**HÀ NỘI 5/2025**

****

**BÁO CÁO**

**THỰC TẬP DOANH NGHIỆP**

CHUYÊN NGÀNH CÔNG NGHỆ PHẦN MỀM

**PHÂN TÍCH THIẾT KẾ HỆ THỐNG PHẦN MỀM**

**GIAO ĐỒ ĂN NHANH**

Sinh viên thực tập: Hồ Viết Trà

Mã sinh viên : 2121051112

Công ty thực tập : Công ty TNHH công nghệ và

thương mại Wintech

Người phụ trách : Dương Huy

Giảng viên hướng dẫn: Nguyễn Thế Lộc

# **MỤC LỤC**

[MỤC LỤC 2](#_Toc199068742)

[DANH MỤC HÌNH VẼ 3](#_Toc199068743)

[LỜI CẢM ƠN 4](#_Toc199068744)

[MỞ ĐẦU 5](#_Toc199068745)

[CHƯƠNG 1 8](#_Toc199068746)

[GIỚI THIỆU VỀ CÔNG TY VÀ VỊ TRÍ THỰC TẬP 8](#_Toc199068747)

[1.Giới thiệu chung 8](#_Toc199068748)

[1.1. Giới thiệu công ty 8](#_Toc199068749)

[1.3. Giá trị cốt lõi 9](#_Toc199068750)

[1.4. Mục tiêu của công ty Wintech 9](#_Toc199068751)

[2. Sản phẩm và dịch vụ 9](#_Toc199068752)

[3. Công nghệ và chất lượng 10](#_Toc199068753)

[4. Văn hóa doanh nghiệp 10](#_Toc199068754)

[4.1 Giới thiệu chung văn hóa danh nghiệp 10](#_Toc199068755)

[4.2 Quy trình phát triển phần mềm của công ty 10](#_Toc199068756)

[5. Vị trí thực tập: 11](#_Toc199068757)

[6. Đề tài thực tập: 12](#_Toc199068758)

[7. Lịch làm việc: 13](#_Toc199068759)

[CHƯƠNG 2 16](#_Toc199068760)

[KHẢO SÁT VÀ MÔ TẢ BÀI TOÁN 16](#_Toc199068761)

[1. Khảo sát hệ thống 16](#_Toc199068762)

[CHƯƠNG 3 18](#_Toc199068763)

[PHÂN TÍCH THIẾT KẾ HỆ THỐNG 18](#_Toc199068764)

[1. Sơ đồ phân rã chức năng 18](#_Toc199068765)

[2. Sơ đồ ngữ cảnh 20](#_Toc199068766)

[3. Sơ đồ luồng dữ liệu mức đỉnh 20](#_Toc199068767)

[4. Các sơ đồ luồng dữ liệu mức dưới đỉnh 21](#_Toc199068768)

[4.1. Sơ đồ mức dưới đỉnh quản lý hệ thống 21](#_Toc199068769)

[4.2. Sơ đồ mức dưới đỉnh quản lý thực đơn 21](#_Toc199068770)

[4.3. Sơ đồ mức dưới đỉnh quản lý đơn hàng 22](#_Toc199068771)

[4.4. Sơ đồ mức dưới đỉnh quản lý giao hàng 23](#_Toc199068772)

[4.5. Sơ đồ mức dưới đỉnh tìm kiếm 23](#_Toc199068773)

[4.6. Sơ đồ mức dưới đỉnh báo cáo thống kê 24](#_Toc199068774)

[CHƯƠNG 4 25](#_Toc199068775)

[NỘI DUNG THỰC TẬP 25](#_Toc199068776)

[1. Giới thiệu về các công cụ đã tìm hiểu và sử dụng trong dự án 25](#_Toc199068777)

[1.1. ReactJs 25](#_Toc199068778)

[1.2. NodeJs 30](#_Toc199068779)

[1.3. MongoDB 34](#_Toc199068780)

[2. Tìm hiểu về API: 36](#_Toc199068781)

[2.1. Giới thiệu về API: 36](#_Toc199068782)

[2.2. Cách thức hoạt động củaAPI 36](#_Toc199068783)

[2.3 Ưu điểm và nhược điểm của API: 38](#_Toc199068784)

[2.4 Một số các ngăn chặn rò rỉ thông tin khi dùng API: 38](#_Toc199068785)

[3. Giao diện trong website giao đồ ăn trực tuyến: 39](#_Toc199068786)

[CHƯƠNG 5: 46](#_Toc199068787)

[ĐÁNH GIÁ KẾT QUẢ VÀ ĐỊNH HƯỚNG PHÁT TRIỂN 46](#_Toc199068788)

[1. Đánh giá kết quả 46](#_Toc199068789)

[2. Định hướng phát triển đề tài 48](#_Toc199068790)

[KẾT LUẬN 48](#_Toc199068791)

[TÀI LIỆU THAM KHẢO 49](#_Toc199068792)

# **DANH MỤC HÌNH VẼ**

[Hình 1-1 Sơ đồ tổ chức công ty TNHH Công Nghệ và thương mại Wintech 9](#_Toc199068205)

[Hình 3- 1 Sơ đồ phân rã chức năng 19](#_Toc199068206)

[Hình 3- 2 Sơ đồ mức ngữ cảnh 21](#_Toc199068207)

[Hình 3- 3 Sơ đồ luồng dữ liệu mức đỉnh 21](#_Toc199068208)

[Hình 3- 4 Sơ đồ mức dưới đỉnh quản lý hệ thống 22](#_Toc199068209)

[Hình 3- 5 Sơ đồ mức dưới đỉnh quản lý thực đơn 23](#_Toc199068210)

[Hình 3- 6 Sơ đồ mức dưới đỉnh quản lý đơn hàng 23](#_Toc199068211)

[Hình 3- 7 Sơ đồ mức dưới đỉnh quản lý giao hàng 24](#_Toc199068212)

[Hình 3- 8 Sơ đồ mức dưới đỉnh tìm kiếm 24](#_Toc199068213)

[Hình 3- 9 Sơ đồ mức dưới đỉnh báo cáo thống kê 25](#_Toc199068214)

[Hình 4- 1 Logo thư viện ReactJs 26](#_Toc199068215)

[Hình 4 – 2: Virtual Dom 27](#_Toc199068216)

[Hình 4 – 3: Logo môi trường NodeJs 31](#_Toc199068217)

[Hình 4 – 4: Cách hoạt động NodeJs 33](#_Toc199068218)

[Hình 4 – 5: Hình ảnh tượng trưng MongooDb 36](#_Toc199068219)

[Hình 4- 6: Cách thức hoạt động của API 38](#_Toc199068220)

[Hình 4- 7 Giao diện “Trang chủ” của website 40](#_Toc199068221)

[Hình 4- 8 Giao diện danh mục món 41](#_Toc199068222)

[Hình 4- 9 Giao diện món ăn ở gần 41](#_Toc199068223)

[Hình 4- 10 Giao diện các app tải trên mobile 42](#_Toc199068224)

[Hình 4- 11 Giao diện chân trang 42](#_Toc199068225)

[Hình 4- 12 Giao diện giỏ hàng 43](#_Toc199068226)

[Hình 4- 13 Giao diện thông tin giao hàng 43](#_Toc199068227)

[Hình 4- 14 Giao diện thanh toán vnpay 44](#_Toc199068228)

[Hình 4- 15 Giao diện các món đã mua 44](#_Toc199068229)

[Hình 4- 16 Giao diện đăng nhập 45](#_Toc199068230)

[Hình 4- 17 Giao diện đăng kí 45](#_Toc199068231)

[Hình 4-18: Giao diện thêm sản phẩm 46](#_Toc199068232)

[Hình 4-19: Giao diện danh sách món ăn 46](#_Toc199068233)

[Hình 4-20: Giao diện quản lý gọi món 47](#_Toc199068234)

# **LỜI CẢM ƠN**

Thời gian thực tập là cơ hội cho em tổng hợp và hệ thống hóa lại các kiến thức đã học, đồng thời kết hợp với thực tế để nâng cao kiến thức chuyên môn. Tuy chỉ có 10 tuần thực tập, nhưng trong quá trình học tập, em đã được mở rộng tầm nhìn và tiếp thu rất nhiều kiến thức thực tế. Từ đó em nhận thấy, việc cọ sát thực tế vô cùng quan trọng, nó giúp sinh viên xây dựng nền tảng lý thuyết được học ở trường vững chắc hơn.

Để hoàn thành kì thực tập này trước tiên em xin gửi lời cảm ơn chân thành đến Ban giám hiệu trường Đại học Mỏ - Địa Chất đã tạo điều kiện cho em được thực tập, cảm ơn các quý thầy cô khoa Công nghệ thông tin vì đã truyền đạt cho em những kiến thức bổ ích và tạo điều kiện để em có thể áp dụng kiến thức đã học vào thực tế.

Đặc biệt, em muốn gửi lời cảm ơn sâu sắc đến thầy Nguyễn Thế Lộc, trường Đại học Mỏ - Địa Chất đã luôn quan tâm, hướng dẫn tận tình và theo dõi sát sao trong suốt quá trình thực tập.

Em cũng muốn cảm ơn đơn vị Wintech và các anh chị tại công ty đã luôn nhiệt tình giúp đỡ và chỉ dạy em các kỹ năng cần thiết để em có được những kinh nghiệm quý báu trong quá trình thực tập tại công ty.

Cuối cùng, em xin gửi lời cảm ơn đến gia đình và bạn bè của mình đã luôn động viên, ủng hộ em trong quá trình thực tập. Nhờ có sự ủng hộ đó, em đã vượt qua được những thử thách và hoàn thành tốt nhiệm vụ được giao.

Trong quá trình thực tập, còn bỡ ngỡ và thiếu kinh nghiệm, em đã gặp rất nhiều khó khăn nhưng với sự giúp đỡ của quý thầy cô, bạn bè, gia đình và đồng nghiệp đã giúp em có những trải nghiệm quý giá .Một lần nữa, em xin chân thành cảm ơn tất cả mọi người đã đồng hành cùng em trong quá trình thực tập này.

# **MỞ ĐẦU**

Ngày nay, do sự tiến triển không ngừng của xã hội, lĩnh vực công nghệ thông tin trở nên vô cùng phổ biến và đã đạt được sự phát triển đáng kể trong nhiều lĩnh vực của cuộc sống. Tin học ngày càng trở thành một yếu tố không thể thiếu trong cuộc sống hàng ngày, đặc biệt trong việc hỗ trợ con người tối ưu hóa các hoạt động tiêu dùng và dịch vụ. Tại các quốc gia phát triển, việc ứng dụng công nghệ thông tin trong các lĩnh vực như kinh tế, thương mại, dịch vụ ăn uống đã mang lại hiệu quả rõ rệt.

Với nhu cầu đặt món ăn nhanh ngày càng cao và sự gia tăng của các dịch vụ giao đồ ăn, việc quản lý đơn hàng, thực đơn và thông tin khách hàng trở nên phức tạp và đòi hỏi hệ thống xử lý nhanh chóng, chính xác. Trước đây, các cửa hàng thường tiếp nhận đơn hàng qua điện thoại hoặc trực tiếp, điều này gây mất nhiều thời gian, dễ dẫn đến sai sót và khó khăn trong việc theo dõi số lượng đơn cũng như nhu cầu của khách hàng.

Do đó, ban quản lý đã quyết định triển khai website đặt đồ ăn nhanh nhằm tin học hóa toàn bộ quy trình từ đặt hàng, xử lý đơn đến giao hàng. Hệ thống mới giúp tăng hiệu suất làm việc, đảm bảo tính đồng bộ và chính xác trong việc xử lý dữ liệu khách hàng và đơn hàng. Với các tính năng như cập nhật thực đơn theo thời gian thực, thống kê doanh thu và hỗ trợ giao hàng nhanh chóng, website không chỉ giúp tiết kiệm chi phí vận hành mà còn góp phần nâng cao trải nghiệm khách hàng và tăng trưởng doanh thu cho doanh nghiệp.

Là sinh viên của trường chúng em luôn luôn mong muốn được học hỏi và áp dụng những kiến thức mà chúng em đã được các thầy cô dạy trong thời gian qua. Báo cáo này giúp chúng em tổng kết lại những điều đã học được và phát triển khả năng tư duy, tính sang tạo trong công việc để sau này chúng em có thể vững bước trên con đường mà chúng em đã chọn.

Xin chân thành cảm ơn các thầy cô đã nhiệt tình dạy chúng em trong suốt những năm học qua và để chúng em có đủ các kiến thức làm được một dự án hoàn chỉnh và là bước khởi đầu cho những dự án trong tương lai. Do thời gian và trình độ chuyên môn còn hạn chế, nên báo cáo của em không tránh khỏi những thiếu sót, em rất mong nhận được nhưng nhận xét, góp ý của quý thầy, cô. Em xin chân thành cảm ơn.

Báo cáo gồm 5 phần:

Phần 1: Giới thiệu chung về Công ty TNHH công nghệ và thương mại Wintech

Phần 2: Khảo sát và mô tả bài toán

Phần 3: Phân tích thiết kế hệ thống

Phần 4: Nội dung thực tập

Phần 5: Đánh giá kết quả và định hướng phát triển

# **CHƯƠNG 1**

# **GIỚI THIỆU VỀ CÔNG TY VÀ VỊ TRÍ THỰC TẬP**

## **1.Giới thiệu chung**

### **1.1. Giới thiệu công ty**

Công ty TNHH Công Nghệ và Thương Mại Wintech được thành lập ngày 26/07/2023 - là doanh nghiệp hoạt động trong lĩnh vực Công Nghệ Thông Tin. Với các dự án phát triển phần mềm thương mại điện tử, phần mềm quản lý doanh nghiệp.. cho công ty và các công ty đối tác.

Equation 1

A diagram of a company

Description automatically generated with medium confidence

Hình 1-1 Sơ đồ tổ chức công ty TNHH Công Nghệ và thương mại Wintech

Góp phần vào sự thành công của doanh nghiệp dựa trên những sản phẩm, dịch vụ và giải pháp công nghệ thông tin được cung cấp bởi Wintech.

Wintech sẽ không ngừng cập nhật và cải tiến công nghệ để cung cấp những sản phẩm hiệu quả và phù hợp nhất theo đúng yêu cầu của khách hàng.

Phục vụ bằng chữ tâm và trách nhiệm trên những sản phẩm, dịch vụ đem đến khách hàng.

### **1.3. Giá trị cốt lõi**

Góp phần vào sự thành công của doanh nghiệp dựa trên những sản phẩm, dịch vụ và giải pháp công nghệ thông tin được cung cấp bởi Wintech.

Wintech sẽ không ngừng cập nhật và cải tiến công nghệ để cung cấp những sản phẩm hiệu quả và phù hợp nhất theo đúng yêu cầu của khách hàng.

Phục vụ bằng chữ tâm và trách nhiệm trên những sản phẩm, dịch vụ đem đến khách hàng.

### **1.4. Mục tiêu của công ty Wintech**

Tìm kiếm và phát triển các sản phẩm từ ý tưởng của các thành viên hoặc từ ý tưởng bên ngoài.

Tìm kiếm đối tác, nhà đầu tư để triển khai các sản phẩm ra thị trường trong và ngoài nước.

Phục vụ như một trung tâm R & D phần mềm từ xa nhưng được tích hợp đầy đủ cho các đối tác.

Phát triển sản phẩm mới theo ý tưởng của đối tác với các kinh nghiệm của các chuyên gia trong nhiều mảng lĩnh vực.

## **2. Sản phẩm và dịch vụ**

Wintech từ lúc ra đời đã đạt được những bước phát triển vượt bậc. Tốc độ phát triển năm sau luôn tăng cao hơn các năm trước và số lượng nhân viên tăng lên liên tục hàng năm. Với chiến lược hoạt động: “đưa tri thức toàn cầu vào phục vụ Việt Nam, và đưa nguồn lực của Việt Nam cung cấp cho thị trường thế giới” – Wintech mong muốn trở thành một công ty về chất lượng dịch vụ và hiệu quả trong các hoạt động trên nền tảng tri thức công nghệ thông tin.

## **3. Công nghệ và chất lượng**

Cam kết mang lại giá trị cao nhất cho khách hàng, Wintech luôn chú trọng đầu tư ngân sách dành cho nghiên cứu và phát triển các công nghệ mới nhất. Trong những năm gần đây, công nghệ Singapore chuyên sản xuất và triển khai các phần mềm và ứng dựng trên nền tảng website và điện thoại di động được coi là trọng tâm và nền tảng để Wintech triển khai, cung cấp các dịch vụ/giải pháp thông minh tới khách hàng.

## **4. Văn hóa doanh nghiệp**

### **4.1 Giới thiệu chung văn hóa danh nghiệp**

Giới thiệu về Wintech không thể bỏ qua mảng cực kỳ quan trọng đó là “Văn hóa doanh nghiệp”, bởi văn hóa chính là yếu tố sống còn, đặt nền tảng cho sự phát triển bền vững của Wintech trong quá khứ, thời điểm hiện tại và tương lai.

Không màu mè hoa lá, không từ ngữ xa hoa, văn hóa doanh nghiệp của Wintech. Là nhân viên của công ty phải đề cao và được đề cao sự tôn trọng cá nhân, tinh thần đổi mới và tinh thần đồng đội; là lãnh đạo phải chí công, gương mẫu, sáng suốt.

Mỗi cán bộ nhân viên Wintech đều có niềm tin sắt đá vào sự nghiệp chính nghĩa, luôn nỗ lực lao động, sáng tạo, góp phần phát triển cá nhân, tập thể, đất nước. “Văn hóa Wintech” có màu sắc riêng, không dễ nhầm lẫn với bất cứ môi trường nào, mỗi cá nhân đều có sự gắn kết chặt chẽ với nhau, tạo thành hệ thống giá trị Wintech – nền tảng vững chắc cho sự phát triển của doanh nghiệp.

### **4.2 Quy trình phát triển phần mềm của công ty**

Có rất nhiều mô hình phát triển phần mềm phổ biến như: Mô hình thác nước (Waterfall), V-Model, Xoắn ốc, mô hình tăng trưởng, mô hình phát triển lặp, mô hình Agile-Scrum... Mô hình Agile-Scrum - một mô hình phát triển phần mềm đang được sử dụng trong quy trình phát triển phần mềm tại công ty Wintech.

Quy trình vận hành:

- Product backlog là danh sách ưu tiên các tính năng hoặc đầu ra khác của dự án, có thể hiểu như là danh sách yêu cầu của dự án.

- Product Owner chịu trách nhiệm sắp xếp độ ưu tiên cho từng hạng mục trong product backlog dựa trên các giá trị do Product Owner định nghĩa.

- Quy trình scrum gồm có 4 cuộc họp:

+ Họp kế hoạch sprint: Thời điểm khi bắt đầu một sprint với mục đích để lên kế hoạch làm việc cho 1 sprint. Công việc lập kế hoạch bao gômg lựa chọn các yêu cầu cần phải phát triển, phân tích và nhận biết các công việc phải làm kèm theo ước lượng thời gian từng phần và tăng theo thời gian, có thích nghi với các tình hình thực tiễn trong tiến trình đi đến sản phẩm.

+ Cuộc họp hàng ngày: Scrum master tổ chức cho đội sản xuất họp hàng ngày trong khoảng 15 phút để nhóm phá triển chia sẻ tiến độ công việc, trong cuộc họp này từng người trong nhóm phát triển báo cáo lần lượt để trình bày 3 câu hỏi: Hôm qua làm gì? Hôm nay sẽ làm gì? Có khó khăn gì không?

+ Họp sơ kết sprint: Sau cuối mỗi sprint nhóm phát triển cũng với product owener sẽ rà soát lại các công việc đã hoàn tất trong sprint vừa qua và đề xuất các chỉnh sửa hoăc thay đổi cần thiết cho sản phẩm.

+ Họp cải tiến sprint: Scrum master hỗ trợ cũng nhóm phát triển sẽ rà soát lại toàn lại toàn diện sprint vừa kết thúc và tìm cách cải tiến quy trình làm việc cũng như bản thân sản phẩm.

## **5. Vị trí thực tập:**

Vị trí thực tập: Thực tập sinh IT

Nội dung công việc:

• Ôn tập kiến thức HTML, CSS, JS, SQLServer.

• Tìm hiểu công nghệ Reactjs, NodeJS, MongooseDb, ….

• Xác định các chức năng chính của website đặt đồ ăn trực tuyến

• Tối ưu và hiệu chỉnh.

• Phân tích cơ sở dữ liệu của dự án

• Setup dự án, Thiết kế giao diện

• Xây dựng backend, kết nối cơ sở dữ liệu

• Thực hiện một vài chức năng cơ bản như đăng nhập, đăng xuất, thêm món, tìm kiếm, ….

• Hoàn thiện đầy đủ tất cả chức năng khác

• Kiểm thử và chỉnh sửa lỗi

• Nộp sản phẩm hoàn chỉnh

• Đánh giá và tổng kết.

## **6. Đề tài thực tập:**

Tên đề tài: Xây dựng website đặt đồ ăn trực tuyến

Mục tiêu của đề tài:

Xây dựng một website có khả năng lưu trữ và quản lý đầy đủ, khoa học các thông tin về món ăn, thực đơn, cửa hàng và đơn đặt hàng.

Website cung cấp thông tin cập nhật về tình hình đặt món và giao hàng tại mọi thời điểm, hỗ trợ hiệu quả cho công tác quản lý hoạt động kinh doanh ẩm thực.

Cung cấp chức năng tra cứu nhanh, tiện lợi và chính xác các món ăn, trạng thái đơn hàng và thông tin khách hàng.

Hỗ trợ tổng hợp, thống kê và xuất báo cáo theo tuần, tháng, quý, năm về số lượng đơn hàng, doanh thu và các chỉ số vận hành khác theo yêu cầu.

## **7. Lịch làm việc:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tuần** | **Công việc** | **Người hướng dẫn** | **Mức độ hoàn thành (SV tự đánh giá)** | **Ghi chú** |
| 1 | • Tìm hiểu văn hoá làm việc, nội quy của công ty  • Tìm hiểu công nghệ, sản phẩm tại công ty, các bước phát triển sản phẩm tại công ty  • Xác định định hướng, ôn tập kiến thức HTML, CSS, JS  • Học thêm các kiến thức về ReactJS và NodeJS:   * Node.js cơ bản: Hiểu về Node.js, cách Node.js hoạt động (Event Loop, V8 Engine), CommonJS vs ES6 Modules * NPM: Quản lý package với NPM (Node Package Manager), tạo và quản lý dự án với package.json. * ES6+ Features: Arrow functions, destructuring, spread/rest operator, template literals, classes, modules, let và const, Map và Set, this và cách xử lý, closures, modules (import/export) * React cơ bản: Hiểu khái niệm về component-based architecture, JSX, Virtual DOM | Dương Huy | Hoàn thành |  |
| 2 | • Học thêm các kiến thức về ReactJS và NodeJS:   * Express.js: Thiết lập Express, tạo API endpoints, middleware, xử lý lỗi. * CRUD Operations: Xây dựng ứng dụng CRUD (Create, Read, Update, Delete). * Routing & Middleware: Hiểu về routing, middlewares cơ bản (cài đặt logger, xử lý lỗi...). * Postman: Sử dụng Postman để kiểm thử API. * Làm việc với Cơ sở Dữ liệu (SQL or NoSQL) * Xác thực/phân quyền (bổ sung) * Functional Components và Class Components: Sự khác biệt, cách sử dụng và khi nào nên sử dụng. * Props và State: Hiểu về Props (Properties) và State, cách chúng quản lý dữ liệu và render lại component. * State Management trong React: useState, useEffect, useReducer. * Lifecycle Methods (Class Components): componentDidMount, componentDidUpdate, componentWillUnmount. * Hooks cơ bản: Hiểu về các hooks phổ biến như useState, useEffect, useContext, useMemo, và useRef. * Routing trong React * Kết hợp React và CSDL Firebase | Dương Huy | Hoàn thành |  |
| 3 | • Xác định các chức năng chính của website đặt đồ ăn trực tuyến  • Phân tích cơ sở dữ liệu của website | Dương Huy | Hoàn thành |  |
| 4 | • Setup dự án, Tạo cấu trúc thư mục  • Thiết kế giao diện  • Khởi tạo project và xây dựng layout UI. | Dương Huy | Hoàn thành |  |
| 5 | • Xây dựng backend  • Kết nối cơ sở dữ liệu | Dương Huy | Hoàn thành |  |
| 6 | • Thực hiện tích hợp fondend – backend  • Thực hiện một vài chức năng cơ bản như tìm kiếm, lọc theo danh mục | Dương Huy | Hoàn thành |  |
| 7 | • Xây dựng chức năng giỏ hàng và đặt món  • Xem lịch sử đơn hàng, hiển thị món đã chọn, giá tiền | Dương Huy | Hoàn thành |  |
| 8 | • Hoàn thiện các chức năng khác  • Kiểm tra các lỗi | Dương Huy | Hoàn thành |  |
| 9 | • Kiểm thử, chỉnh sửa lỗi  • Thực hiện làm báo cáo | Dương Huy | Hoàn thành |  |
| 10 | • Nộp sản phẩm hoàn chỉnh  • Trình bày, báo cáo kết quả và tổng kết thực tập | Dương Huy | Hoàn thành |  |

Bảng 1-1. Lịch làm việc Table

# **CHƯƠNG 2**

# **KHẢO SÁT VÀ MÔ TẢ BÀI TOÁN**

## **1. Khảo sát hệ thống**

Cửa hàng FastFood HVT là một doanh nghiệp vừa và nhỏ chuyên cung cấp đồ ăn nhanh và thức uống mang đi. Trong thời đại công nghệ số, thói quen tiêu dùng của khách hàng đã thay đổi đáng kể: họ ưu tiên đặt hàng trực tuyến để tiết kiệm thời gian thay vì đến mua trực tiếp.

Tuy nhiên, quy trình tiếp nhận và xử lý đơn hàng của FastFood HVT hiện còn nhiều thủ công:

* Khách hàng đặt món qua điện thoại hoặc nhắn tin qua mạng xã hội (Zalo, Facebook).
* Nhân viên ghi đơn hàng bằng tay hoặc nhập vào file Excel, dễ nhầm lẫn.
* Không có hệ thống quản lý tập trung: đơn hàng bị trùng, sai địa chỉ giao hàng, thiếu món.
* Không thể kiểm soát tình trạng món bán chạy, tồn kho, thời gian giao hàng, doanh thu theo ngày.

Từ thực tế đó, cửa hàng nhận thấy nhu cầu cấp thiết cần một website đặt đồ ăn trực tuyến để tối ưu quy trình kinh doanh. Qua khảo sát hoạt động thực tế, các vấn đề nổi bật được ghi nhận như sau:

Các nghiệp vụ hiện tại:

* Khách hàng đặt món:
* Đặt món qua cuộc gọi, mạng xã hội → nhân viên nhập tay → chậm, sai sót cao.
* Không có hệ thống xác nhận tự động hay theo dõi đơn hàng.
* Quản lý thực đơn:
* Khi thay đổi giá hoặc ngừng bán món nào, phải cập nhật thủ công ở nhiều nơi (file Excel, fanpage, nhân viên nhớ).
* Không thể đồng bộ cho khách hàng biết được món đang tạm hết.
* Xử lý đơn hàng và giao hàng:
* Không có chức năng theo dõi đơn hàng đang ở trạng thái nào (chờ xác nhận, đang nấu, đang giao, đã giao).
* Việc điều phối giao hàng chủ yếu do nhân viên ghi nhớ và liên hệ trực tiếp với shipper.
* Thanh toán:
* Hầu hết là thanh toán tiền mặt khi nhận hàng → không linh hoạt với khách hàng thích chuyển khoản, ví điện tử.
* Không có hóa đơn điện tử, lịch sử giao dịch.
* Thống kê và báo cáo:
* Doanh thu, số đơn, món bán chạy, đánh giá khách hàng được ghi chép rời rạc, không có công cụ hỗ trợ phân tích.

Yêu cầu hệ thống mới:

* Xây dựng một website đặt đồ ăn có giao diện đơn giản, dễ sử dụng trên điện thoại và máy tính.
* Cho phép khách hàng:
* Xem thực đơn kèm hình ảnh và giá.
* Thêm món vào giỏ hàng và đặt hàng nhanh chóng.
* Theo dõi trạng thái đơn hàng và lựa chọn hình thức thanh toán.
* Cho phép nhân viên quản trị:
* Quản lý thực đơn, đơn hàng, người dùng, và phân công shipper.
* Thống kê doanh thu, lượt đặt món, món ăn bán chạy.
* Tích hợp thanh toán qua ví điện tử như MoMo, VNPAY, chuyển khoản ngân hàng.

Việc triển khai hệ thống sẽ giúp tăng tốc độ xử lý đơn, giảm chi phí quản lý, nâng cao trải nghiệm khách hàng và hỗ trợ nhà quản lý đưa ra quyết định kinh doanh chính xác.

# **CHƯƠNG 3**

# **PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

## **1. Sơ đồ phân rã chức năng**

Hình 3- 1 Sơ đồ phân rã chức năng

**Mô tả hệ thống**

*Chức năng quản lý hệ thống*

Đăng nhập hệ thống: Giao diện đăng nhập thân thiện, bảo mật, dành cho quản trị viên và người dùng.

Đổi mật khẩu: Người dùng và quản trị viên có thể đổi mật khẩu tài khoản để đảm bảo tính bảo mật.

Phân quyền người dùng:

Quản trị viên: Toàn quyền quản lý hệ thống, thực đơn, đơn hàng, tài khoản...

Khách hàng: Có thể đặt món, theo dõi đơn hàng, đánh giá.

Nhân viên giao hàng: Chỉ xem và cập nhật tình trạng đơn hàng được giao.

*Chức năng quản lý thực đơn*

Cập nhật danh sách món ăn: Thêm, sửa, xoá các món ăn đang phục vụ.

Cập nhật thông tin món ăn: Bao gồm tên món, mô tả, giá, hình ảnh, danh mục (đồ ăn, đồ uống...).

Quản lý danh mục món: Tổ chức món ăn theo các nhóm để khách dễ tìm kiếm.

*Chức năng quản lý đơn hàng*

Theo dõi đơn hàng: Hiển thị danh sách đơn hàng, chi tiết từng đơn.

Cập nhật trạng thái đơn hàng: Chuyển đơn qua các trạng thái như: "Đang chuẩn bị", "Đang giao", "Đã giao", "Đã huỷ".

Xử lý đơn hàng: Gán đơn hàng cho nhân viên giao hàng, liên hệ khách khi cần thiết.

*Chức năng giao hàng*

Theo dõi trạng thái giao hàng: Theo dõi thời gian và trạng thái vận chuyển.

*Chức năng tìm kiếm*

Tìm món ăn theo tên: Hỗ trợ khách hàng tìm kiếm nhanh món ăn.

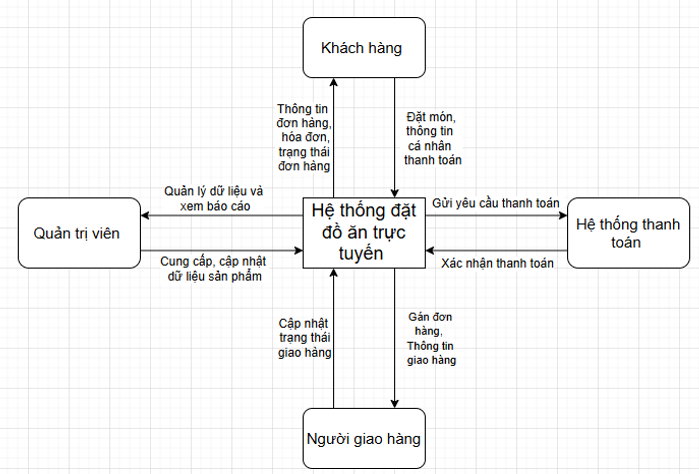
*Chức năng báo cáo*

Báo cáo doanh thu: Thống kê doanh thu theo ngày, tuần, tháng, năm để hỗ trợ quản trị kinh doanh.

Báo cáo món ăn bán chạy: Xác định các món ăn được đặt nhiều nhất để lên kế hoạch nhập nguyên liệu, khuyến mãi.

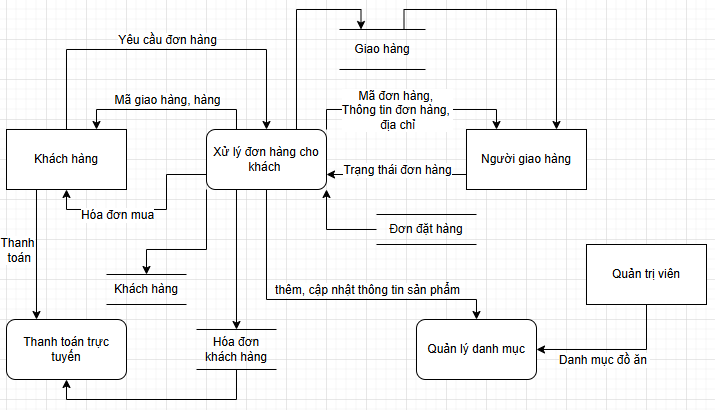
Báo cáo phản hồi của khách hàng: Ghi nhận đánh giá để nâng cao chất lượng phục vụ.

## **2. Sơ đồ ngữ cảnh**

****

Hình 3- 2 Sơ đồ mức ngữ cảnh

## **3. Sơ đồ luồng dữ liệu mức đỉnh**



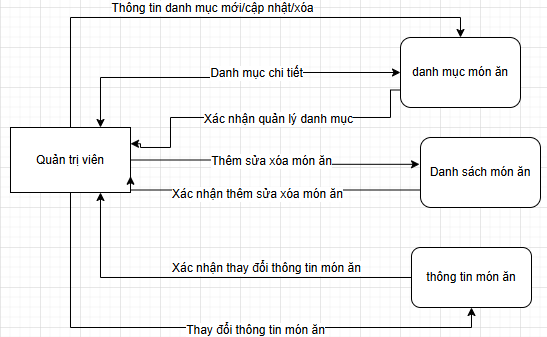
Hình 3- 3 Sơ đồ luồng dữ liệu mức đỉnh

## **4. Các sơ đồ luồng dữ liệu mức dưới đỉnh**

### **4.1. Sơ đồ mức dưới đỉnh quản lý hệ thống**

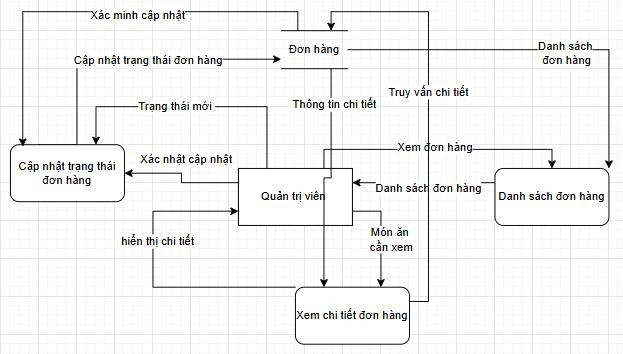
Hình 3- 4 Sơ đồ mức dưới đỉnh quản lý hệ thống

### **4.2. Sơ đồ mức dưới đỉnh quản lý thực đơn**



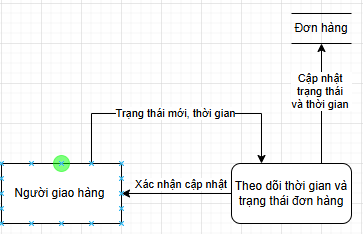
Hình 3- 5 Sơ đồ mức dưới đỉnh quản lý thực đơn

### **4.3. Sơ đồ mức dưới đỉnh quản lý đơn hàng**



Hình 3- 6 Sơ đồ mức dưới đỉnh quản lý đơn hàng

### **4.4. Sơ đồ mức dưới đỉnh quản lý giao hàng**



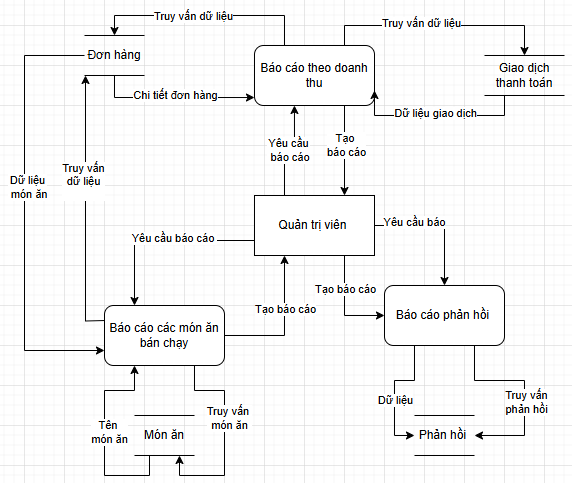
Hình 3- 7 Sơ đồ mức dưới đỉnh quản lý giao hàng

### **4.5. Sơ đồ mức dưới đỉnh tìm kiếm**



Hình 3- 8 Sơ đồ mức dưới đỉnh tìm kiếm

### **4.6. Sơ đồ mức dưới đỉnh báo cáo thống kê**



Hình 3- 9 Sơ đồ mức dưới đỉnh báo cáo thống kê

# **CHƯƠNG 4**

# **NỘI DUNG THỰC TẬP**

## **1. Giới thiệu về các công cụ đã tìm hiểu và sử dụng trong dự án**

### **1.1. ReactJs**

#### undefined**1.1.1. Giới thiệu về ReactJs**

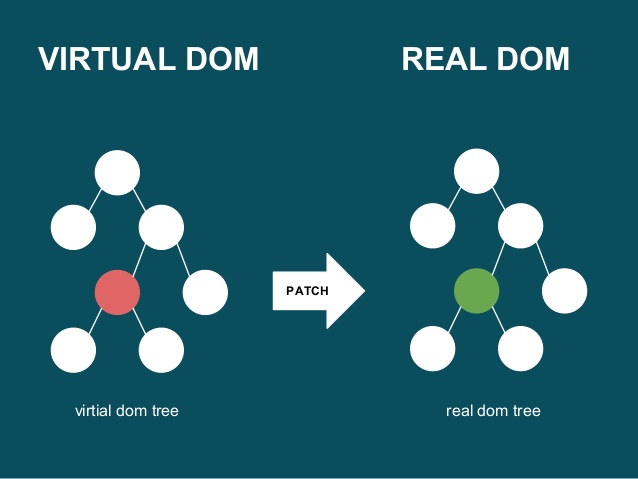
Hình 4- 1 Logo thư viện ReactJs

React là một thư viện UI phát triển tại Facebook để hỗ trợ việc xây dựng những thành phần (components) UI có tính tương tác cao, có trạng thái và có thể sử dụng lại được. React được sử dụng tại Facebook trong production, và www.instagram.com được viết hoàn toàn trên React.

Một trong những điểm hấp dẫn của React là thư viện này không chỉ hoạt động trên phía client, mà còn được render trên server và có thể kết nối với nhau.

React sử dụng khái niệm DOM ảo (Virtual DOM) để chọn lựa và render những phần tử của node dựa tên sự thay đổi trạng thái khiến cho ta chỉ cần thay đổi ít thành phần nhất có thể để giữ DOM update.

#### **1.1.2. Virtual DOM hoạt động như thế nào ?**



Hình 4 – 2: Virtual Dom

Tưởng tượng bạn có một vật thể được thiết kế dựa trên một người. Nó có mọi bộ phận một người có thể có, và phản ánh lại trạng thái hiện tại của người đó. Đây là điều cơ bản React làm với DOM.

Bây giờ thử nghĩ rằng nếu bạn lấy vật thể đó và thay đổi một vài bộ phận. Thêm ria, bắp tay và đôi mắt xanh. Trong React, khi chúng ta tạo ra thay đổi, sẽ diễn ra 2 việc. Đầu tiên, React chạy một thuật toán so sánh sự khác biệt để phát hiện ra thay đổi. Bước thứ 2 là điều hòa bằng cách cập nhật DOM với kết qủa của thuật toán ở bước 1.

Cách React hoạt động là thay vì lấy một người thật và tái tạo lại từ đầu, React chỉ thay đổi khuôn mặt và tay. Điều này có nghĩa là khi bạn nhập vào 1 đoạn text và render xong thì đoạn text này sẽ không bị ảnh hưởng cho tới khi parent node được sắp xếp lại.

Vì React sử dụng một DOM giả nên cũng có khá nhiều ý tưởng thú vị xung quanh thư viện này. Ví dụ như ta có thể render DOM giả này trên server.

#### **1.1.3. Bắt đầu**

Bắt đầu với React đơn giản bằng cách download React Starter Kit

Bạn cũng có thể fork trên React JSFiddle

Thiết lập trang Khi thiết lập trang web, bạn cần thêm vào các thư viện react.js, react-dom.js và JSXTransformer.js, và sau đó thêm component của bạn vào một node script có type="text/jsx":

React DOM là một thư viện hỗ trợ và mới được thêm vào trong React. React DOM trước đó là một phần của thư viện React nhưng được tách ra từ bản 0.14

Trong React, component được cài trên một thành phần nên trong ví dụ này ta có thể sử dụng div như một thành phần bao ngoài.

Đây là cách dễ nhất để bắt đầu, tuy nhiên khi thực sự xây dựng một apps hoàn chỉnh thì tốt hơn hết là sử dụng Browserify hoặc Webpack và thiết lập components trong files riêng biệt.

#### **1.1.4. Cơ bản**

Thành phần cơ bản của React được gọi là components. Syntax để viết HTML khá là kỳ quặc khi sử dụng Javascript để render:

**JSX**

Đoạn Javascript trên được gọi là JSX và nó là syntax chuyển từ Javascript XML. JSX cho phép bạn viết Javascript theo phong cách HTML.

Đối với những tag html bình thường, attribute class được gọi là className và for là htmlFor vì đây là những từ được quy định sẵn. Bạn có thể xem thêm tại JSX Gotchas của Facebook.

**Components**

Khi sử dụng method render với React DOM như trên, argument đầu tiên chính là component chúng ta muốn render, và argument thứ hai là DOM node nó được cài lên. Chúng ta có thể sử dụng method createClass để tạo các class component. Nó có thể nhận một vài giá trị từ một object làm argument.

**Props**

Khi sử dụng components của mình, chúng ta có thể thêm attributes gọi là props. Những attributes này được gọi ra trong components bằng this.props và có thể được sử dụng trong method render để render dữ liệu động.

**Specs, Lifecycle & State**

Method render là spec duy nhất cần có khi tạo một component, nhưng một vài methods lifecycle và spec chúng ta có thể sử dụng vô cùng hữu ích khi bạn thực sự muốn component của bạn thực hiện gì đó. Phương pháp kết xuất là thông số kỹ thuật duy nhất cần thiết để tạo một thành phần, nhưng có một số phương pháp & thông số kỹ thuật vòng đời mà chúng tôi có thể sử dụng rất hữu ích khi bạn thực sự muốn thành phần của mình làm bất cứ điều gì.

**Các method lifecycle**

componentWillMount – được gọi một lần trên cả client và server trước khi render diễn ra. componentDidMount – được gọi một lần chỉ trên client, sau khi render. shouldComponentUpdate – trả lại giá trị quyết định xem component có được update không. componentWillUnmount – được gọi trước khi tách component.

**Các method specs**

getInitialState – trả lại giá là giá trị ban đầu của state. getDefaultProps – lấy props mặc định nếu props không được cung cấp. mixins – một chuỗi các object được sử dụng để extend các tính năng của component. Có một vài method lifecycle và specs khác mà bạn có thể đọc thêm tại đây.

**Trạng thái (state)**

Mọi component đều có một object state và một object props. State được thiết lạp sử dụng method setState. Gọi setState kích hoạt cập nhật UI và gắn liền với hoạt động của React. Nếu chúng ta muốn thiết lập một trạng thái ban đầu trước khi có tương tác thì ta có thể sử dụng method getInitialState.

**Events**

React cũng có một hệ thống events sử dụng được ở các browser khác nhau. Những events này được gắn liền như thành phần của components và có thể kích hoạt các method.

**Dòng dữ liệu một chiều**

Trong React, dòng dữ liệu của application là một chiều thông qua state và props, khác với dòng dữ liệu hai chiều trong những thư viện như Angular. Điều này có nghĩa là trong cấu trúc nhiều tầng components, một component cha có thể quản lý state và truyền xuống component bên dưới thông qua props.

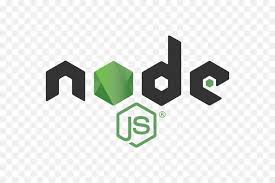
State có thể được update sử dụng method setState để đảm bảo rằng UI được cập nhật liên tục. Kết qủa sẽ được truyền xuống component con sử dụng attributes thông qua this.props.

#### **1.1.5. Kết luận**

React là một thư viện rất thú vị và được phát triển dựa trên rất nhiều cấu trúc phức tạp. Tuy nhiên thư viện này lại rất dễ sử dụng và thêm vào trong nhiều application khác nhau. Các bạn có thể đọc hướng dẫn sử dụng React chi tiết hơn tại đây. Ngoài ra React còn có thư viện React Native được dùng để thiết kế native apps.

### **1.2. NodeJs**

#### **1.2.1. NodeJS là gì?**

NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp. NodeJS là một mã nguồn mở được sử dụng rộng bởi hàng ngàn lập trình viên trên toàn thế giới. NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ WIndow cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.

Hình 4 – 3: Logo môi trường NodeJs

#### **1.2.2 Các đặc tính của NodeJS**

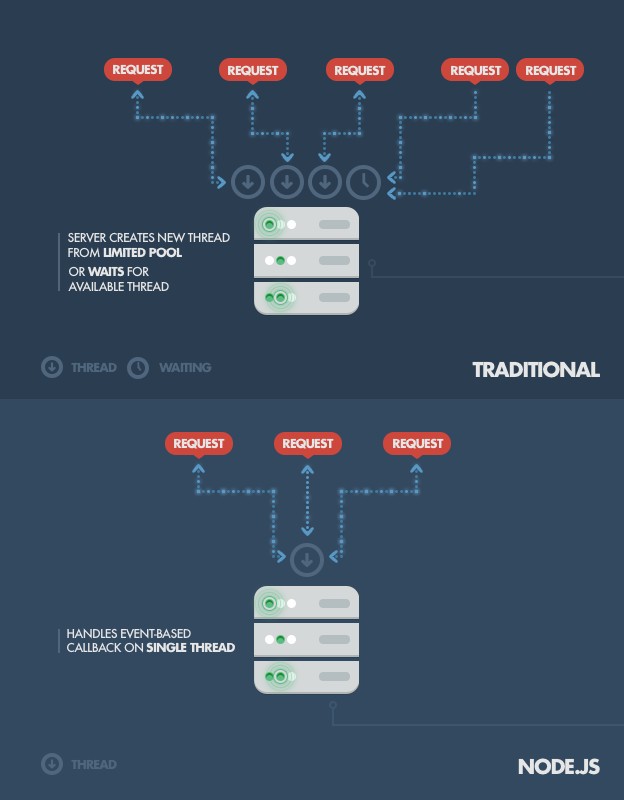
* Không đồng bộ: Tất cả các API của NodeJS đều không đồng bộ (none-blocking), nó chủ yếu dựa trên nền của NodeJS Server và chờ đợi Server trả dữ liệu về. Việc di chuyển máy chủ đến các API tiếp theo sau khi gọi và cơ chế thông báo các sự kiện của Node.js giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước (Realtime).
* Chạy rất nhanh: NodeJ được xây dựng dựa vào nền tảng V8 Javascript Engine nên việc thực thi chương trình rất nhanh.
* Đơn luồng nhưng khả năng mở rộng cao: Node.js sử dụng một mô hình luồng duy nhất với sự kiện lặp. cơ chế tổ chức sự kiện giúp các máy chủ để đáp ứng một cách không ngăn chặn và làm cho máy chủ cao khả năng mở rộng như trái ngược với các máy chủ truyền thống mà tạo đề hạn chế để xử lý yêu cầu. Node.js sử dụng một chương trình đơn luồng và các chương trình tương tự có thể cung cấp dịch vụ cho một số lượng lớn hơn nhiều so với yêu cầu máy chủ truyền thống như Apache HTTP Server.
* Không đệm:

NodeJS không đệm bất kì một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.

* Có giấy phép: NodeJS đã được cấp giấy phép bởi MIT License.

#### **1.2.3.Node js làm việc như thế nào?**

Ý tưởng chính của Node js là sử dụng non-blocking, hướng sự vào ra dữ liệu thông qua các tác vụ thời gian thực một cách nhanh chóng. Bởi vì, Node js có khả năng mở rộng nhanh chóng, khả năng xử lý một số lượng lớn các kết nối đồng thời bằng thông lượng cao. Nếu như các ứng dụng web truyền thống, các request tạo ra một luồng xử lý yêu cầu mới và chiếm RAM của hệ thống thì việc tài nguyên của hệ thống sẽ được sử dụng không hiệu quả. Chính vì lẽ đó giải pháp mà Node js đưa ra là sử dụng luồng đơn (Single-Threaded), kết hợp với non-blocking I/O để thực thi các request, cho phép hỗ trợ hàng chục ngàn kết nối đồng thời.



Hình 4 – 4: Cách hoạt động NodeJs

#### **1.2.4. NPM: The Node Package Manager**

Khi thảo luận về Node js thì một điều chắc chắn không nên bỏ qua là xây dựng package quản lý sử dụng các cộng cụ NPM mà mặc định với mọi cài đặt Node js. Ý tưởng của mô-đun NPM là khá tương tự như Ruby-Gems: một tập hợp các hàm có sẵn có thể sử dụng được, thành phần tái sử dụng, tập hợp các cài đặt dễ dàng thông qua kho lưu trữ trực tuyến với các phiên bản quản lý khác nhau

Danh sách các mô-đun có thể tìm trên web NPM package hoặc có thể truy cập bằng cách sử dụng công cụ NPM CLI sẽ tự động cài đặt với Node js.

Một số các module NPM phổ biến nhất hiện nay là:

* expressjs.com/ - Express.js, một Sinatra-inspired web framework khá phát triển của Node.js, chứa rất nhiều các ứng dụng chuẩn của Node.js ngày nay.
* connect - Connect là một mở rộng của HTTP server framework cho Node.js, cung cấp một bộ sưu tập của hiệu suất cao "plugins" được biết đến như là trung gian; phục vụ như một nền tảng cơ sở cho Express
* socket.io and sockjs - Hai thành phần Server-side websockets components nổi tiếng nhất hiện nay.
* Jade - Một trong những engines mẫu, lấy cảm hứng từ HAML, một phần mặc định trong Express.js.
* mongo and mongojs - MongoDB hàm bao để cung cấp các API cho cơ sở dữ liệu đối tượng trong MongoDB Node.js
* redis - thư viện Redis client.
* coffee-script - CoffeeScript trình biên dịch cho phép developers viết các chương trình Node.js của họ dùng Coffee.
* underscore (lodash, lazy) - Thư viện tiện ích phổ biến nhất trong JavaScript, package được sử dụng với Node.js, cũng như hai đối tác của mình, hứa hẹn hiệu suất tốt hơn bằng cách lấy một cách tiếp cận thực hiện hơi khác nhau.
* forever - Có lẽ là tiện ích phổ biến nhất để đảm bảo rằng một kịch bản nút cho chạy liên tục. Giữ quá trình Node.js của bạn lên trong sản xuất đối mặt với bất kỳ thất bại không ngờ tới.

### **1.3. MongoDB**

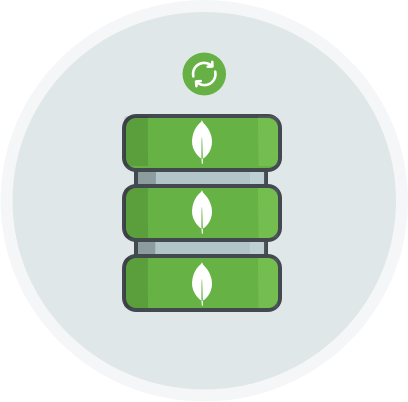
#### **1.3.1. Định nghĩa thêm về MongoDB**

MongoDB lần đầu ra đời bởi MongoDB Inc., tại thời điểm đó là thế hệ 10, vào tháng Mười năm 2007, nó là một phần của sản phẩm PaaS (Platform as a Service) tương tự như Windows Azure và Google App Engine. Sau đó nó đã được chuyển thành nguồn mở từ năm 2009.

MongoDB đã trở thành một trong những NoSQL database nổi trội nhất bấy giờ, được dùng làm backend cho rất nhiều website như eBay, SourceForge và The New York Times.

Các feature của MongoDB gồm có:

* Các ad hoc query: hỗ trợ search bằng field, các phép search thông thường, regular expression searches, và range queries.
* Indexing: bất kì field nào trong BSON document cũng có thể được index.
* Replication: có ý nghĩa là “nhân bản”, là có một phiên bản giống hệt phiên bản đang tồn tại, đang sử dụng. Với cơ sở dữ liệu, nhu cầu lưu trữ lớn, đòi hỏi cơ sở dữ liệu toàn vẹn, không bị mất mát trước những sự cố ngoài dự đoán là rất cao. Vì vậy, người ta nghĩ ra khái niệm “nhân bản”, tạo một phiên bản cơ sở dữ liệu giống hệt cơ sở dữ liệu đang tồn tại, và lưu trữ ở một nơi khác, đề phòng có sự cố.
* Aggregation: Các Aggregation operation xử lý các bản ghi dữ liệu và trả về kết quả đã được tính toán. Các phép toán tập hợp nhóm các giá trị từ nhiều Document lại với nhau, và có thể thực hiện nhiều phép toán đa dạng trên dữ liệu đã được nhóm đó để trả về một kết quả duy nhất. Trong SQL, count(\*) và GROUP BY là tương đương với Aggregation trong MongoDB.
* Lưu trữ file: MongoDB được dùng như một hệ thống file tận dụng những function trên và hoạt động như một cách phân phối qua sharding.



Hình 4 – 5: Hình ảnh tượng trưng MongooDb

#### **1.3.2. Khi nào sử dụng MongoDB?**

Quản lý và truyền tải content – Quản lý đa dạng nhiều product của content chỉ trong một kho lưu trữ data cho phép thay đổi và phản hồi nhanh chóng mà không chịu thêm phức tạp thêm từ hệ thống content.

Cấu trúc Mobile và Social – MongoDB cung cấp một platform có sẵn, phản xạ nhanh, và dễ mở rộng cho phép rất nhiều khả năng đột phá, phân tích real-time, và hỗ trợ toàn cầu.

Quản lý data khách hàng – Tận dụng khả năng query nhanh chóng cho phân tích real-time trên cơ sở dữ liệu người dùng cực lớn vớ các mô hình data phức tạp bằng các schema linh hoạt và tự động sharding cho mở rộng chiều ngang.

#### **1.3.3. Ưu điểm của MongoDB là gì?**

Dữ liệu lưu trữ phi cấu trúc, không có tính ràng buộc, toàn vẹn nên tính sẵn sàng cao, hiệu suất lớn và dễ dàng mở rộng lưu trữ.

Dữ liệu được caching (ghi đệm) lên RAM, hạn chế truy cập vào ổ cứng nên tốc độ đọc và ghi cao.

#### **1.3.4. Nhược điểm của MongoDB**

Không ứng dụng được cho các mô hình giao dịch nào có yêu cầu độ chính xác cao do không có ràng buộc.

Không có cơ chế transaction (giao dịch) để phục vụ các ứng dụng ngân hàng.

Dữ liệu lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn.

Mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

## **2. Tìm hiểu về API:**

### **2.1. Giới thiệu về API:**

API (Application Programming Interface) là một cơ chế cho phép hai phần mềm hoặc ứng dụng giao tiếp và trao đổi dữ liệu với nhau. Một cách trực quan để hiểu API có thể là thông qua ví dụ về dịch vụ dự báo thời tiết. Khi bạn nhận được thông tin về thời tiết hiện tại, đó có thể là do ứng dụng của bạn đang sử dụng API để kết nối với cơ quan dự báo thời tiết trung ương và lấy dữ liệu từ đó.

API có thể được hiểu như một giao diện mà các ứng dụng sử dụng để giao tiếp với nhau. Từ "ứng dụng" ở đây đề cập đến các phần mềm hoặc hệ thống có chức năng cụ thể. "Giao diện" đề cập đến thỏa thuận cụ thể về cách mà các ứng dụng có thể giao tiếp với nhau, thông qua yêu cầu và phản hồi được xác định rõ ràng. Các thông tin chi tiết về cách các nhà phát triển xây dựng nội dung cho các yêu cầu và phản hồi này có thể được tìm thấy trong tài liệu liên quan.

### **2.2. Cách thức hoạt động củaAPI**

Cách thức vận hành của API thường được phân tích ở khía cạnh máy chủ và máy khách. Máy khách được biết đến là các ứng dụng gửi đi yêu cầu, còn máy chủ sẽ là ứng dụng gửi đi phản hồi. Quay trở lại ví dụ thời tiết vừa phân tích ở trên thì có thể hiểu ứng dụng thời tiết trên điện thoại di động chính là máy khách, còn hệ thống cơ sở dữ liệu từ cơ quan thời tiết trung ương chính là máy chủ.

Có 4 cách để API hoạt động, nó phụ thuộc vào lý do và thời điểm tạo ra chúng. Cụ thể là:

* + API SOAP: Đây là một trong những API “đời đầu”, nó dùng giao thức truy cập đối tượng đơn giản. Lúc này, máy chủ sẽ trao đổi thông điệp thông qua XML với máy khách.
  + API RPC: Các API còn được biết đến với tên gọi là Lệnh gọi thủ tục từ xa. API này buộc máy chủ phải hoàn thành được một thủ tục bắt buộc thì máy khách mới có thể được nhận được kết quả.
  + API Websocket: Đây là một phiên bản API web hiện đại chuyển dữ liệu nhờ vào việc sử dụng các đối tượng JSON. Máy khách và máy chủ có thể trao đổi thông tin 2 chiều nhờ vào API này.
  + API REST: Hiện nay thì API REST được cho là linh hoạt nhất và cũng được sử dụng rộng rãi nhất. Với loại API này, máy khách sẽ gửi đến một yêu cầu dưới dạng dữ liệu. Sau đó máy chủ sẽ dùng các dữ liệu này để thực hiện các hàm nội bộ và cho ra các dữ liệu đầu ra rồi gửi lại cho máy khách.



Hình 4- 6: Cách thức hoạt động của API

Ưu điểm củaREST API

* + API REST thường được người dùng sử dụng với mục đích tích hợp các ứng dụng mới với các ứng dụng hiện có. Nhờ đó sẽ làm tăng hiệu quả của các ứng dụng mà không cần phải viết lại chương trình từ đầu.
  + API REST mang đến giúp các doanh nghiệp có thể đáp ứng mọi yêu cầu của khách hàng ở trên các nền tảng khác nhau.
  + API REST giống như một “vách ngăn” ở giữa 2 ứng dụng. Mỗi ứng dụng đó phải thực hiện các thay đổi nội bộ nhưng không được làm ảnh hưởng đến API. Có nghĩa là khi một ứng dụng có sự thay đổi mã thì ứng dụng còn lại cũng không bị tác động đến.

### **2.3 Ưu điểm và nhược điểm của API:**

API có những ưu điểm nổi bật có thể kể đến như:

* Có thể dùng API trên đa dạng các nền tảng của desktop, điện thoại di động mobile hay website.
* Với mỗi nền tảng thì API sẽ linh hoạt các định dạng dữ liệu sao cho phù hợp với các ứng dụng trên các hệ điều hành đó.
* Người dùng có thể thuận tiện xây dựng HTTP server khi dùng API để việc lập trình trở nên dễ dàng hơn.
* API sử dụng các mã nguồn mở với chức năng RESTFUL nên có thể ứng dụng vào bất cứ hệ điều hành nào để hỗ trợ JSON và XML.
* Có thể dùng API để trao đổi thông tin 2 chiều, đảm bảo được độ tin cậy cho các bên.
* Các thành phần MVC như Model Binder, Action, Unit Test, Controller,… đều có thể được API hỗ trợ đầy đủ các tính năng.

Bên cạnh những ưu điểm trên thì API vẫn còn một số điểm hạn chế cần khắc phục trong thời gian tới. Cụ thể là:

* API Website chưa hỗ trợ tất cả các server, nó chỉ mới hỗ trợ POST và GET mà thôi.
* Người dùng cần có hiểu biết kỹ càng và có kinh nghiệm backend thì mới có thể tận dụng triệt để các tính năng mà API mang lại.
* Việc phát triển, vận hành hay nâng cấp API là không hề dễ dàng và cần một thời gian dài để thực hiện. Thậm chí, người dùng còn phải trả một chi phí khá lớn cho việc đó.
* Nếu chủ sở hữu API chưa giới hạn kỹ các điều kiện thì vẫn có thể bị tấn công hệ thống và không đảm bảo được việc bảo mật.

### **2.4 Một số các ngăn chặn rò rỉ thông tin khi dùng API:**

* Xác thực cẩn thận thẩm quyền của người dùng cũng như ứng dụng

Bạn cần xác thực kỹ càng xem ai là người cuối cùng sử dụng cũng như sự tin cậy của ứng dụng. Chẳng hạn như khi bạn dùng các ứng dụng như Uber hay AirBnB thì chúng sẽ thực hiện quy trình gọi API. Khi đó thì ứng dụng sẽ được xác thực chính xác hơn.

* Mã hóa các dữ liệu sẽ được truyền tải từ các nền tảng

Bạn có thể dùng chứng chỉ SSL để thực hiện việc mã hóa này. Khi các dữ liệu đã được mã hóa, bạn sẽ ngăn chặn được các tình trạng như bị đột nhập, tấn công và ăn cắp các dữ liệu quan trọng trong quá trình truyền tải thông tin.

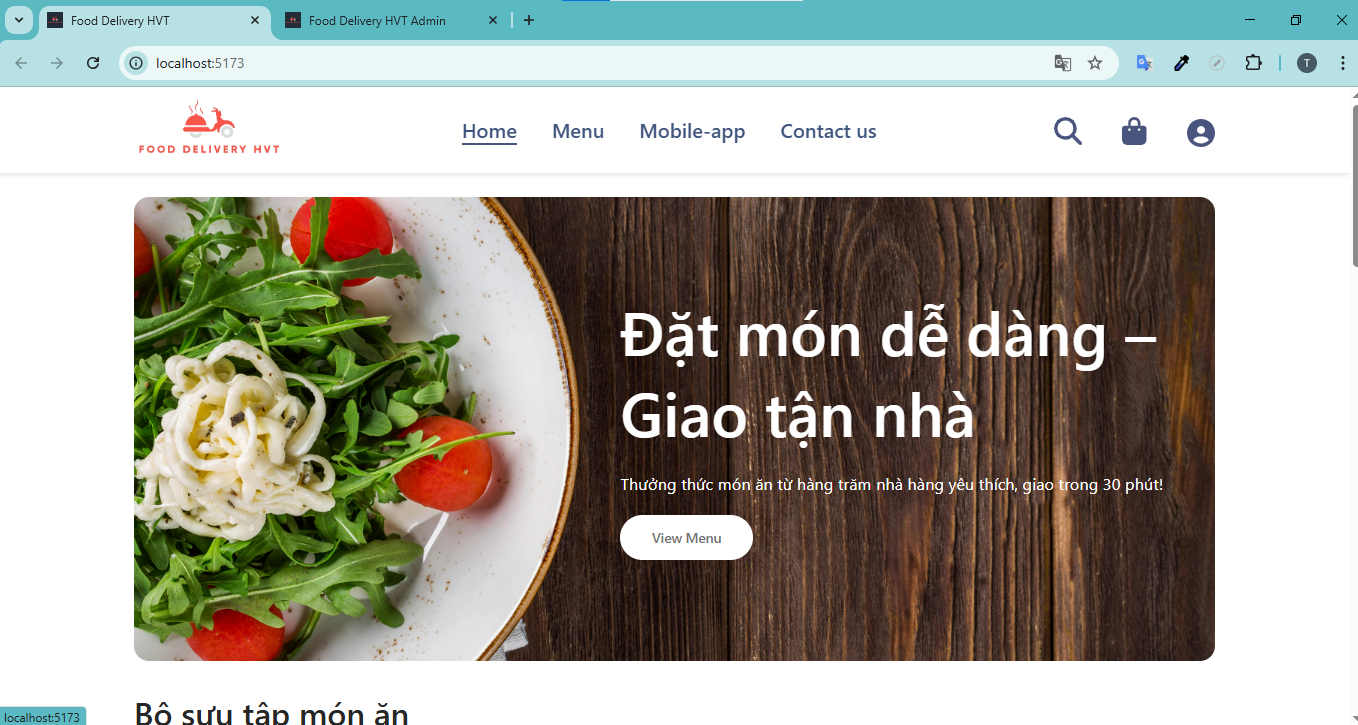
* Hạn chế đặt mật khẩu cố định hoặc quá dễ đoán

Nhiều người vì lười biếng hoặc vì hay quên nên thường chọn mật khẩu cố định hoặc lấy những mật khẩu được lưu cục bộ ở trên các ứng dụng trong điện thoại di động của mình. Thói quen này có thể sẽ làm cho bạn dễ dàng bị đánh cắp thông tin. Vì thế, hãy đặt một mật khẩu có độ khó và không phổ biến nhé.

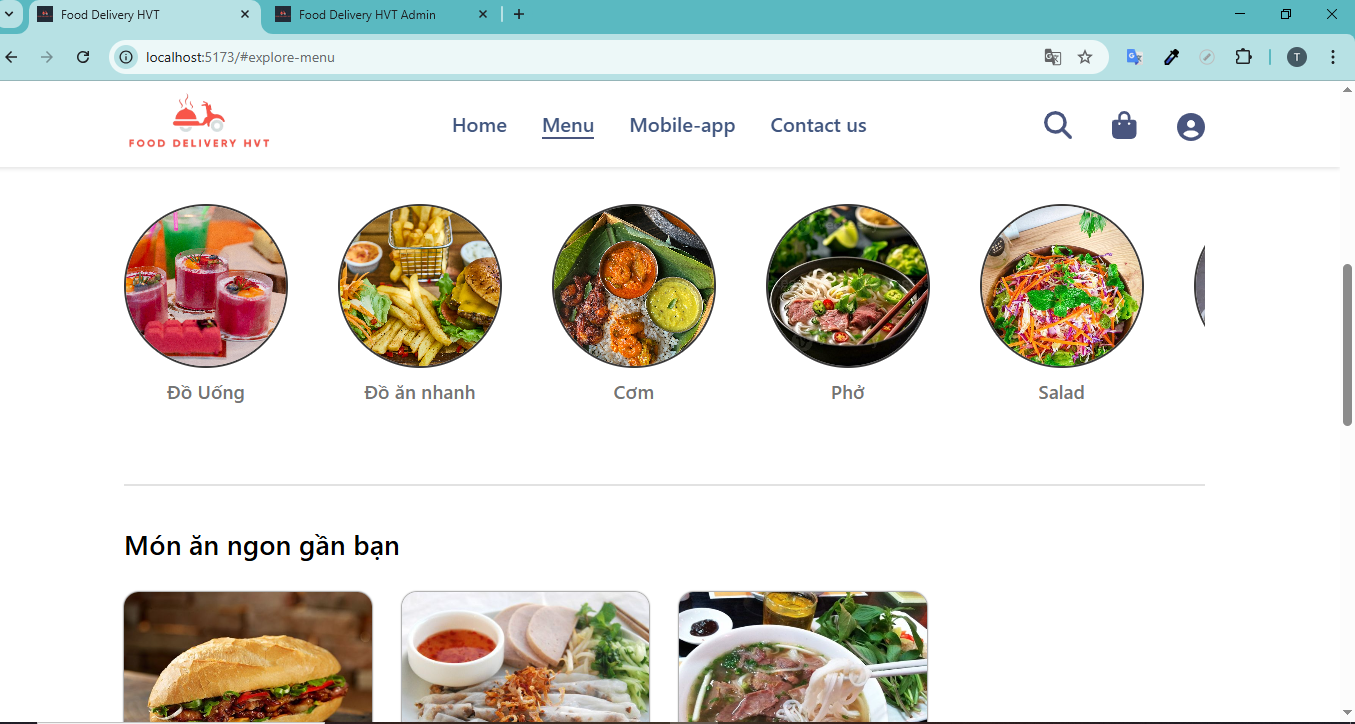
* Dùng chữ ký số để bảo mật

Chữ ký số là chuỗi ký tự đặc trưng và duy nhất của một người dùng. Nó sẽ giúp bạn lưu trữ các chuỗi này ở trong cơ sở dữ liệu và chỉ khi nhập đúng tên và mật khẩu thì mới được cung cấp chữ ký số.

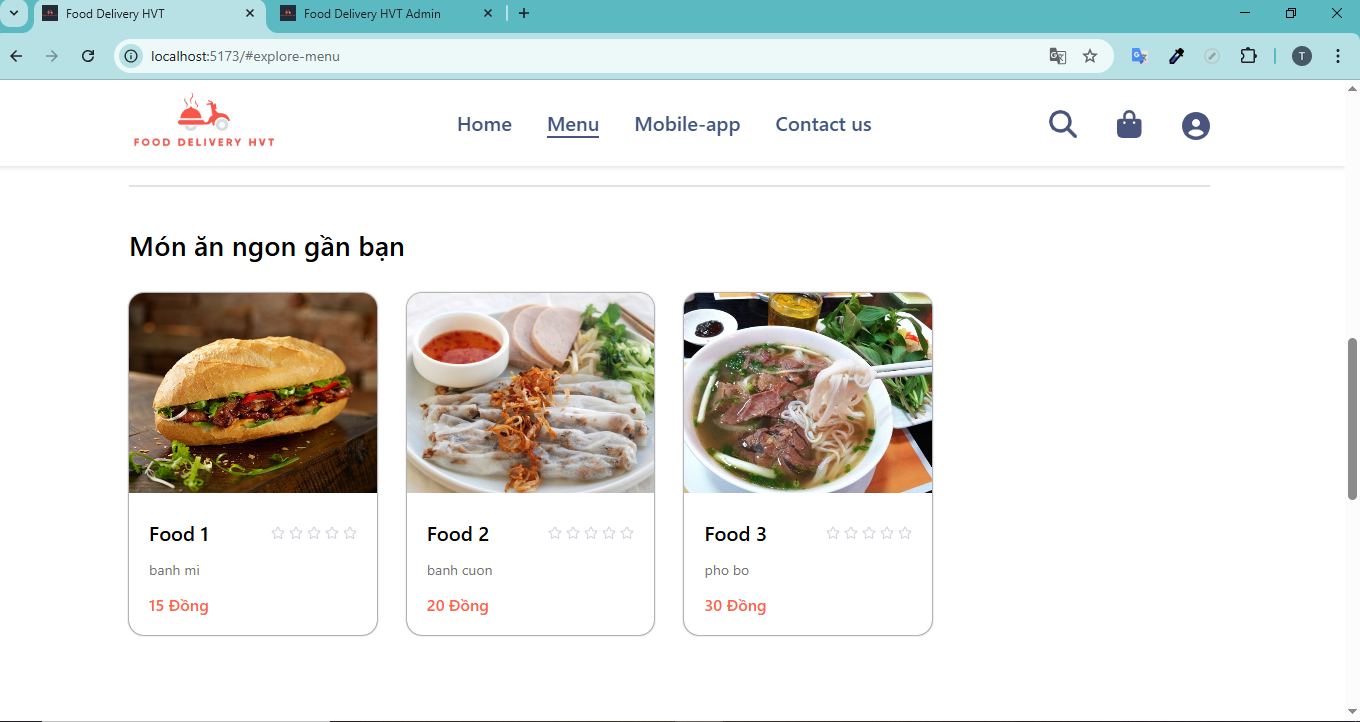
## **3. Giao diện trong website giao đồ ăn trực tuyến:**



Hình 4- 7 Giao diện “Trang chủ” của website



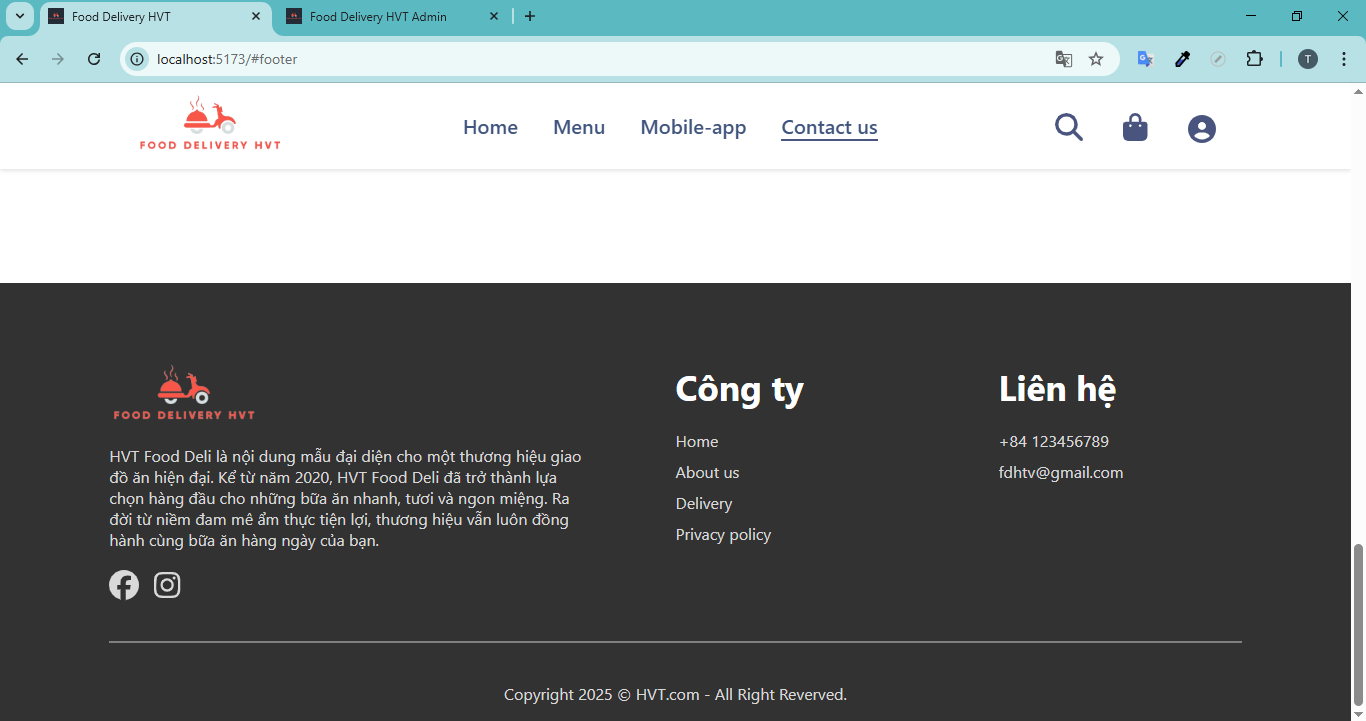
Hình 4- 8 Giao diện danh mục món



Hình 4- 9 Giao diện món ăn ở gần



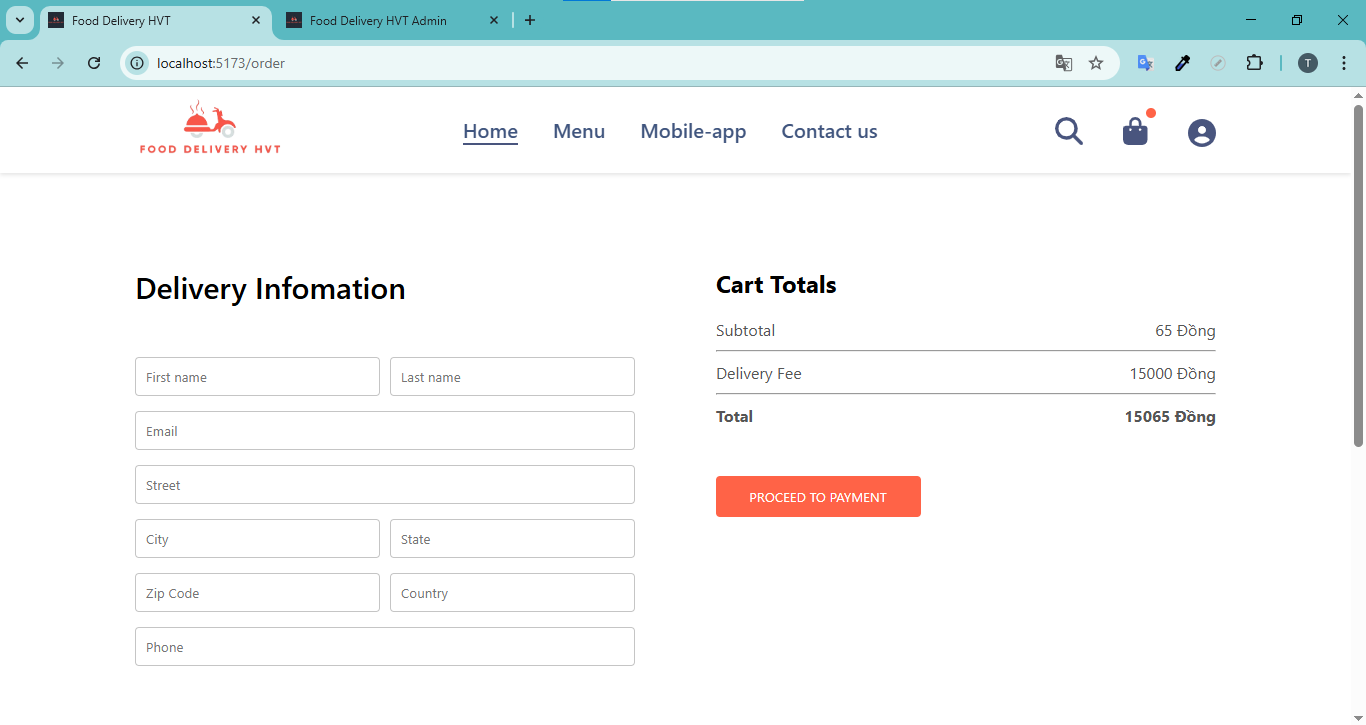
Hình 4- 10 Giao diện các app tải trên mobile



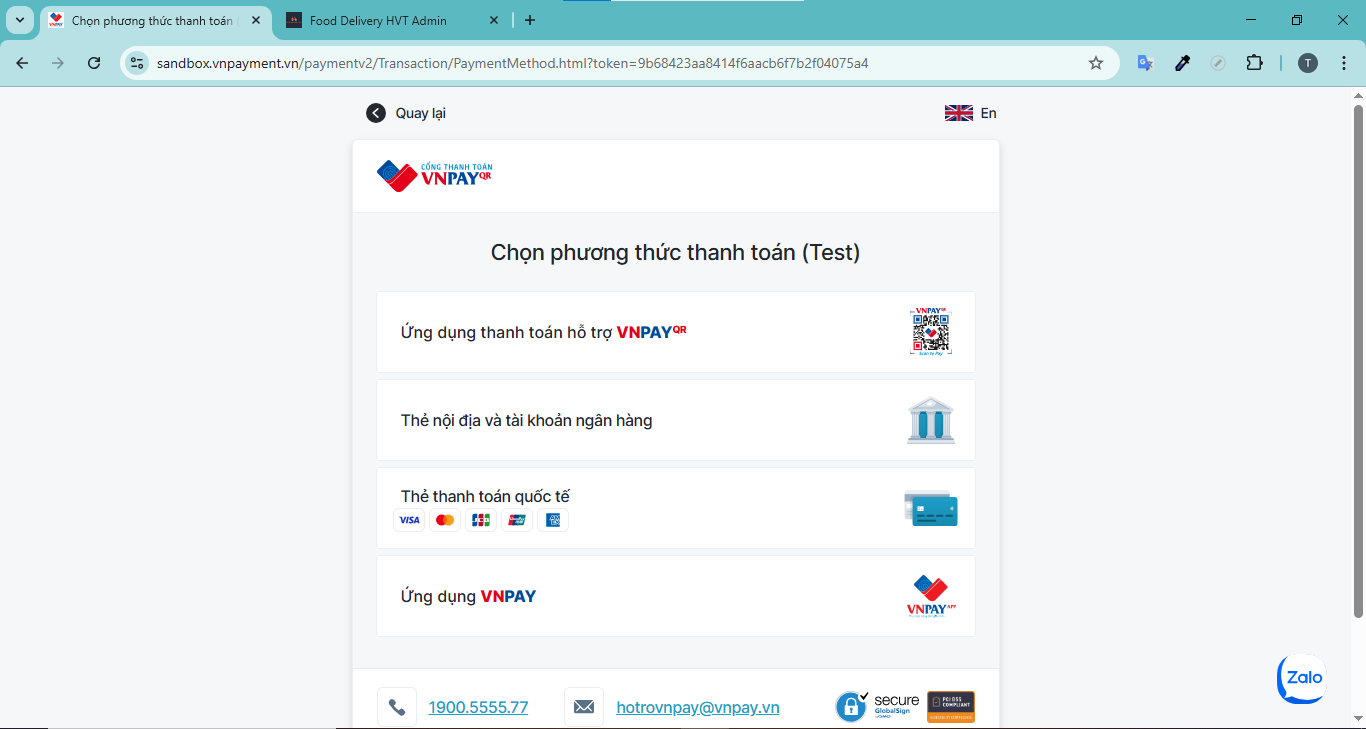
Hình 4- 11 Giao diện chân trang



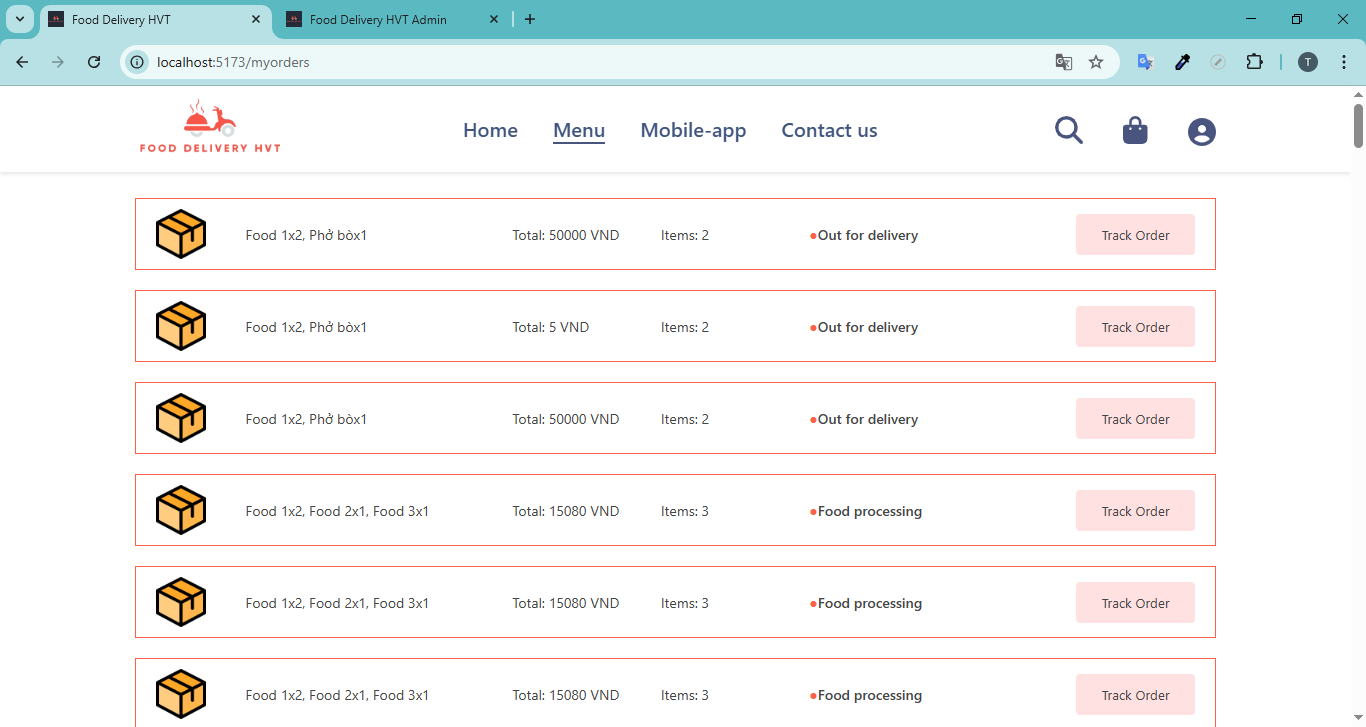
Hình 4- 12 Giao diện giỏ hàng



Hình 4- 13 Giao diện thông tin giao hàng



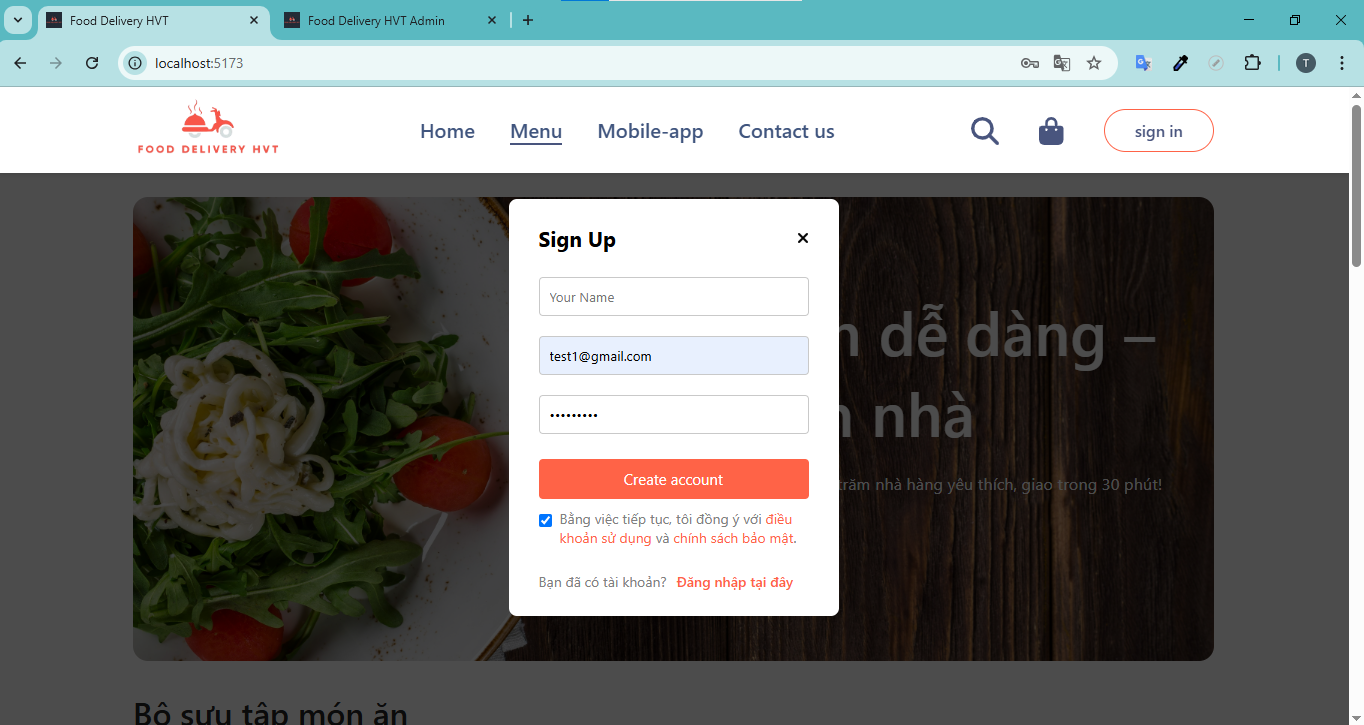
Hình 4- 14 Giao diện thanh toán vnpay



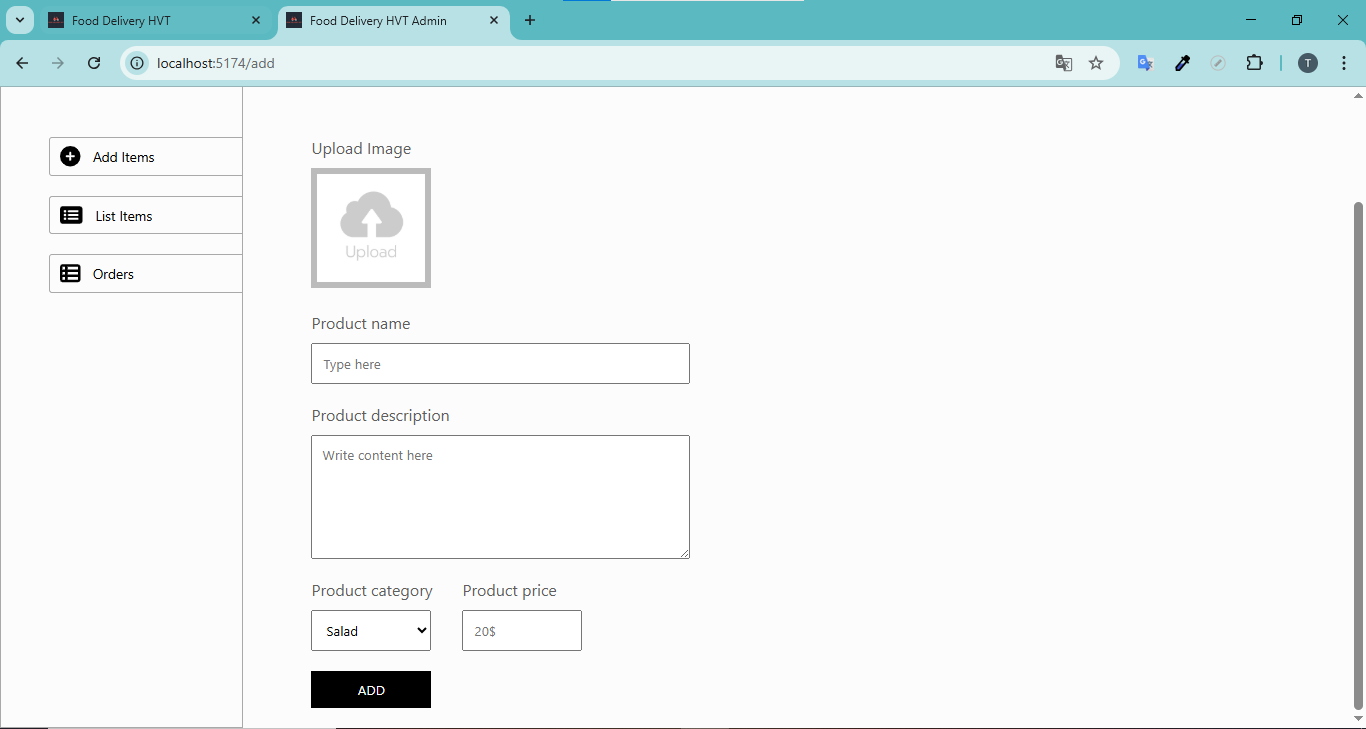
Hình 4- 15 Giao diện các món đã mua



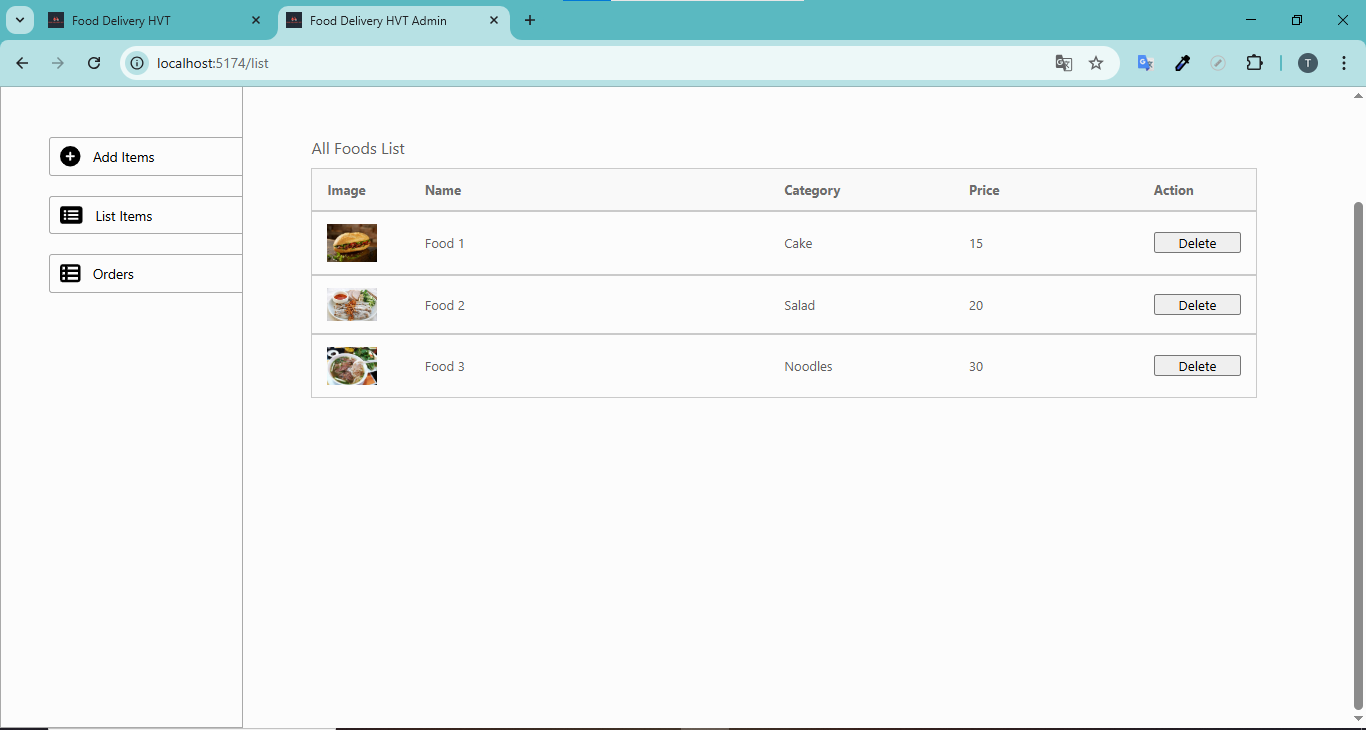
Hình 4- 16 Giao diện đăng nhập



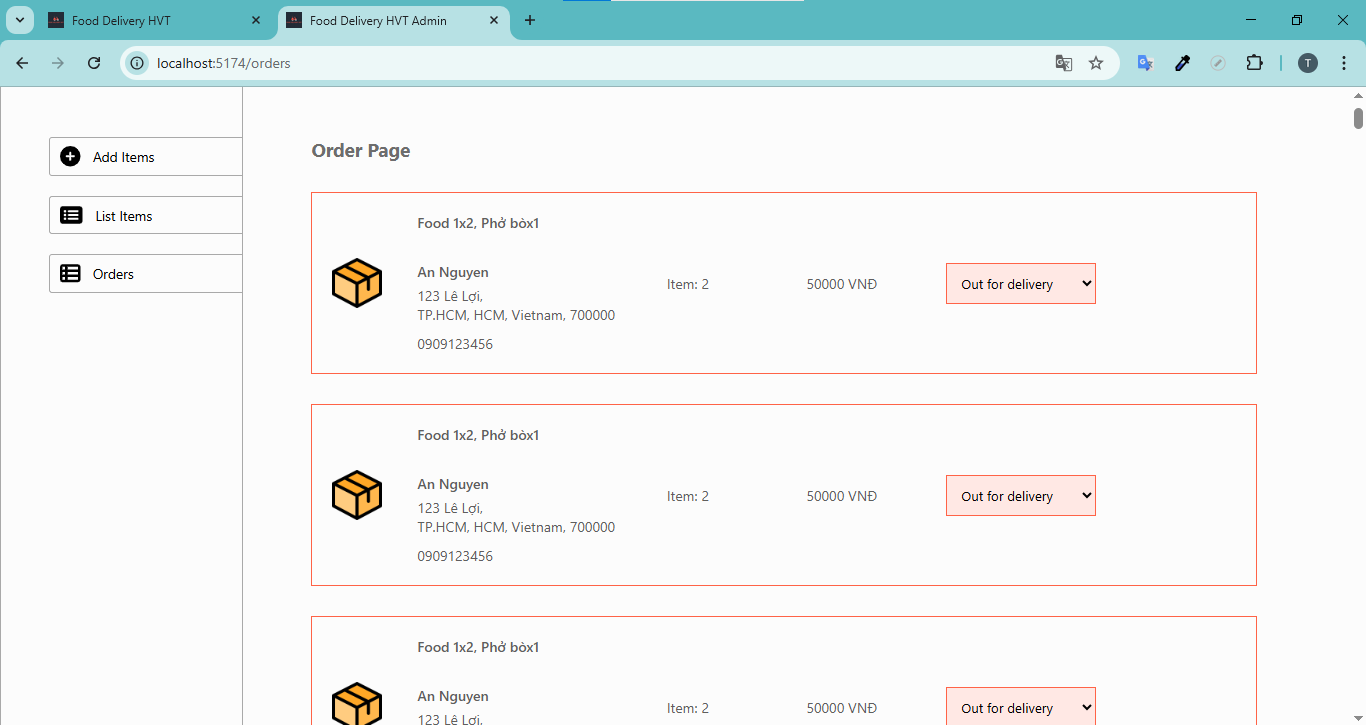
Hình 4- 17 Giao diện đăng kí



Hình 4-18: Giao diện thêm sản phẩm



Hình 4-19: Giao diện danh sách món ăn



Hình 4-20: Giao diện quản lý gọi món

# **CHƯƠNG 5:**

# **ĐÁNH GIÁ KẾT QUẢ VÀ ĐỊNH HƯỚNG PHÁT TRIỂN**

## **1. Đánh giá kết quả**

Qua hai tháng thực tập tại Công ty TNHH công nghệ và thương mại Wintech, em đã tìm hiểu về quy trình làm việc , tổ chức hoạt động của công ty và kiến thức học hỏi từ công ty. Từ đó em đã làm được website giao đồ ăn trực tuyến.

Về kiến thức :

Trong thời gian thực tập và tìm hiểu với sự định hướng và giúp đỡ tận tình của thầy Nguyễn Thế Lộc, em đã đặt được những kết quả như sau:

+ Nắm được tổng quan về các công cụ và thiết kế phần mềm:

Trong quá trình thực hiện dự án website đặt đồ ăn trực tuyến, em đã tìm hiểu và áp dụng các công nghệ hiện đại như ReactJS, Node.js, và MongoDB. Đây đều là các công cụ mạnh mẽ, phổ biến trong việc phát triển web hiện đại, đặc biệt phù hợp cho các ứng dụng theo mô hình client-server và SPA (Single Page Application).

+ Tìm hiểu và nắm được tính năng hỗ trợ kết nối cơ sở dữ liệu trong Node.js:

Node.js cho phép xây dựng RESTful API để xử lý logic phía máy chủ, nhận và trả dữ liệu về cho frontend React. Em đã sử dụng thư viện Mongoose để kết nối và tương tác với MongoDB, giúp quản lý dữ liệu đơn hàng, người dùng, món ăn... Việc thiết kế mô hình dữ liệu linh hoạt trên MongoDB rất phù hợp với tính chất mở rộng và phát triển nhanh của các ứng dụng hiện đại.

+ Nghiên cứu và tạo dựng cơ sở dữ liệu trong MongoDB:

Trong dự án, MongoDB được sử dụng như một cơ sở dữ liệu NoSQL, lưu trữ dữ liệu dưới dạng document (JSON). Điều này giúp xử lý linh hoạt các thông tin như món ăn, đơn hàng, ... Việc sử dụng MongoDB Atlas cũng giúp triển khai dễ dàng trên môi trường thực tế, hỗ trợ tốt cho các ứng dụng web hoạt động trực tuyến 24/7.

+ Nghiên cứu API trong Node.js – Express:

* Trong phần backend, em sử dụng Express.js để xây dựng hệ thống API RESTful, có khả năng kết nối mọi lúc qua Internet. Một số đặc điểm nổi bật gồm:
* Cấu hình đơn giản, hỗ trợ đầy đủ phương thức HTTP (GET, POST, PUT, DELETE) phục vụ cho các chức năng đặt món, quản lý người dùng, xử lý giỏ hàng,...
* Mã nguồn mở, dễ bảo trì và mở rộng.
* Tích hợp dễ dàng với hệ thống ReactJS ở frontend để tạo trải nghiệm mượt mà cho người dùng.
* Hỗ trợ kiểm thử, phân tách rõ ràng theo mô hình MVC, giúp dễ phát triển và bảo trì hệ thống.

+ Nâng cao trải nghiệm người dùng bằng ReactJS:

* ReactJS được sử dụng để xây dựng phần frontend của website, với các tính năng:
* Giao diện hiện đại, tương tác nhanh, phản hồi thời gian thực.
* Hỗ trợ chia component rõ ràng: trang chủ, thực đơn, giỏ hàng, thanh toán,...
* Giao diện responsive chạy tốt trên cả máy tính và thiết bị di động.
* Kết nối API backend thông qua Axios, giúp truyền/nhận dữ liệu nhanh chóng, đồng bộ giữa client và server.

+ Em đã áp dụng được những kiến thức từ bộ môn Phân tích thiết kế hệ thống vào trong công việc để phân tích hệ thống xử lý và tạo lập cơ sở dữ liệu của phần mềm.

**Về kỹ năng :**

* Kỹ năng giải quyết vấn đề: Biết tự giải quyết những vấn đề xuất phát từ bản thân mình, tìm ra nguồn gốc vấn đề, phân tích vấn đề và đưa ra những phương án tối ưu nhất để xử lý vấn đề.
* Kỹ năng hợp tác với người khác: Việc tích cực tham gia vào hội nhóm, cộng đồng lập trình viên có thể giúp bạn dễ dàng bàn bạc, trao đổi những thông tin, kiến thức cần thiết cũng nhận ra những điểm mạnh, điểm yếu của bản thân.
* Kỹ năng quản lý thời gian: Để hoàn thành các công việc được giao, em cần phải quản lý thời gian một cách hiệu quả.
* Kỹ năng tự học: Thường xuyên cập nhật những kiến thức mới, những xu hướng mới trong một thế giới công nghệ không ngừng chuyển mình thay đổi.
* Sự sáng tạo: Những ý tưởng và giải pháp tốt nhất thường đến khi chúng ta tiếp cận mọi thứ từ một góc nhìn khác, vượt ngoài khuôn khổ. Đây là điều làm nên sự khác biệt giữa những người sáng tạo nhất, từ lập trình viên, doanh nhân đến tác giả so với tất cả những người còn lại.

**Hạn chế :**

* Do thời gian nghiên cứu có hạn và khả năng còn hạn chế nên một số vẫn chỉ còn trên ý tưởng, chưa được thực thi.
* Một số chức năng chưa có hoặc chưa hoạt động ổn định.
* Chưa giải quyết chọn vẹn các vấn đề nảy sinh trong quá trình quản lý.
* Website chưa có tính thẩm mỹ chưa cao.

## **2. Định hướng phát triển đề tài**

Trong thời gian tới, em sẽ tiếp tục hoàn thiện các chức năng còn thiếu của hệ thống website đặt đồ ăn trực tuyến, như: theo dõi đơn hàng, đánh giá món ăn và API giao hàng (Be, Ahamove...).

Em cũng sẽ phát triển thêm hệ thống quản lý dành cho nhà hàng đối tác và nâng cao hiệu năng, tối ưu cơ sở dữ liệu MongoDB.

Bên cạnh đó, em sẽ tiếp tục tìm hiểu sâu hơn về ReactJS, Node.js và triển khai hệ thống thực tế bằng các công cụ như Vercel, Render hoặc Docker, làm nền tảng cho đồ án tốt nghiệp sau này.

# **KẾT LUẬN**

Trong quá trình thực tập, em đã học hỏi được nhiều kiến thức và kỹ năng mới về lập trình, từ việc tìm hiểu yêu cầu , nghiên cứu chi tiết chức năng, xây dựng giao diện người dùng, xây dựng cơ sở dữ liệu, cho đến việc xây dựng API, phân luồng và điều hướng cửa sổ.

Dựa trên các nội dung đã trình bày, báo cáo thực tập xây dựng website giao đồ ăn trực tuyến đã đề cập đến các chức năng cơ bản của lập trình website. Đồng thời, báo cáo cũng đề cập đến các kỹ năng cần thiết và kiến thức được học được sau quá trình thiết kế và lập trình. Đây là những kỹ năng rất quan trọng trong lập trình ngày nay và có thể giúp em hoàn thiện bản thân và phát triển sự nghiệp trong tương lai. Ngoài ra, em còn học được nhiều kỹ năng mềm như kỹ năng giao tiếp, thuyết phục, tư duy sáng tạo và sự kiên nhẫn, kiên trì trong công việc. Những kỹ năng này sẽ giúp em tự tin hơn trong việc giao tiếp với khách hàng, đồng nghiệp và xử lý tình huống trong công việc.

Trong bài báo cáo thực tập còn rất nhiều thiếu xót và hạn chế Em rất mong nhận được sự thông cảm và góp ý của quý thầy cô cùng các bạn để bài báo cáo ngày càng hoàn thiện hơn.

Một lần nữa em xin chân thành cảm ơn thầy Nguyễn Thế Lộc đã tận tình giúp đỡ em trong suốt thời gian thực tập và làm báo cáo thực tập. Em xin chân thành cảm ơn thầy và mong thầy giúp đỡ em nhiều hơn nữa trong quá trình phát triển làm đồ án tốt nghiệp.

# **TÀI LIỆU THAM KHẢO**

1. MongooDb là ?

https://topdev.vn/blog/mongodb-la-gi/

2. Giới thiệu về ReatJs

https://viblo.asia/p/gioi-thieu-reactjs-Qbq5QvaGKD8#\_5-ket-luan-11

3. Tổng quan về NodeJs

https://viblo.asia/p/tong-quan-ve-node-js-924lJra0lPM