sự phát triển nhanh chóng của các ứng dụng Web dựa trên Node. Sau đây là một số tính năng cốt lõi của Express framework:

*Các chức năng chính của Express.JS:*

- Xác định bằng định tuyến được sử dụng để thực hiện các hành động khác nhau dựa trên phương thức HTTP và URL.
- Cho phép hiển thị động các trang HTML dựa trên việc chuyển các đối số đến các mẫu.
- Cho phép thiết lập phần mềm trung gian để phản hồi các yêu cầu HTTP.
- Define router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
- Middleware (phần mềm trung gian): Thực thi các lệnh yêu cầu và gửi phản hồi lại các đối tượng.



Hình 2 Sơ đồ hoạt động của Express

## 2.4 Giới thiệu về ReactJS:

ReactJS là một thư viện JavaScript để xây dựng các thành phần giao diện của người dùng (UI) có thể sử dụng lại.

DOM (Document Object Model) là một cấu trúc trừu tượng của text. Mỗi đoạn mã HTML được gọi là HTML DOM và mỗi elements trong HTML là nodes của DOM đó.

Virtual DOM là một DOM ảo được xây dựng dựa trên DOM thật, có một vài thuộc tính của DOM thật nhưng khi thực hiện thay đổi nodes thì không giống với DOM thật.

### 2.4.1 Các thành phần trong React:

Component là phân chia UI thành các thành phần nhỏ để dễ dàng quản lý và tái sử dụng.

Có 2 loại:

- Function components: có ưu điểm là nhanh gọn và không quá phức tạp nhưng nó sẽ không có khái niệm: state, life cycles, events...
- Class components: sẽ sử dụng được hầu hết các chức năng của components như state, life cycles, events... và có thể tổ chức code theo mô hình hướng đối tượng.

JSX được viết tắt là JavaScript XML nó sẽ giúp cho viết các đoạn mã trong HTML có cấu trúc và rõ ràng hơn. Được sử dụng để làm việc với UI trong JavaScript code.

State là một object có thể được sử dụng để chứa dữ liệu hoặc thông tin của component.

State được dùng để phản hồi các yêu cầu từ người dùng, hay lưu trữ một dữ liệu nào đó trong components và dễ dàng thao tác với Class Components.

Props là một object được truyền vào một trong components, mỗi components sẽ nhận vào props và trả về react element. Props cho phép ta giao tiếp giữa các component với nhau bằng cách truyền tham số qua lại giữa components.

Khi một components cha truyền props vào components con thì con chỉ có thể đọc và không có quyền chỉnh sửa bên phía cha.

Sự khác nhau giữa State và Props: State có thể thay đổi bất cứ khi nào mong muốn. Khác với Props có thể truyền sang các components khác nhau thì State chỉ tồn tại trong phạm vi của components mỗi khi thay đổi thì components đó sẽ được render lại.

### 2.5 Giới thiệu về NodeJS:

Là một môi trường chạy mã nguồn mở, đa nền tảng cho phép các nhà phát triển tạo tất cả các loại công cụ và ứng dụng phía máy chủ bằng JavaScript. Thời gian chạy được thiết kế để sử dụng bên ngoài ngữ cảnh trình duyệt (tức là chạy trực tiếp trên máy tính hoặc hệ điều hành máy chủ). Do đó, môi trường bỏ qua các API JavaScript dành riêng cho trình duyệt và bổ sung hỗ trợ cho các API hệ điều hành truyền thống hơn bao gồm HTTP và thư viện hệ thống tệp.

NodeJs là một nền tảng được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở

rộng. NodeJs cung cấp khả năng tùy chỉnh, tính linh hoạt và một thư viện lớn các gói để giúp tạo ra các ứng dụng full – stack giàu tính năng.

Node được thiết kế để tối ưu hóa thông lượng và khả năng mở rộng trong các ứng dụng web và là giải pháp tốt cho nhiều vấn đề phát triển web phổ biến.

JavaScript là một ngôn ngữ lập trình tương đối mới và được hưởng lợi từ những cải tiến trong thiết kế ngôn ngữ khi so sánh với các ngôn ngữ máy chủ web truyền thống khác.

Nhiều ngôn ngữ mới và phổ biến khác được biên dịch / chuyển đổi thành JavaScript nên bạn cũng có thể sử dụng TypeScript, CoffeeScript, ClojureScript, Scala, LiveScript.

Node module là một block code có thể tái sử dụng mà sự tồn tại của nó ảnh hưởng đến các code khác.

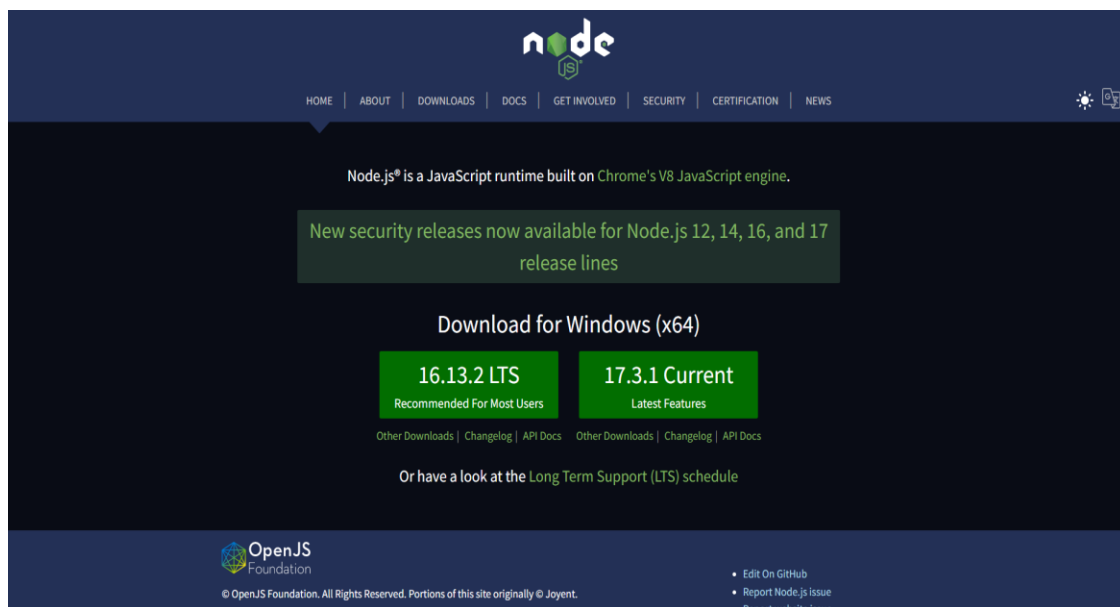
Npm: (Node package manager) là công cụ tạo và quản lý các thư viện lập trình javascript cho nodejs. Npm cũng giống như composer trong PHP, nó sử dụng command line để quản lý (download, update, delete) các package cho NodeJs.

Package.json: để quản lý các gói cài đặt cục bộ bằng npm sẽ thông qua package json là file nằm trong thư mục gốc của project.

## 2.6 Hướng dẫn cài đặt MERN Stack

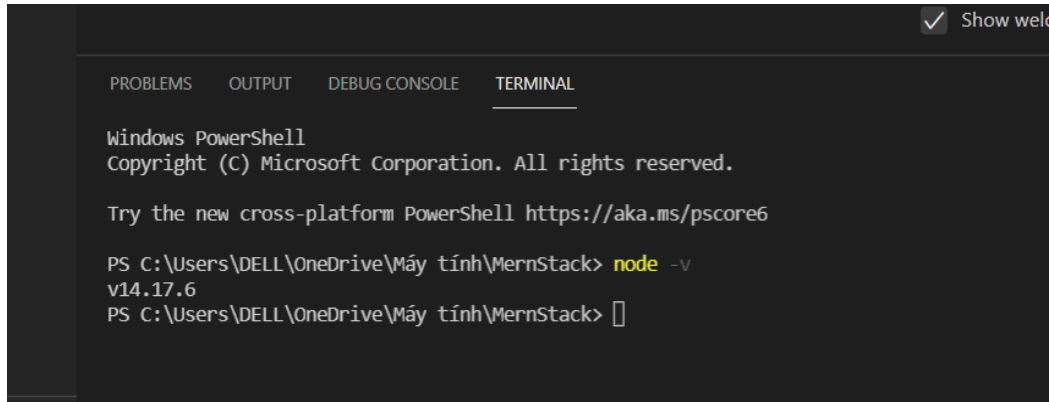
### 2.6.1 Cài đặt NodeJS

Tải NodeJs theo đường dẫn và cài đặt theo hướng dẫn: <https://nodejs.org>



Hình 3 Trang chủ của NodeJS

Sau khi đã tải và cài đặt. Ta kiểm tra bằng cách nhập lệnh **node -v** trên Terminal trong Visual Studio Code.

A screenshot of the Visual Studio Code terminal window. The terminal shows the Windows PowerShell prompt. The user enters the command 'node -v', and the output is 'v14.17.6'. The terminal tabs at the top are PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The terminal text is as follows:

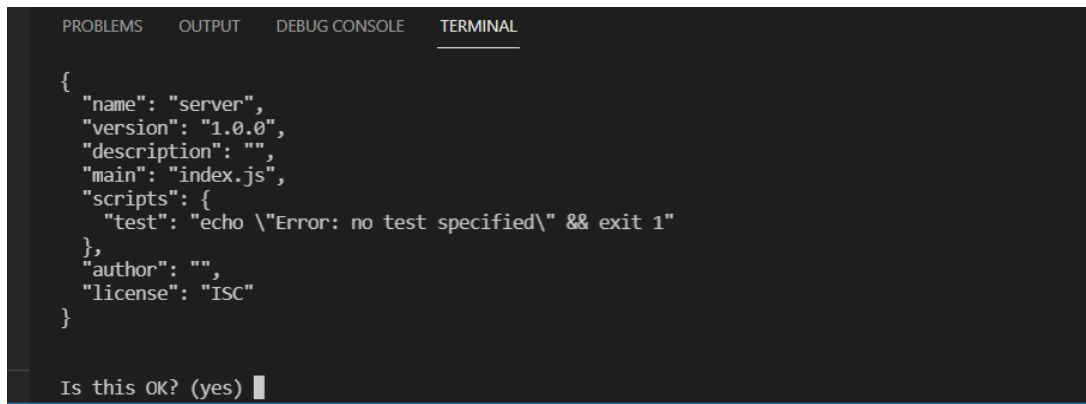
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\DELL\OneDrive\Máy tính\MernStack> node -v
v14.17.6
PS C:\Users\DELL\OneDrive\Máy tính\MernStack> 
```

Hình 4 Kiểm tra phiên bản NodeJS

Tạo Folder App và khởi tạo Server cho ứng dụng ta sẽ sử dụng câu lệnh **npm init** trong terminal.

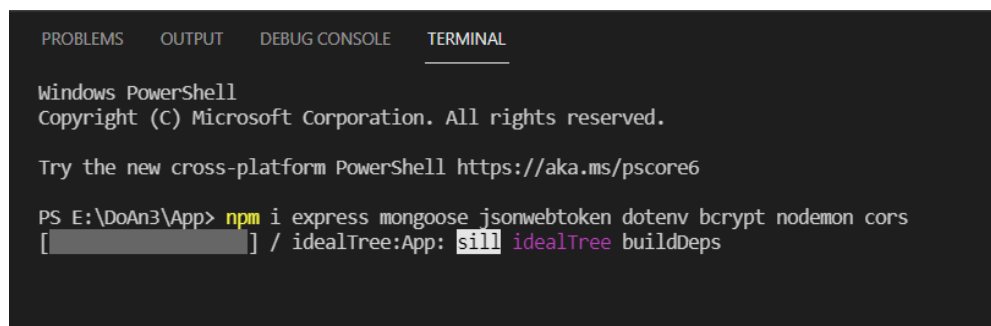
A screenshot of the Visual Studio Code terminal window. The terminal shows the output of the 'npm init' command. It displays a JSON object for the package.json file. The terminal tabs at the top are PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The terminal text is as follows:

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

Is this OK? (yes) 
```

Hình 5 Khởi tạo Server cho ứng dụng

Sử dụng lệnh **npm i express mongoose jsonwebtoken dotenv bcrypt nodemon cors** để cài đặt Express và các thư viện chức năng cơ bản của một ứng dụng web.

A screenshot of the Visual Studio Code terminal window. The terminal shows the command 'npm i express mongoose jsonwebtoken dotenv bcrypt nodemon cors' being entered. The terminal tabs at the top are PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The terminal text is as follows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\DoAn3\App> npm i express mongoose jsonwebtoken dotenv bcrypt nodemon cors
[ ] / idealTree:App: sill idealTree buildDeps
```

Hình 6 Cài đặt Express và các thư viện chức năng

Các thư viện của NodeJS

**Mongoose:** được sử dụng để thao tác giữa NodeJs và cơ sở dữ liệu MongoDB.

**Jsonwebtoken:** là phương tiện chuyển giao giữa hai bên Client – Server được định dạng bằng mã Token nhằm xác thực tài khoản người dùng hoặc trao đổi thông tin.

**Dotenv:** là module được sử dụng để truy cập các biến môi trường trong ứng dụng của mình.

**Bcrypt:** là một thuật toán để mã hóa các dữ liệu quan trọng khi lưu vào cơ sở dữ liệu.

**Nodemon:** là một công cụ NodeJS để tự khởi động lại tập lệnh khi có sự thay đổi trong thư mục.

**Cors:** là một cơ chế cho phép nhiều tài nguyên khác nhau (fonts, Javascript, v.v...) của một trang web có thể được truy vấn từ các domain khác.

Ngoài ra còn rất nhiều thư viện khác tùy theo nhu cầu của người lập trình xây dựng ứng dụng web.

## 2.6.2 Thiết lập cơ sở dữ liệu MongoDB

Truy cập vô đường dẫn <https://www.mongodb.com> để có thể thiết lập cơ sở dữ liệu MongoDB.

The screenshot shows a web-based wizard titled "Connect to Doan-3". It has a progress bar at the top with three steps: "Setup connection security" (completed), "Choose a connection method" (completed), and "Connect" (current step). Step 1, "Select your driver and version", shows "Node.js" selected for the DRIVER and "4.0 or later" for the VERSION. Step 2, "Add your connection string into your application code", includes a checkbox for "Include full driver code example" and a text area containing the connection string: `mongodb+srv://Doan-3:<password>@doan-3.eadiq.mongodb.net/myFirstDatabase?retryWrites=true&w=majority`. Below the text area, instructions state to replace `<password>` with the actual password and `myFirstDatabase` with the database name, ensuring options are URL encoded. A link for "View our troubleshooting documentation" is provided. At the bottom, there are "Go Back" and "Close" buttons.

Hình 7 Thiết lập cơ sở dữ liệu MongoDB

### 2.6.3 Cài đặt ReactJS

Sử dụng câu lệnh `npx create-react-app my-app` để cài đặt ReactJS với tên folder là my-app.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Run create-react-app --help to see all options.
PS E:\DoAn3\App> npx create-react-app my-app

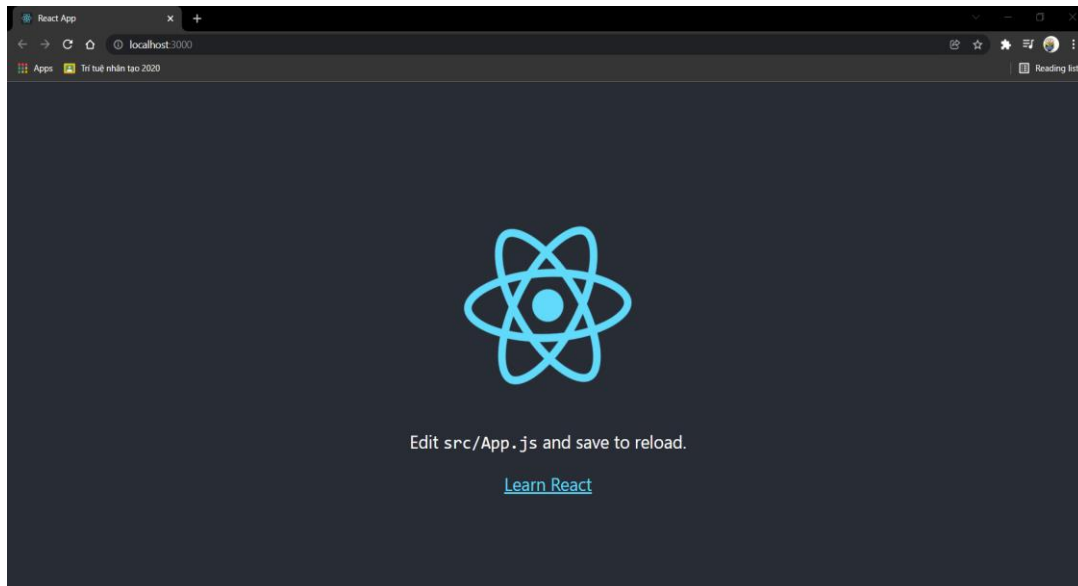
Creating a new React app in E:\DoAn3\App\my-app.

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

[ ] / idealTree:my-app: sill idealTree buildDeps
```

Hình 8 Cài đặt ReactJS

Sau khi đã cài đặt thành công thì ta sử dụng lệnh `npm start` để chạy ứng dụng



Hình 9 Chạy ứng dụng ReactJS

Cài đặt thư viện mở rộng của ReactJS ta sử dụng câu lệnh `npm install tên_thư_viện`

**Bootstrap:** hỗ trợ cho việc import Bootstrap cho giao diện website.

**React-redux:** là thư viện giúp ta quản lý state trong một ứng dụng.

**React-router-dom:** là thư viện định tuyến tiêu chuẩn trong React. Nó giữ cho giao diện đồng bộ với URL trên trình duyệt.

**Axios:** là thư viện HTTP Client dựa trên Promise. Nó sẽ gửi đi các Request HTTP bắt đồng bộ đến Server và thực hiện CRUD.

## **CHƯƠNG III: KHẢO SÁT VÀ ĐẶT TẢ WEBSITE**

### **3.1 Khảo sát thực trạng về chủ thể đề tài:**

Hiện nay với sự phát triển của công nghệ thông tin thì tất cả mọi công việc đều có thể giải quyết bằng một cú click chuột, nhu cầu của con người mỗi ngày đều được nâng cao và phát triển để phục vụ cho cuộc sống như đi lại, chi trả, ăn uống,...

Nhắc đến ăn uống hiện nay có rất nhiều loại hình kinh doanh phát triển để tiếp cận với khách hàng và người bán, cũng như việc ảnh hưởng của đại dịch Covid 19 mọi người càng có nhu cầu ít tiếp xúc, thêm vào đó là sự bận rộn nên nhịp sống cần sự tiện lợi và nhanh chóng, từ đó các trang web đặt thức ăn online ra đời giải quyết được vấn đề di chuyển hoặc tiếp xúc của mọi người.

Trang web đặt thức ăn giúp mọi người có thể theo dõi được gian hàng, giá bán... dễ dàng thao tác và lựa chọn và tìm kiếm các đồ ăn, thức uống mà mình yêu thích. Người bán cũng có thể dễ dàng tiếp cận với khách hàng thông qua trang web.

### **3.2 Các chức năng nghiệp vụ website:**

Hệ thống đặt thức ăn online có các người dùng như: Admin (quản trị hệ thống), người bán, khách hàng.

#### **3.2.1 Admin:**

Khi sử dụng hệ thống cần phải có tài khoản đăng nhập. Sau khi đăng nhập thành công, admin có thể quản lý được các tài khoản của khách hàng đăng kí.

#### **3.2.2 Người bán:**

Cần phải đăng kí tài khoản để sử dụng hệ thống. Sau khi đăng kí thành công, người bán đăng nhập để được đăng tải sản phẩm lên bán.

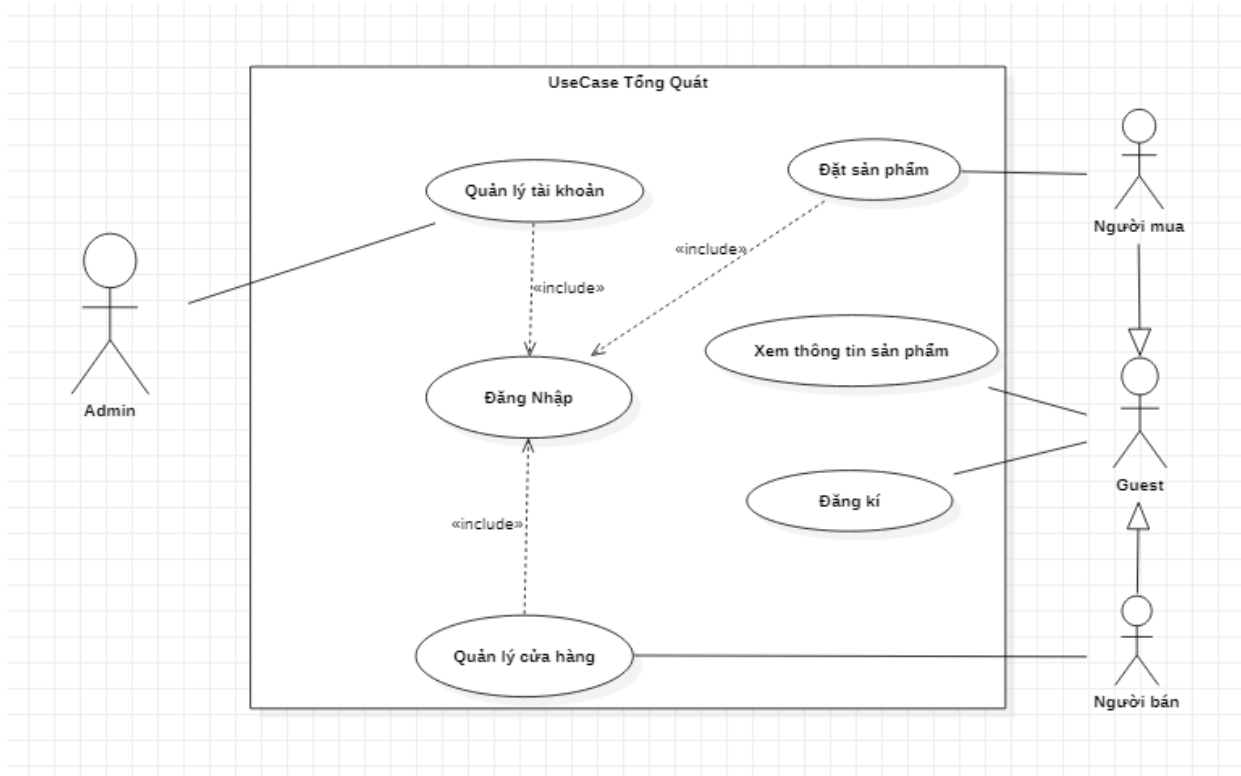
#### **3.2.3 Khách hàng:**

Có thể xem được tất cả sản phẩm của bất kì cửa hàng nào trên website. Nếu muốn đặt sản phẩm thì phải có tài khoản và đăng nhập vào hệ thống với vai trò là khách hàng. Khách hàng có thể xem được những sản phẩm đã được thêm vào giỏ hàng, khách hàng có thể cập nhật được giỏ hàng.

## CHƯƠNG IV: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

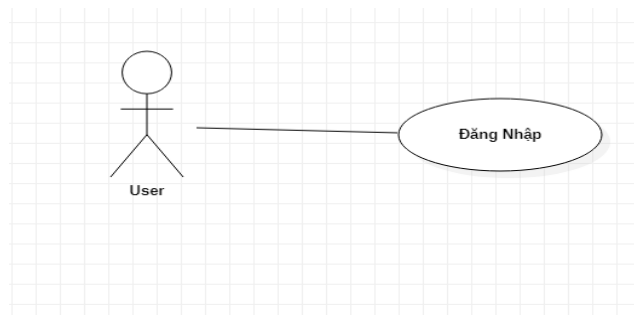
### 4.1 Sơ đồ usecase:

#### 4.1.1 Usecase tổng quát:



Hình 10 : Sơ đồ Usecase tổng quát website đặt thức ăn

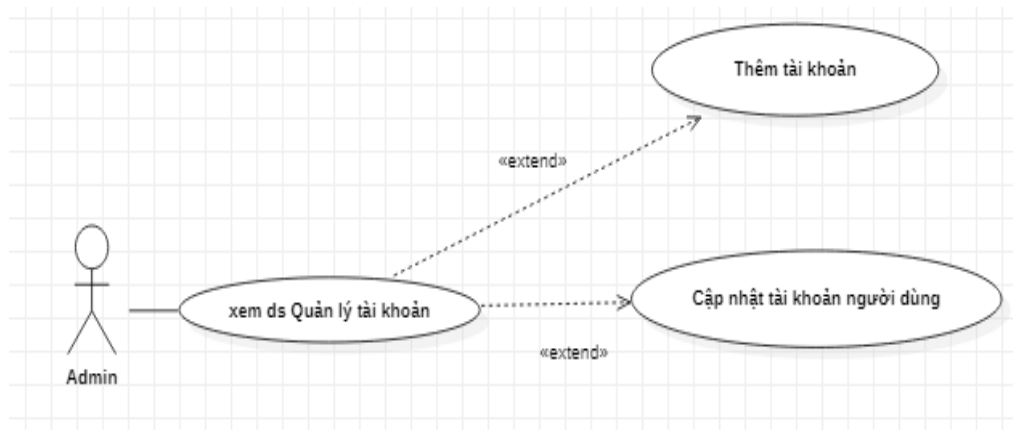
#### 4.1.2 Usecase đăng nhập:



Hình 11 Sơ đồ Usecase đăng nhập website

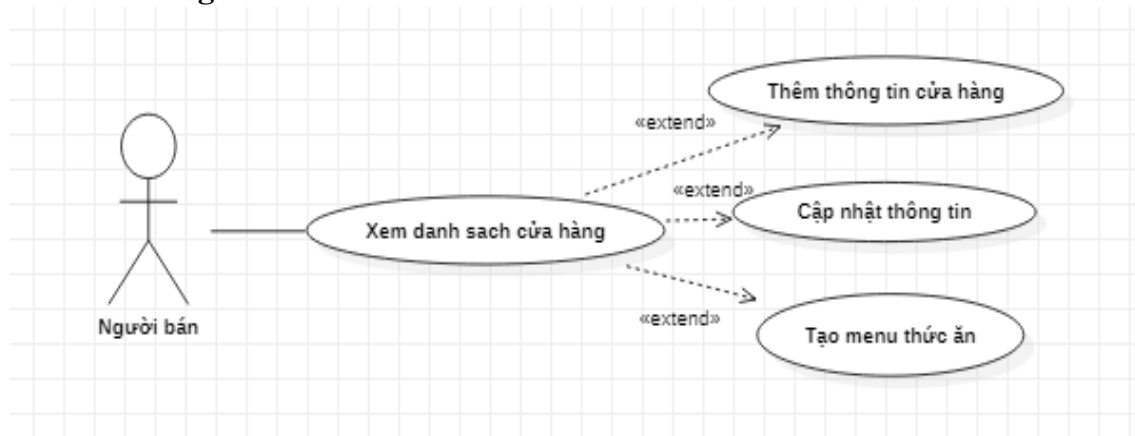


#### 4.1.3 Usecase Admin:



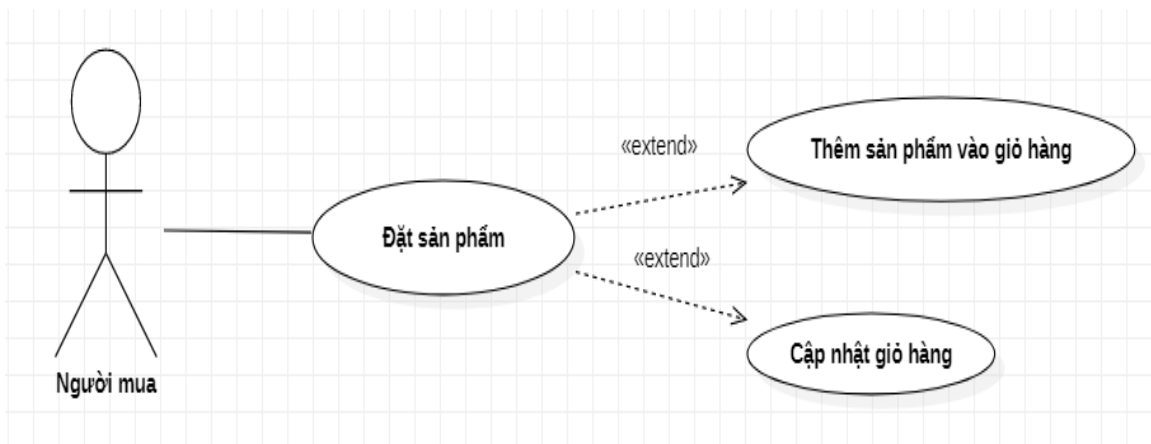
Hình 12 Sơ đồ usecase chức năng Admin

#### 4.1.4 Usecase Người Bán:



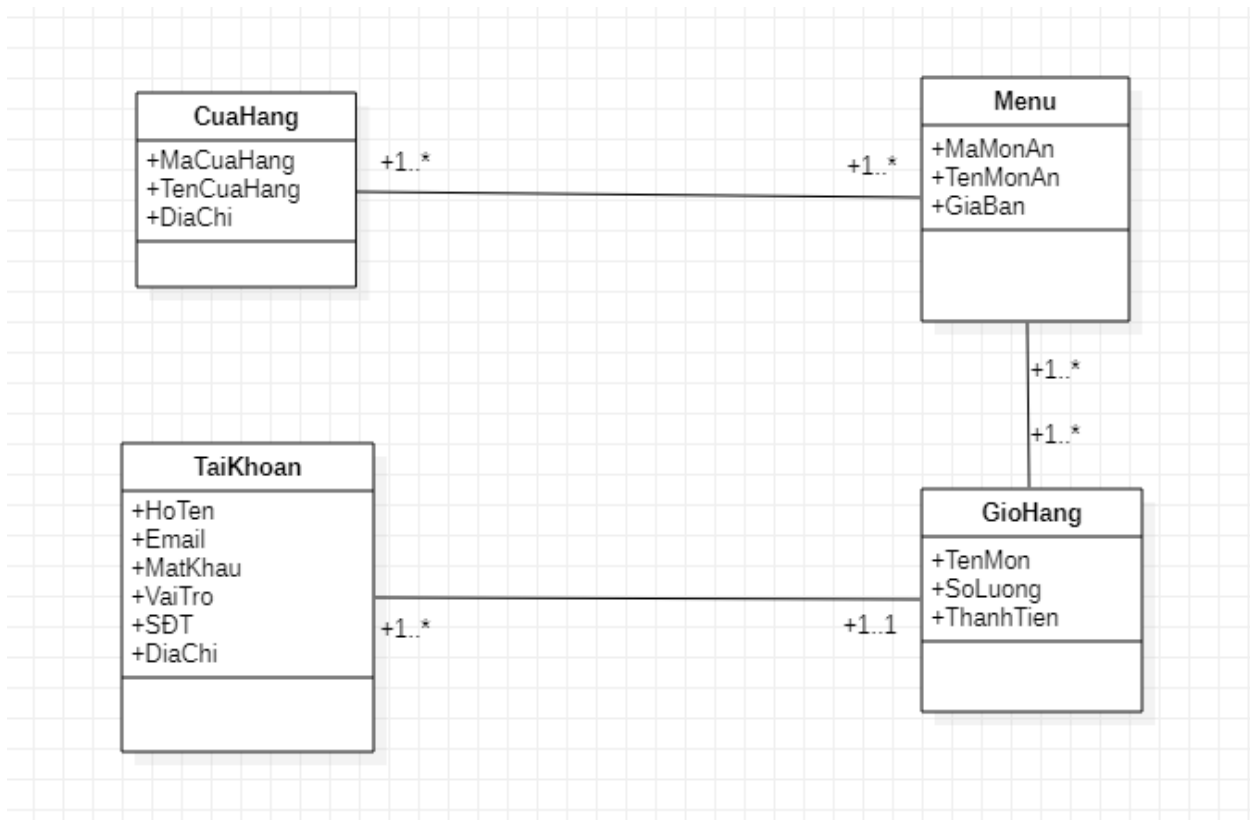
Hình 13 Sơ đồ usecase chức năng Người Bán

#### 4.1.5 Usecase Người Mua:



Hình 14 Sơ đồ usecase chức năng Người Mua


#### 4.2 Sơ đồ class:



Hình 15 Sơ đồ class

## CHƯƠNG V : HƯỚNG DẪN SỬ DỤNG PHẦN MỀM


### 5.1 Đăng ký và đăng nhập vào hệ thống




## Đăng ký

Vui lòng nhập đầy đủ thông tin dưới đây để có thể đăng ký một tài khoản

Nếu bạn là người bán hàng thì có thể bỏ qua địa chỉ





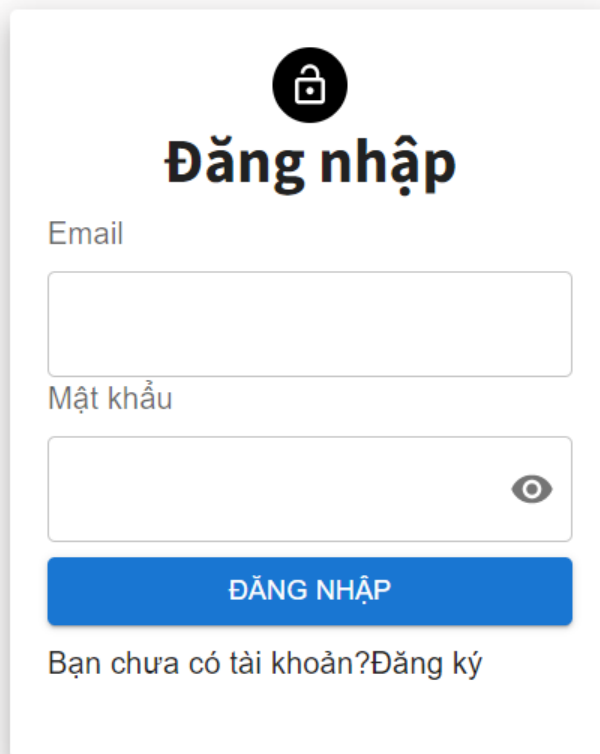
☐ Bạn đăng ký tài khoản như một người bán hàng

ĐĂNG KÝ

[Bạn đã có tài khoản? Đăng nhập](#)

Hình 16 Giao diện đăng ký

Để có thể đăng ký một tài khoản người dùng cần điền đầy đủ các thông tin được yêu cầu, đối với người dùng muốn đăng ký tài khoản như một người bán hàng thì cần lựa chọn ở phía bên dưới và có thể bỏ qua phần địa chỉ.



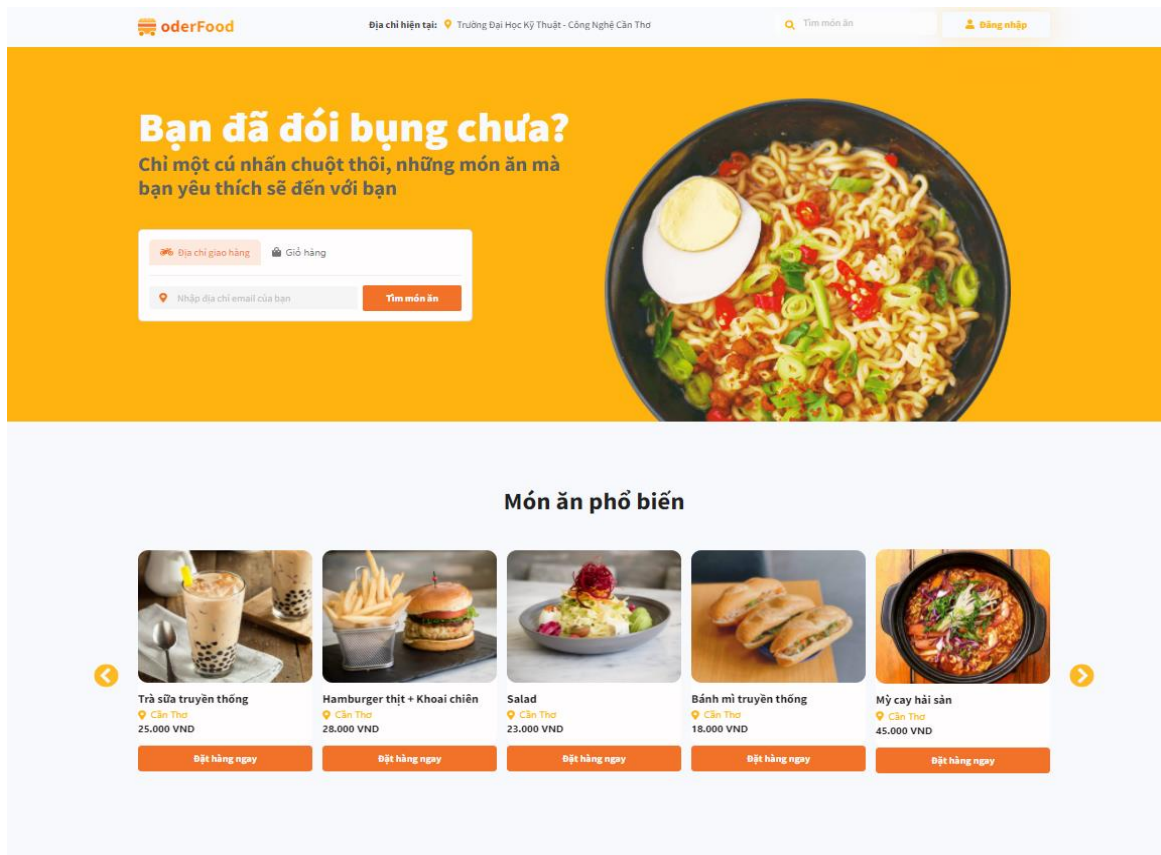
The image shows a login form with a white background and a subtle shadow. At the top center is a black circular icon containing a white padlock. Below this icon is the title "Đăng nhập" in a bold, black, sans-serif font. Under the title are two input fields: the first is labeled "Email" and the second is labeled "Mật khẩu" (Password). The password field includes a small eye icon on its right side to toggle visibility. Below the input fields is a solid blue button with the text "ĐĂNG NHẬP" in white, uppercase letters. At the bottom of the form, there is a link that reads "Bạn chưa có tài khoản? Đăng ký" (Don't have an account? Register).

*Hình 17 Giao diện đăng nhập*

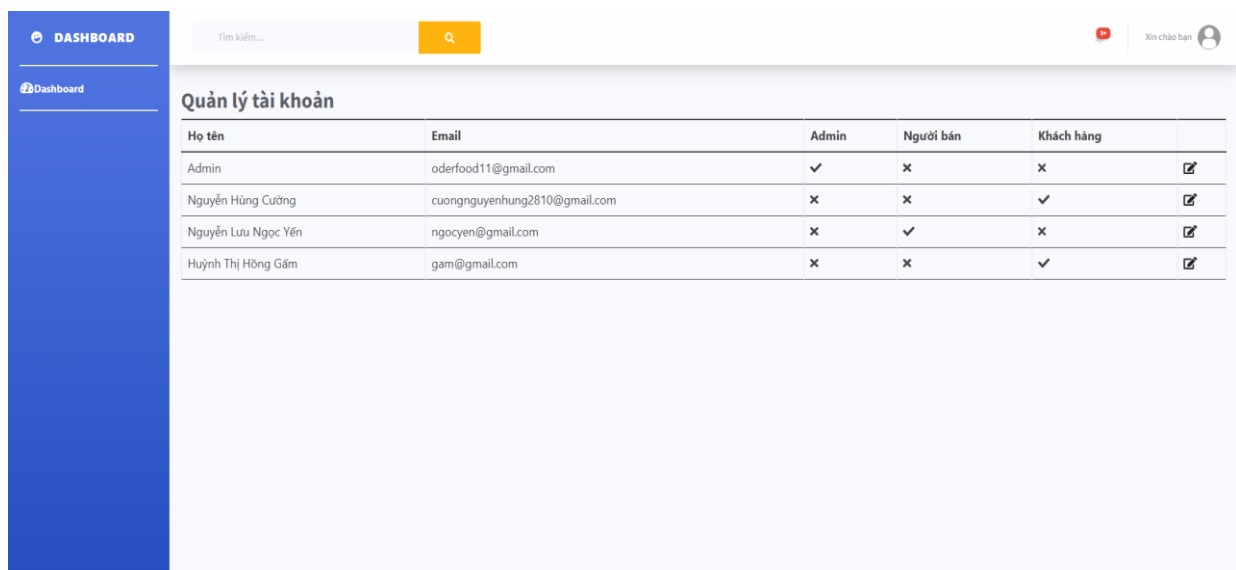
Sau khi đã đăng ký tài khoản thành công thì người dùng sẽ tiến hành đăng nhập theo tài khoản của mình. Nếu đăng nhập thành công hệ thống sẽ chuyển đến trang theo vai trò mà người dùng đã đăng ký trước đó.

Trường hợp người dùng quên mật khẩu thì sẽ liên hệ với quản trị viên để cập nhật mật khẩu lại.

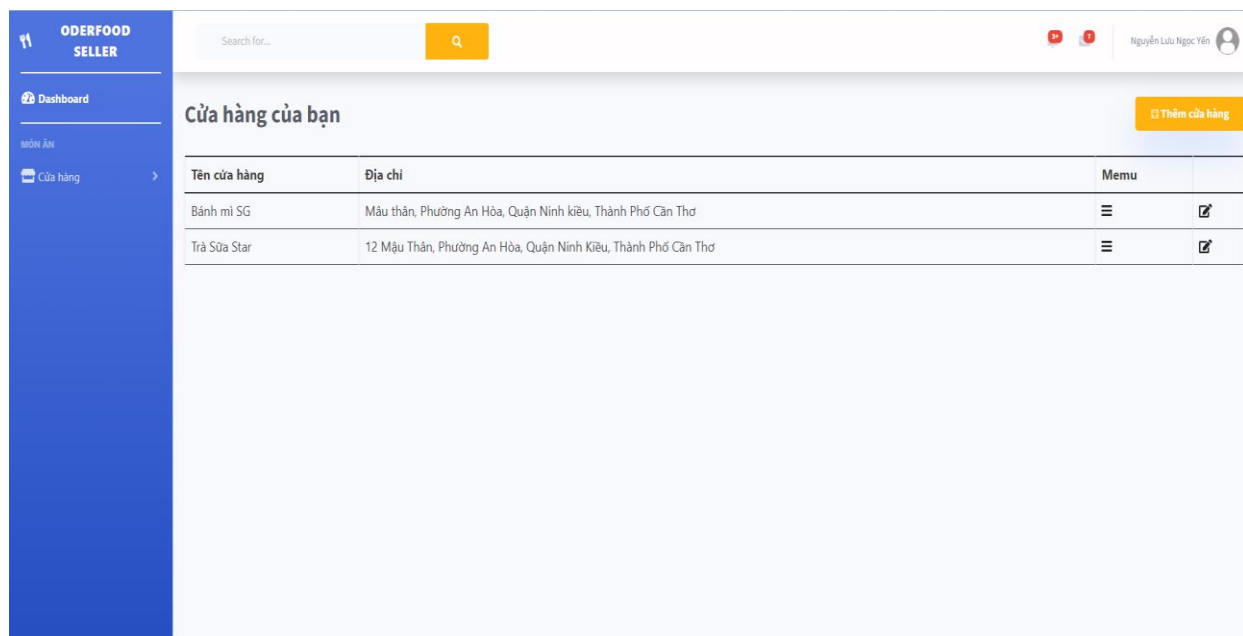
## 5.2 Giao diện của hệ thống



Hình 18 Giao diện chính của hệ thống



Hình 19 Giao diện của quản trị viên



Hình 20 Giao diện của người bán hàng

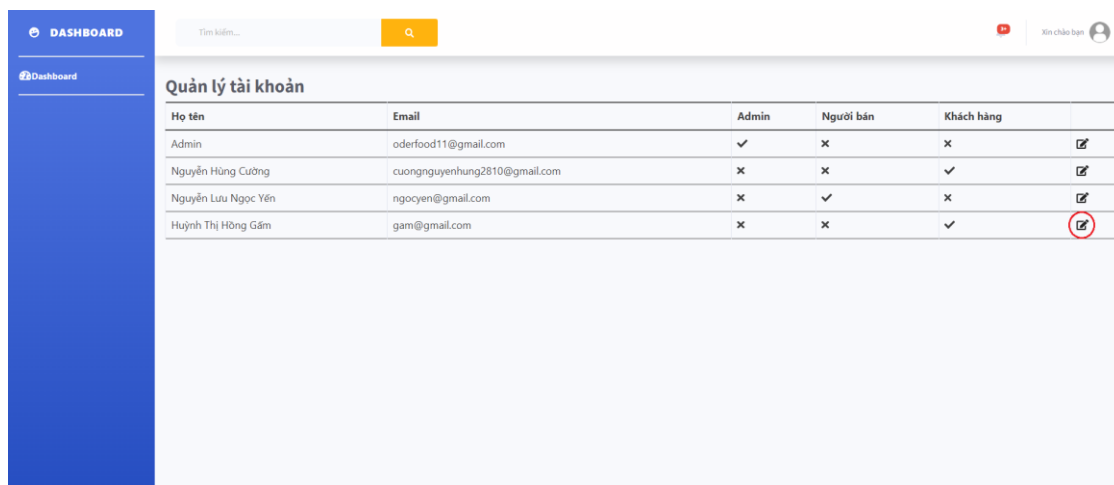
## 5.3 Chức năng nghiệp vụ theo từng vai trò

### 5.3.1 Quản trị viên (Admin)

Các chức năng của Admin bao gồm:

- Quản lý tài khoản của người dùng.
- Cập nhật lại email và vai trò của người dùng.
- Cập nhật lại mật khẩu của người dùng.

Để cập nhật thông tin tài khoản của người dùng. Quản trị viên sẽ chọn chức năng chỉnh sửa như hình dưới đây:



Hình 21 Chỉnh sửa thông tin tài khoản người dùng

**Cập nhật thông tin**

Họ tên  
Huỳnh Thị Hồng Gấm

Email  
honggam@gmail.com

Vai trò  
Người bán

**CẬP NHẬT**

**CẬP NHẬT LẠI MẬT KHẨU**

Hình 22 Cập nhật lại thông tin muốn chỉnh sửa

**Quản lý tài khoản**

Họ tên	Email	Admin	Người bán	Khách hàng	
Admin	oderfood11@gmail.com	✓	✗	✗	✎
Nguyễn Hùng Cường	cuongnguyenhung2810@gmail.com	✗	✗	✓	✎
Nguyễn Lưu Ngọc Yến	ngocyen@gmail.com	✗	✓	✗	✎
Huỳnh Thị Hồng Gấm	honggam@gmail.com	✗	✓	✗	✎

Hình 23 Chỉnh sửa thành công

Khi có người dùng liên hệ yêu cầu cập nhật lại mật khẩu. Tại giao diện cập nhật thông tin, quản trị viên sẽ chọn **CẬP NHẬT LẠI MẬT KHẨU**

Mật khẩu đã được cập nhật lại: 111111



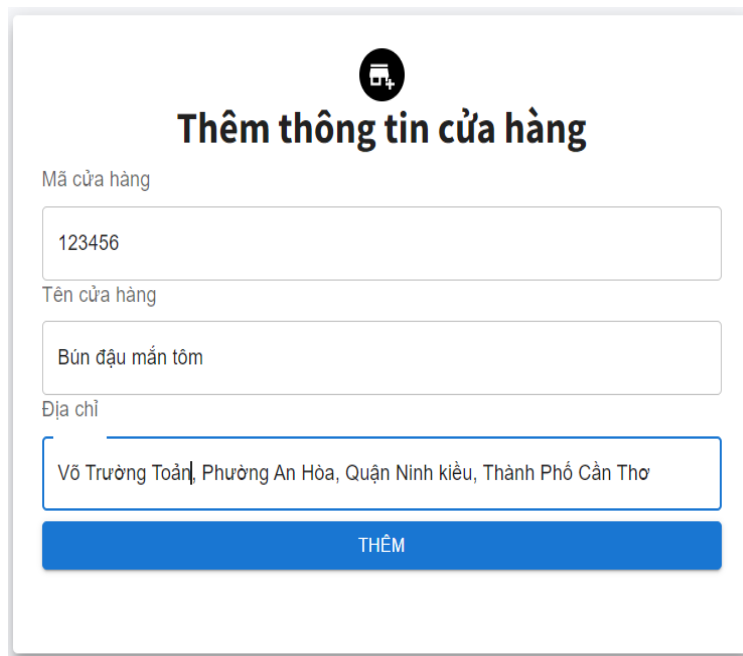
Hình 24 Hệ thống thông báo cập nhật lại thành công

### 5.3.2 Người bán hàng

Chức năng của người bán hàng bao gồm:

- Quản lý thông tin cửa hàng.
- Thêm và cập nhật thông tin cửa hàng.
- Thêm và cập nhật menu món ăn của cửa hàng.
- Cập nhật thông tin cá nhân của mình.

Để thêm thông tin cửa hàng, người dùng chọn  ngay bên góc phải màn hình và làm theo như hình dưới đây:



**Thêm thông tin cửa hàng**

Mã cửa hàng

123456

Tên cửa hàng

Bún đậu mắm tôm

Địa chỉ

Võ Trường Toàn, Phường An Hòa, Quận Ninh Kiều, Thành Phố Cần Thơ


**THÊM**

Hình 25 Nhập thông tin của cửa hàng

Cửa hàng của bạn				
Tên cửa hàng	Địa chỉ	Menu		
Bánh mì SG	Mầu thân, Phường An Hòa, Quận Ninh Kiều, Thành Phố Cần Thơ	≡		
Trà Sữa Star	12 Mậu Thân, Phường An Hòa, Quận Ninh Kiều, Thành Phố Cần Thơ	≡		
Bún đậu mắm tôm	Võ Trường Toàn, Phường An Hòa, Quận Ninh Kiều, Thành Phố Cần Thơ	≡		

Hình 26 Thêm thông tin cửa hàng thành công



Để thêm menu cho cửa hàng, người dùng sẽ chọn  trong phần menu:



**Thêm món ăn**

Mã món ăn

1

Tên món

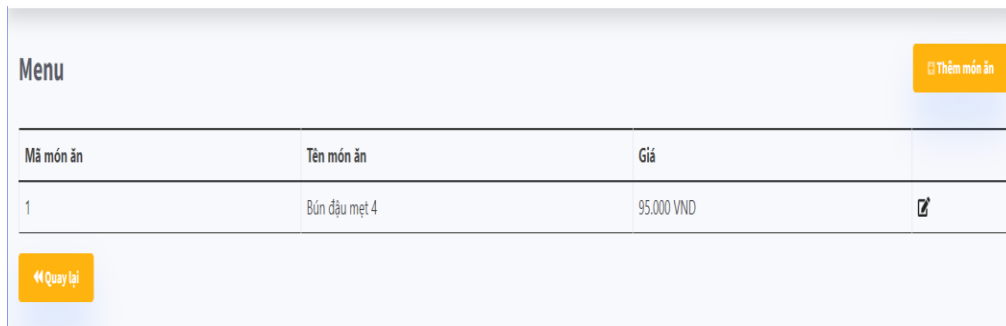
Bún đậu mèt 4

Giá bán


95000


THÊM


Hình 27 Nhập thông tin món ăn



Menu

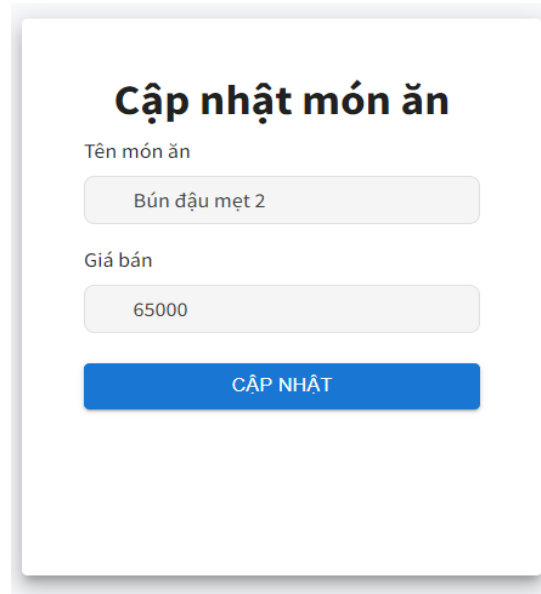


Mã món ăn	Tên món ăn	Giá	
1	Bún đậu mèt 4	95.000 VND	



Hình 28 Thêm món ăn thành công

Để cập nhật lại món ăn, người dùng sẽ chọn chức năng chỉnh sửa như hình bên dưới:



The form is titled "Cập nhật món ăn" (Update dish). It contains two input fields: "Tên món ăn" (Dish name) with the value "Bún đậu mè 2" and "Giá bán" (Selling price) with the value "65000". Below the fields is a blue button labeled "CẬP NHẬT" (Update).

Hình 29 Cập nhật lại món ăn cần chỉnh sửa



The table is titled "Menu" and has a "Thêm món ăn" (Add dish) button in the top right corner. It contains one row of data for a dish named "Bún đậu mè 2" with a price of "65.000 VND". The first column is "Mã món ăn" (Dish code) with the value "1". The last column has an edit icon. Below the table is a "Quay lại" (Go back) button.

Mã món ăn	Tên món ăn	Giá	
1	Bún đậu mè 2	65.000 VND	

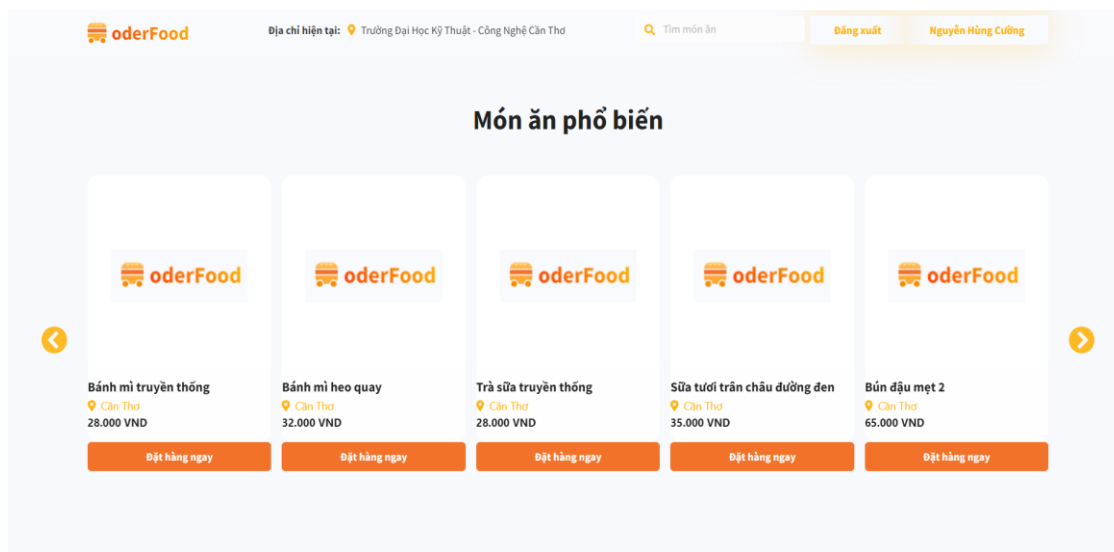
Hình 30 Cập nhật thành công

Cập nhật thông tin cửa hàng ta thao tác tương tự như cập nhật thông tin món ăn

### 5.3.3 Khách hàng

Chức năng của khách hàng bao gồm:

- Thêm tất cả món ăn của tất cả cửa hàng vào giỏ hàng
- Cập nhật lại giỏ hàng.



Hình 31 Giao diện các món ăn của cửa hàng

Để thêm món ăn vào giỏ hàng, người dùng cần làm như hình dưới đây:



Hình 32 Chọn món ăn muốn thêm vào giỏ hàng



Hình 33 Thêm vào giỏ hàng thành công

Người dùng vẫn có thể cập nhật lại số lượng món ăn hoặc xóa ra khỏi giỏ hàng nếu như đổi ý.

## KẾT LUẬN

### **Kết quả đạt được:**

Trong thời gian tìm hiểu và nghiên cứu, chúng em đã phân nào hiểu được khái quát cách hoạt động của MongoDB, Express, ReactJS, NodeJS khi kết hợp với nhau, điểm mạnh và điểm yếu của từng công nghệ.

Xây dựng thành công một ứng dụng minh họa “Website đặt thức ăn” với các chức năng cơ bản như: thêm, sửa thông tin cửa hàng, thêm menu của cửa hàng và thêm món ăn vào giỏ hàng.

**Hạn chế:** Giao diện chưa thân thiện với người dùng, còn một số chức năng chưa được hoàn chỉnh như giỏ hàng và thêm menu và còn mắc một số lỗi logic như bên đăng ký tài khoản và không bắt trùng dữ liệu được bên phần thêm menu.

**Hướng phát triển:** Phát triển thêm về giao diện, thêm những chức năng đặt, giao hàng, thanh toán và xác nhận mail.

### **Source Code ứng dụng minh họa:**

GitHub: <https://github.com/HuynhGam/DA3>

## TÀI LIỆU THAM KHẢO

1. Vasani Subramanian (2019). *Pro MERN Stack Full Stack Web App Development with Mongo, Express, React, and Node*, tái bản lần 2. Apress.
2. Ethan Brown (2014). *Web Development with Node and Express: Leveraging the JavaScript Stack*. O'Reilly Media.
3. Colin J Ihrig (2015). *Full Stack JavaScript Development With MEAN: MongoDB, Express, AngularJS, and Node.JS*. Sitepoint.